

HP-UX Reference

Section 1: User Commands (N-Z)

HP-UX 11i Version 2

Volume 2 of 9



Manufacturing Part Number : B2355-90780

Printed In USA

E0803

Printed in USA

© Copyright 1983-2003 Hewlett-Packard Development Company LP.

Legal Notices

The information in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Use of this document and any supporting software media is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs, in their present form or with alterations, is expressly prohibited.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

U.S. Government License

Proprietary computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Reproduction, adaptation, or translation of this document without prior written permission is prohibited, except as allowed under the copyright laws.

This document and the software it describes may also be protected under one or more of the following copyrights. Additional copyrights are acknowledged in some individual manpages.

Copyright © 1983-2003 Hewlett-Packard Development Company, LP.

Copyright © 1979, 1980, 1983, 1985-1993 The Regents of the University of California.

Copyright © 1980, 1984, 1986 Novell, Inc.

Copyright © 1985, 1986, 1988 Massachusetts Institute of Technology

Copyright © 1986-2000 Sun Microsystems, Inc.

Copyright © 1988 Carnegie Mellon University

Copyright © 1989-1991 The University of Maryland

Copyright © 1989-1993 The Open Software Foundation, Inc.

Copyright © 1990 Motorola, Inc.

Copyright © 1990-1992 Cornell University

Copyright © 1991-2003 Mentat, Inc.

Copyright © 1996 Morning Star Technologies, Inc.

Copyright © 1996 Progressive Systems, Inc.

Trademark Notices

Intel® and Itanium® are registered trademarks of Intel Corporation in the US and other countries and are used under license.

Java™ is a US trademark of Sun Microsystems, Inc.

Microsoft® and MS-DOS® are U.S. registered trademarks of Microsoft Corporation.

OSF/Motif™ is a trademark of The Open Group in the US and other countries.

UNIX® is a registered trademark of The Open Group.

X Window System™ is a trademark of The Open Group.

Revision History

This document's printing date and part number indicate its edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) New editions of this manual incorporate all material updated since the previous edition.

Part Number Date, Release, Format, Distribution

| | |
|----------------|--|
| B2355-60103 | August 2003. HP-UX release 11i version 2, one volume HTML, docs.hp.com and Instant Information. |
| B2355-90779-87 | August 2003. HP-UX release 11i version 2, nine volumes PDF, docs.hp.com and print. |
| B9106-90010 | June 2002. HP-UX release 11i version 1.6, one volume HTML, docs.hp.com and Instant Information. |
| B9106-90007 | June 2001. HP-UX release 11i version 1.5, seven volumes HTML, docs.hp.com and Instant Information. |
| B2355-90688 | December 2000. HP-UX release 11i version 1, nine volumes. |
| B2355-90166 | October 1997. HP-UX release 11.0, five volumes. |
| B2355-90128 | July 1996. HP-UX release 10.20, five volumes, online only. |
| B2355-90052 | July 1995. HP-UX release 10.0, four volumes. |

Conventions

We use the following typographical conventions.

| | |
|------------------|---|
| <i>audit</i> (5) | An HP-UX manpage. <i>audit</i> is the name and 5 is the section in the <i>HP-UX Reference</i> . On the web and on the Instant Information CD, it may be a hot link to the |
|------------------|---|

manpage itself. From the HP-UX command line, you can enter “man audit” or “man 5 audit” to view the manpage. See *man* (1).

| | |
|-------------------|---|
| <i>Book Title</i> | The title of a book. On the web and on the Instant Information CD, it may be a hot link to the book itself. |
| KeyCap | The name of a keyboard key. Note that Return and Enter both refer to the same key. |
| <i>Emphasis</i> | Text that is emphasized. |
| Emphasis | Text that is strongly emphasized. |
| ENVIRONVAR | The name of an environment variable. |
| [ERRORNAME] | The name of an error number, usually returned in the <code>errno</code> variable. |
| Term | The defined use of an important word or phrase. |
| ComputerOutput | Text displayed by the computer. |
| UserInput | Commands and other text that you type. |
| Command | A command name or qualified command phrase. |
| <i>Variable</i> | The name of a variable that you may replace in a command or function or information in a display that represents several possible values. |
| [] | The contents are optional in formats and command descriptions. If the contents are a list separated by , you may choose one of the items. |
| { } | The contents are required in formats and command descriptions. If the contents are a list separated by , you must choose one of the items. |
| ... | The preceding element may be repeated an arbitrary number of times. |
| | Separates items in a list of choices. |

Preface

HP-UX is the Hewlett-Packard Company's implementation of an operating system that is compatible with various industry standards. It is based on the UNIX® System V Release 4 operating system and includes important features from the Fourth Berkeley Software Distribution.

The nine volumes of this manual contain the system reference documentation, made up of individual entries called **manpages**, named for the man command that displays them on the system. The entries are also known as manual pages or reference pages.

General Introduction

For a general introduction to HP-UX and the structure and format of the manpages, please see the *introduction* (9) manpage in volume 9.

Section Introductions

The manpages are divided into sections that also have introduction (*intro*) manpages that describe the contents. These are:

- | | |
|-------------------|---|
| <i>intro</i> (1) | <i>Section 1: User Commands</i> (A-M in volume 1; N-Z in volume 2) |
| <i>intro</i> (1M) | <i>Section 1M: System Administration Commands</i> (A-M in volume 3; N-Z in volume 4) |
| <i>intro</i> (2) | <i>Section 2: System Calls</i> (in volume 5) |
| <i>intro</i> (3C) | <i>Section 3: Library Functions</i> (A-M in volume 6; N-Z in volume 7) |
| <i>intro</i> (4) | <i>Section 4: File Formats</i> (in volume 8) |
| <i>intro</i> (5) | <i>Section 5: Miscellaneous Topics</i> (in volume 9) |
| <i>intro</i> (7) | <i>Section 7: Device (Special) Files</i> (in volume 9) |
| <i>intro</i> (9) | <i>Section 9: General Information</i> (in volume 9) |

Volume Two
Table of Contents

Section 1

Volume Two
Table of Contents

Section 1

Table of Contents Volumes One and Two

Section 1: User Commands

| Entry Name(Section): name | Description |
|--|--|
| intro(1): intro | introduction to command utilities and application programs |
| adb(1): adb | absolute debugger |
| adjust(1): adjust | simple text formatter |
| admin(1): admin | create and administer SCCS files |
| alias: substitute command and/or file name | see sh-posix(1) |
| alias: substitute command and/or filename | see cs(1) |
| alias: substitute command and/or filename | see ksh(1) |
| alloc: show dynamic memory usage | see cs(1) |
| answer(1): answer | phone message transcription system |
| ar(1): ar | maintain portable archives and libraries |
| as(1): as | assembler (Itanium Processor Family) |
| asa(1): asa | interpret ASA carriage control characters |
| at(1): at, batch | execute batched commands immediately or at a later time |
| attributes(1): attributes | describe audio file |
| awk(1): awk | text pattern scanning and processing language |
| banner(1): banner | make posters in large letters |
| basename(1): basename, dirname | extract portions of path names |
| batch: execute batched commands immediately | see at(1) |
| bc(1): bc | arbitrary-precision arithmetic language |
| bdiff(1): bdiff | diff for large files |
| break: exit from enclosing for/next loop | see cs(1) |
| break: exit from enclosing for/next loop | see ksh(1) |
| break: exit from enclosing for/next loop | see sh-posix(1) |
| breaksw: break from switch and resume after endsw | see cs(1) |
| bs(1): bs | a compiler/interpreter for modest-sized programs |
| cal(1): cal | print calendar |
| calendar(1): calendar | reminder service |
| cancel: cancel requests on an LP printer or plotter | see lp(1) |
| case: label in a switch statement | see cs(1) |
| case: label in a switch statement | see ksh(1) |
| case: label in a switch statement | see sh-posix(1) |
| cat(1): cat | concatenate, copy, and print files |
| ccat: cat compacted files | see compact(1) |
| cc_bundled(1): cc | bundled C compiler |
| cd: change working directory | see cs(1) |
| cd: change working directory | see ksh(1) |
| cd: change working directory | see sh-posix(1) |
| cd(1): cd | change working directory |
| cd(1): command | execute a simple command |
| cdc(1): cdc | change the delta commentary of an SCCS delta |
| chacl(1): chacl | add, modify, delete, copy, or summarize access control lists (ACLs) of files |
| chatr(1): chatr | change program's internal attributes |
| chdir: change current working directory | see cs(1) |
| checknr(1): checknr | check nroff/troff files |
| chfn(1): chfn | change user information in password file; used by finger |
| chgrp: change file group | see chown(1) |
| chkey(1): chkey | change user's secure RPC key |
| chmod(1): chmod | change file mode access permissions |
| chown(1): chown, chgrp | change file owner or group |
| chsh(1): chsh | change default login shell |
| ci(1): ci | check in RCS revisions |
| ckconfig(1): ckconfig | verify path names of all FTP configuration files |
| cksum(1): cksum | print file checksum and sizes |
| clear(1): clear | clear terminal screen |
| cmp(1): cmp | compare two files |

Table of Contents

Volumes One and Two

| Entry Name(Section): name | Description |
|--|--|
| co(1): co | check out RCS revisions |
| col(1): col | filter reverse line-feeds and backspaces |
| comb(1): comb | combine SCCS deltas |
| comm(1): comm | select or reject lines common to two sorted files |
| compact(1): compact, uncompact, ccat | compact and uncompact files, and cat them |
| compress(1): compress, compressdir, uncompress, uncompressdir, zcat | compress and expand data |
| compressdir: compress files in a directory | see compress(1) |
| continue: resume execution of nearest while or foreach | see cs(1) |
| continue: resume next iteration of enclosing for/next loop | see ksh(1) |
| continue: resume next iteration of enclosing for/next loop | see sh-posix(1) |
| convert(1): convert | convert audio file |
| cp(1): cp | copy file, files, or directory subtree |
| cpio(1): cpio | copy file archives in and out; duplicate directory trees |
| cpp(1): cpp | the C language preprocessor |
| crontab(1): crontab | user crontab file operations |
| crypt(1): crypt | encode/decode files |
| cs(1): cs | a shell (command interpreter) with C-like syntax |
| csplit(1): csplit | context split |
| ct(1): ct | spawn getty to a remote terminal (call terminal) |
| ctags(1): ctags | create a tags file |
| cu(1): cu | call another UNIX system; terminal emulator |
| cut(1): cut | cut out (extract) selected fields of each line of a file |
| date(1): date | display or set the system-clock date and time |
| dc(1): dc | desk calculator |
| dd(1): dd | convert, reblock, translate, and copy a (tape) file |
| default: label default in switch statement | see cs(1) |
| delta(1): delta | make a delta (change) to an SCCS file |
| deroff(1): deroff | remove nroff, tbl, and neqn constructs |
| diff(1): diff, diffh | differential file comparator |
| diff3(1): diff3 | 3-way differential file comparison |
| diffh: differential file comparator | see diff(1) |
| difflk(1): difflk | mark differences between files |
| dircmp(1): dircmp | directory comparison |
| dirname: extract portions of path names | see basename(1) |
| dirs: print the directory stack | see cs(1) |
| disable: disable LP printers | see enable(1) |
| dmpxlt(1): dmpxlt | dump iconv translation tables to a readable format |
| dnssec-keygen(1): dnssec-keygen | key generation tool for DNSSEC |
| dnssec-makekeyset(1): dnssec-makekeyset | produces a set of DNSSEC keys |
| dnssec-signkey(1): dnssec-signkey | DNSSEC keyset signing tool |
| dnssec-signzone(1): dnssec-signzone | DNSSEC zone signing tool |
| domainname(1): domainname | set or display NIS domain name |
| dos2ux(1): dos2ux, ux2dos | convert ASCII file format |
| doschmod(1): doschmod | change attributes of a DOS file |
| doscp(1): doscp | copy to or from DOS files |
| dosdf(1): dosdf | report number of free disk clusters |
| dosll: list contents of DOS directories | see dosls(1) |
| dosls(1): dosls, dosll | list contents of DOS directories |
| dosmkdir(1): dosmkdir | make a DOS directory |
| dosrm(1): dosrm, dosrmdir | remove DOS files or directories |
| dosrmdir: remove DOS directories | see dosrm(1) |
| du(1): du | summarize disk usage |
| dumpmsg: extract messages from message catalog file | see findmsg(1) |
| echo: echo (print) arguments | see cs(1) |
| echo: echo (print) arguments | see ksh(1) |
| echo: echo (print) arguments | see sh-posix(1) |
| echo(1): echo | echo (print) arguments |
| ed(1): ed, red | line-oriented text editor |
| edit: extended line-oriented text editor | see ex(1) |
| egrep: search a file for a pattern | see grep(1) |
| elfdump(1): elfdump | dump information contained in object files |

Table of Contents Volumes One and Two

| Entry Name(Section): name | Description |
|--|--|
| elm(1): elm | process electronic mail through a screen-oriented interface |
| elmalias(1): elmalias | display/verify elm user and system aliases |
| enable(1): enable, disable | enable/disable LP printers |
| end: terminate foreach or while loop | see cs(1) |
| endsw: terminate switch statement | see cs(1) |
| env(1): env | set environment for command execution |
| eucset(1): eucset | set and get EUC code widths for ldterm |
| eval: read arguments as shell input and execute resulting | see cs(1) |
| eval: read arguments as shell input and execute resulting commands | see ksh(1) |
| eval: read arguments as shell input and execute resulting commands | see sh-posix(1) |
| ex(1): edit, ex | extended line-oriented text editor |
| exec: execute command without creating new process | see cs(1) |
| exec: execute command without creating new process | see ksh(1) |
| exec: execute command without creating new process | see sh-posix(1) |
| exit: exit shell with exit status | see cs(1) |
| exit: exit shell with exit status | see ksh(1) |
| exit: exit shell with exit status | see sh-posix(1) |
| expand(1): expand, unexpand | expand tabs to spaces, and vice versa |
| expand_alias(1): expand_alias | recursively expands the sendmail aliases |
| export: export variable names to environment of subsequent commands | see ksh(1) |
| export: export variable names to environment of subsequent commands | see sh-posix(1) |
| expr(1): expr | evaluate arguments as an expression |
| factor(1): factor, primes | factor a number, generate large primes |
| false: do nothing and return non-zero exit status | see true(1) |
| fastbind(1): fastbind | prepare an incomplete executable for faster program start-up |
| fastmail(1): fastmail | quick batch mail interface |
| fc: edit and execute previous command | see ksh(1) |
| fc: edit and execute previous command | see sh-posix(1) |
| fgrep: search a file for a string (fast) | see grep(1) |
| file(1): file | determine file type |
| find(1): find | find files |
| findmsg(1): findmsg, dumpmsg | create message catalog file for modification |
| findstr(1): findstr | find strings for inclusion in message catalogs |
| finger(1): finger | user information lookup program |
| fmt(1): fmt | format text |
| fold(1): fold | fold long lines for finite width output device |
| for: execute a do list | see ksh(1) |
| for: execute a do list | see sh-posix(1) |
| forder(1): forder | convert file data order |
| foreach: initiate repetitive loop | see cs(1) |
| from(1): from | who is my mail from? |
| fruled(1): fruled | turn on/off attention LEDs |
| ftio(1): ftio | faster tape I/O |
| ftp(1): ftp | file transfer program |
| ftpcount(1): ftpcount | show current number of users for each class |
| ftprestart(1): ftprestart | remove shutdown message file created by ftpshut |
| ftpshut(1): ftpshut | create shutdown message file for ftp servers |
| ftpwho(1): ftpwho | show current process information for each ftp user |
| gencat(1): gencat | generate a formatted message catalog file |
| genxlt(1): genxlt | generate iconv translation |
| get(1): get | get a version of an SCCS file |
| getaccess(1): getaccess | list access rights to file(s) |
| getacl(1): getacl | list access control lists for files, JFS only |
| getconf(1): getconf | get POSIX configuration values |
| getopt(1): getopt | parse command options |
| getopts(1): getopts | parse utility (command) options |
| getprivgrp(1): getprivgrp | get special attributes for group |
| glob: echo without '\\' escapes | see cs(1) |
| goto: continue execution on specified line | see cs(1) |
| gprof(1): gprof | display call graph profile data |
| grep(1): grep, egrep, fgrep | search a file for a pattern |

Table of Contents

Volumes One and Two

| Entry Name(Section): name | Description |
|--|---|
| grget : get password and group information | see pwget(1) |
| groups(1) : groups | show group memberships |
| hashcheck : create hash codes from compressed spelling list | see spell(1) |
| hashmake : convert words to 9-digit hashcodes | see spell(1) |
| hashstat : print hash table effectiveness statistics | see cs(1) |
| head(1) : head | print first few lines in a file |
| history : display event history list | see cs(1) |
| host(1) : host | DNS lookup utility |
| hostname(1) : hostname | set or display name of current host system |
| hp(1) : hp | handle special functions of HP 2640 and HP 2621-series terminals |
| hp-mc680x0 : provide truth value about processor type | see machid(1) |
| hp-pa : provide truth value about processor type | see machid(1) |
| hp9000s800 : provide truth value about processor type | see machid(1) |
| hyphen(1) : hyphen | find hyphenated words |
| iconv(1) : iconv | character code set conversion |
| id(1) : id | print user and group IDs and names |
| ident(1) : ident | identify files in RCS |
| idlookup(1) : idlookup | identify the user of a particular TCP connection |
| ied(1) : ied | input editor and command history for interactive programs |
| if : execute command if expression evaluates true | see cs(1) |
| if : execute command if previous command returns exit status 0 | see ksh(1) |
| if : execute command if previous command returns exit status 0 | see sh-posix(1) |
| insertmsg(1) : insertmsg | use findstr(1) output to insert calls to catgets(3C) |
| inv : make unprintable and non-ASCII characters in a file invisible | see vis(1) |
| iostat(1) : iostat | report I/O statistics |
| ipcrm(1) : ipcrm | remove a message queue, semaphore set or shared memory id |
| ipcs(1) : ipcs | report status of interprocess communication facilities |
| jobs : list active jobs | see cs(1) |
| jobs : list active jobs | see ksh(1) |
| jobs : list active jobs | see sh-posix(1) |
| join(1) : join | relational database operator |
| kdestroy(1) : kdestroy | destroy Kerberos tickets |
| kermit(1) : kermit | C-Kermit 8.0 communications software for serial and network connections |
| keylogin(1) : keylogin | decrypt and store secret key |
| keylogout(1) : keylogout | delete secret key stored with keyser |
| keysh(1) : keysh | context-sensitive softkey shell |
| kill : send termination or specified signal to a process | see cs(1) |
| kill : terminate job or process | see ksh(1) |
| kill : terminate job or process | see sh-posix(1) |
| kill(1) : kill | send a signal to a process; terminate a process |
| kinit(1) : kinit | obtain and cache Kerberos ticket-granting ticket |
| klist(1) : klist | list cached Kerberos tickets |
| kpasswd(1) : kpasswd | change a user's Kerberos password |
| ksh(1) : ksh , rksh | shell, the standard/restricted command programming language |
| ktutil(1) : ktutil | Kerberos keytab file maintenance utility |
| kvno(1) : kvno | print key version numbers of Kerberos principals |
| l : list contents of directories | see ls(1) |
| last(1) : last , lastb | indicate last logins of users and ttys |
| lastb : indicate last bad logins of users and ttys | see last(1) |
| lastcomm(1) : lastcomm | show last commands executed in reverse order |
| lc : list contents of directories | see ls(1) |
| ld(1) : ld | link editor |
| ldd(1) : ldd | list dynamic dependencies of executable files or shared libraries |
| leave(1) : leave | remind you when you have to leave |
| let : evaluate arithmetic expression | see ksh(1) |
| let : evaluate arithmetic expression | see sh-posix(1) |
| lifcp(1) : lifcp | copy to or from LIF files |
| lifinit(1) : lifinit | write LIF volume header on file |
| lifls(1) : lifls | list contents of a LIF directory |
| lifrename(1) : lifrename | rename LIF files |
| lifrm(1) : lifrm | remove a LIF file |

Table of Contents Volumes One and Two

| Entry Name(Section): name | Description |
|--|---|
| limit: limit usage by current process | see cs(1) |
| line(1): line | read one line from user input |
| listusers(1): listusers | display user login data |
| ll: list contents of directories | see ls(1) |
| ln(1): ln | link files and directories |
| locale(1): locale | get locale-specific (NLS) information |
| lock(1): lock | reserve a terminal |
| logger(1): logger | make entries in the system log |
| login: terminate login shell | see cs(1) |
| login(1): login | sign on; start terminal session |
| logname(1): logname | get login name |
| logout: terminate login shell | see cs(1) |
| lorder(1): lorder | find ordering relation for an object library |
| lp(1): cancel, lp, lpalt | print/alter/cancel requests on an LP printer or plotter |
| lpalt: alter requests on an LP printer or plotter | see lp(1) |
| lpfilter(1): lpfilter | filters used by the lp interface scripts |
| lpstat(1): lpstat | print LP status information |
| ls(1): ls, l, lc, ll, lsf, lsr, lsx | list contents of directories |
| lsacl(1): lsacl | list access control lists (ACLs) of files |
| lsf: list contents of directories | see ls(1) |
| lsr: list contents of directories | see ls(1) |
| lsx: list contents of directories | see ls(1) |
| m4(1): m4 | macro processor |
| machid(1): hp9000s800,pdp11,u3b,u3b5,vax | provide truth value about processor type |
| machinfo(1): machinfo | print machine information |
| mail(1): mail, rmail | send mail to users or read mail |
| mailfrom(1): mailfrom | summarize mail folders by subject and sender |
| mailq(1): mailq | prints the mail queue |
| mailstats(1): mailstats | print mail traffic statistics |
| mailx(1): mailx | interactive message processing system |
| make(1): make | maintain, update, and regenerate groups of programs |
| makekey(1): makekey | generate encryption key |
| man(1): man | find manual information by keywords; print out a manual entry |
| mediainit(1): mediainit | initialize disk or cartridge tape media, partition DDS tape |
| merge(1): merge | three-way file merge |
| mesg(1): mesg | permit or deny messages to terminal |
| mkdir(1): mkdir | make a directory |
| mkfifo(1): mkfifo | make FIFO (named pipe) special files |
| mkmf(1): mkmf | make a makefile |
| mkmsgs(1): mkmsgs | create message files for use by gettxt() |
| mkstr(1): mkstr | extract error messages from C source into a file |
| mktemp(1): mktemp | make a name for a temporary file |
| mkuupath: manage the pathalias database | see uupath(1) |
| mm(1): mm, osdd | print/check documents formatted with the mm macros |
| model(1): model | print name of current HP-UX version |
| more(1): more, page | file perusal filter for crt viewing |
| mpsched(1): mpsched | control processor or locality domain on which a specific process executes |
| mt(1): mt | magnetic tape manipulating program |
| mv(1): mv | move or rename files and directories |
| named-checkconf(1): named-checkconf | named configuration file syntax checker |
| named-checkzone(1): named-checkzone | zone validity checking tool |
| neqn(1): neqn | format mathematical text for nroff |
| netstat(1): netstat | show network status |
| newalias(1): newalias | install new elm aliases for user or system |
| newform(1): newform | change or reformat a text file |
| newgrp: equivalent to exec newgrp | see cs(1) |
| newgrp: equivalent to exec newgrp | see ksh(1) |
| newgrp: equivalent to exec newgrp | see sh-posix(1) |
| newgrp(1): newgrp | switch to a new group |
| newmail(1): newmail | notify users of new mail in mailboxes |
| news(1): news | print news items |

Table of Contents

Volumes One and Two

| Entry Name(Section): name | Description |
|--|--|
| nice: alter command priority | see cs(1) |
| nice(1): nice | run a command at nondefault priority |
| nis+(1): nis+ | new version of the network information name service |
| niscat(1): niscat | display NIS+ tables and objects |
| nischgrp(1): nischgrp | change the group owner of an NIS+ object |
| nischmod(1): nischmod | change access rights on an NIS+ object |
| nischown(1): nischown | change the owner of an NIS+ object |
| nischttl(1): nischttl | change the time to live value of an NIS+ object |
| nisdefaults(1): nisdefaults | display NIS+ default values |
| niserror(1): niserror | display NIS+ error messages |
| nisgrep(1): nisgrep | utility for searching NIS+ tables |
| nisgrpadm(1): nisgrpadm | administer NIS+ groups |
| nisl(1): nisl | symbolically link NIS+ objects |
| nisl(1): nisl | list the contents of an NIS+ directory |
| nismatch(1): nismatch | utility for searching NIS+ tables |
| nismkdir(1): nismkdir | create NIS+ directories |
| nisspasswd(1): nisspasswd | change NIS+ password information |
| nism(1): nism | remove NIS+ objects from the namespace |
| nismdir(1): nismdir | remove NIS+ directories |
| nistbladm(1): nistbladm | administer NIS+ tables |
| nistest(1): nistest | return the state of the NIS+ namespace using a conditional expression |
| nl(1): nl | line numbering filter |
| nljust(1): nljust | justify lines, left or right, for printing |
| nm(1): nm | print name list of common object file |
| nohup: ignore hangups during command execution | see cs(1) |
| nohup(1): nohup | run a command immune to hangups |
| notify: notify user of change in job status | see cs(1) |
| nroff(1): nroff | format text |
| nslookup(1): nslookup | query name servers interactively |
| nsquery(1): nsquery | query the Name Service Switch backend libraries |
| nsupdate(1): nsupdate | Dynamic DNS update utility |
| od(1): od, xd | octal and hexadecimal dump |
| odump(1): odump | dump information contained in SOM object files |
| on(1): on | execute a command on a remote host; environment similar to local environment |
| onintr: specify shell's treatment of interrupts | see cs(1) |
| osdd: print/check documents formatted with the mm macros | see mm(1) |
| pack(1): pack, pcat, unpack | compress and expand files |
| page: file perusal filter for crt viewing | see more(1) |
| parstatus(1): parstatus | display information about the Superdome complex |
| passwd(1): passwd | change login password and associated attributes |
| paste(1): paste | merge same lines of several files or subsequent lines of one file |
| patch(1): patch | applying a diff file to an original |
| pathalias(1): pathalias | electronic address router |
| pathchk(1): pathchk | check path names |
| pax(1): pax | portable archive exchange |
| pcat: compress and expand files | see pack(1) |
| pd(1): pd | provide truth value about processor type |
| pd(1): pd | see machid(1) |
| pg(1): pg | file perusal filter for soft-copy terminals |
| piperm(1): piperm | remove a POSIX message queue, semaphore name |
| pipcs(1): pipcs | report status of interprocess communication facilities |
| popd: pop directory stack | see cs(1) |
| pppd(1): pppd | point to point protocol daemon |
| pr(1): pr | format and print files |
| praliases(1): praliases | print system-wide sendmail aliases |
| prealloc(1): prealloc | preallocate disk storage |
| primes: generate large prime numbers | see factor(1) |
| print: output from shell | see ksh(1) |
| print: output from shell | see sh-posix(1) |
| printenv(1): printenv | print out the environment |
| printf(1): echo | print formatted arguments |
| privatepw(1): privatepw | change WU-FTP Group Access File Information |

Table of Contents Volumes One and Two

| Entry Name(Section): name | Description |
|--|--|
| prmail(1): prmail | print out mail in the incoming mailbox file |
| prof(1): prof | display profile data |
| prs(1): prs | print and summarize an SCCS file |
| ps(1): ps | report process status |
| ptx(1): ptx | permuted index |
| pty : get the name of the pseudo-terminal | see tty(1) |
| pushd : push directory stack | see cs(1) |
| pwd : print current working directory | see ksh(1) |
| pwd : print current working directory | see sh-posix(1) |
| pwd(1): pwd | working directory name |
| pwget(1): pwget, grget | get password and group information |
| quota(1): quota | display disk usage and limits |
| ranlib(1): ranlib | regenerate archive symbol table |
| rcp(1): rcp | remote file copy |
| rccs(1): rccs | change RCS file attributes |
| rccsdiff(1): rccsdiff | compare RCS revisions |
| rccsmerge(1): rccsmerge | merge RCS revisions |
| rdist(1): rdist | remote file distribution |
| read : input and parse a line | see ksh(1) |
| read : input and parse a line | see sh-posix(1) |
| read(1): read | read a line from standard input |
| readmail(1): readmail | read mail from a mail folder or incoming mailbox |
| readonly : mark names as unmodifiable | see ksh(1) |
| readonly : mark names as unmodifiable | see sh-posix(1) |
| red : restricted line-oriented text editor | see ed(1) |
| rehash : recompute internal hash table | see cs(1) |
| remsh(1): remsh | execute from a remote shell |
| repeat : execute command more than once | see cs(1) |
| reset : terminal-dependent initialization | see tset(1) |
| return : shell function return to invoking script | see ksh(1) |
| return : shell function return to invoking script | see sh-posix(1) |
| rev(1): rev | reverse lines of a file |
| rksh : restricted Korn shell command programming language | see ksh(1) |
| rlog(1): rlog | print log messages and other information on RCS files |
| rlogin(1): rlogin | remote login |
| rm(1): rm | remove files or directories |
| rmail : send mail to users or read mail | see mail(1) |
| rmdel(1): rmdel | remove a delta from an SCCS file |
| rmdir(1): rmdir | remove directories |
| rmnl(1): rmnl | remove extra new-line characters from file |
| rndc(1): rndc | name server control utility |
| rndc-confgen(1): rndc-confgen | rndc key generation tool |
| rpcgen(1): rpcgen | an RPC protocol compiler |
| rsh : standard and restricted POSIX.2-conformant command shells | see sh-posix(1) |
| rtprio(1): rtprio | execute process with real-time priority |
| rtsched(1): rtsched | execute process with POSIX real-time priority |
| rup(1): rup | show host status of local machines (RPC version) |
| ruptime(1): ruptime | show status of local machines |
| rusers(1): rusers | determine who is logged onto machines on the local network |
| rwho(1): rwho | show who is logged in on local machines |
| sact(1): sact | print current SCCS file editing activity |
| samlog_viewer(1): samlog_viewer | tool for viewing and saving the SAM logfile |
| sccs(1): sccs | utility program for SCCS commands |
| sccsdiff(1): sccsdiff | compare two versions of an SCCS file |
| sccshelp(1): sccshelp | help for SCCS commands |
| script(1): script | make typescript of terminal session |
| sdiff(1): sdiff | side-by-side difference program |
| sed(1): sed | stream text editor |
| send_sound(1): send_sound | play audio file |
| serialize(1): serialize | force target process to run serially with other processes |
| set : set/define flags and arguments | see cs(1) |

Table of Contents

Volumes One and Two

| Entry Name(Section): name | Description |
|---|--|
| set: set/define options and arguments | see ksh(1) |
| set: set/define options and arguments | see sh-posix(1) |
| setacl(1): setacl | modify access control lists for files (JFS only) |
| setenv: define environment variable | see cs(1) |
| sffinger: utility programs for TCP Wrappers | see tryfrom(1) |
| sh(1): sh | overview of various system shells |
| sh-posix(1): sh, rsh | standard and restricted POSIX.2-conformant command shells |
| shar(1): shar | make a shell archive package |
| shift: shift <i>argv</i> members one position to left | see cs(1) |
| shift: shift <i>argv</i> members one position to left | see ksh(1) |
| shift: shift <i>argv</i> members one position to left | see sh-posix(1) |
| shl(1): shl | shell layer manager |
| size(1): size | print section sizes of object files |
| sleep(1): sleep | suspend execution for an interval |
| slp(1): slp | set printing options for a non-serial printer |
| soelim(1): soelim | eliminate .so's from nroff input |
| sort(1): sort | sort or merge files |
| source: define source for command input | see cs(1) |
| spell(1): spell, hashmake, spellin, hashcheck | find spelling errors |
| spellin: create compressed spelling list from hash codes | see spell(1) |
| split(1): split | split a file into pieces |
| ssp(1): ssp | remove multiple line-feeds from output |
| strings(1): strings | find the printable strings in an object or other binary file |
| strip(1): strip | strip symbol and line number information from an object file |
| stty(1): stty | set the options for a terminal port |
| su(1): su | switch user |
| sum(1): sum | print checksum and block count of a file |
| switch: define switch statement | see cs(1) |
| tabs(1): tabs | set tabs on a terminal |
| tail(1): tail | deliver the last part of a file |
| talk(1): talk | talk to another user |
| tar(1): tar | tape file archiver |
| tbl(1): tbl | format tables for nroff |
| tcpdchk(1): tcpdchk | check tcp wrapper configuration |
| tcpdmatch(1): tcpdmatch | evaluate tcp wrapper service requests |
| tee(1): tee | pipe fitting |
| telnet(1): telnet | user interface to the TELNET protocol |
| test: evaluate conditional expression | see cs(1) |
| test: evaluate conditional expression | see ksh(1) |
| test: evaluate conditional expression | see sh-posix(1) |
| test(1): test | condition evaluation command |
| tftp(1): tftp | trivial file transfer program |
| time, times: print summary of time used by processes | see ksh(1) |
| time: print summary of time used by shell and children | see cs(1) |
| time(1): time | time a command |
| time, times: print summary of time used by processes | see sh-posix(1) |
| timex(1): timex | time a command; report process data and system activity |
| top(1): top | display and update information about top processes on system |
| touch(1): touch | update access, modification, and/or change times of file |
| tput(1): tput | query terminfo database |
| tr(1): tr | translate characters |
| trap: trap specified signal | see ksh(1) |
| trap: trap specified signal | see sh-posix(1) |
| true(1): true, false | return zero or non-zero exit status |
| tryfrom(1): tryfrom, sffinger | utility programs for TCP Wrappers |
| tset(1): tset, reset | terminal-dependent initialization |
| tsm(1): tsm | Terminal Session Manager |
| tsm.command(1): tsm.command | send commands to Terminal Session Manager |
| tsm.info(1): tsm.info | get Terminal Session Manager state information |
| tsort(1): tsort | topological sort |
| tty(1): tty, pty | get the name of the terminal |

Table of Contents Volumes One and Two

| Entry Name(Section): name | Description |
|--|--|
| ttytype(1): ttytype | terminal identification program |
| typeset: control leading blanks and parameter handling | see ksh(1) |
| typeset: control leading blanks and parameter handling | see sh-posix(1) |
| u370: provide truth value about processor type | see machid(1) |
| u3b: provide truth value about processor type | see machid(1) |
| u3b10: provide truth value about processor type | see machid(1) |
| u3b2: provide truth value about processor type | see machid(1) |
| u3b5: provide truth value about processor type | see machid(1) |
| ul(1): ul | do underlining |
| ulimit: set size or time limits | see ksh(1) |
| ulimit: set size or time limits | see sh-posix(1) |
| umask: set permissions mask for creating new files | see cs(1) |
| umask: set permissions mask for creating new files | see ksh(1) |
| umask: set permissions mask for creating new files | see sh-posix(1) |
| umask(1): umask | set file-creation mode mask |
| umodem(1): umodem | XMODEM-protocol file transfer program |
| unalias: discard specified alias | see cs(1) |
| unalias: discard specified alias | see ksh(1) |
| unalias: discard specified alias | see sh-posix(1) |
| uname(1): uname | display information about computer system; set node name (system name) |
| uncompact: uncompact files | see compact(1) |
| uncompress: expand compressed data | see compress(1) |
| uncompressdir: expand compressed files in a directory | see compress(1) |
| unexpand: convert spaces to tabs | see expand(1) |
| unset(1): unset | undo a previous get of an SCCS file |
| unhash: disable use of internal hash tables | see cs(1) |
| unifdef(1): unifdef | remove preprocessor lines |
| uniq(1): uniq | report repeated lines in a file |
| units(1): units | conversion program |
| unpack: compress and expand files | see pack(1) |
| unset: remove definition/setting of flags and arguments | see cs(1) |
| unset: remove definition/setting of options and arguments | see ksh(1) |
| unset: remove definition/setting of options and arguments | see sh-posix(1) |
| unsetenv: remove variable from environment | see cs(1) |
| until: execute commands until expression is non-zero | see ksh(1) |
| until: execute commands until expression is nonzero | see sh-posix(1) |
| uptime(1): uptime, w | show how long system has been up |
| users(1): users | compact list of users who are on the system |
| uucp(1): uucp, uulog, uuname, uutry | UNIX system to UNIX system copy |
| uudecode: decode a file encoded by uuencode | see uuencode(1) |
| uuencode(1): uuencode, uudecode | encode/decode a binary file for transmission by mailer |
| uulog: access UUCP summary logs | see uucp(1) |
| uuname: list known UUCP systems | see uucp(1) |
| uupath(1): uupath, mkuupath | access and manage the pathalias database |
| uupick: accept or reject incoming UUCP messages | see uuto(1) |
| uustat(1): uustat | uucp status inquiry and job control |
| uuto(1): uuto, uupick | public UNIX system to UNIX system file copy |
| uutry: test for successful login to remote system | see uucp(1) |
| uux(1): uux | UNIX system to UNIX system command execution |
| ux2dos: convert ASCII file format | see dos2ux(1) |
| vacation(1): vacation | return "I am not here" indication |
| val(1): val | validate SCCS file |
| vax: provide truth value about processor type | see machid(1) |
| vc(1): vc | version control |
| vedit: beginner's screen-oriented text editor | see vi(1) |
| vi(1): vedit, vi, view | extended screen-oriented text editor |
| view: read-only screen-oriented text editor | see vi(1) |
| vis(1): vis, inv | make unprintable and non-ASCII characters in a file visible or invisible |
| vmstat(1): vmstat | report virtual memory statistics |
| vt(1): vt | log in on another system over lan |
| wait: wait for background processes | see cs(1) |

Table of Contents

Volumes One and Two

| Entry Name(Section): name | Description |
|---|--|
| wait: wait for child process | see ksh(1) |
| wait: wait for child process | see sh-posix(1) |
| wait(1): wait | await completion of process |
| wc(1): wc | count words, lines, and bytes or characters in a file |
| what(1): what | get SCCS identification information |
| whence: define interpretation of name as a command | see ksh(1) |
| whence: define interpretation of name as a command | see sh-posix(1) |
| whereis(1): whereis | locate source, binary, and/or manual for program |
| which(1): which | locate a program file including aliases and paths |
| while: execute commands while expression is non-zero | see cs(1) |
| while: execute commands while expression is non-zero | see ksh(1) |
| while: execute commands while expression is nonzero | see sh-posix(1) |
| who(1): who | who is currently logged in on the system |
| whoami(1): whoami | print effective current user id |
| whois(1): whois | Internet user name directory service |
| write(1): write | interactively write (talk) to another user |
| xargs(1): xargs | construct argument list(s) and execute command |
| xd: hexadecimal dump | see od(1) |
| xstr(1): xstr | extract strings from C programs to implement shared strings |
| yes(1): yes | be repetitively affirmative |
| ypcat(1): ypcat | print all Network Information Service map values |
| ypmatch(1): ypmatch | print values of selected keys in Network Information Service map |
| yppasswd(1): yppasswd | change login password in Network Information System (NIS) |
| ypwhich(1): ypwhich | list Network Information System server or map master |
| zcat: expand and cat data | see compress(1) |

Section 1
Part 2

User Commands
N-Z

Section 1

Part 2

User Commands

N-Z

NAME

named-checkconf - named configuration file syntax checking tool

SYNOPSIS

`/usr/sbin/named-checkconf [-t] directory [-v] [filename]`

DESCRIPTION

named-checkconf is a tool to check the syntax, but not semantics, of the configuration file for **named**.

Options

-t *directory*

chroot to *directory* so that **include** directives in the configuration file are processed as if run by a similarly chrooted **named**.

-v Print the version of the **named-checkconf** program and exit.

Operands

filename

The configuration file to be checked. If not specified, it defaults to `/etc/named.conf`.

RETURN VALUES

named-checkconf returns:

0 Success

1 Errors were detected.

AUTHOR

named-checkconf was developed by ISC (Internet Software Consortium).

SEE ALSO

named(1M)

RFC 1035

| n |

NAME

named-checkzone - zone validity checking tool

SYNOPSIS

```
/usr/sbin/named-checkzone [-c class] [-dq] zone [filename]
```

DESCRIPTION

named-checkzone is a tool for performing integrity checks on the zone contents. It uses the same integrity checks as **named**. It mainly checks for syntax errors and also the RR type.

Options

- c** *class*
Specify the class of the zone.
- d** Enable debugging.
- q** Enable quiet mode for exit code only.

Operands

filename

The file that should be used for checking the zone contents.

If *filename* is not specified, it defaults to the file with the zone name. For this to work, the file name for the zone in the **zone** directive of the **named.conf** file must be identical to the zone name.

zone The zone whose contents are to be checked.

RETURN VALUES

named-checkzone returns:

- 0 Success.
- 1 Errors were detected.

n**EXAMPLE**

```
$ named-checkzone -d text.domain db.domain
```

AUTHOR

named-checkzone was developed by ISC (Internet Software Consortium).

SEE ALSO

named(1M)
RFC 1035

NAME

neqn - format mathematical text for nroff

SYNOPSIS

neqn [-*dx*y] [-*sn*] [-*fn*] [-*pn*] [*file*]...

Remarks

The output of **neqn** is very device-dependent. See the *WARNINGS* section.

The examples shown approximate the results in a **dtterm** window. To see the actual results on an output device, send the output of the **man neqn** command directly to the device.

DESCRIPTION

neqn is a preprocessor for **nroff** (see *nroff*(1)) for typesetting mathematical text on typewriter-like terminals. Its invocation is almost always one of the following two forms or equivalent:

```
neqn file... | nroff | col
tbl file... | neqn | nroff | col
```

If no files are specified (or if - is specified instead of *file*), **neqn** reads from standard input. A line beginning with **.EQ** marks the start of an equation. The end of an equation is marked by a line beginning with **.EN**. Neither of these lines is altered, which means that they can be defined in **nroff** macro packages to get centering, numbering, etc.

Delimiters

It is also possible to designate two characters as *delimiters*; subsequent text between delimiters is then treated as **neqn** input. Delimiters can be set to characters *x* and *y* with the command-line argument **-dx**y or (more commonly) with the sequence

```
.EQ
delim xy
.EN
```

The left and right delimiters can be the same character; the dollar sign (\$) is often used as such a delimiter. Delimiters are turned off by **delim off** (see the *WARNINGS* section). All text that is neither between delimiters nor between **.EQ** and **.EN** is passed through untouched.

Separators and Metacharacters

Tokens within **neqn** equations are separated by spaces, tabs, newlines, braces, double quotes, tildes, and circumflexes. Braces ({ }) are used for grouping; generally speaking, anywhere a single character such as *x* can appear, a complicated construction enclosed in braces can be used instead. Tilde (~) represents a full space in the output; circumflex (^), half as much.

Subscripts and Superscripts

Subscripts and superscripts are produced using **sub** and **sup** as follows:

| Source Text | Result |
|----------------------------------|--------------------|
| x sub j | x j |
| a sub k sup 2 | 2 ak |
| e sup {x sup 2 + y sup 2} | 2+y2 ex |

Fractions

Fractions are produced by using **over**:

| Source Text | Result |
|-----------------------|---------------|
| <code>a over b</code> | $\frac{a}{b}$ |

Square Roots

sqrt produces square roots:

| Source Text | Result |
|--|------------------------------|
| <code>1 over sqrt {ax sup 2+bx+c}</code> | $\frac{1}{\sqrt{ax^2+bx+c}}$ |

Upper and Lower Limits

The keywords **from** and **to** specify lower and upper limits:

| Source Text | Result |
|--|--|
| <code>lim from {n -> inf } sum from 0 to n x sub i</code> | $\lim_{n \rightarrow \infty} \sum_{i=0}^n x_i$ |

Brackets and Braces

Left and right brackets, braces, and such, of proper height are made with **left** and **right**:

| Source Text | Result |
|--|---|
| <code>left [{x sup 2 + y sup 2} over alpha right] ~~= 1</code> | $\left[\frac{x^2+y^2}{\alpha} \right] = 1$ |

Legal characters after **left** and **right** are braces, brackets, bars, **c** and **f** for ceiling and floor, and "" for nothing at all (useful for a right-side-only bracket). A **left char** need not have a matching **right char**.

Vertical Piles

Vertical piles of elements are made with **pile**, **lpile**, **cpile**, and **rpile**:

| Source Text | Result |
|--|---|
| <code>pile {a above bb above ccc}</code> | $\begin{array}{c} a \\ bb \\ ccc \end{array}$ |

Piles can have arbitrary numbers of elements; **lpile** left aligns, **pile** and **cpile** center (but with different vertical spacing), and **rpile** right aligns.

| n |

Matrices and Determinants

Matrices are made with **matrix**:

| Source Text | Result |
|--|--|
| <pre>left { matrix { lcol { x sub i above y sub 2 } ccol { 1 above 234 } } } right </pre> | $\left \begin{array}{c} x_i \quad 1 \\ y_2 \quad 234 \end{array} \right $ |

In addition, there is **rcol** for a right-aligned column.

Diacritical Marks

Diacritical marks are made with **dot**, **dotdot**, **hat**, **tilde**, **bar**, **vec**, **dyad**, and **under**:

| Source Text | Result |
|---------------------------------------|---|
| <code>x dot = f(t) bar</code> | $\dot{\bar{x}} = f(t)$ |
| <code>y dotdot bar ~~~ n under</code> | $\ddot{\bar{y}} = \tilde{\underset{\sim}{n}}$ |
| <code>x vec ~~~ y dyad</code> | $\vec{x} = \vec{y}$ |

Point Sizes and Fonts

Point sizes and fonts can be changed with **size** *n* or **size +|-n**, **roman**, **italic**, **bold**, and **font** *n*.

Point sizes and fonts can be changed globally in a document by **gsize** *n* and **gfont** *n*, or by the command-line arguments **-sn** and **-fn**.

Normally, subscripts and superscripts are reduced by 3 points from the previous size; this can be changed by the command-line argument **-pn**.

Vertical Alignment

Successive display arguments can be lined up. Place **mark** before the desired lineup point in the first equation; place **lineup** at the place that is to line up vertically in subsequent equations.

Shorthand Forms

Shorthand forms can be defined or existing keywords redefined with **define**:

```
define thing % replacement %
```

defines a new token called *thing* that is replaced by *replacement* whenever it appears thereafter. The % can be any character that does not occur in *replacement*. The spaces around the % delimiters are required.

Other Keywords

Keywords such as **sum** (displayed as **R**, standing for uppercase sigma), **int** (^, integral sign), **inf** (oo, infinity sign), and shorthands such as **>=** (> overstruck by _), **!=** (= overstruck by /), and **->** (->) are recognized. Greek letters are spelled out in uppercase or lowercase as desired, as in **alpha** (displayed as **A**, standing for lowercase alpha) or **GAMMA** (displayed as **G**, standing for uppercase gamma). Mathematical words such as **sin** (**sin**), **cos** (**cos**), and **log** (**log**) are made Roman automatically. **nroff** four-character escapes, such as **\(dd** (| overstruck by =, standing for double dagger) and **\(bu** (o overstruck by +, standing for bullet), can be used anywhere.

| n |

Verbatim Text

Strings enclosed in double quotes ("*string*") are passed through untouched; this permits keywords to be entered as text, and can be used to communicate with **nroff** when other methods fail. Details are given in the manuals cited below.

Options

neqn accepts the following options:

- dxy** Define the characters *x* and *y* as the start and end in-text delimiter characters. See the *Delimiters* subsection.
- fn** Change the font number globally for the output equations in the document to *n*. See the *Point Sizes and Fonts* subsection.
- pn** Reduce subscript and superscript point sizes by *n* points from the normal size. See the *Point Sizes and Fonts* subsection.
- sn** Change the point size globally for the output equations in the document to *n*. See the *Point Sizes and Fonts* subsection.

Operands

file A file to be processed for **neqn** constructs. If no file is specified, **neqn** reads from standard input. If a *file* is specified as **-**, **neqn** reads from standard input at that point in the sequence of files.

EXTERNAL INFLUENCES**Environment Variables**

LC_CTYPE determines the interpretation of text as single- or multibyte characters.

LANG determines the language in which messages are displayed.

If **LC_CTYPE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **neqn** behaves as if all internationalization variables are set to "C". See *environ(5)*.

| **n** |**International Code Set Support**

Single- and multibyte character code sets are supported.

WARNINGS

To embolden digits, parentheses, etc., it is necessary to quote them, as in **bold "12.3"**. Also see the *WARNINGS* section in *nroff(1)*.

Good practice dictates that if a delimiter is specified in a file, the **delim off** directive should be included at the end of the file to prevent undesirable behavior when processing multiple files where a subsequent file may contain the delimiter character as part of regular text.

To properly display equations on terminal screens and other devices that do not support reverse line feeds, **nroff** output should be piped through **col** (see *col(1)*).

The display on devices that do not support partial line feeds is often difficult to understand; Greek characters and other symbols are often not well supported and can mismatched printing of bold words on the same line (see a printed version of the *Other Keywords* subsection above). Consider using "computer-program" coding instead.

SEE ALSO

col(1), *mm(1)*, *nroff(1)*, *tbl(1)*, *mm(5)*.

Typesetting Mathematics – User's Guide, by B.W. Kernighan and L.L. Cherry.

New Graphic Symbols for EQN and NEQN, by C. Scrocca.

NAME

netstat - show network status

SYNOPSIS

```
netstat [-an] [-f address-family] [system ]
netstat [-an] [-f address-family] [system [core ]] # PA-RISC only
netstat [-Mnrsv] [-f address-family] [-p protocol ] [system ]
netstat [-Mnrsv] [-f address-family] [-p protocol ] [system [core ]] # PA-RISC only
netstat [-ginw] [-I interface ] [interval ] [system ]
netstat [-ginw] [-I interface ] [interval ] [system [core ]] # PA-RISC only
```

DESCRIPTION

netstat displays statistics for network interfaces and protocols, as well as the contents of various network-related data structures. The output format varies according to the options selected. Some options are ignored or invalid when used in combination with other options.

Generally, the **netstat** command takes one of the three forms shown above:

- The first form of the command displays a list of active sockets for each protocol.
- The second form displays the contents of one of the other network data structures according to the option selected.
- The third form displays configuration information for each network interface. It also displays network traffic data on configured network interfaces, optionally updated at each *interval*, measured in seconds.

Options are interpreted as follows:

- a** Show the state of all sockets, including passive sockets used by server processes. When **netstat** is used without any options only active sockets are shown. This option does not show the state of X.25 programmatic access sockets. The option is ignored if the **-g**, **-i**, **-I**, **-M**, **-p**, **-r**, **-s** or *interval* option is specified.
- f *address-family*** Show statistics or address control block for only the specified *address-family*. The following address families are recognized: **inet** for **AF_INET**, **inet6** for **AF_INET6**, and **unix** for **AF_UNIX**. This option with **AF_UNIX** applies to the **-a** and **-s** options. This option with **AF_INET** or **AF_INET6** applies to the **-a**, **-i**, **-n**, and **-s** options.
- g** Show multicast information for network interfaces. Only the address family **AF_INET** is recognized by this option. This option may be combined with the **-i** option to display both kinds of information. The option is ignored if the **-p** option is specified.
- i** Show the state of network interfaces. Only the interfaces that have been configured with an IP address or the **plumb** option using the **ifconfig** command are shown. The output includes both the primary and logical interfaces. (See *ifconfig*(1M)). The counts for Ipkts and Opkts fields are for IP packets only. This option is ignored if the **-p** option is specified. By default, when the **-f** option is not specified, **netstat** shows the state of interfaces configured with either IPv4 or IPv6 addresses. An exception is when the user has not configured any IPv6 address on any interface, **netstat** does not display the state of the IPv6 loopback interface.
- I *interface*** Show information about the specified interface only. This option applies to the **-g** and **-i** options.
- M** Show the multicast routing tables. When **-s** is used with the **-M** option, **netstat** displays multicast routing statistics instead. This option is ignored if the **-p** option is specified.
- n** Show network addresses as numbers. Normally, **netstat** interprets addresses and attempts to display them symbolically. This option applies to the **-a**, **-i**,

| n |

- r** and **-v** options.
- p** *protocol* Show statistics for the specified protocol. The following protocols are recognized: **tcp**, **udp**, **ip**, **icmp**, **igmp**, **ipv6**, and **icmpv6**.
- r** Show the routing tables. When **-v** is used with the **-r** option, **netstat** also displays the network masks in the route entries. This option is ignored if the **-g**, **-i**, **-I**, **-p** or *interval* option is specified and is invalid if the **-s** option is specified.
- s** Show statistics for all protocols. When this option is used with the **-M** option, **netstat** displays multicast routing statistics instead. This option is ignored if the **-g**, **-i**, **-I**, **-p** or *interval* option is specified and is invalid if the **-r** option is specified.
- v** Show additional routing information. When **-v** is used with the **-r** option, **netstat** also displays the network masks in the route entries. This option only applies to the **-r** option.
- w** Show the output in wide format. This option displays all the fields in the output with their maximum resolution in a single line. Thus, the output can be worked upon with text-processing tools. This option works only with the **-i** option and will be ignored when used with any other option.

The *system* argument allows substitution for the default value **/stand/vmunix**. On PA-RISC systems only, the *core* argument allows substitution for the default value **/dev/kmem**.

If no options are specified, **netstat** displays the status of only active sockets. The display of active and passive sockets status shows the local and remote addresses, send and receive queue sizes (in bytes), protocol, and the internal state of the protocol.

Note: The send and receive queue size displayed is usually zero. These fields are displayed only for backward compatibility purposes.

Address formats are in two forms: *host .port*, or *network .port* if the host portion of a socket address is zero. When known, the host and network addresses are displayed symbolically by using **gethostbyname()** and **getnetbyname()**, respectively (see *gethostent(3N)* and *getnetent(3N)*) for IPv4, and **getnameinfo()** for IPv6 (see *getaddrinfo(3N)*). If a symbolic name for an address is unknown, the address is displayed numerically according to the address family. For more information regarding the Internet “dot format” for IPv4 addresses, refer to *inet(3N)*. For more information regarding the Internet “colon format” for IPv6 addresses, refer to *inet6(3N)*. Unspecified or “wildcard” addresses and ports appear as an asterisk (*).

The interface display provides a table of cumulative statistics regarding packets transferred, both inbound and outbound. The network addresses of the interface and the maximum transmission unit (MTU) are also displayed. When the *interval* argument is specified, **netstat** displays a running count of statistics related to network interfaces. This display consists of a column for the first interface found during auto-configuration and a column summarizing information for all interfaces. To display a running count of statistics for a specific interface, use the **-I** option. The first line of each screen of information contains a summary since the system was last rebooted. Subsequent lines of output show values accumulated over the preceding interval.

The routing table display indicates the available routes and their status. Each route consists of a destination host or network, a netmask and a gateway to use in forwarding packets. The **Flags** field shows whether the route is up (**U**), whether the route is to a gateway (**G**), or whether the route is a host or network route (with or without **H**).

The **Netmask** field shows the mask to be applied to the destination IP address of an IP packet to be forwarded. The result will be compared with the destination address in the route entry. If they are the same, then the route is one of the candidates for routing this IP packet. If there are several candidate routes, then the route with the longest **Netmask** field (contiguous 1's starting from the left-most bit position) will be chosen. (see *routing(7)*.)

The **Gateway** field shows the address of the immediate gateway for reaching the destination. It can be the address of the outgoing interface if the destination is on a directly connected network.

The **Interface** field identifies which network interface is used for the route.

The **Pmtu** field displays the path maximum transmission unit (PMTU). If the route is created with a static PMTU value (see *route(1M)*), the corresponding PMTU value permanently overrides the interface

MTU. Otherwise, the PMTU value is the same as the MTU of the network interface used for the route.

The **Prefix** field is for IPv6 only. Its format is similar to the CIDR notation in IPv4. The prefix is an integer between 0 and 128 inclusive. It specifies how many of the leftmost contiguous bits of the address comprise the prefix. A host route has a prefix of 128. A default route has a prefix of 0 (see *route*(1M)). The prefix is also used in selecting a route to forward an IPv6 packet.

DEPENDENCIES

X.25

-a option does not list X.25 programmatic access information.

AUTHOR

netstat was developed by HP and the University of California, Berkeley.

SEE ALSO

ifconfig(1M), lanscan(1M), lanadmin(1M), route(1M), inet(3N), inet6(3N), gethostent(3N), getnetent(3N), getaddrinfo(3N), hosts(4), networks(4), protocols(4), services(4), routing(7).

| **n** |

NAME

newalias - install new elm aliases for user or system

SYNOPSIS

newalias [-g]

Remarks

newalias replaces the former functionality of the **elmaliases** command.

DESCRIPTION

The **newalias** command creates new alias database files from an alias text file for use by **elm** and other programs. For user aliases, this functionality can also be performed from the Alias Menu of the **elm** program (see *elm*(1)).

Options

newalias recognizes the following option:

-g Global. The program updates the system alias files instead of a user's alias files.

Operation

Without the **-g** option, **newalias** updates a user's alias files, based on an input file named

`$HOME/.elm/aliases.text`

Upon finding the file, it creates the output files named

`$HOME/.elm/aliases`
`$HOME/.elm/aliases.dir`
`$HOME/.elm/aliases.pag`

With the **-g** option, **newalias** updates the system alias files, based on an input file named

`/var/mail/.elm/aliases.text`

Upon finding the file, it creates the output files named

`/var/mail/.elm/aliases`
`/var/mail/.elm/aliases.dir`
`/var/mail/.elm/aliases.pag`

In either case, you need read access to the **aliases.text** file and write access to the other files and the **.elm** directory.

Text File Entries

Each entry in either **aliases.text** file is expected to be in the following format:

alias-list = [*lastname* [*;* *firstname*]] [*,* *comment*] = *address-list*

Field Names

The field names are defined as follows:

address-list A blank- or comma-separated list of one or more mail addresses, personal alias names, and/or group alias names.

In practice, each item is tested first as an alias name. If it is not an alias name, it is assumed to be a mail address. A mail address can be in Internet form (**user@host.domain**), in UUCP form (**host.domain!user**), or in **sendmail** alias form (see *sendmail*(1M)). It can also be the name of a local mail user, which is appended with the local host name in Internet form.

alias-list A blank- or comma-separated list of alias names. Each name identifies the same alias entry. An alias name can be made up of letters (**A-Z**, **a-z**), digits (**0-9**), underscores (**_**), dashes (**-**), and periods (**.**). Alias names are not case-sensitive, so **dave** and **Dave** are equivalent.

comment A string containing any information you wish about the entry, such as location and phone numbers. It is displayed in the Alias Menu of the **elm** program, but **elm** does not transmit it in a mail message. This field can contain any characters except an unquoted equal sign (=). See the Quoting Characters subsection.

- firstname* The first name of the person (or group). It is combined with *lastname* to form the *fullname*. This field can contain any characters except an unquoted equal sign (=) or an unquoted comma (,). See the Quoting Characters subsection.
- The only first name under the Personal or Group Aliases subheading below is: **John** in **Smith; John**.
- lastname* The last name of the person (or group). It is combined with *firstname* to form the *fullname*. This field can contain any characters except an unquoted equal sign (=), an unquoted semicolon (;), or an unquoted comma (,). See the Quoting Characters subsection.
- The last names under the Personal or Group Aliases subheading below are: **Dave Taylor, Smith, Unix Gurus, and Unix people**.
- fullname* The combination of *firstname lastname*. It is usually sent in a mail header in parentheses after the address. It is also displayed in the Alias Menu of the **elm** program and by the **elmaliases** command (see *elm*(1) and *elmaliases*(1)).

Delimiters

The delimiters have the following precedence:

- The first and second equal signs (=) mark the end of the *alias-list* and the beginning of the *address-list*, respectively. Both equal signs are required.
- The first comma (,) after the first equal sign and before the second equal sign marks the beginning of the *comment* field.
- The first semicolon (;) after the first equal sign and before the next comma or second equal sign marks the beginning of the *firstname* field.

Personal or Group Aliases

A personal or individual alias has only one address in *address-list*, as in:

```
dave, taylor = Dave Taylor = taylor@company.com
j_smith = Smith; John, 408-555-1212 =johns@pocahontas.gov
```

A group alias has two or more addresses in *address-list*, as in:

```
gurus = Unix Gurus = alan, john, dave, mike,
      richard, larry, t_richardson
unix = Unix people = gurus, taylor, jonboy
```

Other Rules

Entries can be continued over several lines; the continuation lines must start with a blank (a space or tab).

A comment is any line starting with a number sign (#). It is ignored.

Blank lines and comments can be interspersed within entries.

Quoting Characters

You can include normally excluded characters in *firstname*, *lastname*, *comment*, and mail addresses in *address-list* by escaping each character with a backslash (\) or by enclosing the string in quotation marks ("). To include a quotation mark or a backslash, escape it with a backslash, whether inside or outside quotation marks.

FILES

| | |
|--|---------------------------------------|
| <code>\$HOME/.elm/aliases</code> | User alias database data table |
| <code>\$HOME/.elm/aliases.dir</code> | User alias database directory table |
| <code>\$HOME/.elm/aliases.pag</code> | User alias database hash table |
| <code>\$HOME/.elm/aliases.text</code> | User alias source text |
| <code>/var/mail/.elm/aliases</code> | System alias database data table |
| <code>/var/mail/.elm/aliases.dir</code> | System alias database directory table |
| <code>/var/mail/.elm/aliases.pag</code> | System alias database hash table |
| <code>/var/mail/.elm/aliases.text</code> | System alias source text |

AUTHOR

newalias was developed by HP.

SEE ALSO

elm(1), elmalias(1), mail(1), mailx(1).

| **n** |

NAME

newform - change or reformat a text file

SYNOPSIS

```
newform [-i tabspec] [-o tabspec] [-l n] [-b n] [-e n] [-c char] [-p n] [-a n] [-f] [-s]
[files]
```

DESCRIPTION

newform reads lines from the named *files*, or the standard input if no input file is named, and reproduces the lines on the standard output. Lines are reformatted in accordance with command line options in effect.

Except for **-s**, command line options can appear in any order, can be repeated, and can be intermingled with the optional *files*. Command line options are processed in the order specified. This means that option sequences such as **-e 15 -l 60** yield results different from **-l 60 -e 15**. Options are applied to all *files* on the command line.

Options

newform recognizes the following options:

- i***tabspec* Input tab specification: expands tabs to spaces, according to the tab specifications given. *Tabspec* recognizes all tab specification forms described in *tabs*(1). In addition, *tabspec* can be **--**, in which **newform** assumes that the tab specification is to be found in the first line read from the standard input (see *fspec*(4)). If no *tabspec* is given, *tabspec* defaults to **-8**. A *tabspec* of **-0** expects no tabs; if any are found, they are treated as **-1**.
- o***tabspec* Output tab specification: replaces spaces with tabs, according to the tab specifications given. The tab specifications are the same as for **-i***tabspec*. If no *tabspec* is given, *tabspec* defaults to **-8**. A *tabspec* of **-0** means that no spaces will be converted to tabs on output.
- l***n* Set the effective line length to *n* characters. If *n* is not entered, **-l** defaults to 72. The default line length without the **-l** option is 80 characters. Note that tabs and backspaces are treated as single characters (use **-i** to expand tabs to spaces).
- b***n* Truncate *n* characters from the beginning of the line when the line length is greater than the effective line length (see **-l***n*). Default is to truncate the number of characters necessary to obtain the effective line length. The default value is used when **-b** with no *n* is used. This option can be used to delete the sequence numbers from a COBOL program as follows:


```
newform -l1 -b7 file-name
```

The **-l1** must be used to set the effective line length shorter than any existing line in the file so that the **-b** option is activated.
- e***n* Same as **-b***n* except that characters are truncated from the end of the line.
- c***k* Change the prefix/append character to *k*. Default character for *k* is a space.
- p***n* Prefix *n* characters (see **-c***k*) to the beginning of a line when the line length is less than the effective line length. Default is to prefix the number of characters necessary to obtain the effective line length.
- a***n* Same as **-p***n* except characters are appended to the end of a line.
- f** Write the tab specification format line on the standard output before any other lines are output. The tab specification format line which is printed will correspond to the format specified in the *last* **-o** option. If no **-o** option is specified, the line which is printed contains the default specification of **-8**.
- s** Shears off leading characters on each line up to the first tab and places up to 8 of the sheared characters at the end of the line. If more than 8 characters (not counting the first tab) are sheared, the eighth character is replaced by a ***** and any characters to the right of it are discarded. The first tab is always discarded.

An error message and program exit occur if this option is used on a file without a tab on each line. The characters sheared off are saved internally until all other options specified are applied to that line. The characters are then added at the end

of the processed line.

For example, to convert a file with leading digits, one or more tabs, and text on each line, to a file beginning with the text, all tabs after the first expanded to spaces, padded with spaces out to column 72 (or truncated to column 72), and the leading digits placed starting at column 73, the command would be:

```
newform -s -i -l -a -e file-name
```

RETURN VALUE

newform returns one of the following values upon completion:

- 0** No errors encountered.
- 1** An error occurred.

DIAGNOSTICS

All diagnostics are fatal.

usage: ...

newform was called with a bad option.

not -s format

There was no tab on one line.

can't open file

Self-explanatory.

internal line too long

A line exceeds 512 characters after being expanded in the internal work buffer.

tabspec in error

A tab specification is incorrectly formatted, or specified tab stops are not ascending.

tabspec indirection illegal

A *tabspec* read from a file (or standard input) must not contain a *tabspec* referencing another file (or standard input).

n

WARNINGS

newform normally only keeps track of physical characters; however, for the **-i** and **-o** options, **newform** keeps track of backspaces in order to line up tabs in the appropriate logical columns.

newform does not prompt the user if a *tabspec* is to be read from the standard input (by use of **-i--** or **-o--**).

If the **-f** option is used, and the last **-o** option specified was **-o--**, and was preceded by either a **-o--** or a **-i--**, the tab specification format line will be incorrect.

SEE ALSO

fspec(4), csplit(1), tabs(1).

NAME

newgrp - switch to a new group

SYNOPSIS

newgrp [-] [*group*]

DESCRIPTION

The **newgrp** command changes your group ID without changing your user ID and replaces your current shell with a new one.

If you specify *group*, the change is successful if *group* exists and either your user ID is a member of the new *group*, or *group* has a password and you can supply it from the terminal.

If you omit *group*, **newgroup** changes to the group specified in your entry in the password file, */etc/passwd*.

Whether the group is changed successfully or not, or the new group is the same as the old one or not, **newgrp** proceeds to replace your current shell with the one specified in the shell field of your password file entry. If that field is empty, **newgrp** uses the POSIX shell, */usr/bin/sh* (see *sh-posix(1)*).

If you specify - (hyphen) as the first argument, the new shell starts up as if you had just logged in. If you omit -, the new shell starts up as if you had invoked it as a subshell.

You remain logged in and the current directory is unchanged, but calculations of access permissions to files are performed with respect to the new real and effective group IDs.

Exported variables retain their values and are passed to the new shell. All unexported variables are deleted, but the new shell may reset them to default values.

Since the current process is replaced when the new shell is started, exiting from the new shell has the same effect as exiting from the shell in which **newgrp** was executed.

EXTERNAL INFLUENCES**International Code Set Support**

Characters from the 7-bit USASCII code set are supported in group names (see *ascii(5)*).

| n |

DIAGNOSTICS

The **newgrp** command issues the following error messages:

| | |
|--------------------------|---|
| Sorry | Your user ID does not qualify as a group member. |
| Unknown group | The group name does not exist in <i>/etc/group</i> . |
| Permission denied | If a password is required, it must come from a terminal. |
| You have no shell | Standard input is not a terminal file, causing the new shell to fail. |

EXAMPLES

To change from your current group to group **users** without executing the login routines:

```
newgrp users
```

To change from your current group to group **users** and execute the login routines:

```
newgrp - users
```

WARNINGS

There is no convenient way to enter a password into */etc/group*.

The use of group passwords is not recommended because, by their very nature, they encourage poor security practices. Group passwords may be eliminated in future HP-UX releases.

If the specified group to **newgrp** has multiple inconsistent entries (i.e. the group id or/and password are different) in the group database, **newgrp** will consider the group id and password of the first matched group entry as the correct group id and password for the group.

FILES

| | |
|--------------------|----------------------|
| <i>/etc/group</i> | System group file |
| <i>/etc/passwd</i> | System password file |

SEE ALSO

csh(1), ksh(1), login(1), sh-posix(1), group(4), passwd(4), environ(5).

STANDARDS CONFORMANCE

newgrp: SVID2, SVID3, XPG2, XPG3, XPG4

| **n** |

NAME

newmail - notify users of new mail in mailboxes

SYNOPSIS

```
newmail [-i interval] [-w] [file-spec]...
```

DESCRIPTION

The **newmail** utility monitors your incoming mailbox or specified mail folders.

The basic operation is that the program checks the folders each *interval* seconds (default 60) and lists any new mail that has arrived in any of the mailboxes, indicating the sender's name, and the subject of the message.

Without any options, **newmail** runs in the background at a default interval of 60 seconds to monitor the user's incoming mailbox. So that they are suitable for display on an already active screen, messages are prefixed with a pair of pointer characters as follows:

```
>> Mail from sender-name - subject-of-message
>> Priority sender-name - subject-of-message
```

If there is no subject, the message (**No Subject Specified**) is displayed. If there is more than one folder, output lines are prefixed by the *folder-name* or the prefix string specified by *file-spec*.

newmail runs until you log out or explicitly kill it. It can internally reset itself if the mailbox shrinks in size and then grows again.

Options

newmail recognizes the following options:

-i *interval* Set the time interval between mailbox checks to the value specified, in seconds. The default is 60.

interval must be less than 2^{32} seconds. If it is set to less than 10 seconds, **newmail** warns that such short intervals are not recommended.

-w Run the program within the current window in the foreground with a more succinct output format.

The output formats become:

```
Mail from sender-name - subject-of-message
Priority sender-name - subject-of-message
```

Operands

newmail recognizes the following operand:

file-spec Specifies the name of a folder and an optional prefix string, in the form:

```
foldername[=prefix-string]
```

Metacharacters such as +, =, and % indicate the folder directory. The default is the value of the environment variable **MAILDIR** or **\$HOME/Mail**.

EXAMPLES

Check incoming mailbox every 60 seconds:

```
newmail
```

Check incoming mailboxes of **joe** and **root** every 15 seconds for new messages.

```
newmail -i 15 joe root
```

Monitor the incoming mailbox of user **mary** and the folder in your mail directory called **postmaster**. Prefix all new messages in the incoming mailbox of **mary** with the string **Mary**, and the new messages in the folder **postmaster** with **POBOX**. Also, monitor folder **/tmp/mbox**:

```
newmail "mary=Mary" +postmaster=POBOX /tmp/mbox
```

AUTHOR

newmail was developed by HP.

NAME

news - print news items

SYNOPSIS

news [-a] [-n] [-s] [*items*]

DESCRIPTION

news is used to keep the user informed of current events. By convention, these events are described by files in the directory **/var/news**.

When invoked without arguments, **news** prints the contents of all current files in **/var/news**, most recent first, with each preceded by an appropriate header. **news** stores the “currency” time as the modification date of a file named **.news_time** in the user’s home directory (the identity of this directory is determined by the environment variable **\$HOME**); only files more recent than this currency time are considered “current.”

Options

news recognizes the following options:

- a** Print all items, regardless of currency. The stored time is not changed.
- n** Report the names of the current items without printing their contents, and without changing the stored time.
- s** Report how many current items exist without printing their names or contents, and without changing the stored time. It is useful to include such an invocation of *news* in one’s **.profile** file, or in the system’s **/etc/profile**.

All other arguments are assumed to be specific news items that are to be printed.

If an interrupt is typed during the printing of a news item, printing stops and the next item is started. Another interrupt within one second of the first causes the program to terminate.

EXTERNAL INFLUENCES**International Code Set Support**

Single- and multi-byte character code sets are supported.

FILES

/var/news/*
\$HOME/.news_time
/etc/profile

SEE ALSO

mail(1), profile(4), environ(5).

STANDARDS CONFORMANCE

news: SVID2, SVID3, XPG2

NAME

nice - run a command at nondefault priority

SYNOPSIS

nice [-*priority_change*] *command* [*command_args*]

nice [-**n** *priority_change*] *command* [*command_args*]

DESCRIPTION

The **nice** command executes *command* at a nondefault CPU scheduling priority. (The name is derived from being "nice" to other system users by running large programs at lower priority.)

Arguments

The command-line arguments are as follows:

-n *priority_change*

priority_change The difference between the system nice value (relative priority) of the current (or parent) process and the actual system nice value at which *command* is to run.

An unsigned value increases the system nice value for *command*, causing it to run at lower priority.

A negative value requires superuser privileges, and assigns a lower system nice value (higher priority) to *command*. If the current process is not privileged, the value is silently treated as if it were 0.

If the value of *priority_change* would result in a system nice value outside the range 0 through 39, the corresponding limit value of 0 or 39 is used instead.

Note that a positive *priority_change* (lower priority) has a single - option character before the numeric value; a negative (higher priority) *priority_change* has two: the option character followed by the minus sign (--). If *-priority_change* is not specified, it defaults to **10**.

command A program, HP-UX command, user shell script, etc. to be executed at the nondefault priority. *command* can be run as a foreground or background process.

If *command* is run as a background process, any nice *priority_change* made by the shell (**ksh** executes all background processes via **nice -4**) is in addition to that specified in the **nice** command line.

command_args Any arguments recognized by *command*.

Process Priorities

All processes have an associated system nice value which is used to compute the instantaneous-priority of the process when it is scheduled to run. Normally, all processes inherit the system nice value of their parent process when they are spawned. The shell (**sh**, **cs****sh**, **ksh**, etc.) can create a child process with a different priority from the current shell process by spawning the child process via the **nice** command. If the *priority_change* value is unsigned (positive), the child process is nicer (lower in priority) relative to the parent. If the *priority_change* value is negative, the child process runs at a higher priority with a greater share of available system resources. To spawn a higher priority child process, the parent process must be owned by a user who has the appropriate privileges.

At boot-up, the system starts the **init** process at a system nice value of 20 (system default). On most systems, all processes (down to the login shells) inherit this priority. Starting from their individual login shell processes, users can alter the system nice value of descendent processes to as much as 39, or, with appropriate privileges, as little as 0. A system nice value of 0 establishes an extremely high priority, whereas a value of 39 indicates a very low priority.

Ordinary users can only increase the system nice value of any child process relative to the current process; i.e., *priority_change* must be a positive (unsigned) value, resulting in a lower priority. To start a child process at a lower system nice value (higher priority) than the current process, the user must have the appropriate privileges, regardless of the relative nice-priority value desired.

For example, using the command

```
nice ksh
```

from a login shell whose current nice value is 20 spawns a subshell with a system nice value of 30. Attempting to use

```
nice --2 ksh
```

from the new shell to spawn another subshell whose system nice value would be 28, is rejected (unless the user has appropriate privileges), even though the resulting system nice value would be less than the priority of the original login shell process.

The system nice value for current processes is listed under the **NI** column produced by the **ps -l** command (see *ps(1)*).

Background Processes

Foreground processes are run at same system nice value as the parent shell. Background processes spawned by **ksh** run at the equivalent of a **nice -4** by default. If a background process is started via **nice** from **ksh**, any *priority_change* specified in the **nice** command is added to default **nice -4**. Thus the command

```
nice 12 command &
```

runs at a system nice value of 36 if executed from **ksh**.

EXTERNAL INFLUENCES

Environment Variables

LC_MESSAGES determines the language in which messages are displayed.

If **LC_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **nice** behaves as if all internationalization variables are set to "C". See *environ(5)*.

International Code Set Support

Single- and multi-byte character code sets are supported.

| n |

RETURN VALUE

nice returns the value returned by *command*.

EXAMPLES

The following examples assume the current process is running with a system nice value of 20 and **nice** is executed from the Korn shell (see *ksh(1)*).

Run a program named **prog** in the current directory at the default *priority_change* of 10 (system nice value of 30):

```
nice ./prog prog_args
```

Run the same program in the background using a system nice value of 36 (*priority_change*=12 plus 4 for the Korn shell):

```
nice -12 ./prog prog_args &
```

As a user with appropriate privileges, run **prog** as a foreground process with a system nice value of 6:

```
nice --14 ./prog prog_args
```

WARNINGS

The C shell, **cs**, has a built-in **nice** command with different syntax. See *cs(1)* for details.

SEE ALSO

cs(1), *ksh(1)*, *nohup(1)*, *sh-posix(1)*, *sh(1)*, *renice(1M)*, *nice(2)*.

STANDARDS CONFORMANCE

nice: SVID2, SVID3, XPG4

NAME

nis+, NIS+, nis - a new version of the network information name service

DESCRIPTION

NIS+ is a new version of the network information name service. This version differs in several significant ways from version 2, which is referred to as NIS or YP in earlier releases. Specific areas of enhancement include the ability to scale to larger networks, security, and the administration of the service.

The man pages for NIS+ are broken up into three basic categories. Those in section 1 are the user commands that are most often executed from a shell script or directly from the command line. Section 1M man pages describe utility commands that can be used by the network administrator to administer the service itself. The NIS+ programming API is described by man pages in section 3N.

All commands and functions that use NIS version 2 are prefixed by the letters **yp** as in *ypmatch(1)*, *ypcat(1)*, *yp_match(3C)*, and *yp_first(3C)*. Commands and functions that use the new replacement software NIS+ are prefixed by the letters **nis** as in *nismatch(1)*, *nischown(1)*, *nis_list(3N)*, and *nis_add_entry(3N)*. A complete list of NIS+ commands is in the *LIST OF COMMANDS* section.

This man page introduces the NIS+ terminology. It also describes the NIS+ namespace, authentication, and authorization policies.

NIS+ NAMESPACE

The naming model of NIS+ is based upon a tree structure. Each node in the tree corresponds to an NIS+ object. There are six types of NIS+ objects: *directory*, *table*, *group*, *link*, *entry*, and *private*.

NIS+ Directory Object

Each NIS+ namespace will have at least one NIS+ directory object. An NIS+ directory is like a UNIX file system directory which contains other NIS+ objects including NIS+ directories. The NIS+ directory that forms the root of the NIS+ namespace is called the root directory. There are two special NIS+ directories: **org_dir** and **groups_dir**. The **org_dir** directory consists of all the system-wide administration tables, such as **passwd**, **hosts**, and **mail_aliases**. The **groups_dir** directory consists of NIS+ group objects which are used for access control. The collection of **org_dir**, **groups_dir** and their parent directory is referred to as an NIS+ domain. NIS+ directories can be arranged in a tree-like structure so that the NIS+ namespace can match the organizational or administrative hierarchy.

| n |

NIS+ Table Object

NIS+ tables (not files), contained within NIS+ directories, store the actual information about some particular type. For example, the **hosts** system table stores information about the IP address of the hosts in that domain. NIS+ tables are multicolumn and the tables can be searched through any of the searchable columns. Each table object defines the schema for its table. The NIS+ tables consist of NIS+ entry objects. For each entry in the NIS+ table, there is an NIS+ entry object. NIS+ entry objects conform to the schema defined by the NIS+ table object.

NIS+ Group Object

NIS+ group objects are used for access control at group granularity. NIS+ group objects, contained within the **groups_dir** directory of a domain, contain a list of all the NIS+ principals within a certain NIS+ group. An NIS+ principal is a user or a machine making NIS+ requests.

NIS+ Link Object

NIS+ link objects are like UNIX symbolic file-system links—they are typically used for shortcuts in the NIS+ namespace.

Refer to *nis_objects(3N)* for more information about the NIS+ objects.

NIS+ NAMES

The NIS+ service defines two forms of names, **simple** names and **indexed** names. Simple names are used by the service to identify NIS+ objects contained within the NIS+ namespace. Indexed names are used to identify NIS+ entries contained within NIS+ tables. Furthermore, entries within NIS+ tables are returned to the caller as NIS+ objects of type *entry*. NIS+ objects are implemented as a union structure which is described in the file `<rpcsvc/nis_object.h>`. The differences between the various types and the meanings of the components of these objects are described in *nis_objects(3N)*.

Simple Names

Simple names consist of a series of labels that are separated by the '.'(dot) character. Each label is composed of printable characters from the ISO Latin 1 set. Each label can be of any nonzero length, provided that the fully qualified name is fewer than **NIS_MAXNAMELEN** octets including the separating dots. (See `<rpcsvc/nis.h>` for the actual value of **NIS_MAXNAMELEN** in the current release.) Labels that contain special characters (see *Grammar*) must be quoted.

The NIS+ namespace is organized as a singly rooted tree. Simple names identify nodes within this tree. These names are constructed such that the leftmost label in a name identifies the leaf node and all of the labels to the right of the leaf identify that object's parent node. The parent node is referred to as the leaf's *directory*. This is a naming directory and should not be confused with a file system directory.

For example, the name *example.simple.name.* is a simple name with three labels, where *example* is the leaf node in this name, the directory of this leaf is *simple.name.* which by itself is a simple name. The leaf of which is *simple* and its directory is simply *name.*

The function *nis_leaf_of(3N)* returns the first label of a simple name. The function *nis_domain_of(3N)* returns the name of the directory that contains the leaf. Iterative use of these two functions can break a simple name into each of its label components.

The name '.' (dot) is reserved to name the *global root* of the namespace. For systems that are connected to the Internet, this global root will be served by a Domain Name Service. When an NIS+ server is serving a root directory whose name is not '.'(dot), this directory is referred to as a *local root*.

NIS+ names are said to be *fully qualified* when the name includes all of the labels identifying all of the directories, up to the global root. Names without the trailing dot are called *partially* qualified.

Indexed Names

Indexed names are compound names that are composed of a search criterion and a simple name. The search criterion component is used to select entries from a table; the simple name component is used to identify the NIS+ table that is to be searched. The search criterion is a series of column names and their desired values enclosed in bracket '[']' characters. These criteria take the following form:

```
[column_name=value , column_name=value , ... ]
```

A search criterion is combined with a simple name to form an indexed name by concatenating the two parts, separated by a ','(comma) character as follows.

```
[ search-criterion ],table.directory.
```

When multiple column name/value pairs are present in the search criterion, only those entries in the table that have the appropriate value in all columns specified are returned. When no column name/value pairs are specified in the search criterion, [], *all* entries in the table are returned.

Grammar

The following text represents a context-free grammar that defines the set of legal NIS+ names. The terminals in this grammar are the characters '.' (dot), '[' (open bracket), ']' (close bracket), ',' (comma), '=' (equals) and whitespace. Angle brackets ('<' and '>'), which delineate non-terminals, are not part of the grammar. The character '|' (vertical bar) is used to separate alternate productions and should be read as "this production *OR* this production".

```
name           ::= . | <simple name> | <indexed name>
simple name     ::= <string>. | <string>.<simple name>
indexed name   ::= <search criterion>,<simple name>
search criterion ::= [ <attribute list> ]
attribute list ::= <attribute> | <attribute>,<attribute list>
attribute      ::= <string> = <string>
string         ::= ISO Latin 1 character set except the
                  character '/' (slash). The initial character
                  may not be a terminal character or the
                  characters '@' (at), '+' (plus), or '-'
                  hyphen.
```

Terminals that appear in strings must be quoted with "" (double quote). The "" character may be quoted by quoting it with itself "" "".

Name Expansion

The NIS+ service only accepts fully qualified names. However, since such names may be unwieldy, the NIS+ commands in section 1 employ a set of standard expansion rules that will attempt to fully qualify a partially qualified name. This expansion is actually done by the NIS+ library function *nis_getnames*(3N) which generates a list of names using the default NIS+ directory search path or the **NIS_PATH** environment variable. The default NIS+ directory search path includes all the names in its path. *nis_getnames*() is invoked by the functions *nis_lookup*(3N) and *nis_list*(3N) when the **EXPAND_NAME** flag is used.

The **NIS_PATH** environment variable contains an ordered list of simple names. The names are separated by the ':' (colon) character. If any name in the list contains colons, the colon should be quoted as described in the *Grammar* section. When the list is exhausted, the resolution function returns the error **NIS_NOTFOUND**. This may mask the fact that the name existed but a server for it was unreachable. If the name presented to the list or lookup interface is fully qualified, the **EXPAND_NAME** flag is ignored.

In the list of names from the **NIS_PATH** environment variable, the '\$' (dollar sign) character is treated specially. Simple names that end with the label '\$' have this character replaced by the default directory (see *nis_local_directory*(3N)). Using "\$" as a name in this list results in this name being replaced by the list of directories between the default directory and the global root that contain at least two labels.

Below is an example of this expansion. Given the default directory of *some.long.domain.name.*, and the **NIS_PATH** variable set to **fred.bar.:org_dir.\$:\$**. This path is initially broken up into the list:

- 1 **fred.bar.**
- 2 **org_dir.\$**
- 3 **\$**

The dollar sign in the second component is replaced by the default directory. The dollar sign in the third component is replaced with the names of the directories between the default directory and the global root that have at least two labels in them. The effective path value becomes:

- 1 **fred.bar.**
- 2a **org_dir.some.long.domain.name.**
- 3a **some.long.domain.name.**
- 3b **long.domain.name.**
- 3c **domain.name.**

Each of these simple names is appended to the partially qualified name that was passed to the *nis_lookup*(3N) or *nis_list*(3N) interface. Each is tried in turn until **NIS_SUCCESS** is returned or the list is exhausted.

If the **NIS_PATH** variable is not set, the path "\$" is used.

The library function *nis_getnames*(3N) can be called from user programs to generate the list of names that would be attempted. The program *nisdefaults*(1) with the **-s** option can also be used to show the fully expanded path.

Concatenation Path

Normally all the entries for a certain type of information are stored within the table itself. However, there are times when it is desirable for the table to point to other tables where entries can be found. For example, you may want to store all the IP addresses in the host table for their own domain, and yet want to be able to resolve hosts in some other domain without explicitly specifying the new domain name. NIS+ provides a mechanism for concatenating different but related tables with an "NIS+ Concatenation Path". With a concatenation path, you can create a sort of flat namespace from a hierarchical structure. You can also create a table with no entries and just point the hosts or any other table to its parent domain. Note that with such a setup, you are moving the administrative burden of managing the tables to the parent domain. The concatenation path will slow down the request response time because more tables and more servers are searched. It will also decrease the availability if all the servers are incapacitated for a particular directory in the table path.

The NIS+ Concatenation Path is also referred to as the "table path". This path is set up at table creation time through *nistbladm*(1). You can specify more than one table to be concatenated and they will be searched in the given order. Note that the NIS+ client libraries, by default, will not follow the concatena-

tion path set in site-specific tables. Refer to *nis_list*(3N) for more details.

Namespaces

The NIS+ service defines two additional *disjoint* namespaces for its own use. These namespaces are the NIS+ *Principal* namespace, and the NIS+ *Group* namespace. The names associated with the group and principal namespaces are syntactically identical to simple names. However, the information they represent *cannot* be obtained by directly presenting these names to the NIS+ interfaces. Instead, special interfaces are defined to map these names into NIS+ names so that they may then be resolved.

Principal Names

NIS+ principal names are used to uniquely identify users and machines that are making NIS+ requests. These names have the form:

principal.domain

Here *domain* is the fully qualified name of an NIS+ directory where the named principal's credentials can be found. See *Directories and Domains* for more information on domains. Note that in this name, *principal*, is not a leaf in the NIS+ namespace.

Credentials are used to map the identity of a host or user from one context such as a process UID into the NIS+ context. They are stored as records in an NIS+ table named *cred*, which always appears in the *org_dir* subdirectory of the directory named in the principal name.

This mapping can be expressed as a replacement function:

principal.domain -> [**cname=principal.domain**],**cred.org_dir.domain**

This latter name is an NIS+ name that can be presented to the *nis_list*(3N) interface for resolution. NIS+ principal names are administered using the *nisaddcred*(1M) command.

The *cred* table contains five columns named *cname*, *auth_name*, *auth_type*, *public_data*, and *private_data*. There is one record in this table for each identity mapping for an NIS+ principal. The current service supports two such mappings:

LOCAL This mapping is used to map from the UID of a given process to the NIS+ principal name associated with that UID. If no mapping exists, the name *nobody* is returned. When the effective UID of the process is 0 (for example, the super-user), the NIS+ name associated with the host is returned. Note that UIDs are sensitive to the context of the machine on which the process is executing.

DES This mapping is used to map to and from a Secure RPC "netname" into an NIS+ principal name. See *secure_rpc*(3N) for more information on netnames. Note that since netnames contain the notion of a domain, they span NIS+ directories.

The NIS+ client library function *nis_local_principal*(3N) uses the *cred.org_dir* table to map the UNIX notion of an identity, a process' UID, into an NIS+ principal name. Shell programs can use the program *nisdefaults*(1) with the **-p** switch to return this information.

Mapping from UIDs to an NIS+ principal name is accomplished by constructing a query of the form:

[**auth_type=LOCAL, auth_name=uid**],**cred.org_dir.default-domain** .

This query will return a record containing the NIS+ principal name associated with this UID in the machine's default domain.

The NIS+ service uses the DES mapping to map the names associated with Secure RPC requests into NIS+ principal names. RPC requests that use Secure RPC include the *netname* of the client making the request in the RPC header. This netname has the form:

unix.UID@domain

The service constructs a query using this name of the form:

[**auth_type=DES, auth_name=netname**],**cred.org_dir.domain** .

where the domain part is extracted from the netname rather than using the default domain. This query is used to look up the mapping of this netname into an NIS+ principal name in the domain where it was created.

This mechanism of mapping UID and netnames into an NIS+ principal name guarantees that a client of the NIS+ service has only one principal name. This principal name is used as the basis for authorization which is described below. All objects in the NIS+ namespace and all entries in NIS+ tables must have an

owner specified for them. This owner field always contains an NIS+ principal name.

Group Names

Like NIS+ principal names, NIS+ group names take the form:

group_name .**domain**

All objects in the NIS+ namespace and all entries in NIS+ tables may optionally have a *group owner* specified for them. This group owner field, when filled in, always contains the fully qualified NIS+ group name.

The NIS+ client library defines several interfaces (*nis_groups*(3N)) for dealing with NIS+ groups. These interfaces internally map NIS+ group names into an NIS+ simple name which identifies the NIS+ group object associated with that group name. This mapping can be shown as follows:

group.domain -> *group.groups_dir.domain*

This mapping eliminates collisions between NIS+ group names and NIS+ directory names. For example, without this mapping, a directory with the name *engineering.foo.com.*, would make it impossible to have a group named *engineering.foo.com.*. This is due to the restriction that within the NIS+ namespace, a name unambiguously identifies a single object. With this mapping, the NIS+ *group* name *engineering.foo.com.* maps to the NIS+ *object* name *engineering.groups_dir.foo.com.*

The contents of a group object is a list of NIS+ principal names and the names of other NIS+ groups. See *nis_groups*(3N) for a more complete description of their use.

NIS+ SECURITY

NIS+ defines a security model to control access to information managed by the service. The service defines access rights that are selectively granted to individual clients or groups of clients. Principal names and group names are used to define clients and groups of clients that may be granted or denied access to NIS+ information. These principals and groups are associated with NIS+ domains as defined below.

The security model also uses the notion of a class of principals called *nobody*, which contains all clients, whether or not they have authenticated themselves to the service. The class *world* includes any client who has been authenticated.

| n |

Directories and Domains

Some directories within the NIS+ namespace are referred to as NIS+ *Domains*. Domains are those NIS+ directories that contain the subdirectories *groups_dir* and *org_dir*. Further, the subdirectory *org_dir* should contain the table named *cred*. NIS+ Group names and NIS+ Principal names *always* include the NIS+ domain name after their first label.

Authentication

The NIS+ name service uses Secure RPC for the integrity of the NIS+ service. This requires that users of the service and their machines must have a Secure RPC key pair associated with them. This key is initially generated with either the *nisaddcred*(1M) or *nisclient*(1M) commands and modified with the *chkey*(1) or *nispassud*(1) commands.

The use of Secure RPC allows private information to be stored in the name service that will not be available to untrusted machines or users on the network.

In addition to the Secure RPC key, users need a mapping of their UID into an NIS+ principal name. This mapping is created by the system administrator using the *nisclient*(1M) or *nisaddcred*(1M) command.

Users that will be using machines in several NIS+ domains must insure that they have a *local* credential entry in each of those domains. This credential should be created with the NIS+ principal name of the user in their “home” domain. For the purposes of NIS+ and Secure RPC, the home domain is defined to be the one where your Secure RPC key pair is located.

Authorization

The NIS+ service defines four access rights that can be granted or denied to clients of the service. These rights are *read*, *modify*, *create*, and *destroy*. These rights are specified in the object structure at creation time and may be modified later with the *nischmod*(1) command. In general, the rights granted for an object apply only to that object. However, for purposes of authorization, rights granted to clients reading *directory* and *table* objects are granted to those clients for all of the objects “contained” by the parent object. This notion of containment is abstract. The objects do not actually contain other objects within them. Note that *group* objects do contain the list of principals within their definition.

Access rights are interpreted as follows:

- read** This right grants read access to an object. For directory and table objects, having read access on the parent object conveys read access to all of the objects that are direct children of a directory, or entries within a table.
- modify** This right grants modification access to an existing object. Read access is not required for modification. However, in many applications, one will need to read an object before modifying it. Such modify operations will fail unless read access is also granted.
- create** This right gives a client permission to create new objects where one had not previously existed. It is only used in conjunction with directory and table objects. Having create access for a table allows a client to add additional entries to the table. Having create access for a directory allows a client to add new objects to an NIS+ directory.
- destroy** This right gives a client permission to destroy or remove an existing object or entry. When a client attempts to destroy an entry or object by removing it, the service first checks to see if the table or directory containing that object grants the client destroy access. If it does, the operation proceeds. If the containing object does not grant this right then the object itself is checked to see if it grants this right to the client. If the object grants the right, then the operation proceeds; otherwise the request is rejected.

Each of these rights may be granted to any one of four different categories.

owner A right may be granted to the *owner* of an object. The owner is the NIS+ principal identified in the owner field. The owner can be changed with the *nischown*(1) command. Note that if the owner does not have modification access rights to the object, the owner cannot change any access rights to the object, unless the owner has modification access rights to its parent object.

group owner

A right may be granted to the *group owner* of an object. This grants the right to any principal that is identified as a member of the group associated with the object. The group owner may be changed with the *nischgrp*(1) command. The object owner need not be a member of this group.

world A right may be granted to everyone in the *world*. This grants the right to all clients who have authenticated themselves with the service.

nobody A right may be granted to the *nobody* principal. This has the effect of granting the right to any client that makes a request of the service, regardless of whether they are authenticated or not.

Note that for bootstrapping reasons, directory objects that are NIS+ domains, the *org_dir* subdirectory and the *cred* table within that subdirectory must have *read* access to the *nobody* principal. This makes navigation of the namespace possible when a client is in the process of locating its credentials. Granting this access does not allow the contents of other tables within *org_dir* to be read (such as the entries in the password table) unless the table itself gives "read" access rights to the *nobody* principal.

Directory Authorization

Additional capabilities are provided for granting access rights to clients for directories. These rights are contained within the *object access rights* (OAR) structure of the directory. This structure allows the NIS+ service to grant rights that are not granted by the directory object to be granted for objects contained by the directory of a specific type.

An example of this capability is a directory object which does not grant create access to all clients, but does grant create access in the OAR structure for *group* type objects to clients who are members of the NIS+ group associated with the directory. In this example the only objects that could be created as children of the directory would have to be of the type *group*.

Another example is a directory object that grants create access only to the owner of the directory, and then additionally grants create access through the OAR structure for objects of type *table*, *link*, *group*, and *private* to any member of the directory's group. This has the effect of giving nearly complete create access to the group with the exception of creating subdirectories. This restricts the creation of new NIS+ domains because creating a domain requires creating both a *groups_dir* and *org_dir* subdirectory.

Note that there is currently no command line interface to set or change the OAR of the directory object.

Table Authorization

As with directories, additional capabilities are provided for granting access to entries within tables. Rights granted to a client by the access rights field in a table object apply to the table object and all of the entry objects “contained” by that table. If an access right is not granted by the table object, it may be granted by an entry within the table. This holds for all rights except *create*.

For example, a table may not grant read access to a client performing a *nis_list*(3N) operation on the table. However, the access rights field of entries within that table may grant read access to the client. Note that access rights in an entry are granted to the owner and group owner of the *entry* and not the owner or group of the table. When the list operation is performed, all entries that the client has read access to are returned. Those entries that do not grant read access are not returned. If none of the entries that match the search criterion grant read access to the client making the request, no entries are returned and the result status contains the **NIS_NOTFOUND** error code.

Access rights that are granted by the rights field in an entry are granted for the entire entry. However, in the table object an additional set of access rights is maintained for each column in the table. These rights apply to the equivalent column in the entry. The rights are used to grant access when neither the table nor the entry itself grant access. The access rights in a column specification apply to the owner and group owner of the entry rather than the owner and group owner of the table object.

When a read operation is performed, if read access is not granted by the table and is not granted by the entry but *is* granted by the access rights in a column, that entry is returned with the correct values in all columns that are readable and the string ***NP*** (No Permission) in columns where read access is not granted.

As an example, consider a client that has performed a list operation on a table that does not grant read access to that client. Each entry object that satisfied the search criterion specified by the client is examined to see if it grants read access to the client. If it does, it is included in the returned result. If it does not, then each column is checked to see if it grants read access to the client. If any columns grant read access to the client, data in those columns is returned. Columns that do not grant read access have their contents replaced by the string ***NP***. If none of the columns grant read access, then the entry is not returned.

LIST OF COMMANDS

The following lists all commands and programming functions related to NIS+:

| n |

NIS+ User Commands

| | |
|--------------------------|---|
| <i>nisaddent</i> (1) | add /etc files and NIS maps into their corresponding NIS+ tables |
| <i>niscat</i> (1) | display NIS+ tables and objects |
| <i>nischgrp</i> (1) | change the group owner of a NIS+ object |
| <i>nischmod</i> (1) | change access rights on a NIS+ object |
| <i>nischown</i> (1) | change the owner of a NIS+ object |
| <i>nischttl</i> (1) | change the time to live value of a NIS+ object |
| <i>nisdefaults</i> (1) | display NIS+ default values |
| <i>niserror</i> (1) | display NIS+ error messages |
| <i>nisgrep</i> (1) | utilities for searching NIS+ tables |
| <i>nisgrpadm</i> (1) | NIS+ group administration command |
| <i>nisln</i> (1) | symbolically link NIS+ objects |
| <i>nisls</i> (1) | list the contents of a NIS+ directory |
| <i>nismatch</i> (1) | utilities for searching NIS+ tables |
| <i>nismkdir</i> (1) | create NIS+ directories |
| <i>nisspasswd</i> (1) | change NIS+ password information |
| <i>nismrm</i> (1) | remove NIS+ objects from the namespace |
| <i>nismrmdir</i> (1) | remove NIS+ directories |
| <i>nisshowcache</i> (1M) | NIS+ utility to print out the contents of the shared cache file |
| <i>nistbladm</i> (1) | NIS+ table administration command |
| <i>nistest</i> (1) | return the state of the NIS+ namespace using a conditional expression |

NIS+ Administrative Commands

| | |
|--------------------------|---|
| <i>nis_cachemgr</i> (1M) | NIS+ utility to cache location information about NIS+ servers |
| <i>nisaddcred</i> (1M) | create NIS+ credentials |
| <i>nisaddent</i> (1M) | create NIS+ tables from corresponding /etc files or NIS maps |
| <i>nisclient</i> (1M) | initialize NIS+ credentials for NIS+ principals |
| <i>nisd</i> (1M) | NIS+ service daemon |

| | |
|-----------------------------|---|
| <i>nisd_resolv</i> (1M) | NIS+ service daemon |
| <i>nisinit</i> (1M) | NIS+ client and server initialization utility |
| <i>nislog</i> (1M) | display the contents of the NIS+ transaction log |
| <i>nisping</i> (1M) | send ping to NIS+ servers |
| <i>nispopulate</i> (1M) | populate the NIS+ tables in a NIS+ domain |
| <i>nisserver</i> (1M) | set up NIS+ servers |
| <i>nissetup</i> (1M) | initialize a NIS+ domain |
| <i>nisshowcache</i> (1M) | NIS+ utility to print out the contents of the shared cache file |
| <i>nisstat</i> (1M) | report NIS+ server statistics |
| <i>nisupdkeys</i> (1M) | update the public keys in a NIS+ directory object |
| <i>rpc.nisd</i> (1M) | NIS+ service daemon |
| <i>rpc.nisd_resolv</i> (1M) | NIS+ service daemon |

NIS+ Programming API

| | |
|---------------------------------|--|
| <i>_nis_map_group</i> (3N) | NIS+ group manipulation functions |
| <i>db_add_entry</i> (3N) | NIS+ Database access functions |
| <i>db_checkpoint</i> (3N) | NIS+ Database access functions |
| <i>db_create_table</i> (3N) | NIS+ Database access functions |
| <i>db_destroy_table</i> (3N) | NIS+ Database access functions |
| <i>db_first_entry</i> (3N) | NIS+ Database access functions |
| <i>db_free_result</i> (3N) | NIS+ Database access functions |
| <i>db_initialize</i> (3N) | NIS+ Database access functions |
| <i>db_list_entries</i> (3N) | NIS+ Database access functions |
| <i>db_next_entry</i> (3N) | NIS+ Database access functions |
| <i>db_remove_entry</i> (3N) | NIS+ Database access functions |
| <i>db_reset_next_entry</i> (3N) | NIS+ Database access functions |
| <i>db_standby</i> (3N) | NIS+ Database access functions |
| <i>db_table_exists</i> (3N) | NIS+ Database access functions |
| <i>db_unload_table</i> (3N) | NIS+ Database access functions |
| <i>nis_add</i> (3N) | NIS+ namespace functions |
| <i>nis_add_entry</i> (3N) | NIS+ table functions |
| <i>nis_addmember</i> (3N) | NIS+ group manipulation functions |
| <i>nis_checkpoint</i> (3N) | misc NIS+ log administration functions |
| <i>nis_clone_object</i> (3N) | NIS+ subroutines |
| <i>nis_creategroup</i> (3N) | NIS+ group manipulation functions |
| <i>nis_db</i> (3N) | NIS+ Database access functions |
| <i>nis_destroy_object</i> (3N) | NIS+ subroutines |
| <i>nis_destroygroup</i> (3N) | NIS+ group manipulation functions |
| <i>nis_dir_cmp</i> (3N) | NIS+ subroutines |
| <i>nis_domain_of</i> (3N) | NIS+ subroutines |
| <i>nis_error</i> (3N) | display NIS+ error messages |
| <i>nis_first_entry</i> (3N) | NIS+ table functions |
| <i>nis_freenames</i> (3N) | NIS+ subroutines |
| <i>nis_freeresult</i> (3N) | NIS+ namespace functions |
| <i>nis_freeservlist</i> (3N) | miscellaneous NIS+ functions |
| <i>nis_freetags</i> (3N) | miscellaneous NIS+ functions |
| <i>nis_getnames</i> (3N) | NIS+ subroutines |
| <i>nis_getservlist</i> (3N) | miscellaneous NIS+ functions |
| <i>nis_groups</i> (3N) | NIS+ group manipulation functions |
| <i>nis_ismember</i> (3N) | NIS+ group manipulation functions |
| <i>nis_leaf_of</i> (3N) | NIS+ subroutines |
| <i>nis_lerror</i> (3N) | display some NIS+ error messages |
| <i>nis_list</i> (3N) | NIS+ table functions |
| <i>nis_local_directory</i> (3N) | NIS+ local names |
| <i>nis_local_group</i> (3N) | NIS+ local names |
| <i>nis_local_host</i> (3N) | NIS+ local names |
| <i>nis_local_names</i> (3N) | NIS+ local names |
| <i>nis_local_principal</i> (3N) | NIS+ local names |
| <i>nis_lookup</i> (3N) | NIS+ namespace functions |
| <i>nis_map_group</i> (3N) | NIS+ group manipulation functions |
| <i>nis_mkdir</i> (3N) | miscellaneous NIS+ functions |

| | |
|-----------------------------------|--|
| <i>nis_modify</i> (3N) | NIS+ namespace functions |
| <i>nis_modify_entry</i> (3N) | NIS+ table functions |
| <i>nis_name_of</i> (3N) | NIS+ subroutines |
| <i>nis_names</i> (3N) | NIS+ namespace functions |
| <i>nis_next_entry</i> (3N) | NIS+ table functions |
| <i>nis_objects</i> (3N) | NIS+ object formats |
| <i>nis_perror</i> (3N) | display NIS+ error messages |
| <i>nis_ping</i> (3N) | misc NIS+ log administration functions |
| <i>nis_print_group_entry</i> (3N) | NIS+ group manipulation functions |
| <i>nis_print_object</i> (3N) | NIS+ subroutines |
| <i>nis_remove</i> (3N) | NIS+ namespace functions |
| <i>nis_remove_entry</i> (3N) | NIS+ table functions |
| <i>nis_removemember</i> (3N) | NIS+ group manipulation functions |
| <i>nis_rmdir</i> (3N) | miscellaneous NIS+ functions |
| <i>nis_server</i> (3N) | miscellaneous NIS+ functions |
| <i>nis_servstate</i> (3N) | miscellaneous NIS+ functions |
| <i>nis_sperrno</i> (3N) | display NIS+ error messages |
| <i>nis_sperror</i> (3N) | display NIS+ error messages |
| <i>nis_sperror_r</i> (3N) | display NIS+ error messages |
| <i>nis_stats</i> (3N) | miscellaneous NIS+ functions |
| <i>nis_subr</i> (3N) | NIS+ subroutines |
| <i>nis_tables</i> (3N) | NIS+ table functions |
| <i>nis_verifygroup</i> (3N) | NIS+ group manipulation functions |

NIS+ Files and Directories

| | |
|---------------------|---|
| <i>nisfiles</i> (4) | NIS+ database files and directory structure |
|---------------------|---|

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

NIS+ was developed by Sun Microsystems, Inc.

FILES

| | |
|--|--|
| <code><rpcsvc/nis_object.h></code> | Protocol description of an NIS+ object. |
| <code><rpcsvc/nis.h></code> | Defines the NIS+ protocol using the RPC language. It should be included by all clients of the NIS+ service |

SEE ALSO

nischown(1), *nisdefaults*(1), *nismatch*(1), *nisspasswd*(1), *newkey*(1M), *nisaddcred*(1M), *nisclient*(1M), *nispopulate*(1M), *nissserver*(1M), *nis_add_entry*(3N), *nis_domain_of*(3N), *nis_getnames*(3N), *nis_groups*(3N), *nis_leaf_of*(3N), *nis_list*(3N), *nis_local_directory*(3N), *nis_lookup*(3N), *nis_objects*(3N).

NAME

niscat - display NIS+ tables and objects

SYNOPSIS

```
niscat [ -AhLMv ] tablename ...
```

```
niscat [ -ALMP ] -o name ...
```

DESCRIPTION

In the first synopsis, **niscat** displays the contents of the NIS+ tables named by *tablename*. In the second synopsis, it displays the internal representation of the NIS+ objects named by *name*.

Options

- A** Display the data within the table and all of the data in tables in the initial table's concatenation path.
- h** Display the header line prior to displaying the table. The header consists of the '#' (hash) character followed by the name of each column. The column names are separated by the table separator character.
- L** Follow links. When this option is specified, if *tablename* or *name* names a LINK type object, the link is followed and the object or table named by the link is displayed.
- M** Master server only. This option specifies that the request should be sent to the master server of the named data. This guarantees that the most up-to-date information is seen at the possible expense of increasing the load on the master server and increasing the possibility of the NIS+ server being unavailable or busy for updates.
- P** Follow concatenation path. This option specifies that the request should follow the concatenation path of a table if the initial search is unsuccessful. This option is only useful when using an indexed name for *name* and the **-o** option.
- v** Display binary data directly. This option displays columns containing binary data on the standard output. Without this option, binary data is displayed as the string ***BINARY***.
- o name** Display the internal representation of the named NIS+ object(s). If *name* is an indexed name (see *nismatch*(1)), then each of the matching entry objects is displayed. This option is used to display access rights and other attributes of individual columns.

Notes

Columns without values in the table are displayed by two adjacent separator characters.

EXAMPLES

Display the contents of the hosts table:

```
niscat -h host.org_dir
# cname name      addr          comment
client1 client1 129.144.201.100 Joe Smith
crunchy crunchy 129.144.201.44  Jane Smith
crunchy softy   129.144.201.44
```

The string ***NP*** is returned in those fields where the user has insufficient access rights.

Display the **passwd.org_dir** on the standard output:

```
niscat passwd.org_dir
```

Display the contents of table **frodo** and the contents of all tables in its concatenation path:

```
niscat -A frodo
```

Display the entries in the table **group.org_dir** as NIS+ objects (note that the brackets are protected from the shell by single quotes):

```
niscat -o '[ ]group.org_dir'
```

Display the table object of the **passwd.org_dir** table:

```
niscat -o passwd.org_dir
```

The previous example displays the passwd table object and not the passwd table. The table object includes information such as the number of columns, column type, searchable or not searchable,

separator, access rights, and other defaults.

Display the directory object for **org_dir**, which includes information such as the access rights and replica information:

```
niscat -o org_dir
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ table name is not fully qualified, each directory specified will be searched until the table is found (see *nisdefaults(1)*).

RETURN VALUE

niscat returns **0** on success and **1** on failure.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

niscat was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), *nismatch(1)*, *nistbladm(1)*, *nisdefaults(1)*, *nis_objects(3N)*, *nis_tables(3N)*.

| n |

NAME

nischgrp - change the group owner of an NIS+ object

SYNOPSIS

```
nischgrp [ -AfLP ] group name ...
```

DESCRIPTION

nischgrp changes the group owner of the NIS+ objects or entries specified by *name* to the specified NIS+ *group*. Entries are specified using indexed names (see *nismatch(1)*). If *group* is not a fully qualified NIS+ group name, it will be resolved using the directory search path (see *nisdefaults(1)*).

The only restriction on changing an object's group owner is that you must have modify permissions for the object.

This command will fail if the master NIS+ server is not running.

Options

- A** Modify all entries in all tables in the concatenation path that match the search criterion specified in *name*. This option implies the **-P** switch.
- f** Force the operation and fail silently if it does not succeed.
- L** Follow links and change the group owner of the linked object or entries rather than the group owner of the link itself.
- P** Follow the concatenation path within a named table. This option only makes sense when either *name* is an indexed name or the **-L** switch is also specified and the named object is a link pointing to entries.

Notes

The NIS+ server will check the validity of the group name prior to effecting the modification.

EXAMPLES

Change the group owner of an object to a group in a different domain, and how to change it to a group in the local domain, respectively:

```
nischgrp newgroup.remote.domain. object
nischgrp my-buds object
```

Change the group owner for a password entry:

```
nischgrp admins '[uid=99],passwd.org_dir'
```

In the above example, **admins** is an NIS+ group in the same domain.

Change the group owner of the object or entries pointed to by a link, and the group owner of all entries in the **hobbies** table:

```
nischgrp -L my-buds linkname
nischgrp my-buds '[],hobbies'
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see *nisdefaults(1)*).

RETURN VALUE

nischgrp returns 0 on success and 1 on failure.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nischgrp was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), nischmod(1), nischown(1), nisdefaults(1), nisgrpadm(1), nis_objects(3N).

NAME

nischmod - change access rights on an NIS+ object

SYNOPSIS

nischmod [**-AfLP**] *mode name* ...

DESCRIPTION

nischmod changes the access rights (mode) of the NIS+ objects or entries specified by *name* to *mode*. Entries are specified using indexed names (see *nismatch*(1)). Only principals with modify access to an object may change its mode.

mode has the following form:

rights [,*rights*] ...

rights has the form:

[*who*] *op permission* [*op permission*] ...

who is a combination of:

| | |
|----------|-----------------------|
| n | Nobody's permissions. |
| o | Owner's permissions. |
| g | Group's permissions. |
| w | World's permissions. |
| a | All, or owg . |

If *who* is omitted, the default is **a**.

op is one of:

| | |
|----------|------------------------------------|
| + | To grant the <i>permission</i> . |
| - | To revoke the <i>permission</i> . |
| = | To set the permissions explicitly. |

permission is any combination of:

| | |
|----------|----------|
| r | Read. |
| m | Modify. |
| c | Create. |
| d | Destroy. |

Options

- A** Modify all entries in all tables in the concatenation path that match the search criteria specified in *name*. This option implies the **-P** switch.
- f** Force the operation and fail silently if it does not succeed.
- L** Follow links and change the permission of the linked object or entries rather than the permission of the link itself.
- P** Follow the concatenation path within a named table. This option is only applicable when either *name* is an indexed name or the **-L** switch is also specified and the named object is a link pointing to an entry.

Notes

Unlike the system **chmod** command, this command does not accept an octal notation.

EXAMPLES

Give everyone read access to an object (that is, access for owner, group, and all):

```
nischmod a+r object
```

Deny create and modify privileges to **group** and unauthenticated clients (**nobody**):

```
nischmod gn-cm object
```

Set a complex set of permissions for an object:

```
nischmod o=rmcd,g=rm,w=rc,n=r object
```

Set the permissions of an entry in the password table so that the group owner can modify them:

| **n** |


```
nischmod g+m '[uid=55],passwd.org_dir'
```

Change the permissions of a linked object:

```
nischmod -L w+mr linkname
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see *nisdefaults(1)*).

RETURN VALUE

nischmod returns **0** on success and **1** on failure.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nischmod was developed by Sun Microsystems, Inc.

SEE ALSO

chmod(1), *nis+(1)*, *nischgrp(1)*, *nischown(1)*, *nisdefaults(1)*, *nis_objects(3N)*.

| n |

NAME

nischown - change the owner of an NIS+ object

SYNOPSIS

```
nischown [ -AfLP ] owner name ...
```

DESCRIPTION

nischown changes the owner of the NIS+ objects or entries specified by *name* to *owner*. Entries are specified using indexed names (see *nismatch*(1)). If *owner* is not a fully qualified NIS+ principal name (see *nisaddcred*(1M)), the default domain (see *nisdefaults*(1)) will be appended to it.

The only restriction on changing an object's owner is that you must have modify permissions for the object. Note: If you are the current owner of an object and you change ownership, you may not be able to regain ownership unless you have modify access to the new object.

The command will fail if the master NIS+ server is not running.

Options

- A** Modify all entries in all tables in the concatenation path that match the search criteria specified in *name*. It implies the **-P** option.
- f** Force the operation and fail silently if it does not succeed.
- L** Follow links and change the owner of the linked object or entries rather than the owner of the link itself.
- P** Follow the concatenation path within a named table. This option is only meaningful when either *name* is an indexed name or the **-L** option is also specified and the named object is a link pointing to entries.

Notes

The NIS+ server will check the validity of the name before making the modification.

EXAMPLES

n Change the owner of an object to a principal in a different domain, and to change it to a principal in the local domain, respectively:

```
nischown bob.remote.domain. object
nischown skippy object
```

Change the owner of an entry in the passwd table:

```
nischown bob.remote.domain. '[uid=99],passwd.org_dir'
```

Change the object or entries pointed to by a link:

```
nischown -L skippy linkname
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set, and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see *nisdefaults*(1)).

RETURN VALUE

nischown returns 0 on success and 1 on failure.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nischown was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), nischgrp(1), nischmod(1), nischttl(1), nisdefaults(1), nisaddcred(1M), nis_objects(3N).

NAME

nischttl - change the time to live value of an NIS+ object

SYNOPSIS

```
nischttl [ -AFLP ] time name ...
```

DESCRIPTION

nischttl changes the time to live value (**ttl**) of the NIS+ objects or entries specified by *name* to *time*. Entries are specified using indexed names (see *nismatch*(1)).

The time to live value is used by object caches to expire objects within their cache. When an object is read into the cache, this value is added to the current time in seconds yielding the time when the cached object would expire. The object may be returned from the cache as long as the current time is earlier than the calculated expiration time. When the expiration time has been reached, the object will be flushed from the cache.

The time to live *time* may be specified in seconds or in days, hours, minutes, seconds format. The latter format uses a suffix letter of **d**, **h**, **m**, or **s** to identify the units of time. See the examples below for usage.

The command will fail if the master NIS+ server is not running.

Options

- A** Modify all tables in the concatenation path that match the search criterion specified in *name*. This option implies the **-P** switch.
- f** Force the operation and fail silently if it does not succeed.
- L** Follow links and change the time to live of the linked object or entries rather than the time to live of the link itself.
- P** Follow the concatenation path within a named table. This option only makes sense when either *name* is an indexed name or the **-L** switch is also specified and the named object is a link pointing to entries.

Notes

Setting a high **ttl** value allows objects to stay persistent in caches for a longer period of time and can improve performance. However, when an object changes, in the worst case, the number of seconds in this attribute must pass before that change is visible to all clients. Setting a **ttl** value of **0** means that the object should not be cached at all.

A high **ttl** value is a week, a low value is less than a minute. Password entries should have **ttl** values of about 12 hours (easily allows one password change per day), entries in the RPC table can have **ttl** values of several weeks (this information is effectively unchanging).

Only directory and group objects are cached in this implementation.

EXAMPLES

Change the **ttl** of an object using the seconds format and the days, hours, minutes, seconds format (the **ttl** of the second object is set to 1 day and 12 hours):

```
nischttl 184000 object
nischttl 1d12h object
```

Change the **ttl** for a password entry:

```
nischttl 1h30m '[uid=99],passwd.org_dir'
```

Change the **ttl** of the object or entries pointed to by a link, and the **ttl** of all entries in the **hobbies** table:

```
nischttl -L 12h linkname
nischttl 3600 '[],hobbies'
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see *nisdefaults*(1)).

RETURN VALUE

nischttl returns **0** on success and **1** on failure.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nischttl was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), **nischgrp(1)**, **nischmod(1)**, **nischown(1)**, **nisdefaults(1)**, **nis_objects(3N)**.

| n |

NAME

nisdefaults - display NIS+ default values

SYNOPSIS

```
nisdefaults [ -adghprstv ]
```

DESCRIPTION

nisdefaults prints the default values that are returned by calls to the NIS+ local name functions (see *nis_local_names(3N)*). With no options specified, all defaults will be printed in a verbose format. With options, only that option is displayed in a terse form suitable for shell scripts. See the example below.

Options

- a** Print all defaults in a terse format.
- d** Print the default domain name.
- g** Print the default group name.
- h** Print the default host name.
- p** Print the default principal name.
- r** Print the default access rights with which new objects will be created.
- s** Print the default directory search path.
- t** Print the default time to live value.
- v** Print the defaults in a verbose format. This prepends an identifying string to the output.

EXAMPLES

Print the NIS+ defaults for a root process on machine **example** in the **foo.bar.** domain:

```
example# nisdefaults
Principal Name : example.foo.bar.
Domain Name   : foo.bar.
Host Name     : example.foo.bar.
Group Name    :
Access Rights : ----rmcdr---r---
Time to live  : 12:00:00
Search Path   : foo.bar.
```

Set a variable in a shell script to the default domain:

```
DOMAIN=`nisdefaults -d`
```

Print out the default time to live in a verbose format:

```
nisdefaults -tv
Time to live : 12:00:00
```

Print out the time to live in the terse format:

```
nisdefaults -t
43200
```

EXTERNAL INFLUENCES**Environment Variables**

Several environment variables affect the defaults associated with a process.

NIS_DEFAULTS This variable contains a defaults string that will override the NIS+ standard defaults. The defaults string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. All of the legal tokens are described below.

tvl=*time*

This token sets the default time to live for objects that are created. The value *time* is specified in the format as defined by the **nischttl** command. (See *nischttl(1)*). The default value is 12 hours.

owner=*ownername*

This token specifies that the NIS+ principal *ownername* should own created

objects. The default for this value is the principal who is executing the command.

group=groupname

This token specifies that the group *groupname* should be the group owner for created objects. The default is **NULL**.

access=rights

This token specifies the set of access rights that are to be granted for created objects. The value *rights* is specified in the format as defined by the **nischmod** command. (See *nischmod*(1)). The default value is **---rmcdr---r---**.

NIS_GROUP This variable contains the name of the local NIS+ group. If the name is not fully qualified, the default domain will be appended to it.

NIS_PATH This variable overrides the default NIS+ directory search path. It contains an ordered list of directories separated by ':' (colon) characters. The '\$' (dollar sign) character is treated specially. Directory names that end in '\$' have the default domain appended to them, and a '\$' by itself is replaced by the list of directories between the default domain and the global root that are at least two levels deep. The default NIS+ directory search path is '\$'.

Refer to the *Name Expansion* subsection in *nis+(1)* for more details.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nisdefaults was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), *nis_local_names(3N)*.

n

NAME

niserror - display NIS+ error messages

SYNOPSIS

niserror *error-num*

DESCRIPTION

niserror prints the NIS+ error associated with status value *error-num* on the standard output. It is used by shell scripts to translate NIS+ error numbers that are returned into text messages.

EXAMPLES

Print the error associated with the error number 20:

```
niserror 20  
Not Found, no such name
```

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

niserror was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), nis_error(3N).

| n |

NAME

nisgrpadm - NIS+ group administration command

SYNOPSIS

```
nisgrpadm -a | -r | -t [ -s ] group principal...
```

```
nisgrpadm -c | -d | -l [ -M ] [ -s ] group
```

DESCRIPTION

nisgrpadm is used to administer NIS+ groups. This command administers both groups and the groups' membership lists. **nisgrpadm** can create, destroy, or list NIS+ groups. **nisgrpadm** can be used to administer a group's membership list. It can add or delete principals to the group, or test principals for membership in the group.

The names of NIS+ groups are syntactically similar to names of NIS+ objects but they occupy a separate namespace. A group named "a.b.c.d." is represented by a NIS+ group object named "a.groups_dir.b.c.d."; the functions described here all expect the name of the group, not the name of the corresponding group object.

There are three types of group members:

- An **explicit** member is just a NIS+ principal-name, for example "wickedwitch.west.oz."
- An **implicit** ("domain") member, written "*.west.oz.", means that all principals in the given domain belong to this member. No other forms of wildcarding are allowed: "wickedwitch.*.oz." is invalid, as is "wickedwitch.west.*.". Note that principals in subdomains of the given domain are *not* included.
- A **recursive** ("group") member, written "@cowards.oz.", refers to another group; all principals that belong to that group are considered to belong here.

Any member may be made **negative** by prefixing it with a minus sign ('-'). A group may thus contain explicit, implicit, recursive, negative explicit, negative implicit, and negative recursive members.

A principal is considered to belong to a group if it belongs to at least one non-negative group member of the group and belongs to no negative group members.

| n |

Options

- a Add the list of NIS+ principals specified to *group*. The principal name should be fully qualified.
- c Create *group* in the NIS+ namespace. The NIS+ group name should be fully qualified.
- d Destroy (remove) *group* from the namespace.
- l List the membership list of the specified *group*. (See **-M**.)
- M Master server only. Send the lookup to the master server of the named data. This guarantees that the most up to date information is seen at the possible expense that the master server may be busy. Note that the **-M** flag is applicable only with the **-l** flag.
- r Remove the list of principals specified from *group*. The principal name should be fully qualified.
- s Work silently. Results are returned using the exit status of the command. This status can be translated into a text string using the *niserror*(1) command.
- t Display whether the principals specified are members in *group*.

Notes

Principal names *must* be fully qualified, whereas groups can be abbreviated on all operations *except* create.

EXAMPLES**Administering Groups**

Create a group in the **foo.com.** domain:

```
nisgrpadm -c my_buds.foo.com.
```

Remove the group from the current domain:

```
nisgrpadm -d fred's_group
```


Administering Members

Add two principals, **bob** and **betty** to the group **my_buds.foo.com**:

```
nisgrpadm -a my_buds.foo.com. bob.bar.com. betty.foo.com.
```

Remove **betty** from **freds_group**:

```
nisgrpadm -r freds_group betty.foo.com.
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ group name is not fully qualified, each directory specified will be searched until the group is found (see *nisdefaults(1)*).

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

DIAGNOSTICS

NIS_SUCCESS On success, this command returns an exit status of 0.

NIS_PERMISSION

When you do not have the needed access right to change the group, the command returns this error.

NIS_NOTFOUND This is returned when the group does not exist.

NIS_TRYAGAIN This error is returned when the server for the group's domain is currently checkpointing or otherwise in a read-only state. The command should be retried at a later date.

NIS_MODERROR This error is returned when the group was modified by someone else during the execution of the command. Reissue the command and optionally recheck the group's membership list.

AUTHOR

nisgrpadm was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), *nischgrp(1)*, *nisdefaults(1)*, *niserror(1)*, *nis_groups(3N)*.

| n |

NAME

nisl - symbolically link NIS+ objects

SYNOPSIS

```
nisl [ -L ] [ -D defaults ] name linkname
```

DESCRIPTION

The **nisl** command links an NIS+ object named *name* to an NIS+ name *linkname*. If *name* is an indexed name (see *nismatch*(1)), the link points to entries within an NIS+ table. Clients wishing to look up information in the name service can use the **FOLLOW_LINKS** flag to force the client library to follow links to the name they point to. Further, all of the NIS+ administration commands accept the **-L** switch indicating they should follow links (see *nis_names*(3N) for a description of the **FOLLOW_LINKS** flag).

Options

- L** When present, this option specifies that this command should follow links. If *name* is itself a link, then this command will follow it to the linked object that it points to. The new link will point to that linked object rather than to *name*.
- D *defaults*** Specify a different set of defaults to be used for the creation of the link object. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. All of the legal tokens are described below.
- ttl=*time*** This token sets the default time to live for objects that are created by this command. The value *time* is specified in the format as defined by the *nischttl*(1) command. The default is 12 hours.
- owner=*ownername*** This token specifies that the NIS+ principal *ownername* should own the created object. The default for this value is the the principal who is executing the command.
- group=*groupname*** This token specifies that the group *groupname* should be the group owner for the object that is created. The default is **NULL**.
- access=*rights*** This token specifies the set of access rights that are to be granted for the given object. The value *rights* is specified in the format as defined by the *nischmod*(1) command. The default value is **---rmcdr---r---**.

Notes

When creating the link, **nisl** verifies that the linked object exists. Once created, the linked object may be deleted or replaced and the link will not be affected. At that time the link will become invalid and attempts to follow it will return **NIS_LINKNAMEERROR** to the client. When the path attribute in tables specifies a link rather than another table, the link will be followed if the flag **FOLLOW_LINKS** was present in the call to **nis_list** () (see *nis_tables*(3N)) and ignored if the flag is not present. If the flag is present and the link is no longer valid, a warning is sent to the system logger and the link is ignored.

EXAMPLES

Create a link in the domain **foo.com.** named **hosts** that points to the object **hosts.bar.com:**

```
nisl hosts.bar.com. hosts.foo.com.
```

Make a link *example.sun.com.* that points to an entry in the hosts table in *eng.sun.com.:*

```
nisl '[name=example],hosts.eng.sun.com.' example.sun.com:
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see *nisdefaults*(1)).

RETURN VALUE

nisl returns 0 on success and 1 on failure.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nisln was developed by Sun Microsystems, Inc.

SEE ALSO

nisdefaults(1), nismatch(1), nisrm(1), nistbladm(1), nis_names(3N), nis_tables(3N).

| n |

NAME

nisls - list the contents of an NIS+ directory

SYNOPSIS

```
nisls [ -dglLmMR ] [ name ... ]
```

DESCRIPTION

For each *name* that is an NIS+ directory, **nisls** lists the contents of the directory. For each *name* that is an NIS+ object other than a directory, **nisls** simply echos the name. If no *name* is specified, the first directory in the search path (see *nisdefaults*(1)) is listed.

Options

- d** Treat NIS+ directories like other NIS+ objects, rather than listing their contents.
- g** Display group owner instead of owner when listing in long format.
- l** List in long format. This option displays additional information about the objects such as their type, creation time, owner, and access rights.
The access rights are listed in the following order in long mode: nobody, owner, group owner, and world.
- L** This option specifies that links are to be followed. If *name* actually points to a link, it is followed to the linked object.
- m** Display modification time instead of creation time when listing in long format.
- M** Master only. This specifies that information is to be returned from the master server of the named object. This guarantees that the most up-to-date information is seen at the possible expense that the master server may be busy.
- R** List directories recursively. This option will reiterate the list for each subdirectory found in the process of listing each *name*.

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see *nisdefaults*(1)).

RETURN VALUE

nisls returns 0 on success and 1 on failure.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nisls was developed by Sun Microsystems, Inc.

SEE ALSO

nisdefaults(1), *nisgrpadm*(1), *nismatch*(1), *nistbladm*(1), *nis_objects*(3N).

NAME

nismatch, nisgrep - utilities for searching NIS+ tables

SYNOPSIS

```
nismatch [ -AchMoPv ] key tablename
nismatch [ -AchMoPv ] colname=key... tablename
nismatch [ -AchMoPv ] indexedname
nisgrep [ -AchMov ] keypat tablename
nisgrep [ -AchMov ] colname=keypat... tablename
```

DESCRIPTION

nismatch and **nisgrep** can be used to search NIS+ tables. The command **nisgrep** differs from the **nismatch** command in its ability to accept regular expressions *keypat* for the search criteria rather than simple text matches.

Because **nisgrep** uses a callback function, it is not constrained to searching only those columns that are specifically made searchable at the time of table creation. This makes it more flexible, but slower than **nismatch**.

In **nismatch**, the server does the searching; whereas in **nisgrep**, the server returns all the readable entries and then the client does the pattern-matching.

In both commands, the parameter *tablename* is the NIS+ name of the table to be searched. If only one key or key pattern is specified without the column name, then it is applied searching the first column. Specific named columns can be searched by using the *colname=key* syntax. When multiple columns are searched, only entries that match in all columns are returned. This is the equivalent of a logical join operation.

nismatch accepts an additional form of search criteria, *indexedname*, which is a NIS+ indexed name of the form:

```
[ colname=value , . . . ],tablename
```

Options

- A** All data. Return the data within the table and all of the data in tables in the initial table's concatenation path.
- c** Print only a count of the number of entries that matched the search criteria.
- h** Display a header line before the matching entries that contains the names of the table's columns
- M** Master server only. Send the lookup to the master server of the named data. This guarantees that the most up to date information is seen at the possible expense that the master server may be busy.
- o** Display the internal representation of the matching NIS+ object(s).
- P** Follow concatenation path. Specify that the lookup should follow the concatenation path of a table if the initial search is unsuccessful.
- v** Verbose. Do not suppress the output of binary data when displaying matching entries. Without this option, binary data is displayed as the string ***BINARY***.

RETURN VALUES

- 0** Successfully matches some entries.
- 1** Successfully searches the table and no matches are found.
- 2** An error condition occurs. An error message is also printed.

EXAMPLES

This example searches a table named **passwd** in the **org_dir** subdirectory of the **zotz.com.** domain. It returns the entry that has the username of **skippy**. In this example, all the work is done on the server.

```
nismatch name=skippy passwd.org_dir.zotz.com.
```

This example is similar to the one above except that it uses **nisgrep** to find all users in the table named **passwd** that are using either *ksh*(1) or *csh*(1).

```
nisgrep 'shell=[ck]sh' passwd.org_dir.zotz.com.
```

EXTERNAL INFLUENCES

Environment Variables

NIS_PATH If this variable is set and the NIS+ table name is not fully qualified, each directory specified will be searched until the table is found (see *nisdefaults(1)*).

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

DIAGNOSTICS

No memory

An attempt to allocate some memory for the search failed.

tablename is not a table

The object with the name *tablename* was not a table object.

Can't compile regular expression

The regular expression in *keypat* was malformed.

column not found: *colname*

The column named *colname* does not exist in the table named *tablename*.

AUTHOR

nismatch and **nisgrep** were developed by Sun Microsystems, Inc.

SEE ALSO

niscat(1), *nisdefaults(1)*, *nisls(1)*, *nistbladm(1)*, *nis_objects(3N)*.

| **n** |

NAME

nismkdir - create NIS+ directories

SYNOPSIS

```
nismkdir [ -D defaults ] [ -m hostname | -s hostname ] dirname
```

DESCRIPTION

The **nismkdir** command creates new NIS+ subdirectories within an existing domain. It can also be used to create replicated directories. Without options, this command will create a subdirectory with the same master and the replicas as its parent directory.

It is advisable to use *nisserver*(1M) to create an NIS+ domain which consists of the specified directory along with the **org_dir** and **groups_dir** subdirectories.

The two primary aspects that are controlled when making a directory are its access rights, and its degree of replication.

dirname is the fully qualified NIS+ name of the directory that has to be created.

Options

-D *defaults* Specify a different set of defaults to be used when creating new directories. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. All of the legal tokens are described below.

ttl=*time*

This token sets the default time to live for objects that are created by this command. The value *time* is specified in the format as defined by the *nischttl*(1) command. The default value is **12h** (12 hours).

owner=*ownername*

This token specifies that the NIS+ principal *ownername* should own the created object. The default for this value is the principal who is executing the command.

group=*groupname*

This token specifies that the group *groupname* should be the group owner for the object that is created. The default value is **NULL**.

access=*rights*

This token specifies the set of access rights that are to be granted for the given object. The value *rights* is specified in the format as defined by the *nischmod*(1) command. The default value is **---rmcdr--r---**.

-m *hostname* If the directory named by *dirname* does not exist, then a new directory that is *not* replicated is created with host *hostname* as its master server.

If the directory name by *dirname* does exist, then the host named by *hostname* is made its master server.

-s *hostname* Specify that the host *hostname* will be a replica for an existing directory named *dirname*.

Notes

A host that serves an NIS+ directory *must be* an NIS+ client in a directory above the one it is serving. The exceptions to this rule are the root NIS+ servers which are both clients and servers of the same NIS+ directory.

When the host's default domain is different from the default domain on the client where the command is executed, the hostname supplied as an argument to the **-s** or **-m** options must be fully qualified.

RETURN VALUES

This command returns **0** if successful and **1** otherwise.

EXAMPLES

Create a new directory **bar** under the **foo.com.** domain that shares the same master and replicas as the **foo.com.** directory:

```
nismkdir bar.foo.com.
```

Create a new directory *bar.foo.com.* that is not replicated under the **foo.com.** domain:

```
nismkdir -m myhost.foo.com. bar.foo.com.
```

Add a replica server of the *bar.foo.com.* directory:

```
nismkdir -s replica.foo.com. bar.foo.com.
```

EXTERNAL INFLUENCES

Environment Variables

NIS_DEFAULTS This variable contains a defaults string that will override the NIS+ standard defaults. If the **-D** switch is used, those values will then override both the **NIS_DEFAULTS** variable and the standard defaults.

NIS_PATH If this variable is set and the NIS+ directory name is not fully qualified, each directory specified will be searched until the directory is found (see *nisdefaults*(1)).

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nismkdir was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), *nischmod*(1), *nisdefaults*(1), *nisl*s(1), *nismrmdir*(1), *nissserver*(1M).

| **n** |

NAME

nispasswd - change NIS+ password information

SYNOPSIS

```
nispasswd [ -ghs ] [ -D domainname ] [ username ]
nispasswd -a
nispasswd -D domainname [ -d [ username ] ]
nispasswd [ -l ] [ -f ] [ -n min ] [ -x max ] [ -w warn ] [ -D domainname ] username
```

DESCRIPTION

nispasswd changes a password, gecos (finger) field (**-g**option), home directory (**-h**option), or login shell (**-s**option) associated with the *username* (invoker by default) in the NIS+ passwd table.

Additionally, the command can be used to view or modify aging information associated with the user specified if the invoker has the right NIS+ privileges.

nispasswd uses secure RPC to communicate with the NIS+ server, and therefore, never sends unencrypted passwords over the communication medium.

nispasswd does not read or modify the local password information stored in the `/etc/passwd` file.

When used to change a password, **nispasswd** prompts non-privileged users for their old password. It then prompts for the new password twice to forestall typing mistakes. When the old password is entered, **nispasswd** checks to see if it has aged sufficiently. If aging is insufficient, **nispasswd** terminates; see *getpwent(3C)*.

The old password is used to decrypt the username's secret key. If the password does not decrypt the secret key, **nispasswd** prompts for the old secure-RPC password. It uses this password to decrypt the secret key. If this fails, it gives the user one more chance. The old password is also used to ensure that the new password differs from the old by at least three characters. Assuming aging is sufficient, a check is made to ensure that the new password meets construction requirements described below. When the new password is entered a second time, the two copies of the new password are compared. If the two copies are not identical, the cycle of prompting for the new password is repeated twice. The new password is used to re-encrypt the user's secret key. Hence, it also becomes their secure-RPC password.

Passwords must be constructed to meet the following requirements:

- Each password must have at least six characters. Only the first eight characters are significant.
- Each password must contain at least two alphabetic characters and at least one numeric or special character. In this case, "alphabetic" refers to all upper or lower case letters.
- Each password must differ from the user's login *username* and any reverse or circular shift of that login *username*. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.
- New passwords must differ from the old by at least three characters. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.

Network administrators, who own the NIS+ password table, may change any password attributes if they establish their credentials (see *keylogin(1)*) before invoking **nispasswd**. Hence, **nispasswd** does not prompt these privileged-users for the old password and they are not forced to comply with password aging and password construction requirements.

Any user may use the **-d** option to display password attributes for his or her own login name. The format of the display will be:

```
username status mm/dd/yy min max warn
```

or, if password aging information is not present,

```
username status
```

where

username The login ID of the user.

status The password status of *username*: "PS" stands for password exists or locked, "LK" stands for locked, and "NP" stands for no password.

| | |
|-----------------|--|
| <i>mm/dd/yy</i> | The date password was last changed for <i>username</i> . (Note that all password aging dates are determined using Greenwich Mean Time and, therefore, may differ by as much as a day in other time zones.) |
| <i>min</i> | The minimum number of days required between password changes for <i>username</i> . |
| <i>max</i> | The maximum number of days the password is valid for <i>username</i> . |
| <i>warn</i> | The number of days relative to <i>max</i> before the password expires that the <i>username</i> will be warned. |

Options

| | |
|-------------------------------|---|
| -g | Change the <i>gecos</i> (finger) information. |
| -h | Change the home directory. |
| -s | Change the login shell. By default, only the NIS+ administrator can change the login shell. User will be prompted for the new login shell. |
| -a | Show the password attributes for all entries. This will show only the entries in the NIS+ passwd table in the local domain that the invoker is authorized to "read". |
| -d [<i>username</i>] | Display password attributes for the caller or the user specified if the invoker has the right privileges. |
| -l | Locks the password entry for <i>username</i> . Subsequently, <i>login</i> (1) would disallow logins with this NIS+ password entry. |
| -f | Force the user to change password at the next login by expiring the password for <i>username</i> . |
| -n <i>min</i> | Set minimum field for <i>username</i> . The <i>min</i> field contains the minimum number of days between password changes for <i>username</i> . If <i>min</i> is greater than <i>max</i> , the user may not change the password. Always use this option with the -x option, unless <i>max</i> is set to -1 (aging turned off). In that case, <i>min</i> need not be set. |
| -x <i>max</i> | Set maximum field for <i>username</i> . The <i>max</i> field contains the number of days that the password is valid for <i>username</i> . The aging for <i>username</i> will be turned off immediately if <i>max</i> is set to -1. If it is set to 0, then the user is forced to change the password at the next login session and aging is turned off. |
| -w <i>warn</i> | Set <i>warn</i> field for <i>username</i> . The <i>warn</i> field contains the number of days before the password expires that the user will be warned whenever he or she attempts to log in. |
| -D <i>domainname</i> | Consult the passwd.org_dir table in <i>domainname</i> . If this option is not specified, the default <i>domainname</i> returned by nis_local_directory() will be used. This <i>domainname</i> is the same as that returned by <i>domainname</i> (1). |

Notes

The login program, file access display programs (for example, **ls -l**) and network programs that require user passwords (for example, *rlogin*(1), *ftp*(1), etc.) use the standard *getpwent*(3C) interface to get password information. These programs will get the NIS+ password information, which is modified by **nispasswd**, only if the **passwd:** entry in the **/etc/nsswitch.conf** file includes **nisplus**. See *nsswitch.conf*(4) for more details.

RETURN VALUE

The **nispasswd** command exits with one of the following values:

- 0 SUCCESS.
- 1 Permission denied.
- 2 Invalid combination of options.
- 3 Unexpected failure. NIS+ passwd table unchanged.
- 4 NIS+ passwd table missing.
- 5 NIS+ is busy. Try again later.
- 6 Invalid argument to option.

7 Aging is disabled.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nispasswd was developed by Sun Microsystems, Inc.

SEE ALSO

keylogin(1), login(1), nis+(1), nistbladm(1), passwd(1), domainname(1), getpwent(3C), nsswitch.conf(4), passwd(4).

| n |

NAME

nisrm - remove NIS+ objects from the namespace

SYNOPSIS

nisrm [**-if**] *name* ...

DESCRIPTION

The **nisrm** command removes NIS+ objects named *name* from the NIS+ namespace.

This command will fail if the NIS+ master server is not running.

Options

- i** Interactive mode. Like the system *rm(1)* command, the **nisrm** command will ask for confirmation prior to removing an object. If the name specified by *name* is a non-fully qualified name, this option is forced on. This prevents the removal of unexpected objects.
- f** Force. The removal is attempted, and if it fails for permission reasons, an *nischmod(1)* is attempted and the removal retried. If the command fails, it fails silently.

Notes

This command will not remove directories (see *nisrmdir(1)*) nor will it remove non-empty tables (see *nistbladm(1)*).

EXAMPLES

Remove the objects *foo*, *bar*, and *baz* from the namespace:

```
nisrm foo bar baz
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see *nisdefaults(1)*).

n**RETURN VALUE**

nisrm returns 0 on success and 1 on failure.

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nisrm was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), *nischmod(1)*, *nisdefaults(1)*, *nisrmdir(1)*, *nistbladm(1)*, *rm(1)*.

NAME

nisrmdir - remove NIS+ directories

SYNOPSIS

```
nisrmdir [ -if ] [ -s hostname ] dirname
```

DESCRIPTION

nisrmdir deletes existing NIS+ subdirectories. It can remove a directory outright, or simply remove replicas from serving a directory.

This command modifies the object that describes the directory *dirname*, and then notifies each replica to remove the directory named *dirname*. If the notification of any of the affected replicas fails, the directory object is returned to its original state unless the **-f** option is present.

This command will fail if the NIS+ master server is not running.

Options

-i Interactive mode. Like the system *rm*(1) command, the **nisrmdir** command will ask for confirmation prior to removing a directory. If the name specified by *dirname* is a non-fully qualified name, this option is forced on. This prevents the removal of unexpected directories.

-f Force the command to succeed even though it may not be able to contact the affected replicas. This option should be used when a replica is known to be down and will not be able to respond to the removal notification. When the replica is finally rebooted, it will read the updated directory object, note that it is no longer a replica for that directory, and stop responding to lookups on that directory. Cleanup of the files that held the now removed directory can be accomplished manually by removing the appropriate files in the **/var/nis** directory (see *nisfiles*(4) for more information).

-s *hostname*

Specify that the host *hostname* should be removed as a replica for the directory named *dirname*. If this option is not present, *all* replicas and the master server for a directory are removed and the directory is removed from the namespace.

RETURN VALUE

This command returns 0 if it is successful, and 1 otherwise.

EXAMPLES

Remove a directory **bar** under the **foo.com.** domain:

```
nisrmdir bar.foo.com.
```

Remove a replica that is serving directory **bar.foo.com.:**

```
nisrmdir -s replica.foo.com. bar.foo.com.
```

Force the removal of directory **bar.foo.com.** from the namespace:

```
nisrmdir -f bar.foo.com.
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ directory name is not fully qualified, each directory specified will be searched until the directory is found (see *nisdefaults*(1)).

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nisrmdir was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), nisdefaults(1), nisrm(1), nisfiles(4).

NAME

nistbladm - NIS+ table administration command

SYNOPSIS

```
nistbladm -a | -A [ -D defaults ] colname=value ... tablename
nistbladm -a | -A [ -D defaults ] indexedname
nistbladm -c [ -D defaults ] [ -p path ] [ -s sep ] type colname=[flags][,access] ...
    tablename
nistbladm -d tablename
nistbladm -e | -E colname=value ... indexedname
nistbladm -m colname=value ... indexedname
nistbladm -r | -R [ colname=value ... ] tablename
nistbladm -r | -R indexedname
nistbladm -u [ -p path ] [ -s sep ] [ -t type ] [ colname=access ... ] tablename
```

DESCRIPTION

The **nistbladm** command is used to administer NIS+ tables. There are five primary operations that it performs: creating and deleting tables, adding entries to, modifying entries within, and removing entries from tables.

Though NIS+ does not place restrictions on the size of tables or entries, the size of data has an impact on the performance and the disk space requirements of the NIS+ server. NIS+ is not designed to store huge pieces of data, such as files; instead pointers to files should be stored in NIS+.

NIS+ design is optimized to support 10,000 objects with a total size of 10M bytes. If the requirements exceed the above, it is suggested that a domain hierarchy be created, or the data stored in the tables be pointers to the actual data, instead of the data itself.

When creating tables, a table type, *type*, and a list of column definitions must be provided.

type is a string that is stored in the table and later used by the service to verify that entries being added to it are of the correct type.

Syntax for column definitions is:

```
colname=[flags] [,access]
```

flags is a combination of:

- S** Searchable. Specifies that searches can be done on the column's values (see *nismatch*(1)).
- I** Case-insensitive (only makes sense in combination with **S**). Specifies that searches should ignore case.
- C** Crypt. Specifies that the column's values should be encrypted.
- B** Binary data (does not make sense in combination with **S**). If not set, the column's values are expected to be null terminated ASCII strings.
- X** XDR encoded data (only makes sense in combination with **B**).

access is specified in the format as defined by the *nischmod*(1) command.

When manipulating entries, this command takes two forms of entry name. The first uses a series of space separated *colname=value* pairs that specify column values in the entry. The second is an NIS+ indexed name, *indexedname*, of the form:

```
[ colname=value , . . . ],tablename
```

Options

- a | A** Add entries to a NIS+ table. The difference between the lowercase 'a' and the uppercase 'A' is in the treatment of preexisting entries. The entry's contents are specified by the *column=value* pairs on the command line. Note: Values for *all* columns must be specified when adding entries to a table.

Normally, NIS+ reports an error if an attempt is made to add an entry to a table that would overwrite an entry that already exists. This prevents multiple parties from adding duplicate entries and having one of them get overwritten. If you wish to force the add, the uppercase 'A' specifies that the entry is to be added, even if it

- already exists. This is analogous to a modify operation on the entry.
- c** Create a table named *tablename* in the namespace. The table that is created must have at least one column and at least one column must be searchable.
- d *tablename*** Destroy the table named *tablename*. The table that is being destroyed must be empty. The table's contents can be deleted with the **-R** option below.
- e | E** Edit the entry in the table that is specified by *indexdname*. *indexdname* must uniquely identify a single entry. It is possible to edit the value in a column that would change the indexed name of an entry.
- The change (*colname=value*) may affect other entries in the table if the change results in an entry whose indexed name is different from *indexedname* and which matches that of another existing entry. In this case, the **-e** option will fail and an error will be reported. The **-E** option will force the replacement of the existing entry by the new entry (effectively removing two old entries and adding a new one).
- m** Modify an entry in the table that is specified by *indexedname*. Note: Since it is possible to modify the value in a column that would change the indexed name for an entry, both the column value pair and the indexed name are required. It uses the indexed name to look up the entry, modify it, and write it back with the new value. The indexed name must uniquely identify a single entry.
- r | R** Remove entries from a table. The entry is specified by either a series of *column=value* pairs on the command line, or an indexed name that is specified as *entryname*. The difference between the interpretation of the lowercase **r** versus the uppercase **R** is in the treatment of non-unique entry specifications. Normally the NIS+ server will disallow an attempt to remove an entry when the search criterion specified for that entry resolves to more than one entry in the table. However, it is sometimes desirable to remove more than one entry, as when you are attempting to remove all of the entries from a table. In this case, using the uppercase **R** will force the NIS+ server to remove all entries matching the passed search criterion. If that criterion is null and no column values specified, then all entries in the table will be removed.
- u** Update attributes of a table. This allows the concatenation path (**-p**), separation character (specified with the (**-s**)), column access rights, and table type string (**-t**) of a table to be changed. Neither the number of columns, nor the columns that are searchable may be changed.
- D *defaults*** When creating objects, this option specifies a different set of defaults to be used during this operation. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. All of the legal tokens are described below.
- ttl=time*** This token sets the default time to live for objects that are created by this command. The value *time* is specified in the format as defined by the *nischttl*(1) command. The default value is 12 hours.
- owner=ownername***
This token specifies that the NIS+ principal *ownername* should own the created object. Normally this value is the same as the principal who is executing the command.
- group=groupname***
This token specifies that the group *groupname* should be the group owner for the object that is created. The default value is NULL.
- access=rights***
This token specifies the set of access rights to be granted for the given object. The value *rights* is specified in the format as defined by the *nischmod*(1) command. The default value is **---rmcdr---r---**.
- p *path*** When creating or updating a table, this option specifies the table's search path. When an **nislislist**() function is invoked, the user can specify the flag FOLLOW_PATH to tell the client library to continue searching tables in the table's path if the search criteria used does not yield any entries. The path consists of an ordered list of table names, separated by colons. The names in the path must be

| n |

- fully qualified.
- s** *sep* When creating or updating a table, this option specifies the table's separator character. The separator character is used by *niscat*(1) when displaying tables on the standard output. Its purpose is to separate column data when the table is in ASCII form. The default value is a space.
 - t** *type* When updating a table, this option specifies the table's type string.

RETURN VALUE

This example returns 0 on success and 1 on failure.

EXAMPLES

Create a table named **hobbies** in the directory **foo.com.** of the type **hobby_tbl** with two searchable columns, **name** and **hobby**:

```
nistbladm -c hobby_tbl name=S,a+r,o+m hobby=S,a+r
hobbies.foo.com.
```

The column **name** has read access for all (that is, **owner**, **group**, and **world**) and modify access for only the owner. The column **hobby** is readable by all, but not modifiable by anyone.

In this example, if the access rights had not been specified, the tables access rights would have come from either the standard defaults or the **NIS_DEFAULTS** variable (see below).

Add entries to this table:

```
nistbladm -a name=bob hobby=skiing hobbies.foo.com.
nistbladm -a name=sue hobby=skiing hobbies.foo.com.
nistbladm -a name=ted hobby=swimming hobbies.foo.com.
```

Add the concatenation path:

```
nistbladm -u -p hobbies.bar.com.:hobbies.baz.com. hobbies
```

Delete the skiers from our list:

```
nistbladm -R hobby=skiing hobbies.foo.com.
```

Note: The use of the **-r** option would fail because there are two entries with the value of **skiing**.

To create a table with a column that is named with no flags set, you supply only the name and the equal sign (=) as follows:

```
nistbladm -c notes_tbl name=S,a+r,o+m note= notes.foo.com.
```

This example created a table, named *notes.foo.com.*, of type *notes_tbl* with two columns **name** and **note**. The **note** column is not searchable.

When entering data for columns in the form of a *value* string, it is essential that terminal characters be protected by single or double quotes. These are the characters equals (=), comma (,), left bracket ([), right bracket (]), and space (.). These characters are parsed by NIS+ within an indexed name. These characters are protected by enclosing the entire value in double quote (") characters as follows:

```
nistbladm -a fullname="Joe User" nickname=Joe nicknames
```

If there is any doubt about how the string will be parsed, it is better to enclose it in quotes.

EXTERNAL INFLUENCES

Environment Variables

NIS_DEFAULTS This variable contains a defaults string that will override the NIS+ standard defaults. If the **-D** switch is used, those values will then override both the **NIS_DEFAULTS** variable and the standard defaults.

NIS_PATH If this variable is set and the NIS+ table name is not fully qualified, each directory specified will be searched until the table is found (see *nisdefaults*(1)).

WARNINGS

To modify one of the entries, say, for example, from **bob** to **robert**:

```
nistbladm -m name=robert [name=bob],hobbies
```


Note that `[name=bob],hobbies` is an indexed name, and that the characters '[' (open bracket) and ']' (close bracket) are interpreted by the shell. When typing entry names in the form of NIS+ indexed names, the name must be protected by using single quotes.

It is possible to specify a set of defaults such that you cannot read or modify the table object later.

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

`nistbladm` was developed by Sun Microsystems, Inc.

SEE ALSO

`nis+(1)`, `niscat(1)`, `nischmod(1)`, `nischown(1)`, `nisdefaults(1)`, `nismatch(1)`, `nissetup(1M)`.

| n |

NAME

nistest - return the state of the NIS+ namespace using a conditional expression

SYNOPSIS

```
nistest [ -ALMP ] [ -a rights | -t type ] object
```

```
nistest [ -ALMP ] [ -a rights ] indexedname
```

DESCRIPTION

nistest provides a way for shell scripts and other programs to test for the existence, type, and access rights of objects and entries. Entries are named using indexed names (see *nismatch*(1)).

Options

- A** All data. This option specifies that the data within the table and all of the data in tables in the initial table's concatenation path be returned. This option is only valid when using indexed names or following links.
- L** Follow links. If the object named by *object* or the tablename component of *indexedname* names a LINK type object, the link is followed when this switch is present.
- M** Master server only. This option specifies that the lookup should be sent to the master server of the named data. This guarantees that the most up-to-date information is seen at the possible expense that the master server may be busy.
- P** Follow concatenation path. This option specifies that the lookup should follow the concatenation path of a table if the initial search is unsuccessful. This option is only valid when using indexed names or following links.
- a rights** This option is used to verify that the current process has the desired or required access rights on the named object or entries. The access rights are specified in the same way as the **nischmod** command.
- t type** This option tests the type of *object*. The value of *type* can be one of the following:
 - G** Return true if the object is a group object.
 - D** Return true if the object is a directory object.
 - T** Return true if the object is a table object.
 - L** Return true if the object is a link object.
 - P** Return true if the object is a private object.

| **n** |**RETURN VALUE**

- 0 Success.
- 1 Failure due to object not present, not of specified type and/or no such access.
- 2 Failure due to illegal usage.

EXAMPLES

When testing for access rights, **nistest** returns success (0) if the specified rights are granted to the current user. Thus, testing for access rights

```
nistest -a w=mr skippy.domain
```

Tests that all authenticated NIS+ clients have read and modify access to the object named *skippy.domain*.

Testing for access on a particular entry in a table can be accomplished using the indexed name syntax. The following example tests to see if an entry in the password table can be modified.

```
nistest -a o=m '[uid=99],passwd.org_dir'
```

EXTERNAL INFLUENCES**Environment Variables**

NIS_PATH If this variable is set and the NIS+ name is not fully qualified, each directory specified will be searched until the object is found (see *nisdefaults*(1)).

WARNINGS

HP-UX 11i Version 2 is the last HP-UX release on which NIS+ is supported.

LDAP is the recommended replacement for NIS+. HP fully supports the industry standard naming services based on LDAP.

AUTHOR

nistest was developed by Sun Microsystems, Inc.

SEE ALSO

nis+(1), nischmod(1), nisdefaults(1).

| n |

NAME

nl - line numbering filter

SYNOPSIS

nl [-*btype*] [-*htype*] [-*ftype*] [-*p*] [-*vstart#*] [-*incr*] [-*ssep*] [-*width*] [-*nformat*] [-*lnum*] [-*d**delim*] [*file*]

DESCRIPTION

nl reads lines from the named *file* or the standard input if no *file* is named and reproduces the lines on the standard output. Lines are numbered on the left in accordance with the command options in effect.

nl views the text it reads in terms of logical pages. Line numbering is reset at the start of each logical page. A logical page consists of a header, a body, and a footer section. Empty sections are valid. Different line numbering options are independently available for header, body, and footer (e.g., no numbering of header and footer lines while numbering blank lines only in the body).

The start of logical page sections are signaled by input lines containing nothing but the following delimiter character(s):

| Line contents | Start of |
|---------------|----------|
| \\: \\: | header |
| \\: \\: | body |
| \\: | footer |

Unless told otherwise, **nl** assumes the text being read is in a single logical page body.

Command options can appear in any order and can be intermingled with an optional file name. Only one file can be named. **nl** recognizes the following options:

-btype Specifies which logical page body lines are to be numbered. Recognized *types* and their meanings are:

| | |
|----------------|---|
| a | number all lines; |
| t | number lines with printable text only; |
| n | no line numbering; |
| pstring | number only lines that contain the regular expression specified in <i>string</i> . Basic Regular Expression syntax is supported (see <i>regex</i> (5)). |

The default *type* for logical page body is **t** (text lines numbered).

-htype Same as **-btype** except for header. Default *type* for logical page header is **n** (no lines numbered).

-ftype Same as **-btype** except for footer. Default for logical page footer is **n** (no lines numbered).

-p Do not restart numbering at logical page delimiters.

-vstart# *start#* is the initial value used to number logical page lines. Default is **1**.

-incr *incr* is the increment value used to number logical page lines. Default is **1**.

-ssep *sep* is the character or characters used in separating the line number and the corresponding text line. Default *sep* is a tab.

-width *width* is the number of character columns to be used for the line number. Default *width* is **6**.

-nformat *format* is the line numbering format. Recognized values are:

| | |
|-----------|---|
| ln | left justified, leading zeroes suppressed; |
| rn | right justified, leading zeroes suppressed; |
| rz | right justified, leading zeroes kept. |

Default *format* is **rn** (right justified).

-lnum *num* is the number of consecutive blank lines to be treated and numbered as a single line. For example, **-13** results in every third adjacent blank line being numbered if the appropriate **-ha**, **-ba**, and/or **-fa** option is set. Default is **1**.

-dxx The delimiter characters specifying the start of a logical page section can be changed from the default characters (\\:) to two user-specified characters. If only one

character is entered, the second character remains the default character (:). No space should appear between the **-d** and the delimiter characters, however, this restriction is not there for **XPG4** compliant **nl**. To define a backslash as the delimiter, use two backslashes.

EXTERNAL INFLUENCES

Environment Variables

LC_COLLATE determines the collating sequence used in evaluating regular expressions.

LC_CTYPE determines the characters matched by character class expressions in regular expressions.

If **LC_COLLATE** or **LC_CTYPE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **nl** behaves as if all internationalization variables are set to "C". See *environ(5)*.

International Code Set Support

Single-byte character code sets are supported.

EXAMPLES

Number **file1** starting at line number 10, using an increment of ten. The logical page delimiters are **!** and **+**:

```
nl -v10 -i10 -d!+ file1
```

SEE ALSO

pr(1), *environ(5)*, *lang(5)*, *regex(5)*.

STANDARDS CONFORMANCE

nl: SVID2, SVID3, XPG2, XPG3, XPG4

| **n** |

NAME

nljust - justify lines, left or right, for printing

SYNOPSIS

nljust [-**acilnt**] [-**d** *digits*] [-**e** *seq*] [-**j** *just*] [-**m** *mode*] [-**o** *order*] [-**r** *margin*] [-**w** *width*] [-**x** *ck*] [*file ...*]

DESCRIPTION

nljust formats for printing data written in languages with a right-to-left orientation. It is designed to be used with the **pr** and the **lp** commands (see **pr(1)** and **lp(1)**).

nljust reads the concatenation of input files (or standard input if none are given) and produces on standard output a right-to-left formatted version of its input. If **-** appears as an input file name, **nljust** reads standard input at that point. Use **--** to delimit the end of options.

nljust formats input files for all languages that are read from right to left. For languages that have a left-to-right orientation, the command merely copies input files to standard output.

Options

nljust recognizes the following options:

- a** Justify data for all languages, including those having a left-to-right text orientation. By default only right-to-left language data is justified. For all other languages, input files are directly copied to standard output.
- c** Select enhanced printer shapes for some Arabic characters. With this option, two-character combinations of laam and alif are replaced by a single character.
- i** Triggers ISO 8859-6 interpretation of the data.
- d** *digits* Processes digits for output as hindi, western, or both. *digits* can be **h**, **w**, or both.
- e** *seq* Use *seq* as the escape sequence to select the primary character set. This escape sequence is used by languages that have too many characters to be accommodated by ASCII in a single 256-character set. In these cases, the *seq* escape sequence can be used to select the non-ASCII character set. The *escape* character itself (0x1b) is not given on the command line. Hewlett-Packard escape sequences are used by default.
- j** *just* If *just* is **l**, left justify print lines. If *just* is **r**, right-justify print lines starting from the (designated or default) print width column. The default is right justification.
- l** Replace leading spaces with alternative spaces. Some right-to-left character sets have a non-ASCII or alternative space. This option can be useful when filtering **pr -n** output (see **pr(1)**). With right justification, the **-l** option causes line numbers to be placed immediately to the right of the tab character. Without the **-l** option, right justification causes line numbers to be placed at the print-width column. By default, leading spaces are not replaced by alternative spaces.
- m** *mode* Indicate *mode* of any file to be formatted. Mode refers to the text orientation of the file when it was created. If *mode* is **l**, assume Latin mode. If *mode* is **n**, assume non-Latin mode. By default, mode information is obtained from the **LANGOPTS** environment variable.
- n** Do not terminate lines containing printable characters with a new-line. By default, print lines are terminated by new-lines.
- o** *order* Indicate data *order* of any file to be formatted. The text orientation of a file can affect the way its data is arranged. If *order* is **k**, assume keyboard order. If *order* is **s**, assume screen order. By default, order information is obtained from the **LANGOPTS** environment variable.
- t** Truncate print lines that do not fit the designated or default line length. Print lines are folded (that is, wrapped to next line) by default.
- x** *ck* Expand input tabs to column positions $k+1$, $2*k+1$, $3*k+1$, etc. Tab characters in the input are expanded to the appropriate number of spaces. If *k* is 0 or is omitted, default tab settings at every eighth position is assumed. If *cd* (any non-digit character) is given, it is treated as the input tab character. The default for *c* is the tab character. **nljust** always expands input tabs. This option provides a way to change the tab character and setting. If this option is specified, at least one of the parameters *c* or *k*

must be given.

- r** *margin* Designate a number as the print *margin*. The print margin is the column where truncation or folding takes place. The print margin determines how many characters appear on a single line and can never exceed the print width. The print margin is relative to the justification. If the print margin is 80, folding or truncation occurs at column 80 starting from the right during a right justification. Similarly, folding or truncation occurs at column 80 starting from the left during a left justification. By default, the print margin is set to column 80.
- w** *width* Designates a number as the print *width*. The print width is the maximum number of columns in the print line. Print width determines the start of text during a right justification. The larger the print width, the further to the right the text will start. By default, an 80-column print width is used.

EXTERNAL INFLUENCES

Environment Variables

The **LANGOPTS** environment variable determines the mode and order of the file. The syntax of **LANGOPTS** is [*mode*][_*order*]. *mode* describes the mode of a file where **l** represents Latin mode and **n** represents non-Latin mode. Non-Latin mode is assumed for values other than **l** and **n**. *order* describes the data order of a file where **k** is keyboard and **s** is screen. Keyboard order is assumed for values other than **k** and **s**. Mode and order information in **LANGOPTS** can be overridden from the command line.

The **LC_ALL** environment variable determines the direction of a language (left-to-right or right-to-left) and whether context analysis of characters is necessary.

The **LC_NUMERIC** environment variable determines whether a language has alternative numbers.

The **LANG** environment variable determines the language in which messages are displayed.

International Code Set Support

Single-byte character code sets are supported.

EXAMPLES

Right justify **file1** on a 132-column printer with a print margin at column 80 (the default):

```
nljust -w 132 file1 | lp
```

Right justify **pr** output of **file2** with line numbers on a 132-column printer with a print margin at column 132:

```
pr -n file2 | nljust -w 132 -r 132 | lp
```

WARNINGS

If **pr** with line numbers (**-n** option) is piped to **nljust**, the separator character must be a tab (0x09).

It is the user's responsibility to ensure that the **LANGOPTS** environment variable accurately reflects the status of the file.

Mode and justification must be consistent. Only non-Latin-mode files can be right justified in a meaningful way. Similarly, only Latin-mode files can be safely left justified. If mode and justification do not match, the results are undefined.

If present, alternative numbers always have a left-to-right orientation.

The **nljust** command is HP proprietary, not portable to other vendors' systems, and will not be provided in future HP-UX releases.

AUTHOR

nljust was developed by HP.

SEE ALSO

forder(1), lp(1), pr(1), strord(3C).

| n |

NAME

nm - print name list of common object file

SYNOPSIS

```
/usr/ccs/bin/nm [-ACefghlnNqrsTuUvV] [-d|-o|-x] [-p|-P] [-t format] file ...
```

DESCRIPTION

The **nm** command displays the symbol table of each object file, *file*.

file can be a relocatable object file or an executable object file, or it can be an archive of relocatable or executable object files.

There are three general output formats: the default (neither **-p** nor **-P** specified), **-p** specified, and **-P** specified. The output formats are described after the "Options" subsection.

By default, **nm** prints the entire name of the symbols listed. Since object files can have symbol names with an arbitrary number of characters, a name that is longer than the width of the column set aside for names will overflow its column, forcing every column after the name to be misaligned.

Options

nm recognizes the following options:

- | | |
|-----------|--|
| -A | Prefix each output line with the name of the object file or archive, <i>file</i> . Equivalent to -r . |
| -C | Demangle C++ names before printing them (ELF only). |
| -d | Display the <i>value</i> and <i>size</i> of a symbol in decimal. This is the default for the default format or the -p format. Equivalent to -t d . |
| -e | Display only external and static symbols. This option is ignored (see -f). |
| -f | Display full output. This option is in force by default. |
| -g | Display only external (global) symbol information. |
| -h | Do not display the output header data. |
| -l | Distinguish between weak and global symbols by appending * to the key letter of weak symbols. Only takes effect with -p and/or -P . |
| -n | Sort symbols by <i>name</i> , in ascending collation order, before they are printed. This is the default. See "Environment Variables" in EXTERNAL INFLUENCES below. |
| -N | Display symbols in the order in which they appear in the symbol table. |
| -o | Display the <i>value</i> and <i>size</i> of a symbol in octal. Equivalent to -t o . |
| -p | Display information in a blank-separated output format. Each symbol <i>name</i> is preceded by its value (blanks if undefined) and a letter to indicate type. A lowercase letter indicates a local (nonexternal) symbol. |
-
- | | |
|----------|------------------|
| A | (absolute) |
| B | (bss symbol) |
| C | (common symbol) |
| D | (data symbol) |
| M | (milli symbol) |
| N | (notype) |
| R | (section region) |
| T | (text symbol) |
| U | (undefined) |
- If the symbol is a secondary definition, the type letter is followed by the letter **S**.
- Note that **-p** is not compatible with **-P**.
- | | |
|-----------|---|
| -P | Display information in a portable output format, as specified below, to standard output. Note that -P is not compatible with -p . |
|-----------|---|

- r** Prefix each output line with the name of the object file or archive, *file*. Equivalent to **-A**.
- s** Print the section index instead of the section name (ELF only).
- t *format*** Display each numeric value in the specified format. *format* can be one of:
 - d** Display the *value* and *size* of a symbol in decimal. This is the default for the default format or the **-p** format. Equivalent to **-d**.
 - o** Display the *value* and *size* of a symbol in octal. Equivalent to **-o**.
 - x** Display the *value* and *size* of a symbol in hexadecimal. This is the default for the **-P** format. Equivalent to **-x**.
- u** Display undefined symbols only.
- U** Print the usage menu.
- v** Sort symbols by *value* before they are printed.
- V** Display the executing version of the **nm** command on standard error.
- x** Display the *value* and *size* of a symbol in hexadecimal. This is the default for the **-P** format. Equivalent to **-t x**.

Default Output Format - 32 bit

If the default (neither the **-p** nor the **-P** option) output format is specified, each symbol has the following columns, separated by vertical bars (|). The default for numbers is decimal (**-d** or **-t d**).

If decimal:

```
"%20s|%10d|%6s|%7s|%s", name, value, scope, type, subspace
```

If octal:

```
"%20s|%012o|%6s|%7s|%s", name, value, scope, type, subspace
```

If hexadecimal:

```
"%20s|0x%08x|%6s|%7s|%s", name, value, scope, type, subspace
```

| n |

Default Output Format - 64 bit

If the default (neither the **-p** nor the **-P** option) output format is specified, each symbol has the following columns, separated by vertical bars (|). The default for numbers is decimal (**-d** or **-t d**).

If decimal:

```
"[%u]s|%2211u|%8u|s|s|%1d|s|s",
  index, value, size, type, bind, O, shndx, name
```

If octal:

```
"[%u]s|%02211o|%010o|s|s|%1o|s|s",
  index, value, size, type, bind, O, shndx, name
```

If hexadecimal:

```
"[%u]s|0x%01611x|0x%08x|s|s|%1x|s|s",
  index, value, size, type, bind, O, shndx, name
```

The descriptions are explained below:

| | |
|-----------------|---|
| <i>name</i> | The name of the symbol. |
| <i>value</i> | Its value expressed as an offset or an address depending on its storage class. |
| <i>scope</i> | The scope of the symbol (external , sdef , static , or undefined). The sdef scope indicates an external symbol that is flagged as a secondary definition. |
| <i>type</i> | The type of the symbol (absolute , arg_ext , code , data , entry , milli_ext , millicode , module , null , oct_dis , plabel , pri_prog , sec_prog , storage , stub , sym_ext , tstor). |
| <i>subspace</i> | The subspace to which the symbol belongs. |

| | |
|--------------|---|
| <i>bind</i> | Specifies the symbol binding type (local, weak, global). |
| <i>O</i> | This field is used for files that have large section tables (>65K sections). For smaller files, the value of this field is 0. |
| <i>Shndx</i> | Identifies the index of the section that the symbol belongs to. |
| | Identifies the index of the symbol in the symbol table. |

Output Format for -p

If the **-p** option is specified, information is displayed using the following portable C-language formats. The default for numbers is decimal (**-d** or **-t d**).

If decimal: `"%010d %s %s", value, type, name`

If octal: `"%012o %s %s", value, type, name`

If hexadecimal: `"0x%08x %s %s", value, type, name`

If **-A** or **-r**, the line is preceded by: `"%20s:", file`

Output Format for -P

If the **-P** option is specified, information is displayed using the following portable C-language formats. The default for numbers is hexadecimal (**-x** or **-t x**). In the format string, **%s** represents string output; **%d** represents decimal output; **%o** represents octal output; **%x** represents hexadecimal output; **\n** represents newline; all other characters represent themselves.

- If decimal is specified:

`"%s %s %d %d\n", library-object, name, type, value, size`

- If octal is specified:

`"%s%s %s %o %o\n", library-object, name, type, value, size`

- If hexadecimal is specified, or by default:

`"%s%s %s %x %x\n", library-object, name, type, value, size`

where *library-object* is a string preformatted as follows:

- If **-A** and **-r** are not specified, *library-object* is an empty string.
- If **-A** or **-r** is specified, and the corresponding *file* operand does not name a library:

`"%s:", file`

- If **-A** or **-r** is specified and the corresponding *file* operand names a library, *object-file* names the object file in the library containing the symbol being described:

`"%s[%s]:", file, object-file`

If **-A** and **-r** are not specified, and if more than one *file* operand is specified, or if a single *file* operand that names a library is specified, then **nm** prints a line identifying the object containing the symbols before the lines containing those symbols, in one of the following forms:

- If the corresponding *file* operand does not name a library:

`"%s:\n", file`

- If the corresponding *file* operand names a library, *object-file* names the object file in the library containing the symbol being described:

`"%s[%s]:\n", file, object-file`

EXTERNAL INFLUENCES

Environment Variables

The following internationalization variables affect the execution of **nm**:

LANG determines the locale category for native language, local customs and coded character set in the absence of **LC_ALL** or other **LC_*** environment variables. If **LANG** is not specified or is null, it defaults to **C** (see *lang*(5)).

LC_ALL, if set to a nonempty string value, determines the values for all locale categories and has precedence over **LANG** and other **LC_*** environment variables.

LC_COLLATE determines the locale category for character collation.

LC_CTYPE determines the locale category for character handling functions.

LC_MESSAGES determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

LC_NUMERIC determines the locale category for numeric formatting.

ST_NMCAT and **NLSPATH** determine the location of message catalogues for processing **LC_MESSAGES**.

If an internationalization variable is not specified or is null, it defaults to the value of **LANG**.

If **LANG** is not specified or is null, it defaults to **C** (see *lang*(5)).

If any internationalization variable contains an invalid setting, then all internationalization variables default to **C** (see *environ*(5)).

International Code Set Support

Single-byte character code sets are supported.

EXAMPLES

Display which object files have undefined references for the symbol **leap**:

```
nm -rup *.o | grep leap
```

Display which object files have a definition for the text symbol **leap**:

```
nm -rp *.o | awk '{ if ($3 == "T" && $4 == "leap") { print $0 } }'
```

WARNINGS

By default, **nm** now sorts symbols by name (the **-n** option). To turn off sorting, use the **-N** option.

Some options added for standards conformance duplicate the functionality of options that previously existed. This duplication has been retained for backward compatibility.

SEE ALSO

System Tools

cc_bundled(1) HP-UX C compiler
ld(1) Link editor

Miscellaneous

crt0(3) Execution startup routine
end(3C) Symbol of the last locations in program

STANDARDS CONFORMANCE

nm: SVID2, SVID3, XPG2, XPG3, XPG4

| n |

NAME

nohup - run a command immune to hangups

SYNOPSIS

nohup *command* [*arguments*]

DESCRIPTION

nohup executes *command* with hangups and quits ignored. If output is not redirected by the user, both standard output and standard error are sent to **nohup.out**. If **nohup.out** is not writable in the current directory, output is redirected to **\$HOME/nohup.out**; otherwise, **nohup** fails. If a file is created, the file's permission bits will be set to **S_IRUSR | S_IWUSR**.

If output from **nohup** is redirected to a terminal, or is not redirected at all, the output is sent to **nohup.out**.

EXTERNAL INFLUENCES**Environment Variables**

LC_MESSAGES determines the language in which messages are displayed.

If **LC_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **nohup** behaves as if all internationalization variables are set to "C". See *environ(5)*.

International Code Set Support

Single- and multi-byte character code sets are supported.

EXAMPLES

It is frequently desirable to apply **nohup** to pipelines or lists of commands. This can be done only by placing pipelines and command lists in a single file, called a shell script. To run the script using **nohup**:

```
nohup sh file
```

nohup features apply to the entire contents of *file*. If the shell script *file* is to be executed often, the need to type **sh** can be eliminated by setting execute permission on *file*. The script can also be run in the background with interrupts ignored (see *sh(1)*):

```
nohup file &
```

file typically contains normal keyboard command sequences that one would want to continue running in case the terminal disconnects, such as:

```
tbl ofile | eqn | nroff > nfile
```

WARNINGS

Be careful to place punctuation properly. For example, in the command form:

```
nohup command1; command2
```

nohup applies only to *command1*. To correct the problem, use the command form:

```
nohup (command1; command2)
```

Be careful of where standard error is redirected. The following command may put error messages on tape, making it unreadable:

```
nohup cpio -o <list >/dev/rmt/c0t0d0BEST&
```

whereas

```
nohup cpio -o <list >/dev/rmt/c0t0d0BEST 2>errors&
```

puts the error messages into file **errors**.

EXIT STATUS

The following exit values are returned:

126 The command specified by *command* was found but could not be invoked

127 An error occurred in the nohup utility or the specified *command* could not be found. Otherwise, the exit status of nohup will be that of the command specified.

SEE ALSO

chmod(1), nice(1), sh(1), signal(5).

STANDARDS CONFORMANCE

nohup: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| n |

NAME

nroff - format text

SYNOPSIS

nroff [*options*] *file* ...

DESCRIPTION

nroff is a text formatting program that interprets source text contained in *file* and prepares it for printing on typewriter-like devices and line printers. If *file* name is - or not specified, standard input is used as source text.

If the file contains plain text with no formatter requests, **nroff** uses default line lengths and page dimensions to produce readable output, outputting a blank line for each blank line encountered in the input, and filling and adjusting text to both margins. **nroff** ignores any lines in the source text that begin with a period (.) but are not valid **nroff** formatter requests.

nroff formatting capabilities are described in the tutorial cited below.

Source File Preparation

Document source file preparation is usually easier when text is coded using macro packages such as *mm*(1) which provide a high-level interface for headings, page footers, lists, and other features, rather than coding the file with inherently low-level **nroff** requests.

Options

nroff recognizes the following command-line *options*, which can appear in any order but must appear before the *file* argument:

- olist** Print only pages whose page numbers appear in the *list* of numbers and ranges, separated by commas. A range *n-m* means pages *n* through *m*; an initial *-n* means from the beginning to page *n*; and a final *n-* means from *n* to the end. (See WARNINGS below.)
- nn** Number first generated page *n*.
- sn** Stop every *n* pages. **nroff** halts *after* every *n* pages (default *n*=1) to allow paper loading or changing, and resumes upon receipt of a line-feed or new-line (new-lines do not work in pipelines, such as with **mm**). When **nroff** halts between pages, an ASCII **BEL** is sent to the terminal.
- raN** Set register *a* (which must have a one-character name) to *N*.
- i** Read standard input after *files* are exhausted.
- q** Invoke the simultaneous input-output mode of the **.rd** request.
- z** Print only messages generated by **.tm** (terminal message) requests.
- mname** Precede the input *files* with the non-compiled (ASCII text) macro file
 - `/usr/lib/nls/LANG/tmac/tmac.name`
 where *LANG* is the value of the **LANG** environment variable. If **LANG** is not set or
 - `/usr/lib/nls/LANG/tmac/tmac.name`
 does not exist, the following file is used instead:
 - `/usr/share/lib/tmac/tmac.name`
- Tname** Prepare output for specified terminal. Known *names* are as follows:
 - 37** for the (default) TELETYPE Model 37 terminal
 - tn300** for the GE TermiNet 300 (or any terminal without half-line capability)
 - 300s** for the DASI 300s
 - 300** for the DASI 300
 - 450** for the DASI 450
 - lp** for a (generic) ASCII line printer
 - 382** for the DTC-382
 - 4000A** for the Trendata 4000A

- 832** for the Anderson Jacobson 832
 - X** for a (generic) EBCDIC printer
 - 2631** for the Hewlett-Packard 2631 line printer
 - klp** for a (generic) 16-bit character printer having ratio of 2 to 3 in 8-bit and 16-bit character width
 - lj** for Hewlett-Packard PCL3 and newer laser printers.
- e** Produce equally-spaced words in adjusted lines, using the full resolution of the particular terminal.
 - h** Use output tabs during horizontal spacing to speed output and reduce output character count. Tab settings are assumed to be every eight nominal character widths.
 - un** Set the emboldening factor (number of character overstrikes) for the third font position (bold) to *n*, or to zero if *n* is missing.
 - P** If this option is specified on the command line, it allows the use of the special feature provided by some Asian printers which prints two column wide characters in 3/2 column wide boxes.

EXTERNAL INFLUENCES

Environment Variables

LC_CTYPE determines the interpretation of text as single and/or multi-byte characters.

LANG is used to determine the search path for the **-m** option. **LANG** also determines the language in which messages are displayed.

If **LC_CTYPE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **nroff** behaves as if all internationalization variables are set to "C". See *environ(5)*.

EXAMPLES

The following command prints the first five pages of the document whose **nroff** source file is *filename*:

```
nroff -o-5 filename
```

Note that there should not be a space between the **o** and the **-** or the **-** and the **5**.

To print only pages 1, 3, and 4 type:

```
nroff -o1,3,4 filename
```

WARNINGS

When **nroff** is used with the **-olist** option inside a pipeline, it may cause a harmless "broken pipe" diagnostic if the last page of the document is not specified in *list*.

FILES

| | |
|-----------------------------------|--|
| /usr/share/lib/macros/* | Standard macro files |
| /usr/share/lib/term/* | Terminal driving tables for nroff |
| /usr/share/lib/suftab | Suffix hyphenation tables |
| /usr/share/lib/tmac/tmac.* | Standard macro files and pointers |

SEE ALSO

col(1), *mm(1)*, *neqn(1)*, *soelim(1)*, *ul(1)*, *man(5)*.

NAME

nslookup - query name servers interactively

SYNOPSIS

nslookup [-option ...] *host-to-find* [*server*]

nslookup [-option ...] [- [*server*]]

DESCRIPTION

nslookup is a program to query Internet domain name servers. **nslookup** has been extended to follow the configured name resolution algorithm of the host and to query NIS, as well as, DNS and host tables.

Both an interactive and non-interactive mode are available with **nslookup**. Interactive mode allows the user to query a name server for information about various hosts and domains, or print a list of hosts in the domain. Non-interactive mode is used to query a name server for information about one host or domain.

By default, **nslookup** accesses name services for name and address resolution based on the policy information obtained from the switch configuration file `/etc/nsswitch.conf`. When the policy is set to use NIS or `/etc/hosts` first, or when DNS is first but unavailable, then **nslookup** will only provide a limited command set (a **help** command while in this situation will show what actions are possible when querying NIS or `/etc/hosts`). To override the switch policy and query DNS servers directly, the **server** command can be used to specify a nameserver. This same overriding of the switch policy can also be done by providing a nameserver as the second argument on the command line. In this case, **nslookup** will ignore the switch policy and directly query nameservers, until a *reset* command is issued. Whenever an action is taken that causes the switch policy to be overridden, a warning message is displayed.

Note, NIS+ is not supported by **nslookup**. If the hosts source **nisplus** is found in the `/etc/nsswitch.conf` file, it will be ignored.

ARGUMENTS

Interactive mode is entered in the following cases:

- No arguments are given.
- The first argument is a hyphen (-). The optional second argument is a host name or Internet (IP) address of a name server.

Non-interactive mode is used when the name of the host to be looked up is given as the first argument. The optional second argument is a host name or Internet address of a name server.

Options listed under the **set** command below can be specified one per line in the `.nslookuprc` file in the user's home directory. Alternatively, these options may be specified on the command line by prefixing them with a hyphen and they must precede other command line arguments. For example, to change the default query type to host information, and the initial timeout to 10 seconds, type:

```
nslookup -query=hinfo -timeout=10
```

The command line option **-swdebug** may be used to debug syntactic errors in the switch configuration file. This option turns on tracing during initialization, causing the switch module to print out a trace of the scan and parse actions on the "hosts" entry (see `nsswitch.conf(4)`) in the `/etc/nsswitch.conf` file.

Interactive Commands

Commands can be interrupted at any time by using the interrupt character. To exit, type a Ctrl-D (EOF) or type **exit**. To treat a built-in command as a host name, precede it with an escape character (\). When using NIS or the host table, only host names and Internet addresses are allowed as commands. An unrecognized command is interpreted as a host name.

host [*server*] Look up information for *host* using the current default server or using *server* if specified. If *host* is an Internet address and the query type is **A** or **PTR**, the name of the host is returned. If *host* is a name and does not have a trailing period, one or more domains are appended to the name (this behavior depends on the state of the **set** options **domain**, **srchlist**, **defname**, and **search**). Answers from a name server's cache are labeled "non-authoritative."

server *domain*

lserver *domain*

Change the default server to *domain*. **lserver** uses the initial server to look up information about *domain* while **server** uses the current default server. When **server** is used while the current name service being pointed to is either NIS or `/etc/hosts`, then the switch policy will be overridden until a **reset** is issued.

root

Changes the default server to the server for the root of the domain name space. Currently, the host `ns.nic.ddn.mil` is used (this command is a synonym for **lserverns.nic.ddn.mil**). The name of the root server can be changed with the **set root** command.

policy

Prints out the policy read from the switch configuration file. The number of name services specified in the file are shown, as well as the order and criteria on how the name services are to be used. The four statuses of the criteria are represented by the four positions within the square brackets. The order of the statuses are: SUCCESS, NOTFOUND, UNAVAIL and TRYAGAIN. The two actions of the criteria are represented by the two possible letters used in the four status positions: **R** for return and **C** for continue. However, if no criteria is specified between two sources, then the default actions are assigned to the statuses:

```
SUCCESS=      return
NOTFOUND=     continue
UNAVAIL=      continue
TRYAGAIN=     continue
```

finger [*name*] [> *filename*]

finger [*name*] [>> *filename*]

Connects with the finger server on the current host. The current host is defined when a previous lookup for a host was successful and returned address information (see the **set querytype=A** command). *name* is optional. **>** and **>>** can be used to redirect output in the usual manner.

ls [*option*] *domain* [> *filename*]

ls [*option*] *domain* [>> *filename*]

List the information available for *domain*, optionally creating or appending to *filename*. The default output contains host names and their Internet addresses. *option* can be one of the following:

- t** *querytype* lists all records of the specified type (see *querytype* below).
- a** lists aliases of hosts in the domain (synonym for **-t CNAME**).
- d** lists all records for the domain (synonym for **-t ANY**).
- h** lists CPU and operating system information for the domain (synonym for **-t HINFO**).
- s** lists well-known services of hosts in the domain (synonym for **-t WKS**).

When output is directed to a file, **#** characters are printed for every 50 records received from the server.

view *filename* Sorts and lists the output of previous **ls** command(s) using **more** (see *more(1)*).

help

? Prints a brief summary of commands.

exit

Exits the program.

reset

Returns to the use of the configured name service switch policy and resets to use the original nameservers.

set *keyword*[=*value*]

This command is used to change state information that affects the lookups. Valid keywords are:

all Prints the current values of the various options to **set**. Information about the current default server and host is also printed.

cl[*ass*]=*value* Change the query class to one of:

| n |

IN the Internet class.
CHAOS the Chaos class.
HESIOD the MIT Athena Hesiod class.
ANY wildcard (any of the above).

The class specifies the protocol group of the information. (Default = **IN**)

[no]deb[ug] Turn debugging mode on. More information is printed about the packet sent to the server and the resulting answer. (Default = **nodebug**)

[no]d2 Turn exhaustive debugging mode on. Essentially all fields of every packet are printed. (Default = **nod2**)

[no]def[name] If set, append the default domain name to a single-component lookup request (i.e., one that does not contain a period character). (Default = **defname**)

do[main]=name Change the default domain name to *name*. The default domain name is appended to a lookup request, depending on the state of the **defname** and **search** options. The domain search list contains the parents of the default domain if it has at least two components in its name. For example, if the default domain is **CC.Berkeley.EDU**, the search list is **CC.Berkeley.EDU** and **Berkeley.EDU**. Use the **set srchlist** command to specify a different list. Use the **set all** command to display the list. (Default = value from `hostname`, `/etc/resolv.conf` or **LOCALDOMAIN**)

[no]ig[nore] Ignore truncation errors. (Default = **noignore**)

q[querytype]=value

ty[pe]=value Change the type of information returned from a query to one of:

A Host's IPv4 address
AAAA Host's IPv6 address
ANY All types of data
CNAME Canonical name for an alias
GID Group ID
HINFO Host CPU and operating system type
MB Mailbox domain name
MG Mail group member
MINFO Mailbox or mail list information
MR Mail rename domain name
MX Mail exchanger
NS Name server for the named zone
PTR Host name if the query is an Internet address, otherwise the pointer to other information.
SOA Start of authority record
TXT Text information
UID User ID
UINFO User information
WKS Well-known service description

po[rt]=value Change the default TCP/UDP name server port to *value*. (Default = 53)

[no]rec[urse] Tell the name server to query other servers if it does not have the

- information. (Default = **recurse**)
- ret[ry]=number** Set the number of retries to *number*. When a reply to a request is not received within a certain amount of time (changed with **set timeout**), the timeout period is doubled and the request is resent. The retry value controls how many times a request is resent before giving up. (Default = 4)
- ro[ot]=host** Change the name of the root server to *host*. This affects the **root** command. (Default = **ns.nic.ddn.mil**)
- [no]sea[rch]** If the lookup request contains at least one period but doesn't end with a trailing period, append the domain names in the domain search list to the request until an answer is received. See *hostname(5)*. (Default = **search**)
- srchl[ist]=name1/name2/...** Change the default domain name to *name1* and the domain search list to *name1*, *name2*, etc. A maximum of 6 names separated by slashes (/) can be specified. For example,
- ```
set srchlist=lcs.MIT.EDU/ai.MIT.EDU/MIT.EDU
```
- sets the domain to **lcs.MIT.EDU** and the search list to the three names. This command overrides the default domain name and search list of the **set domain** command. Use the **set all** command to display the list. (Default = value based on *hostname*, */etc/resolv.conf* or **LOCALDOMAIN**)
- [no]swtr[ace]** When set, this flag causes **nslookup** to print out information about the sources used for resolving a name or an address lookup. This flag traces the behavior generated by the switch policy. (Default = **noswtrace**)
- t[imeout]=number** Change the initial timeout interval for waiting for a reply to *number* seconds. Each retry doubles the timeout period. (Default = 5 seconds)
- [no]v[c]** Always use a virtual circuit when sending requests to the server. (Default = **novc**)

The default behaviour of **nslookup** in the absence of **hosts** entry in the */etc/nsswitch.conf* file or in the absence of the */etc/nsswitch.conf* file is as follows:

```
hosts: dns nis files
```

## DIAGNOSTICS

If the lookup request was not successful, an error message is printed. Possible errors are:

### Time-out

The server did not respond to a request after a certain amount of time (changed with **set timeout=value**) and a certain number of retries (changed with **set retry=value**).

### No response from server

No name server is running on the server machine.

### No records

The server does not have resource records of the current query type for the host, although the host name is valid. The query type is specified with the **set querytype** command.

### Non-existent domain

The host or domain name does not exist.

### Connection refused

### Network is unreachable

The connection to the name server could not be made at the present time.

### Server failure

The name server found an internal inconsistency in its database and could not return a valid answer.

**Refused**

The name server refused to service the request.

**Format error**

The name server found that the request packet was not in the proper format.

**AUTHOR**

**nslookup** was developed by the University of California, Berkeley.

**FILES**

|                           |                                               |
|---------------------------|-----------------------------------------------|
| <b>/etc/resolv.conf</b>   | Initial domain name and name server addresses |
| <b>\$HOME/.nslookuprc</b> | User's initial options                        |

**SEE ALSO**

named(1M), resolver(3N), resolver(4), nsswitch.conf(4), hostname(5),  
RFC1034, RFC1035

**| n |**

**NAME**

nsquery - query the Name Service Switch backend libraries

**SYNOPSIS**

```
nsquery lookup_type lookup_query [lookup_policy]
```

**DESCRIPTION**

**nsquery** is used to find the Name Service that returned the response to a **gethostbyname()**, **gethostbyaddr()**, **getpwnam()**, **getpwuid()**, **getgrnam()**, or **getgrgid()** function call. This application is Name Service Switch aware and follows the lookup policies in **/etc/nsswitch.conf**. The lookup types supported are:

**hosts** Used to resolve host name or IP Address lookups.

**passwd** Used to resolve user name or UID lookups.

**group** Used to resolve group name or GID lookups.

The lookup query can either be a host name, IP Address, user name, user ID, group name or group ID.

The lookup policy must be a valid lookup policy described in **nsswitch.conf(4)**. If the policy is invalid, the system default policy will be used.

The default policies are:

**hosts** dns [NOTFOUND=return TRYAGAIN=return] nis [NOTFOUND=return] files

**passwd** files nis

**group** files nis

**nsquery** will display the lookup policy being used, the name of the service being queried, and the result of the query.

If the result of the query was successful, the appropriate structure will be displayed.

**Notes**

Changing the default behavior for SUCCESS is not recommended.

| n |

**EXAMPLES**

```
nsquery hosts hondo
```

```
nsquery hosts 15.204.204.204 "dns files"
```

```
nsquery passwd dog "nisplus"
```

```
nsquery passwd 105
```

```
nsquery group wayne "nis [NOTFOUND=RETURN] files"
```

```
nsquery group 22
```

**RETURN VALUE**

0: Success.

1: Invalid Usage.

2: Unknown ACTION.

3: No match found in any name services queried.

**AUTHOR**

**nsquery** was developed by Hewlett-Packard.

**SEE ALSO**

nsswitch.conf(4).

**NAME**

nsupdate - Dynamic DNS update utility

**SYNOPSIS**

```
nsupdate [-d] [-v] [-y keyname:secret | -k keyfile]
```

**DESCRIPTION**

**nsupdate** is used to submit Dynamic DNS Update requests as defined in RFC2136 to a name server. This allows resource records to be added or removed from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

Zones that are under dynamic control via **nsupdate** or a DHCP server should not be edited by hand. Manual edits could conflict with dynamic updates and cause data to be lost.

The resource records that are dynamically added or removed with **nsupdate** have to be in the same zone. Requests are sent to the zone's master server. This is identified by the MNAME field of the zone's SOA record.

**Options**

- d** This option makes **nsupdate** operate in debug mode. This provides tracing information about the update requests that are made and the replies received from the name server.
- Transaction signatures can be used to authenticate the Dynamic DNS updates. These use the TSIG resource record type described in RFC2845. The signatures rely on a shared secret that should only be known to **nsupdate** and the name server. Currently, the only supported encryption algorithm for TSIG is HMAC-MD5, which is defined in RFC 2104.
- Once other algorithms are defined for TSIG, applications will need to ensure they select the appropriate algorithm as well as the key when authenticating each other. For instance, suitable **key{}** and **server{}** statements would be added to **/etc/named.conf** so that the name server can associate the appropriate secret key and algorithm with the IP address of the client application that will be using TSIG authentication. **nsupdate** does not read **/etc/named.conf**.
- k** This option is used to provide the shared secret needed to generate a TSIG record for authenticating Dynamic DNS update requests. With this option, **nsupdate** reads the shared secret from the file *keyfile*, whose name is of the form **Kname.+157.+random.private**. For historical reasons, the file **Kname.+157.+random.key** must also be present. This option is mutually exclusive with the **-y** option.
- v** This option makes **nsupdate** use a TCP connection to send update requests to the name server. By default, **nsupdate** uses UDP to send update requests. This may be preferable when a batch of update requests is made.
- y** This option is used to generate a signature from *keyname:secret*. *keyname* is the name of the key, and *secret* is the base64 encoded shared secret. Use of the **-y** option is discouraged because the shared secret is supplied as a command line argument in clear text. This may be visible in the output from *ps(1)* or in a history file maintained by the user's shell.

**Input Format**

**nsupdate** reads commands from its standard input. Each command is supplied on exactly one line of input. Some commands are for administrative purposes. The others are either update instructions or prerequisite checks on the contents of the zone. These checks set conditions that some name or set of resource records (RRset) either exists or is absent from the zone. These conditions must be met if the entire update request is to succeed. Updates will be rejected if the tests for the prerequisite conditions fail.

Every update request consists of zero or more prerequisites and zero or more updates. This allows a suitably authenticated update request to proceed if some specified resource records are present or missing from the zone. A blank input line causes the accumulated commands to be sent as one Dynamic DNS update request to the name server.

The command formats and their meaning are as follows:

- server** *servername port*  
Sends all dynamic update requests to the name server *servername*. When no **server** statement is provided, **nsupdate** will send updates to the master server of the correct zone. The MNAME field of that zone's SOA record will identify the master server for that zone. *port* is the port number on *servername* where the dynamic update requests get sent. If no port number is specified, the default DNS port number of 53 is used.
- zone** *zonename*  
Specifies that all updates are to be made to the zone *zonename*. If no **zone** statement is provided, **nsupdate** will attempt to determine the correct zone to update based on the rest of the input.
- prereq nxdomain** *domain-name*  
Requires that no resource record of any type exists with name *domain-name*.
- prereq yxdomain** *domain-name*  
Requires that **domain-name** exists (has as at least one resource record, of any type).
- prereq nxrrset** *domain-name class type*  
Requires that no resource record exists of the specified *type*, *class*, and *domain-name*. If *class* is omitted, **IN** (Internet) is assumed.
- prereq yxrrset** *domain-name class type*  
This requires that a resource record of the specified *type*, *class*, and *domain-name* must exist. If *class* is omitted, **IN** (Internet) is assumed.
- prereq yxrrset** *domain-name class type data...*  
The *data* from each set of prerequisites of this form sharing a common *type*, *class*, and *domain-name* are combined to form a set of RRs (resource records). This set of RRs must exactly match the set of RRs existing in the zone at the given *type*, *class*, and *domain-name*. The *data* are written in the standard text representation of the resource record's RDATA.
- update delete** *domain-name class type data...*  
Deletes any resource records named *domain-name*. If *type* and *data* is provided, only matching resource records will be removed. The Internet class (**IN**) is assumed if *class* is not supplied.
- update add** *domain-name ttl class type data..*  
Adds a new resource record with the specified *ttl*, *class* and *data*.
- local** *address port*  
Sends all dynamic update requests using the local address. When no **local** statement is provided, **nsupdate** will send updates using an *address* and *port* chosen by the system. *port* can additionally be used to set a specific port from where requests are sent. If port number is not specified, the system will assign one.
- key** *name secret*  
Specifies that all updates need to be TSIG signed using the *keyname keysecret* pair. The **key** command overrides any key specified on the command line via **-y** or **-k**.
- show**  
Displays the current message, containing all the pre-requisites and updates specified since the last **send** operation.
- send**  
Sends the current message. This is equivalent to entering a blank line.

**EXAMPLES**

The examples below show how **nsupdate** could be used to insert and delete resource records from the **example.com** zone. Notice that the input in each example contains a trailing blank line so that a group of commands are sent as one dynamic update request to the master name server for **example.com**.

```
nsupdate
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
>
```

Any A records for **oldhost.example.com** are deleted. An A record for **newhost.example.com** with IP address 172.16.1.1 is added. The newly-added record has a 1 day TTL (86400 seconds)

```
nsupdate
> prereq nxdomain nickname.example.com
> update add nickname.example.com CNAME somehost.example.com
>
```

The prerequisite condition gets the name server to check that there are no resource records of any type for **nickname.example.com**. If there are, the update request fails. If this name does not exist, a CNAME for it is added. This ensures that when the CNAME record is added, it cannot conflict with the long-standing rule in RFC1034 that a name must not exist as any other record type if it exists as a CNAME.

(The rule has been updated for DNSSEC in RFC2535 to allow CNAMEs to have SIG, KEY and NXT records.)

#### FILES

|                                   |                                                                    |
|-----------------------------------|--------------------------------------------------------------------|
| <b>/etc/resolv.conf</b>           | Used to identify default name server.                              |
| <b>Kname.+157.+random.key</b>     | Base-64 encoding of HMAC-MD5 key created by <b>dnssec-keygen</b> . |
| <b>Kname.+157.+random.private</b> | Base-64 encoding of HMAC-MD5 key created by <b>dnssec-keygen</b> . |

#### SEE ALSO

RFC2136 , RFC2137 , RFC2104 , RFC2845 , RFC1034 , RFC2535 , named(1M), dnssec-keygen(1).

#### BUGS

The TSIG key is redundantly stored in two separate files. This is a consequence of **nsupdate** using the DST library for its cryptographic operations, and may change in future.

| **n** |



**NAME**

od, xd - octal and hexadecimal dump

**SYNOPSIS**

**od** [-v] [-A *address\_base*] [-j *skip*] [-N *count*] [-t *type\_string*] ... [*file* ...]

**xd** [-v] [-A *address\_base*] [-j *skip*] [-N *count*] [-t *type\_string*] ... [*file* ...]

**Supported Pre-POSIX Usage**

**od** [-bcdosx] [*file*] [[+][0x]offset[.][b]]

**xd** [-bcdosx] [*file*] [[+][0x]offset[.][b]]

**DESCRIPTION**

**od** and **xd** concatenate one or more input *files* and write their contents to standard output in a user-specified format. If *file* is not specified, the standard input is used.

**Options and Arguments**

**od** and **xd** recognize the following options and command-line arguments:

**-A** *address\_base* Specify the input offset base. *address\_base* is a single character that defines which format the offset base is written in:

- d** Decimal format.
- o** Octal format.
- x** Hexadecimal format.
- n** Do not write the offset.

**-j** *skip* Jump over *skip* bytes from the beginning of the input. **od** seeks past the first *skip* bytes in the concatenated input files. If the combined input is not at least *skip* bytes long, **od** writes a diagnostic message to standard error and exits with a non-zero exit status. By default, *skip* is interpreted as a *decimal* number. If *skip* has a leading **0x** or **0X**, it is interpreted as a *hexadecimal* number; a leading **0** indicates that *skip* is an *octal* number.

If the value of *skip* is followed by a **b**, **k**, or **m**, it is interpreted as a multiple of 512, 1024, or 1048576, respectively.

**-N** *count* Format no more than *count* bytes of input.

By default, *count* is interpreted as a *decimal* number. A leading **0x** or **0X** indicates that *count* is a *hexadecimal* number; a leading **0** identifies an *octal* value.

If *count* bytes of input are not available (after successfully skipping if **-jskip** is specified), the input that is available is formatted.

**-t** *type\_string* *type\_string* is a string defining the types to be used when writing the input data.

The string can contain any of the following type-specification characters:

- a** named character ,
- c** character ,
- d** signed decimal ,
- f** floating point ,
- o** octal ,
- u** unsigned decimal ,
- x** hexadecimal ,

Type specification characters **d**, **f**, **o**, **u**, and **x** can be followed by an optional *unsigned decimal* integer specifying the number of bytes to be transformed by each instance of the output type, or by an optional **C**, **S**, **I**, or **L** indicating that the conversion should be applied to an item of type *char*, *short*, *int*, or *long*, respectively.

Type specification character **f** can be followed by an optional **F**, **D**, or **L** indicating that the conversion should be applied to an item of type *float*, *double*, or *long double*, respectively.

Multiple types can be concatenated within the same *type\_string* and multiple **-t** options can be specified. Output lines are written for each type specified in

- the order in which the type specification characters appear.
- v** Write all input data. Without the **-v** option, any number of groups of output lines, that would be identical to the immediately preceding group of output lines (except for the byte offsets), are replaced with a line containing only an asterisk (\*).
- file* Pathname of one or more input files to be processed. If *file* is not specified, the standard input is used.
- Input files can be any file type.

### DESCRIPTION OF PRE-POSIX USAGE

**od** and **xd** dump *file* in one or more formats as selected by the first argument. If the first argument is missing, the default is **-o** for **od**; **-x** for **xd**. An offset field is inserted at the beginning of each line. For **od**, the offset is in octal, for **xd**, the offset is in hexadecimal.

### Options

**od** and **xd** recognize the following format options:

- b** Interpret bytes in octal (hexadecimal).
- c** Interpret bytes in ASCII. Certain non-graphic characters appear as C escapes: null=**\0**, backspace=**\b**, form-feed=**\f**, new-line=**\n**, return=**\r**, tab=**\t**; others appear as 3-digit octal numbers.
- d** Interpret 16-bit words in decimal.
- o** Interpret 16-bit words in octal.
- s** Interpret 16-bit words in signed decimal.
- x** Interpret 16-bit words in hexadecimal.

*file* specifies which file is to be dumped. If *file* is not specified, the standard input is used.

*offset* specifies the offset in the file where dumping is to commence, and is normally interpreted as octal bytes. Interpretation can be altered as follows:

- *offset* must be preceded by **+** if the file argument is omitted.
- *offset* preceded by **0x** is interpreted in hexadecimal.
- *offset* followed by **.** is interpreted in decimal.
- *offset* followed by **b** is interpreted in blocks of 512 bytes.

Dumping continues until end-of-file.

### EXAMPLES

Write hexadecimal bytes and the corresponding octal values to the standard output in blocks of 16 bytes in one line, by transforming the data from the input file **file1**:

```
od -tx1oC file1
```

The following commands write one line each of the types *character*, *signed decimal integer*, and *float*, in the order given, transforming 100 bytes of data starting from fifteenth byte offset in the file **file1**:

```
od -j14 -N100 -tc -tdfF file1
od -j0xe -N100 -tcd4fF file1
```

Write one line each of the types *unsigned integer*, *named character*, and *long double*, with the offsets written in hexadecimal and forcing a write, even on lines that are identical to the immediately preceding group of output lines:

```
od -v -Ax -tuafL file1
```

### WARNINGS

When the output format is of floating-point type; i.e., when using the **-t fD**, **-t fL**, or **-t f** options:

- If the input bytes cannot be transformed into a valid floating point number, a floating point exception might occur. In that case, the output is printed as a string containing some non-numeric characters and program execution continues.
- When the number of input bytes used for transformation is set to 1 with the type specifier characters **d**, **o**, **u**, or **x**, only the least-significant seven bits of each byte are used.

- When one or more of the **-A**, **-j**, **-N**, or **-t** options is specified, an operand starting with the first character as a plus-sign (+) or the first character as numeric is interpreted as a file name.

(XPG4 only. Multiple types can be specified by using multiple **-bcdox** options. Output lines are written for each type specified in the order in which the types are specified.)

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** provides a default value for the internationalization variables that are unset or null. If **LANG** is unset or null, the default value of "C" (see *lang(5)*) is used. If any of the internationalization variables contains an invalid setting, **od** will behave as if all internationalization variables are set to "C". See *environ(5)*.

**LC\_ALL** If set to a non-empty string value, overrides the values of all the other internationalization variables.

**LC\_CTYPE** determines the interpretation of text as single and/or multi-byte characters, the classification of characters as printable, and the characters matched by character class expressions in regular expressions.

**LC\_MESSAGES** determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

**NLSPATH** determines the location of message catalogues for the processing of **LC\_MESSAGES**.

### International Code Set Support

Single- and multi-byte character code sets are supported. Multi-byte data is displayed as multi-byte values.

## RETURN VALUE

Exit values are:

- 0 Successful completion.
- >0 Error condition occurred.

## SEE ALSO

adb(1).

## STANDARDS CONFORMANCE

**od**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| o |

**NAME**

odump - dump information contained in SOM object files

**SYNOPSIS**

**odump** [-option ...] *file*...

**DESCRIPTION**

**odump** takes one or more 32-bit native code *files* (object files, shared and archive libraries, executables) and dumps information about them.

**Note:** **odump** skips non-SOM input files and issues a warning.

```
$ odump -all /usr/lib/pa20_64/libc.sl
```

**File is not a SOM, skipping: /usr/lib/pa20\_64/libc.sl.**

**Options**

Some **odump** options have additional support or provide additional information when used in combination with the modifier options:

- V (VERIFY)** Indicate that the structure specified by its corresponding option is verified when run with the **-verifyall** option.
- T (TITLES)** Provides additional output format information for the structure specified by its corresponding option when run with the **-titles** option.
- B (VERBOSE)** Prints a more descriptive message for the structure specified by its corresponding option when run with the **-verbose** option.

The table below indicates the available support for each option.

**odump** recognizes the following options and their modifiers:

| <b>V</b> | <b>T</b> | <b>B</b> | <b>-Option</b>         | <b>Description</b>                                                    |
|----------|----------|----------|------------------------|-----------------------------------------------------------------------|
| -        | -        | -        | <b>-all</b>            | Print all information                                                 |
| V        | -        | -        | <b>-auxheader</b>      | Print SOM auxiliary headers                                           |
| V        | -        | -        | <b>-auxunwind</b>      | Print the stack auxiliary unwind table                                |
| -        | -        | -        | <b>-compunit</b>       | Print compilation unit dictionary                                     |
| -        | -        | -        | <b>-doclinetable</b>   | Print the line tables for debugging optimized code                    |
| -        | -        | -        | <b>-filtertable</b>    | Print the list of implementation libraries for a given filter library |
| -        | -        | B        | <b>-fixups</b>         | Print fixups                                                          |
| V        | -        | -        | <b>-header</b>         | Print SOM header                                                      |
| -        | -        | -        | <b>-help</b>           | Print this help screen                                                |
| V        | T        | -        | <b>-initpointers</b>   | Print initialization pointers                                         |
| -        | -        | -        | <b>-linetable</b>      | Print the line number table                                           |
| -        | -        | -        | <b>-linkmap</b>        | Print the linkmap contents                                            |
| -        | -        | -        | <b>-linkmapbss</b>     | Print the linkmap bss contents                                        |
| -        | -        | -        | <b>-linkmapfile</b>    | Print the linkmap object file table                                   |
| -        | -        | -        | <b>-linkmapstr</b>     | Print the linkmap string table                                        |
| -        | -        | -        | <b>-loaderfixups</b>   | Print loader fixup records                                            |
| -        | -        | -        | <b>-lstauxheader</b>   | Print LST auxheader                                                   |
| -        | -        | -        | <b>-lstexportlist</b>  | Print LST exportlist                                                  |
| -        | -        | -        | <b>-lstheader</b>      | Print LST header                                                      |
| -        | -        | -        | <b>-lstimportlist</b>  | Print LST importlist                                                  |
| -        | -        | -        | <b>-lstsomdir</b>      | Print LST som directory                                               |
| -        | -        | -        | <b>-lststringtable</b> | Print LST stringtbl                                                   |

|   |   |   |                        |                                                        |
|---|---|---|------------------------|--------------------------------------------------------|
| - | - | - | <b>-nclist</b>         | Treat remaining options as nclist options              |
| - | - | - | <b>-octoutput</b>      | Treat remaining input files as OCT output              |
| V | - | - | <b>-recover</b>        | Print the recover table                                |
| - | - | - | <b>-silent</b>         | Don't print tables, to be used with verify option      |
| - | - | - | <b>-sldatalinktbl</b>  | Print the shared lib data linkage table                |
| V | - | - | <b>-sldlexporthash</b> | Print the shared lib export entries hashed values      |
| V | - | - | <b>-sldlheader</b>     | Print the shared lib DL header                         |
| V | - | - | <b>-sldlheaderext</b>  | Print the shared lib dl_header_ext structure           |
| V | - | - | <b>-sldlimporthash</b> | Print the shared lib import entries hashed values      |
| V | - | - | <b>-sldlplabelhash</b> | Print the shared lib plabel entries hashed values      |
| V | - | - | <b>-sldynamrel</b>     | Print the shared lib dynamic relocation records        |
| V | - | - | <b>-slexportext</b>    | Print the shared lib export list extension             |
| V | - | - | <b>-slexportlist</b>   | Print the shared lib export list                       |
| - | - | - | <b>-slfastbind</b>     | Print the fastbind tables                              |
| - | - | - | <b>-slfbdynamrel</b>   | Print the fastbind dynamic relocation list             |
| - | - | - | <b>-slfbhdr</b>        | Print the fastbind header                              |
| - | - | - | <b>-slfbimpexp</b>     | Print the fastbind import export list                  |
| - | - | - | <b>-slfbllibver</b>    | Print the fastbind library version list                |
| - | - | - | <b>-slfbmodbound</b>   | Print the fastbind module bound list                   |
| - | - | - | <b>-slhashtbl</b>      | Print the shared lib export hash table                 |
| V | - | - | <b>-slimportlist</b>   | Print the shared lib import list                       |
| - | T | - | <b>-sllliblist</b>     | Print the list of shared libraries used in build       |
| - | - | - | <b>-slllibloadlist</b> | Print the shared lib load list, including dependencies |
| - | T | - | <b>-slllibsearch</b>   | Print the shared lib dynamic search status             |
| V | - | - | <b>-slmodule</b>       | Print the shared lib module table                      |
| V | - | - | <b>-slmoduledyn</b>    | Print the shared lib module table relocation list      |
| V | - | - | <b>-slmoduleimp</b>    | Print the shared lib module table import list          |
| - | - | - | <b>-slproclinktbl</b>  | Print the shared lib procedure linkage table           |
| - | - | - | <b>-slstringtable</b>  | Print the shared lib string table                      |
| - | - | - | <b>-slunwindhdr</b>    | Print the shared lib unwind header                     |
| V | - | B | <b>-somap</b>          | Print map of SOM structures                            |
| V | - | - | <b>-spaces</b>         | Print space list                                       |
| - | - | - | <b>-spstringtable</b>  | Print string table for space dictionary                |
| V | - | - | <b>-subspaces</b>      | Print subspace list                                    |
| V | - | - | <b>-symbols</b>        | Print all symbols                                      |
| - | - | - | <b>-symstringtable</b> | Print string table for symbol dictionary               |
| - | - | - | <b>-titles</b>         | Print full names in all titles                         |
| - | - | - | <b>-tlssym</b>         | Print the TLS symbol references                        |
| - | - | - | <b>-toolbssmap</b>     | Print the linker bss subspace map                      |
| - | - | - | <b>-toolcompmap</b>    | Print the linker compilation unit map                  |

| o |

```

- - - -toolsubspmap Print the linker subspace map
V T - -unwind Print the stack unwind table
- - - -usage Print the help screen
- - - -verbose # Set verbosity level for tables that follow,
 0=default
- - - -verify Verify correctness of tables before they are
 printed
- - - -verifyall Verify correctness of tables that have verify
 support
- - - -version Print date of current version

```

## RETURN VALUE

**odump** exits with one of the following values:

- 0: All input files were listed successfully.
- >0: **odump** aborted because errors occurred when accessing files. The following conditions cause an error:
- Specified file not found.
  - User has no permission to read the directory.
  - Process could not get enough memory.
  - Invalid option specified.

## EXTERNAL INFLUENCES

### Environment Variables

The following variable affects the execution of **odump**:

**SHLIB\_PATH** A colon separated list of path names used to search for shared libraries. Used by **-sllibsearch** and **-sllibloadlist** options.

## EXAMPLES

To see what compiler options and version an executable was compiled with:

```
$ odump -compunit a.out
```

To list the shared libraries (.sl) that a program loads, even if **SHLIB\_PATH** is set:

```
$ odump -sllibload a.out
```

To see how much space a program needs in Quadrant 2 to hold the data, bss, stack, and other areas, look at the values in the **som\_exec\_auxhdr**:

```
$ odump -auxheader a.out
```

### Auxiliary Header Record (HP-UX) for a.out:

```

mandatory = 0 copy = 0
append = 0 ignore = 1
type = 4 length = 40
exec_tsize = 0x000023ec exec_tmemo = 0x00001000
exec_tfile = 0x00002000 exec_dsize = 0x00001000
exec_dmem = 0x40001000 exec_dfile = 0x00005000
exec_bsize = 0x3bfff120 exec_entry = 0x000029c8
exec_flags = 0x00000004 exec_bfill = 0x00000000

```

To see if a program was built for debugging with **-g** and not for optimization with **-O**, look for a **\$DEBUG\$** space (programs built for both show **\$PINFOS\$** instead of **\$DEBUG\$**):

```
$ odump -spaces a.out
```

```
Space dictionary for a.out:
```

| Ind | LDPIT | Sort | Space | Subspaces | Ldr | Fixups | Init | Ptrs | Name        |
|-----|-------|------|-------|-----------|-----|--------|------|------|-------------|
| 0   | LD... | 8    | 1     | 0 8       |     | -1 0   | 0    | 1    | \$TEXT\$    |
| 1   | LDP.. | 16   | 0     | 8 9       |     | -1 0   | 1    | 1    | \$PRIVATE\$ |
| 2   | .DP.. | 80   | 2     | 17 5      |     | -1 0   | -1   | 0    | \$DEBUG\$   |

#### AUTHOR

`odump` was developed by HP.

#### SEE ALSO

##### System Tools

*ld*(1) Invoke the link editor

##### Miscellaneous

*a.out*(4) Assembler, compiler, and linker output  
*elfdump*(1) Dump utility for ELF object format  
*elf*(3E) Executable and Linking Format

##### Texts and Tutorials

*HP-UX Linker and Libraries Online User Guide*  
 (See the **+help** option)  
*HP-UX Linker and Libraries User's Guide*  
 (See *manuals*(5) for ordering information)

**NAME**

**on** - execute command on remote host with environment similar to local

**SYNOPSIS**

**on** [-i | -n] [-d] *host* [*command* [*argument*] ... ]

**DESCRIPTION**

**on** executes a command on a remote host, using an environment similar to that of the invoking user where:

- host* specifies the name of the host on which to execute the command.
- command* specifies the command to execute on *host*

If *command* is not specified, **on** starts a shell on *host*. *argument* ... is a list of arguments for *command*.

The user's environment variables are copied to the remote host, and the file system containing the user's current working directory is NFS mounted on the remote host (see *nfs(7)*). The command is executed on the remote host in the user's current working directory.

Commands using relative path names that reference file system objects within the user's current working file system have the same behavior as running the command on the client. The behavior of commands using relative path names that cross the file system boundary or commands using absolute path names depends on the organization of the remote host's file system.

Implicit and explicit use of environment variables may also cause a command's behavior to be dependent on the organization of the remote host's file system. For example, the **\$PATH** environment variable usually contains absolute path names.

Standard input, output and error of the remote command are connected to the appropriate file descriptors on the client.

The remote execution daemon (**rex**d) does not allow **root** to execute a remote command.

The signals **SIGINT**, **SIGTERM**, and **SIGQUIT** are propagated to the remote command. **SIGTSTP** and **SIGSTOP** are ignored by the remote command. All other signals are delivered to the **on** command.

In order to execute a remote command, the remote host must be configured to execute **rex**d (see *rex*d(1M)).

**Options**

**on** recognizes the following options:

- i** Interactive mode. This option is required for commands that must communicate with a terminal such as **vi**, **ksh**, or **more**. Terminal mode changes are propagated to the **rex**d server. The standard input for an interactive **on** command must be a tty device. The **-i** and **-n** options are mutually exclusive.
- d** Debug mode. Print diagnostic messages during startup of the **on** command. These messages are useful for detecting configuration problems if the **on** command to a specific host is failing.
- n** No input mode. This option causes the remote command to get end-of-file (EOF) when it reads from standard input, instead of connecting the standard input of the **on** command to the standard input of the remote command. The **-n** option is required when running commands in the background. The **-n** and **-i** options are mutually exclusive.

**DIAGNOSTICS**

**on: unknown host** *host*

The host name *host* was not found in the hosts database.

**on: cannot connect to server on** *host*

The host *host* is down, unreachable on the network, or not running **rex**d.

**on: can't find** *current\_dir*

A problem occurred trying to find the user's current working directory (*current\_dir*).

**on: can't locate mount point for** *current\_dir*

A problem occurred trying to determine the mount point of the user's current working directory (*current\_dir*).



**on: standard input (stdin) is not a tty**

The standard input (stdin) of the **on** command with the **-i** option is not a tty device.

**on server : rexd: message**

Errors that occur on the server *server* are propagated back to the client. These messages are documented in the DIAGNOSTICS section of *rexd(1M)*.

**AUTHOR**

**on** was developed by Sun Microsystems, Inc.

**SEE ALSO**

*exports(4)*, *rexd(1M)*.

**| o |**

**NAME**

pack, pcat, unpack - compress and expand files

**SYNOPSIS**

**pack** [-] [-f] *name* ...

**pcat** *name* ...

**unpack** *name* ...

**DESCRIPTION**

**pack** attempts to store the specified files in a compressed form. Wherever possible, each input file *name* is replaced by a packed file *name.z* with the same ownership, modes, and access and modification times. The **-f** option forces packing of *name*. This is useful for causing an entire directory to be packed even if some of the files do not benefit. If **pack** is successful, *name* is removed. Packed files can be restored to their original form using **unpack** or **pcat**.

**pack** uses Huffman (minimum redundancy) codes on a byte-by-byte basis. If the **-** argument is used, an internal flag is set that causes the number of times each byte is used, its relative frequency, and the code for the byte to be printed on the standard output. Additional occurrences of **-** in place of *name* cause the internal flag to be set and reset.

The amount of compression obtained depends on the size of the input file and the character frequency distribution. Because a decoding tree forms the first part of each *.z* file, it is usually not worthwhile to pack files smaller than three blocks unless the character frequency distribution is very skewed such as in printer plots or pictures.

Typically, text files are reduced to 60-75% of their original size. Load modules, which use a larger character set and have a more uniform distribution of characters, show little compression, the packed versions being about 90% of the original size.

**pack** returns a value that is the number of files that it failed to compress.

No packing occurs if:

- The file appears to be already packed.
- The file name has more than 12 characters and the file system is configured as a short filename system.
- The file has links.
- The file is a directory.
- The file cannot be opened.
- The file is empty.
- No disk storage blocks will be saved by packing.
- A file called *name.z* already exists.
- The *.z* file cannot be created.
- An I/O error occurred during processing.

On short-filename systems, the last segment of the file name must contain no more than 12 characters to allow space for the appended *.z* extension. Directories cannot be compressed.

**pcat** does for packed files what *cat*(1) does for ordinary files, except that **pcat** cannot be used as a filter. The specified files are unpacked and written to the standard output. Thus to view a packed file named *name.z* use:

```
pcat name.z
```

or simply:

```
pcat name
```

To make an unpacked copy (named *file*) of a packed file named *name.z* without destroying *name.z* use the command:

```
pcat name >file
```

**pcat** returns the number of files it was unable to unpack. Failure may occur if:

- The file name (exclusive of the *.z*) has more than 12 characters;
- The file cannot be opened;
- The file does not appear to have been created by *pack*.

**unpack** expands files created by **pack**. For each file *name* specified in the command, a search is made for a file called *name.z* (or just *name* if *name* ends in *.z*). If this file appears to be a packed file, it is replaced by its expanded version. The new file has the *.z* suffix stripped from its name, and has the same access modes, access and modification dates, and owner as those of the packed file.

**unpack** returns a value that is the number of files it was unable to unpack. Failure may occur for the reasons given for **pcat**, as well as for the following:

- A file with the “unpacked” name already exists;
- The unpacked file cannot be created.

#### Access Control Lists (ACLs)

**pack** retains all entries in a file’s access control list when compressing and expanding it (see *acl(5)*).

#### DEPENDENCIES

##### NFS

Optional access control list entries of networked files are summarized (as returned in **st\_mode** by **stat()**), but not copied to the new file (see *stat(2)*).

#### WARNINGS

This command is likely to be withdrawn from X/Open standards. Applications using this command might not be portable to other vendors’ systems. Instead of **pack** it is recommended to use **compress** utility as it has the following advantages:

- The algorithm used to create the output files is frequently more effective in reducing the size of files
- The compress utility can compress data from its standard input, not just a named regular file. Thus it is useful in pipelines

**zcat** is recommended instead of **pcat** and, **uncompress** is recommended instead of **unpack**.

#### SEE ALSO

*cat(1)*, *compact(1)*, *compress(1)*, *acl(5)*.

#### STANDARDS CONFORMANCE

**pack**: SVID2, SVID3, XPG2, XPG3

**pcat**: SVID2, SVID3, XPG2, XPG3

**unpack**: SVID2, SVID3, XPG2, XPG3

**NAME**

parstatus - display information about a hardware partitionable complex

**SYNOPSIS**

```

parstatus
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

parstatus -s
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

parstatus -w
 [-u username :[passwd] -h IPaddress | hostname]

parstatus -X
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

parstatus -C | -I [-A] [-M]
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

parstatus -B | -P [-M]
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

parstatus -i I/Ochassis [-i ...] [-M]
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

parstatus -c cell [-c ...] [-V] [-M]
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

parstatus -b cabinet [-b ...] [-V] [-M]
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

parstatus -p PartitionNumber [-p ...] [-V] [-M]
 [-u username :[passwd] -h IPaddress | hostname
 | -g [passwd] -h IPaddress | hostname]

```

**DESCRIPTION**

The **parstatus** command displays information about a system complex. If no arguments are supplied, all major components that are physically present in the local complex are listed. A specific component (cell, I/O chassis, cabinet, or partition) may be specified in order to restrict the output to information about that component. Either the **-u** option or the **-g** option can be specified to access a remote complex.

Please refer to the *HP System Partitions Guide* for a description of the partition management terms used in this man page.

**Options and Arguments**

**parstatus** recognizes the following command line options and arguments:

- s** Indicates if the command is being executed on a hardware partitionable platform. When this option is specified, the command will either succeed or fail (see *RETURN VALUE* below) and nothing will be displayed on stdout or stderr.
- u *username* :*[passwd]***  
Specifies the required authorization to access a partition other than the local partition (but can also be used as a loopback access to the local partition). If complex attributes are displayed, the complex is the one in which this target partition resides.  
  
The **-h** option is required if this option is used.  
  
*username* specifies a configured user name on the target partition.  
  
*passwd* specifies the password associated with the *username*. If this field is empty, the command prompts for the password.

*Note:* This command is a Web-Based Enterprise Management (WBEM) Client Application. The **-u** option accesses the target partition using a Secure Sockets Layer (SSL) connection. If errors are reported, check that the conditions described in the *DEPENDENCIES* section are satisfied.

**-h** *IPaddress|hostname*

This option should only be used in combination with either the **-u** or **-g** option. *IPaddress|hostname* specifies the IP address or hostname of the target partition (**-u**) or complex (**-g**).

**-g** [*passwd*]

Allows access to the complex specified by the **-h** option. The accessed complex is then considered the target complex. Access is via the service processor's LAN port.

The **-h** option is required if this option is used.

*passwd* specifies the IPMI password of the service processor. If this field is omitted, the command prompts for the password.

If an error is reported when you attempt to connect using this option, check to see that IPMI LAN access has not been disabled on the remote service processor. Access to the complex via IPMI over LAN can be enabled or disabled by logging on to the service processor and using the **SA** command from the Command Menu.

The **-u** and **-g** options are mutually exclusive.

**-w** Display the local partition number in a full sentence. If the **-u** option connects to another partition, that partition number is reported instead. The **-M** option restricts the display to just the partition number.

**-X** Display the complex attributes.

**-A** Only display the available resources in the complex.

**-V** Verbose mode. Increase the amount of information displayed. Additional information displayed with **-V** includes:

For Cabinet:

- \* Status of each of the cabinet blower(s)
- \* Status of each of the I/O fan(s)
- \* Status of each of the bulk power supply(s)
- \* Status of each of the backplane power board(s)

For Partition:

- \* Primary, alternate and secondary boot path settings
- \* PDC, IODC version numbers
- \* Memory interleaving / local memory settings
- \* Cells summary info
- \* I/O chassis summary info

For Cell:

- \* Details of each cpu on the cell
- \* Details of memory on the cell
- \* Memory interleaving / local memory settings

**-M** Produce a machine readable/parseable output, or restrict the output of the **-w** option to a partition number. The machine readable output will have the columns separated by a single colon character (:) and will not have a column header.

**-C** Show information for all cells in the complex.

**-I** Show information for all I/O chassis in the complex.

**-B** Show information for all cabinets in the complex.

**-P** Show information for all partitions in the complex.

**-c** *cell* Show information about the specified cell. The *cell* id can be specified either in the local (*cabinet#/slot#*) or global (*cell\_ID*) format. For example, the cell located in cabinet 0, slot 1 is locally identified as 0/1 or globally as 1.

**-i I/Ochassis**

Show information about the specified *I/Ochassis*. The *I/Ochassis* id can be specified in the form of *cabinet#/enclosure#/chassis#*. For example, the I/O chassis located in cabinet 0, enclosure 1 and I/O chassis slot 3 is identified as 0/1/3.

**-b cabinet**

Show information about the specified cabinet.

**-p PartitionNumber**

Show information about the specified partition. *PartitionNumber* is the unique partition number (integer) which was assigned to the partition when it was created.

**Mapping of Global Cell Numbers to Local Cell Numbers**

The cabinets in a complex are numbered starting from 0. The cell slots in each cabinet are also numbered starting from 0. Each cabinet can have a maximum of 8 cells. For example, the cells located in cabinet 0 will have the following cell numbers in global format: 0, 1, 2, 3, 4, 5, 6, 7. The cell numbers in corresponding local format will be 0/0, 0/1, 0/2, 0/3, 0/4, 0/5, 0/6, 0/7.

Similarly the cells located in cabinet 1 will have the following cell numbers in global format: 8, 9, 10, 11, 12, 13, 14, 15. The cell numbers in corresponding local format will be 1/0, 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7.

From the above convention the cell located in cabinet 1, slot 0 is identified in the local format as 1/0 or in the global format as 8. The *parstatus*(1) command will display the above cell as "cab1,cell0". The cell located in cabinet 1, slot 4 is identified in the local format as 1/4 or in the global format as 12. The *parstatus*(1) command will display the above cell as "cab1,cell4".

**RETURN VALUE**

The **parstatus** command exits with one of the following values:

- 0 Successful completion.
- 1 Error condition occurred.

**EXAMPLES**

**Example 1**

Display information for all the I/O chassis in the complex:

```
parstatus -I
```

The output format is as shown below.

```
[Chassis]
```

| Hardware Location  | Usage    | Core IO | Connected To | Par Num |
|--------------------|----------|---------|--------------|---------|
| cab0,bay0,chassis0 | Absent   | -       | -            | -       |
| cab0,bay0,chassis1 | Active   | yes     | cab0,cell10  | 0       |
| cab0,bay0,chassis2 | Absent   | -       | -            | -       |
| cab0,bay0,chassis3 | Active   | yes     | cab0,cell14  | 2       |
| cab0,bay1,chassis0 | Absent   | -       | -            | -       |
| cab0,bay1,chassis1 | Active   | yes     | cab0,cell16  | 1       |
| cab0,bay1,chassis2 | Absent   | -       | -            | -       |
| cab0,bay1,chassis3 | Inactive | yes     | cab0,cell12  | 1       |

**Example 2**

The following example displays information for all major components that are physically present in the local complex. No action is specified for the command; the default behaviour is to display all.

```
parstatus
```

```
[Complex]
```

```
Complex Name : SuperDomeComplex
Complex Capacity
 Compute Cabinet (8 cell capable) : 1
Active MP Location : cabinet 0
Model : 9000/800/SD32000
Original Serial Number : USR4001WXY
Current Product Order Number : 12345B
```

Original Manufacturer : HP  
 Complex Profile Revision : 1.0  
 The total number of partitions present : 3

[Cabinet]

| Cabinet          | I/O      | Bulk Power | Backplane    |          |        |
|------------------|----------|------------|--------------|----------|--------|
| Blowers          | Fans     | Supplies   | Power Boards |          |        |
| OK/              | OK/      | OK/        | OK/          |          |        |
| Cab              | Failed/  | Failed/    | Failed/      | Failed/  |        |
| Num Cabinet Type | N Status | N Status   | N Status     | N Status | MP     |
| ====             | =====    | =====      | =====        | =====    | =====  |
| 0 8 cell slot    | 4/0/N+   | 5/0/NA     | 5/0/N+       | 3/0/N+   | Active |

Notes: N+ = There are one or more spare items (fans/power supplies).  
 N = The number of items meets but does not exceed the need.  
 N- = There are insufficient items to meet the need.  
 ? = The adequacy of the cooling system/power supplies is unknown.  
 HO = Housekeeping only; The power is in a standby state.  
 NA = Not Applicable.

[Cell]

| Hardware Location | Actual Usage | CPU OK/ Deconf/ Max | Memory (GB) OK/ Deconf | Connected To       | Core Cell Capable | Use On Next Boot | Par Num |
|-------------------|--------------|---------------------|------------------------|--------------------|-------------------|------------------|---------|
| ====              | =====        | =====               | =====                  | =====              | =====             | =====            | =====   |
| cab0,cell10       | Active Core  | 4/0/4               | 2/0                    | cab0,bay0,chassis1 | yes               | yes              | 0       |
| cab0,cell11       | Active Base  | 2/0/4               | 2/0                    | -                  | no                | yes              | 2       |
| cab0,cell12       | Inactive     | 4/0/4               | 2/0                    | cab0,bay1,chassis3 | yes               | yes              | 1       |
| cab0,cell13       | Inactive     | 2/0/4               | 2/0                    | -                  | no                | -                | -       |
| cab0,cell14       | Active Core  | 2/0/4               | 2/0                    | cab0,bay0,chassis3 | yes               | yes              | 2       |
| cab0,cell15       | Powered off  | 0/0/4               | ?                      | ?                  | ?                 | no               | 0       |
| cab0,cell16       | Active Core  | 2/0/4               | 2/0                    | cab0,bay1,chassis1 | yes               | yes              | 1       |
| cab0,cell17       | Powering on  | 0/0/4               | ?                      | ?                  | ?                 | -                | -       |

Notes: \* = Cell has no interleaved memory.

[Chassis]

| Hardware Location  | Usage    | Core IO | Connected To | Par Num |
|--------------------|----------|---------|--------------|---------|
| ====               | =====    | =====   | =====        | =====   |
| cab0,bay0,chassis0 | Absent   | -       | -            | -       |
| cab0,bay0,chassis1 | Active   | yes     | cab0,cell10  | 0       |
| cab0,bay0,chassis2 | Absent   | -       | -            | -       |
| cab0,bay0,chassis3 | Active   | yes     | cab0,cell14  | 2       |
| cab0,bay1,chassis0 | Absent   | -       | -            | -       |
| cab0,bay1,chassis1 | Active   | yes     | cab0,cell16  | 1       |
| cab0,bay1,chassis2 | Absent   | -       | -            | -       |
| cab0,bay1,chassis3 | Inactive | yes     | cab0,cell12  | 1       |

[Partition]

| Par Num | Status | # of Cells | # of Chassis | I/O Core cell | Partition Name (first 30 chars) |
|---------|--------|------------|--------------|---------------|---------------------------------|
| ====    | =====  | =====      | =====        | =====         | =====                           |
| 0       | Active | 2          | 1            | cab0,cell10   | cup2000                         |
| 1       | Active | 3          | 2            | cab0,cell16   | cup2006                         |
| 2       | Active | 2          | 1            | cab0,cell14   | cup2004                         |

DEPENDENCIES

This command uses the *Web-Based Enterprise Management (WBEM)* product and certain of its configuration settings. If you encounter connection errors when using the **-u** option, check that the

following two conditions are satisfied:

- Use the *cimconfig*(1M) command to verify (and correct if necessary) the setting of the following two variables:
  - **enableRemotePrivilegedUserAccess=true**
  - **enableHttpsConnection=true**
- You must have appended the target partition's digital certificate to the local partition's Trust Store file. For the nPartition commands, the Trust Store file is **/var/opt/wbem/client.pem**.

Refer to the WBEM documents specified in the *SEE ALSO* section below for further information.

**AUTHOR**

**parstatus** was developed by the Hewlett-Packard Company.

**SEE ALSO**

*cplxmodify*(1M), *fruled*(1), *frupower*(1M), *parcreate*(1M), *parmgr*(1M), *parmodify*(1M), *parremove*(1M), *parunlock*(1M), *partition*(5),

*HP System Partitions Guide* on **docs.hp.com**,

*HP WBEM Services for HP-UX System Administrator's Guide* on **docs.hp.com**,

*HP WBEM Services for HP-UX 11i v2.0 on Integrity Servers Version A.01.05 Release Notes* on **docs.hp.com**.

**| P |**



**NAME**

passwd - change login password and associated attributes

**SYNOPSIS**

```
passwd [name]
passwd -r files [-F file] [name]
passwd -r files [-e [shell]] [-gh] [name]
passwd -r files -s [-a]
passwd -r files -s [name]
passwd -r files [-d|-l] [-f] [-n min] [-w warn] [-x max] name
passwd -r nis [-e [shell]] [-gh] [name]
passwd -r nisplus [-e [shell]] [-gh] [-D domain] [name]
passwd -r nisplus -s [-a]
passwd -r nisplus -s [-D domain] [name]
passwd -r nisplus [-l] [-f] [-n min] [-w warn] [-x max] [-D domain] name
passwd -r dce [-e [shell]] [-gh] [name]
```

**DESCRIPTION**

The **passwd** command modifies the password as well as the attributes associated with the login *name*. If *name* is omitted, it defaults to the invoking user's login *name*, which is determined using **getuid**. See *getuid*(2).

Ordinary users can only change passwords corresponding to their login *name*. If an old password has been established, it is requested from the user. If valid, a new password is obtained. Once the new password is entered, it is determined if the old password has "aged" sufficiently. If password aging is not sufficient, the new password is rejected and **passwd** terminates. See *passwd*(4).

If password aging and construction requirements are met, the password is re-entered to ensure consistency. If the new copy differs, **passwd** repeats the new password prompting cycle, at most twice.

A superuser, whose effective user ID is zero, (see *id*(1) and *su*(1)), is allowed to change any password and is not forced to comply with password aging. Superusers are not prompted for old passwords, unless they are attempting to change a superuser's password in a trusted system. On untrusted systems, superusers are not forced to comply with password construction requirements. Null passwords can be created by entering a carriage return in response to the prompt for a new password.

For the **files** (local system) repository, if no **/etc/shadow** file exists, then the encrypted password is stored in the password field of **/etc/passwd**. If the **/etc/shadow** file exists, then the encrypted password is stored there, and an 'x' is added to the password field of **/etc/passwd**.

The DCE repository (**-r dce**) is only available if Integrated Login has been configured. See *auth.adm*(1M). If Integrated Login has been configured, other considerations apply. A user with appropriate DCE privileges is capable of modifying a user's password, shell, gecos or home directory and this is not dependent upon superuser privileges.

If the repository is not specified, i.e. **passwd [name]**, the password is changed in all existing repositories configured in **/etc/nsswitch.conf**. If password options are used, and no repository is specified, the default repository is **files**.

**Options**

The following options are recognized:

- D domain** Use the **passwd.org\_dir** in the specified *domain*. This option is for **nisplus** repositories only. If not specified, the default *domain* is returned.
- e shell** Modify the default shell for the user's login *name* in the password file. If the *shell* is not provided, the user will be prompted to enter the default login shell.
- F name** The default password file is **/etc/passwd**. The **-F** option can be used to choose an alternate password file, where read and write permissions are required. This option is only available using the **files** repository, and it is not intended for trusted mode.

- g** Change the `gecos` information in the password file, which is used by the `finger` command. The user is prompted for each subfield: name, location, work phone, and home phone.
- r repository** Specify the repository to which the operation is to be applied. Supported repositories include `files`, `nis`, `nisplus`, and `dce`. If repository is not specified, the default is `files`.
- s name** Display some password attributes associated with the specified `name`. Superuser privilege and non-trusted mode is required if the `files` repository is specified. For `nisplus`, there are no restrictions.  
The format of the display will be:  

```
name status mm/dd/yy min max warn
```

or, if password aging information is not present  

```
name status
```

where `status` means: `PS` =passworded; `LK` =locked; and `NP` =no password.
- a** Display some password attributes for all users in the password file. The `-a` option must be used in conjunction with the `-s` option, with no `name` specified. For `nisplus`, this will display entries in the NIS+ `passwd` table in the local domain. For `files`, this is restricted to superuser, and is only valid in non-trusted mode. For a more complete display of attributes use the command `logins -x`.

### Privileged User Options

A superuser can modify characteristics associated with the user `name` using the following options:

- d** Allow user to login without a password by deleting it. In untrusted mode this unlocks/activates the user account if found locked/deactivated.
- f** Force user to change password upon next login by expiring the current password.
- h** Modify the default home directory in the password file.
- l** Lock user account. In untrusted mode this replaces the encrypted password with `*`.
- n min** Determine the minimum number of days, `min`, that must transpire before the user can change the password. If the `-f` option was used in a previous invocation of `passwd` to immediately expire a password, the effect of the `-f` option is cancelled. The effect of the `-f` option is not cancelled if the `-x` option and `-f` option are specified on the same command line or if the system has been converted to a trusted system.
- w warn** Specify the number of days, `warn`, prior to the password expiring when the user will be notified that the password needs to be changed. This option is not allowed for systems in non-shadowed standard mode.
- x max** Determine the maximum number of days, `max`, a password can remain unchanged. The user must enter another password after that number of days has transpired, known as the password *expiration time*. If the `-f` option was used in a previous invocation of `passwd` to immediately expire a password, the effect of the `-f` option is cancelled, and the password will not expire until `max` days. The effect of the `-f` option is not cancelled if the `-x` option and the `-f` option are specified on the same command line or if the system has been converted to a trusted system.

The `min` and `max` arguments are each represented in units of days. These arguments will be rounded up to the nearest week on a nontrusted HP-UX system. If the system is then converted to a trusted system, the number of days will be based on those weeks. If only one of the two arguments is supplied, and the other argument does not exist, then the number of days is set to zero.

### Password Aging

The following description applies to all repositories except `nis`, which does not support password aging.

The system requires a minimum time to elapse before a password can be changed. This prevents reuse of an old password within too brief a period of time. System warnings are displayed as the expiration time approaches.

A password is no longer usable after a time period known as the password *lifetime*. After the lifetime passes, the account is locked until it is re-enabled by a system administrator. Once unlocked, the user is

forced to change the password before using the account.

The **-n min** and **-x max** arguments are each represented in units of days. These arguments are rounded up to the nearest week on a standard system. If only one of the two arguments is supplied and the other argument does not exist, then the number of days is set to zero.

Default values may be set in the `/etc/default/security` file for the **-n min**, **-x max**, and **-w warn** options. See *security*(4). The parameters to select password aging defaults are:

**PASSWORD\_MINDAYS**

**PASSWORD\_MAXDAYS**

**PASSWORD\_WARNDAYS**

### Password Construction Requirements

Passwords must be constructed to meet the following requirements:

- On an untrusted system, only the first eight characters of a password are significant.
- On an untrusted system, passwords of non-root users must have at least six characters. On a trusted system, passwords of all users must have at least six characters. This restriction on the password length can be increased to a value larger than six. Refer to the *security*(4) manual page for detailed information on configurable parameters that affect the behavior of this command. The parameter to select the minimum password length is

**MIN\_PASSWORD\_LENGTH**

- Characters must be from the 7-bit US-ASCII character set; letters from the English alphabet.
- A password must contain at least two letters and at least one numeric or special character.
- A password must differ from the user's login *name* and any reverse or circular shift of that login *name*. For comparison purposes, an uppercase letter and its corresponding lowercase equivalent are treated as identical.
- A new password must differ from the old one by at least three characters (one character for non super user if changed by the super user in a trusted system).

### Repository Configuration

The `/etc/nsswitch.conf` file specifies the repositories for which the password must be modified. The following configurations are supported:

- passwd: files
- passwd: files nisplus
- passwd: files nis
- passwd: compat (--> files nis)
- passwd: compat (--> files nisplus)
- passwd\_compat: nisplus

### Smart Card Login

If the user account is configured to use a Smart Card, the user password is stored in the card. This password has characteristics identical to a normal password stored on the system.

The Smart Card must be inserted into the Smart Card reader. The user is prompted for a PIN instead of a password during authentication.

**Enter PIN:**

The password is retrieved automatically from the Smart Card when a valid PIN is entered. Therefore, it is not necessary to know the password, only the PIN.

If the system retrieves a valid old password from the card, a new password is requested (twice). If the new password meets all requirements, the system automatically overwrites the old password stored on the card with the new password.

Therefore, the new dialog resembles:

**Enter PIN:**

**New password:**

**Re-enter new password:**

A Smart Card account can be shared among users. If one user modifies the password, other users must use the **scsync** command to write the new password onto their cards.

The **scpin** command is used to change the Smart Card PIN.

## SECURITY FEATURES

This section applies only to trusted systems. It describes additional capabilities and restrictions.

When **passwd** is invoked on a trusted system, the existing password is requested (if one is present). This initiates the password solicitation dialog which depends upon the type of password generation (format policy) that has been enabled on the account doing the **passwd** command. There are four possible options for password generation:

|                          |                                                                                  |
|--------------------------|----------------------------------------------------------------------------------|
| <b>Random syllables</b>  | A pronounceable password made up of meaningless syllables.                       |
| <b>Random characters</b> | An unpronounceable password made up of random characters from the character set. |
| <b>Random letters</b>    | An unpronounceable password made up of random letters from the alphabet.         |
| <b>User-supplied</b>     | A user-supplied password, subject to length and triviality restrictions.         |

Passwords can be greater than eight characters, but it is recommended that they be less than 40 characters. System warnings are displayed if passwords lengths are either too long or short. The system administrator can specify a maximum password length guideline for the system generated options (random syllables, random characters, and random letters). The actual maximum password length depends upon several parameters in the authentication database and in the algorithm.

The system requires a *minimum time* to elapse before a password can be changed. This prevents reuse of an old password within an undesirable period of time.

A password expires after a period of time known as the *expiration time*. System warnings are displayed as expiration time approaches.

A password dies after a time period known as the *password lifetime*. After the lifetime passes, the account is locked until it is re-enabled by a system administrator. Once unlocked, the user is forced to change the password before account use.

The system administrator can enable accounts without passwords. If a user account is allowed to function without a password, the user can choose a null password by typing a carriage-return when prompted for a new password.

The system administrator can enable the password history feature to discourage users from reusing previously used passwords. Refer to the *security(4)* manual page for detailed information on configurable parameters that affect the behavior of this command. The parameter for password history is:

**PASSWORD\_HISTORY\_DEPTH**

## EXTERNAL INFLUENCES

### International Code Set Support

Characters from single-byte character code sets are supported in passwords.

## EXAMPLES

Change the password expiration date of **user** to 42 days in the **files** repository:

```
passwd -r files -x 42 user
```

Modify the minimum time between password changes of **user1** to 7 days in the **nisplus** repository:

```
passwd -r nisplus -n 7 user1
```

Force **user2** to establish a new password on the next login which will expire in 70 days and prohibit the user from changing the password until 7 days have transpired:

```
passwd -r files -f -x 70 -n 7 user2
```

## DEPENDENCIES

### Pluggable Authentication Modules (PAM)

PAM is an Open Group standard for user authentication, password modification, and account validation. In particular, **pam\_chauthtok()** is invoked to perform all functions related to **passwd**. This includes establishing and changing a password, using **passwd** options, and displaying error messages.

**WARNINGS**

Avoid password characters which have special meaning to the tty driver, such as # (erase) and @ (kill). You may not be able to login with these characters.

Multiple superusers are allowed, but are strongly discouraged. That is because the system often stores user ID rather than user name. Having unique IDs for all users will guarantee a consistent mapping between user name and user ID.

**FILES**

|                                    |                                                                              |
|------------------------------------|------------------------------------------------------------------------------|
| <code>/etc/passwd</code>           | Standard password file used by HP-UX.                                        |
| <code>/etc/shadow</code>           | Shadow password file.                                                        |
| <code>/tcb/files/auth/**</code>    | Protected password database used when system is converted to trusted system. |
| <code>/etc/nsswitch.conf</code>    | Repository Configuration.                                                    |
| <code>/etc/default/security</code> | Security defaults configuration file.                                        |

**SEE ALSO**

chfn(1), id(1), login(1), su(1), logins(1M), pwconv(1M), getuid(2), crypt(3C), passwd(4), security(4), shadow(4), auth(5), auth.adm(1M), auth.dce(5).

*Managing Systems and Workgroups*

**Pluggable Authentication Modules (PAM)**

*pam\_chauthtok(3), pam(3), pam.conf(4), pam\_user.conf(4).*

**HP-UX Smart Card Login**

*scpin(1), scsync(1).*

**STANDARDS CONFORMANCE**

**passwd:** SVID2, SVID3, XPG2

**NAME**

paste - merge same lines of several files or subsequent lines of one file

**SYNOPSIS**

```
paste file1 file2 ...
paste -d list file1 file2 ...
paste -s [-d list] file1 file2 ...
```

**DESCRIPTION**

In the first two forms, **paste** concatenates corresponding lines of the given input files *file1*, *file2*, etc. It treats each file as a column or columns in a table and pastes them together horizontally (parallel merging). In other words, it is the horizontal counterpart of *cat*(1) which concatenates vertically; i.e., one file after the other. In the **-s** option form above, **paste** replaces the function of an older command with the same name by combining subsequent lines of the input file (serial merging). In all cases, lines are glued together with the *tab* character, or with characters from an optionally specified *list*. Output is to standard output, so **paste** can be used as the start of a pipe, or as a filter if **-** is used instead of a file name.

**paste** recognizes the following options and command-line arguments:

- d** Without this option, the new-line characters of all but the last file (or last line in case of the **-s** option) are replaced by a *tab* character. This option allows replacing the *tab* character by one or more alternate characters (see below).
- list* One or more characters immediately following **-d** replace the default *tab* as the line concatenation character. The list is used circularly; i.e., when exhausted, it is reused. In parallel merging (that is, no **-s** option), the lines from the last file are always terminated with a new-line character, not from the *list*. The list can contain the special escape sequences: **\n** (new-line), **\t** (tab), **\\** (backslash), and **\0** (empty string, not a null character). Quoting may be necessary if characters have special meaning to the shell. (For example, to get one backslash, use **-d"\\\\"**).
- s** Merge subsequent lines rather than one from each input file. Use *tab* for concatenation, unless a *list* is specified with the **-d** option. Regardless of the *list*, the very last character of the file is forced to be a new-line.
- Can be used in place of any file name to read a line from the standard input (there is no prompting).

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the locale for the interpretation of text as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **paste** behaves as if all internationalization variables are set to "C". See *environ*(5).

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**RETURN VALUE**

These commands return the following values upon completion:

- 0 Completed successfully.
- >0 An error occurred.

**EXAMPLES**

List directory in one column:

```
ls | paste -d" " -
List directory in four columns
ls | paste - - - -
Combine pairs of lines into lines
paste -s -d"\t\n" file
```

**Notes**

**pr -t -m...** works similarly, but creates extra blanks, tabs and new-lines for a nice page layout.

**DIAGNOSTICS**

**too many files** Except for the **-s** option, no more than **OPEN\_MAX** - 3 input files can be specified (see *limits*(5)).

**AUTHOR**

**paste** was developed by OSF and HP.

**SEE ALSO**

**cut(1)**, **grep(1)**, **pr(1)**.

**STANDARDS CONFORMANCE**

**paste**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| p |**

**NAME**

patch - a program for applying a diff file to an original

**SYNOPSIS****Non-XPG4 version**

```
patch [options] orig patchfile [+ [options] orig]
```

```
patch <patchfile # usual form
```

**XPG4 version**

```
patch [-bLNR] [-c | -e | -n] [-d dir] [-D define] [-i patchfile] [-o outfile] [-p num]
 [-r rejectfile] [file]
```

**DESCRIPTION**

**patch** will take a patch file containing any of the three forms of difference listing produced by the *diff* program (normal, context or in the style of *ed*) and apply those differences to an original file, producing a patched version. By default, the patched version is put in place of the original, with the original file backed up to the same name with the extension “.orig”, or as specified by the **-b** switch. Note that functionality of this option varies for XPG4 version. You may also specify where you want the output to go with a **-o** switch. If *patchfile* is omitted, or is a hyphen, the patch will be read from standard input. For XPG4 version, *patchfile* has to be specified as argument to **-i** switch. If this option is omitted or a hyphen is specified as argument, the patch will read from standard input.

Upon startup, *patch* will attempt to determine the type of the diff listing, unless over-ruled by a **-c**, **-e**, or **-n** switch. Context diffs and normal diffs are applied by the *patch* program itself, while *ed* diffs are simply fed to the *ed* editor via a pipe.

*patch* will try to skip any leading garbage, apply the diff, and then skip any trailing garbage. Thus you could feed an article or message containing a diff listing to *patch*, and it should work. If the entire diff is indented by a consistent amount, this will be taken into account.

With context diffs, and to a lesser extent with normal diffs, *patch* can detect when the line numbers mentioned in the patch are incorrect, and will attempt to find the correct place to apply each hunk of the patch. As a first guess, it takes the line number mentioned for the hunk, plus or minus any offset used in applying the previous hunk. If that is not the correct place, *patch* will scan both forwards and backwards for a set of lines matching the context given in the hunk. First *patch* looks for a place where all lines of the context match. If no such place is found, and it's a context diff, and the maximum fuzz factor is set to 1 or more, then another scan takes place ignoring the first and last line of context. If that fails, and the maximum fuzz factor is set to 2 or more, the first two and last two lines of context are ignored, and another scan is made. (The default maximum fuzz factor is 2.) Note that for XPG4 version maximum fuzz factor can not be specified as an option and the default maximum fuzz factor is used. If *patch* cannot find a place to install that hunk of the patch, it will put the hunk out to a reject file, which normally is the name of the output file plus “.rej”. (Note that the rejected hunk will come out in context diff form whether the input patch was a context diff or a normal diff. If the input was a normal diff, many of the contexts will simply be null.) The line numbers on the hunks in the reject file may be different than in the patch file: they reflect the approximate location *patch* thinks the failed hunks belong in the new file rather than the old one.

As each hunk is completed, you will be told whether the hunk succeeded or failed, and which line (in the new file) *patch* thought the hunk should go on. If this is different from the line number specified in the diff you will be told the offset. A single large offset MAY be an indication that a hunk was installed in the wrong place. You will also be told if a fuzz factor was used to make the match, in which case you should also be slightly suspicious. Note that XPG4 version does not support verbose option. So, most of the diagnostic messages are not printed for this version. However user queries will always be displayed.

If no original file is specified on the command line, *patch* will try to figure out from the leading garbage what the name of the file to edit is. In the header of a context diff, the filename is found from lines beginning with “\*\*\*” or “---”, with the shortest name of an existing file winning. Only context diffs have lines like that, but if there is an “Index:” line in the leading garbage, *patch* will try to use the filename from that line. The context diff header takes precedence over an Index line. If no filename can be intuited from the leading garbage, you will be asked for the name of the file to patch.

(If the original file cannot be found, but a suitable SCCS or RCS file is handy, *patch* will attempt to get or check out the file.)



Additionally, if the leading garbage contains a “**Prereq:** ” line, *patch* will take the first word from the prerequisites line (normally a version number) and check the input file to see if that word can be found. If not, *patch* will ask for confirmation before proceeding.

The upshot of all this is that you should be able to say, while in a news interface, the following:

```
| patch -d /usr/src/local/blurfl
```

and patch a file in the blurfl directory directly from the article containing the patch.

If the patch file contains more than one patch, *patch* will try to apply each of them as if they came from separate patch files. This means, among other things, that it is assumed that the name of the file to patch must be determined for each diff listing, and that the garbage before each diff listing will be examined for interesting things such as filenames and revision level, as mentioned previously. You can give switches (and another original file name) for the second and subsequent patches by separating the corresponding argument lists by a “+”. (The argument list for a second or subsequent patch may not specify a new patch file, however.)

With XPG4 version, processing of multiple patches varies considerably. You can not specify different options for different patches. Options remain same for all the patches. This also affects the contents of output file specified with the **-o** option. See the description of this option for more details.

**patch** recognizes the following switches:

- b** causes the next argument to be interpreted as the backup extension, to be used in place of “.orig”. (For XPG4 version this option varies. With this option no argument is required and the option only enables the backup process. Always default extension is used.)
- c** forces *patch* to interpret the patch file as a context diff.
- d** causes *patch* to interpret the next argument as a directory, and cd to it before doing anything else.
- D** causes *patch* to use the “#ifdef...#endif” construct to mark changes. The argument following will be used as the differentiating symbol. Note that, unlike the C compiler, there must be a space between the **-D** and the argument. (For XPG4 version this option varies. With this version “#ifndef” constructor is not used.)
- e** forces *patch* to interpret the patch file as an ed script.
- f** forces *patch* to assume that the user knows exactly what he or she is doing, and to not ask any questions. It does not suppress commentary, however. Use **-s** for that. (This option is not supported by XPG4 version.)
- F number**  
sets the maximum fuzz factor. This switch only applied to context diffs, and causes *patch* to ignore up to that many lines in looking for places to install a hunk. Note that a larger fuzz factor increases the odds of a faulty patch. The default fuzz factor is 2, and it may not be set to more than the number of lines of context in the context diff, ordinarily 3. (This option is not supported by XPG4 version.)
- i** This option is supported only by XPG4 version. It causes next argument to be interpreted as the patch file name.
- l** causes the pattern matching to be done loosely, in case the tabs and spaces have been munged in your input file. Any sequence of whitespace in the pattern line will match any sequence in the input file. Normal characters must still match exactly. Each line of the context must still match a line in the input file.
- n** forces *patch* to interpret the patch file as a normal diff.
- N** causes *patch* to ignore patches that it thinks are reversed or already applied. See also **-R**.
- o** causes the next argument to be interpreted as the output file name. There are some added features for the XPG4 version. Multiple patches for a single file will be applied to the intermediate versions of the file created by any previous patches, and will result in multiple, concatenated versions of the file being written to output file.
- p number**  
sets the pathname strip count, which controls how pathnames found in the patch file are treated, in case the you keep your files in a different directory than the person who sent out the patch. The strip count specifies how many backslashes are to be stripped from the front of the pathname. (Any intervening directory names also go away.) For example, supposing the filename in the patch file

| p |

was

```
/u/howard/src/blurfl/blurfl.c
```

setting **-p** or **-p0** gives the entire pathname unmodified, **-p1** gives

```
u/howard/src/blurfl/blurfl.c
```

without the leading slash, **-p4** gives

```
blurfl/blurfl.c
```

and not specifying **-p** at all just gives you "blurfl.c". Whatever you end up with is looked for either in the current directory, or the directory specified by the **-d** switch.

- r** causes the next argument to be interpreted as the reject file name.
- R** tells *patch* that this patch was created with the old and new files swapped. (Yes, I'm afraid that does happen occasionally, human nature being what it is.) *patch* will attempt to swap each hunk around before applying it. Rejects will come out in the swapped format. The **-R** switch will not work with ed diff scripts because there is too little information to reconstruct the reverse operation.

If the first hunk of a patch fails, *patch* will reverse the hunk to see if it can be applied that way. If it can, you will be asked if you want to have the **-R** switch set. If it can't, the patch will continue to be applied normally. (Note: this method cannot detect a reversed patch if it is a normal diff and if the first command is an append (i.e. it should have been a delete) since appends always succeed, due to the fact that a null context will match anywhere. Luckily, most patches add or change lines rather than delete them, so most reversed normal diffs will begin with a delete, which will fail, triggering the heuristic.)

- s** makes *patch* do its work silently, unless an error occurs. (This option is not supported by XPG4 version.)
- S** causes *patch* to ignore this patch from the patch file, but continue on looking for the next patch in the file. Thus
 

```
patch -S + -S + <patchfile
```

 will ignore the first and second of three patches. (This option is not supported by XPG4 version.)
- v** causes *patch* to print out it's revision header and patch level. (This option is not supported by XPG4 version.)
- x number** sets internal debugging flags, and is of interest only to *patch* patchers. (This option is not supported by XPG4 version.)

## EXTERNAL INFLUENCES

### Environment Variables

**UNIX95** determines which version of patch is used. If this variable is set, patch exhibits XPG4 behaviour.

## RETURN VALUE

The following exit values are returned for XPG4 version:

- 0 Successful completion.
- 1 One or more lines were written to a reject file.
- >1 An error occurred.

For non-XPG4 version exit values vary as follows:

- 0 Successful completion or one or more lines were written to a reject file.
- 1 An error occurred.

## DIAGNOSTICS

Too many to list here, but generally indicative that *patch* couldn't parse your patch file.

The message "Hm..." indicates that there is unprocessed text in the patch file and that *patch* is attempting to intuit whether there is a patch in that text and, if so, what kind of patch it is.

Note that only few diagnostic messages are printed for XPG4 version, since it does not support verbose option.

**WARNINGS**

**patch** cannot tell if the line numbers are off in an ed script, and can only detect bad line numbers in a normal diff when it finds a “change” or a “delete” command. A context diff using fuzz factor 3 may have the same problem. Until a suitable interactive interface is added, you should probably do a context diff in these cases to see if the changes made sense. Of course, compiling without errors is a pretty good indication that the patch worked, but not always.

**patch** usually produces the correct results, even when it has to do a lot of guessing. However, the results are guaranteed to be correct only when the patch is applied to exactly the same version of the file that the patch was generated from.

The result obtained from the XPG4 options **-c**, **-e**, **-n** which forces the patch command to interpret the diff file either as a context diff or as an ed script or as a normal diff respectively is unspecified. For example, if one forces the patch command to treat the context diff file as an ed script, the result is unspecified. The same is true if one forces patch to treat an ed script as a context file and so on.. When a diff is forced with the above options, the diff file is searched for patterns that are specific to that type of diff file. If the diff file is not what was specified by the option, the file is checked for ed commands. If ed commands are present in the diff file, then the file is assumed to be an ed\_diff file and the patch proceeds.

**FILES**

`/var/tmp/patch*`

**SEE ALSO**

diff(1), ed(1).

**NOTES FOR PATCH SENDERS**

There are several things you should bear in mind if you are going to be sending out patches. First, you can save people a lot of grief by keeping a patchlevel.h file which is patched to increment the patch level as the first diff in the patch file you send out. If you put a Prereq: line in with the patch, it won't let them apply patches out of order without some warning. Second, make sure you've specified the filenames right, either in a context diff header, or with an Index: line. If you are patching something in a subdirectory, be sure to tell the patch user to specify a **-p** switch as needed. Third, you can create a file by sending out a diff that compares a null file to the file you want to create. This will only work if the file you want to create doesn't exist already in the target directory. Fourth, take care not to send out reversed patches, since it makes people wonder whether they already applied the patch. Fifth, while you may be able to get away with putting 582 diff listings into one file, it is probably wiser to group related patches into separate files in case something goes haywire.

**BUGS**

Could be smarter about partial matches, excessively deviant offsets and swapped code, but that would take an extra pass.

If code has been duplicated (for instance with `#ifdef OLDPCODE ... #else ... #endif`), *patch* is incapable of patching both versions, and, if it works at all, will likely patch the wrong one, and tell you that it succeeded to boot.

If you apply a patch you've already applied, *patch* will think it is a reversed patch, and offer to un-apply the patch. This could be construed as a feature.

One more thing to be noted with respect to XPG4 version of *patch*. If you are using multiple patches for different files, group patches that have to be applied to a single file. Otherwise, intermediate versions of the previous patches of a file will not be used for the current patch.

**STANDARDS CONFORMANCE**

**patch:** XPG4

| p |

**NAME**

pathalias - electronic address router

**SYNOPSIS**

```
pathalias [-ivcDf] [-l host] [-d link] [-t link] [files]
```

**DESCRIPTION**

**pathalias** computes the shortest paths and corresponding routes from one host (computer system) to all other known, reachable hosts. **pathalias** reads host-to-host connectivity information on standard input or in the named *files*, and writes a list of host-route pairs on the standard output.

**Options**

**pathalias** recognizes the following options and command-line arguments:

- i** Ignore case: map all host names to lowercase. By default, case is significant.
- c** Print costs. Print the path cost (see below) before each host-route pair.
- v** Verbose. Report some statistics on the standard error output.
- D** Terminal domains. Domain members are terminal.
- f** First hop cost. The printed cost is the cost to the first relay in a path instead of the cost of the path itself; implies (and overrides) the **-c** option.
- l host** Set local host name to *host*. By default, **pathalias** discovers the local host name in a system-dependent way.
- d link** Declare a dead link, host, or network (see below). If *link* is of the form **host1!host2**, the link from host1 to host2 is treated as an extremely high cost (i.e., **DEAD**) link. If *link* is a single host name, that host is treated as dead and is used as an intermediate host of last resort on any path. If *link* is a network name, the network requires a gateway.
- t link** Trace input for link, host, or network on the standard error output. The form of *link* is as above.

The public domain version of **pathalias** includes two undocumented options that are briefly described in the Special Options section below.

**p****Input Format**

A line beginning with white space continues the preceding line. Anything following **#** on an input line is ignored.

A list of host-to-host connections consists of a “from” host in column 1, followed by white space, followed by a comma-separated list of “to” hosts, called *links*. A link may be preceded or followed by a network character to use in the route. Valid network characters are **!** (default), **@**, **:**, and **%**. A link (and network character, if present) may be followed by a “cost” enclosed in parentheses. Costs can be arbitrary arithmetic expressions involving numbers, parentheses, **+**, **-**, **\***, and **/**. Negative costs are prohibited. The following symbolic costs are recognized:

|                  |       |                                  |
|------------------|-------|----------------------------------|
| <b>LOCAL</b>     | 25    | (local-area network connection)  |
| <b>DEDICATED</b> | 100   | (high speed dedicated link)      |
| <b>DIRECT</b>    | 200   | (toll-free call)                 |
| <b>DEMAND</b>    | 300   | (long-distance call)             |
| <b>HOURLY</b>    | 500   | (hourly poll)                    |
| <b>EVENING</b>   | 2000  | (time restricted call)           |
| <b>DAILY</b>     | 5000  | (daily poll, also called POLLED) |
| <b>WEEKLY</b>    | 30000 | (irregular poll)                 |

In addition, **DEAD** is a very large number (effectively infinite), and **HIGH** and **LOW** are **-5** and **+5** respectively, for baud-rate or quality bonuses/penalties, and **FAST** is **-80**, for adjusting costs of links that use high-speed (9.6 Kbaud or more) modems. These symbolic costs represent an imperfect measure of bandwidth, monetary cost, and frequency of connections. For most mail traffic, it is important to minimize the number of hosts in a route, thus, e.g., **HOURLY** is far greater than **DAILY** divided by 24. If no cost is given, a default of 4000 is used.

For the most part, arithmetic expressions that mix symbolic constants other than **HIGH**, **LOW**, and **FAST** make no sense. For example, if a host calls a local neighbor whenever there is work, and additionally polls every evening, the cost is **DIRECT**, *not* **DIRECT+EVENING**.

Some examples:

```
down princeton!(DEDICATED), tilt,
 %thrash(LOCAL)
princeton topaz!(DEMAND+LOW)
topaz @rutgers(LOCAL+1)
```

If a link is encountered more than once, the least-cost occurrence dictates the cost and network character. Links are treated as bidirectional but asymmetric: for each link declared in the input, a **DEAD** reverse link is assumed.

If the “to” host in a link is surrounded by angle brackets, the link is considered *terminal*, and further links beyond this one are heavily penalized. For example, with input

```
seismo <research>(10), research(100), ihnp4(10)
research allegra(10)
ihnp4 allegra(50)
```

the path from **seismo** to **research** is direct, but the path from **seismo** to **allegra** uses **ihnp4** as a relay; not **research**.

The set of names by which a host is known by its neighbors is called its *aliases*. Aliases are declared as follows:

```
name=alias , alias ...
```

The name used in the route to or through aliased hosts is the name by which the host is known to its predecessor in the route.

Fully connected networks, such as the ARPANET or a local-area network, are declared as follows:

```
net = {host, host, ... }
```

The host-list can be preceded or followed by a routing character (! by default), and can be followed by a cost (4000 by default). The network name is optional; if not given, **pathalias** creates one.

```
etherhosts = {rahway, milan, joliet}!(LOCAL)
ringhosts = @{gimli, alida, almo}(DEDICATED)
 = {etherhosts, ringhosts}(0)
```

The routing character used in a route to a network member is the one encountered when “entering” the network. See also the sections on *gateways* and *domains*.

Connection data can be given while hiding host names by declaring

```
private {host, host, ... }
```

**pathalias** does not generate routes for private hosts, but can produce routes through them. The scope of a private declaration extends from the declaration to the end of the input file in which it appears, or to a private declaration with an empty host list, whichever comes first. The latter scope rule offers a way to retain the semantics of private declarations when reading from the standard input.

Dead hosts, links, or networks can be presented in the input stream by declaring

```
dead {arg, ... }
```

where *arg* has the same form as the argument to the **-d** option.

To force a specific cost for a link, delete all prior declarations with

```
delete {host1 !host2 }
```

and declare the link as desired. To delete a host and all its links, use

```
delete {host }
```

Error diagnostics refer to the file in which the error was found. To alter the file name, use

```
file {filename }
```

Fine-tuning is possible by adjusting the weights of all links from a given host, as in

```
adjust {host1 , host-2 (LOW) , host3 (-1) }
```

If no cost is given, a default of 4000 is used.

| p |

Input from compressed (and uncompressed) files can be piped into **pathalias** with the following script.

```
for i in $*; do
 case $i in
 .Z) echo "file {\`expr $i : '\(.\).Z'\`}"
 zcat $i ;;
 *) echo "file {$i}"
 cat $i ;;
 esac
 echo "private {"
done
```

### Output Format

A list of host-route pairs is written to the standard output, where route is a string appropriate for use with **printf()** (see *printf(3S)*), such as

```
rutgers princeton!topaz!%s@rutgers
```

The **%s** in the route string should be replaced by the user name at the destination host (this task is normally performed by a mailer).

Except for *domains* (see below), the name of a network is never used in routes. Thus, in the earlier example, the path from **rahway** to **milan** would be **milan!%s**, not **etherhosts!milan!%s**.

### Gateways

A network is represented by a pseudo-host and a set of network members. Links from the members to the network have the weight given in the input, while the cost from the network to the members is zero. If a network is declared dead, the member-to-network links are marked dead, which effectively prohibits access to the network from its members.

However, if the input also shows an explicit link from any host to the network, then that host can be used as a gateway (in particular, the gateway need not be a network member).

For example, suppose **CSNET** is declared dead on the command line and the input contains

```
CSNET = {...}
csnet-relay CSNET
```

Then routes to **CSNET** hosts will use **csnet-relay** as a gateway.

### Domains

A network whose name begins with **.** is called a domain. Domains are presumed to require gateways; i.e., they are **DEAD**. The route given by a path through a domain is similar to that for a network, but here the domain name is appended to the end of the name of the next host. Subdomains are permitted. For example:

```
harvard .EDU # harvard is gateway to .EDU domain
.EDU = { .BERKELEY, .UMICH }
.BERKELEY = { ernie }
```

yields

```
ernie ...!harvard!ernie.BERKELEY.EDU!%s
```

Output is given for the nearest gateway to a domain; e.g., the example above gives

```
.EDU ...!harvard!%s
```

Output is given for a subdomain if it has a different route than its parent domain, or if all its ancestor domains are private.

If the **-D** option is given on the command line, **pathalias** treats a link from a domain to a host member of that domain as terminal. This property extends to host members of subdomains, etc., and discourages routes that use any domain member as a relay.

### Special Options

The public domain version of **pathalias** includes two undocumented options that rewrite named files with intermediate data of limited usage. Here are brief descriptions:

```
-g file Dump graph edges into file in the form host>host for simple connections and
 host@<tab>host for network connections (from hosts to networks only).
```

**-s** *file* Dump shortest path tree into *file* in the form *host*<tab>[@]*host* [!](*cost*), including both connections from hosts to networks and from networks to hosts. This data may be useful for generating lists of one-way connections.

## BUGS

The **-i** option should be the default.

The order of arguments is significant. In particular, **-i** and **-t** should appear early in the command line.

**pathalias** can generate hybrid (i.e., ambiguous) routes, which are abhorrent and most certainly should not be given as examples in a manual entry. Experienced mappers largely shun '@' when preparing input; this is historical, but also reflects UUCP's simplistic syntax for source routes.

Mixed-mode paths are ambiguous because the precedence of @ versus ! is not specified, varies from host to host, and is configurable. They should rarely be used.

Multiple @s in routes are prohibited by many mailers. To circumvent this restriction, mailers instead support the "magic %" rule, described below. When **pathalias** would otherwise generate a path containing multiple @s, it instead generates a path to which the "magic %" rule can be correctly applied.

Basically, the "magic %" rule for generating paths is "when constructing a path that would require multiple @s, replace all but the right-most @ with %".

When a mailer that supports the "magic %" rule receives a message that was routed to it via `..path..@host`, it processes the route as follows:

1. Remove the trailing "@host" part of the route.
2. Examine the remaining route from right to left, proceeding to the next step when a "!" is seen. If a '%' is seen, change it to '@' and proceed to the next step immediately.
3. Continue processing the message using the modified route. If the modified route contains both '!' and '@' characters, the exact selection of the next host to route the message is governed by the specific precedence of '!' vs. '@' at this host.

For example, if a host, **jazz.nonesuch.com**, received a message with a path **foo!joe%castle.hrh.gov.uk@jazz.nonesuch.com**, the mailer would convert the path to **foo!joe@castle.hrh.gov.uk**, and then forward it appropriately. If the host were configured such that '!' were of higher precedence than '@', the message would be forwarded to host **foo**, which would then deliver the message to **joe@castle.hrh.gov.uk**. If instead **jazz.nonesuch.com** were configured with '@' as higher in precedence, it would forward the message to host **castle.hrh.gov.uk**, which would then deliver it to **foo!joe**. (Clearly, **pathalias** could only correctly generate such a path if it knew the precedence at host **jazz.nonesuch.com**; since the database does not contain that information, such paths from **pathalias** should be viewed with suspicion.)

The **-D** option suppresses insignificant routes to domain members. This is benign, perhaps even beneficial, but confusing, since the behavior is undocumented and somewhat unpredictable.

## AUTHOR

**pathalias** was developed by Peter Honeyman and Steven M. Bellovin.

## FILES

**newsgroup comp.mail.maps** Likely location of some input files.

## SEE ALSO

P.Honeyman and S.M. Bellovin, *PATHALIAS or The Care and Feeding of Relative Addresses*, in *Proc. Summer USENIX Conf.*, Atlanta, 1986.

**NAME**

pathchk - check path names

**SYNOPSIS**

pathchk [-p] *pathname* ...

**DESCRIPTION**

The **pathchk** command checks that one or more path names are valid and portable. By default, the **pathchk** command checks each component of each path name specified by the *pathname* parameter based on the underlying file system. An error message is written for each path name operand that:

- is longer than that allowed by the system.
- contains any component longer than that allowed by the system.
- contains any component in a directory that is not searchable.
- contains any character in any component that is not valid in its containing directory.

It is not considered an error if one or more components of a path name do not exist, as long as a file matching the path name specified by the *pathname* parameter could be created that does not violate any of the checks above.

More extensive portability checks are performed when the **-p** flag is specified.

**Options**

The **pathchk** command supports the following option:

- p** Performs path name checks based on POSIX portability standards instead of the underlying file system. An error message is written for each path name that:
  - is longer than `_POSIX_PATH_MAX` bytes.
  - contains any component longer than `_POSIX_NAME_MAX` bytes.
  - contains any character in any component that is not in the portable file name character set.

**EXTERNAL INFLUENCES****Environment Variables**

`LC_MESSAGES` determines the language in which messages are displayed.

If `LC_MESSAGES` is not specified in the environment or is set to the empty string, the value of `LANG` is used as a default for each unspecified or empty variable. If `LANG` is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of `LANG`.

If any internationalization variable contains an invalid setting, **pathchk** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single-byte and multi-byte character code sets are supported.

**RETURN VALUE**

Upon successful completion, **pathchk** returns zero; otherwise it returns nonzero to indicate an error.

**EXAMPLES**

To check the validity and portability of the

```
/users/mary/work/tempfiles
```

path name on your system, use:

```
pathchk /users/mary/work/tempfiles
```

To check the validity and portability of the

```
/users/mary/temp
```

path name for POSIX standards, use:

```
pathchk -p /users/mary/temp
```



**STANDARDS CONFORMANCE**

**pathchk:** XPG4, POSIX.2

**NAME**

pax - Extracts, writes, and lists archive files; copies files and directory hierarchies

**SYNOPSIS****Listing Member Files of Archived Files**

```
pax [-cdnv] [-f archive] [-s replstr] ... [pattern ...]
```

**Extracting Archive Files**

```
pax -r [-cdiknuvy] [-f archive] [-p string] ... [-s replstr] ... [pattern ...]
```

**Writing Archive Files**

```
pax -w [-adituvXy] [-b blocking] [-f archive] [-s replstr] ... [-x format] [file ...]
```

**Copying Files**

```
pax -r -w [-diklntuvXy] [-p string] ... [-s replstr] ... [file ...] directory
```

**DESCRIPTION**

The **pax** command extracts and writes member files of archive files; writes lists of the member files of archives; and copies directory hierarchies. The **-r** and **-w** flags specify the archive operation performed by the **pax** command.

The *pattern* argument specifies a pattern that matches one or more paths of archive members. A \ (backslash) character is not recognized in the *pattern* argument and it prevents the subsequent character from having any special meaning. If no *pattern* argument is specified, all members are selected in the archive.

If a *pattern* argument is specified, but no archive members are found that match the pattern specified, the **pax** command detects the error, exits with a nonzero exit status, and writes a diagnostic message.

The **pax** command can read both **tar** and **cpio** archives. In the case of **cpio**, this means that **pax** can read ASCII archives (which are created with **cpio -c**) and binary archives (which are created without the **-c** flag). The supported archive formats are automatically detected on input.

**pax** can also write archives that **tar** and **cpio** can read; by default, **pax** writes archives in the **ustar** extended **tar** interchange format. **pax** also writes ASCII **cpio** archives; use the **-x cpio** flag to specify this extended **cpio** output format.

**P****Options**

- a** Appends files to the end of the archive. Certain devices might not support appending.
- b blocking** Specifies the block size for output to be the positive decimal integer of bytes specified by the *blocking* argument. The block size value cannot exceed 32,256. Blocking is automatically determined on input.  
Do not specify a value for the *blocking* argument larger than 32768. Default blocking when creating archives depends on the archive format. (See the **-x** flag description.)
- c** Matches all file or archive members except those specified by the *pattern* or *file* arguments.
- d** Causes directories being copied or archived, or archived directories being extracted, to match only the directory or archived directory itself and not the contents of the directory or archived directory.
- f archive** Specifies the path of an archive file to be used instead of standard input (when the **-w** flag is not specified) or the standard output (when the **-w** flag is specified but the **-r** flag is not). When specified with the **-a** flag, any files written to the archive are appended to the end of the archive.
- i** Renames files or archives interactively. For each archive member that matches the *pattern* argument or file that matches a *file* argument, a prompt is written to the terminal (**/dev/tty**) that contains the name of a file or archive member. A line is then read from the terminal. If this line is empty, the file or archive member is skipped. If this line consists of a dot, the file or archive member is processed with no modification to its name. Otherwise, its name is replaced with the contents of the line. The **pax** command immediately exits with a nonzero exit status if an End-of-File is encountered when reading a response or if it cannot read or write to the terminal.

- k** Prevents the **pax** command from writing over existing files.
- l** Links files when copying files. When both **-r** and **-w** are specified, hard links are established between the source and destination file hierarchies whenever possible.
- n** Selects the first archive member that matches each *pattern* argument. No more than one archive member is matched for each pattern (although members of type directory will still match the file hierarchy rooted at that file).
- p string** Specifies one or more file characteristics to be retained or discarded on extraction. The *string* argument consists of the characters **a**, **e**, **m**, **o**, and **p**. Multiple characteristics can be concatenated within the same string and multiple **-p** flags can be specified. The specification flags have the following meanings:
  - a** Does not retain file-access times.
  - e** Retains the user ID, group ID, access permission, access time, and modification time.
  - m** Does not retain file-modification times.
  - o** Retains the user ID and the group ID.
  - p** Retains the access permission.

Note that "retain" means that an attribute stored in the archive is given to the extracted file, subject to the permissions of the invoking process; otherwise, the attribute is determined as part of the normal file creation action.

If neither the **e** nor the **o** flag is specified, or the user ID and group ID are not retained, the **pax** command does not set the **S\_ISUID** and **S\_ISGID** bits of the access permission. If the retention of any of these items fails, the **pax** command writes a diagnostic message to standard error. Failure to retain any of the items affects the exit status, but does not cause the extracted file to be deleted. If specification flags are duplicated or conflict with each other, the ones given last take precedence. For example, if **-p eme** is specified, file-modification times are retained.

- r** Reads an archive file from the standard input.
- s** Modifies file-member or archive-member names specified by the *pattern* or *file* arguments according to the substitution expression *replstr*, using the syntax of the **ed** command. The substitution expression has the following format:

**-s/old/new/[gp]**

where as in the **ed** command, *old* is a basic regular expression and *new* can contain an **&** (ampersand), **\n** (*n* is a digit) back references, or subexpression matching. The *old* string can also contain newline characters.

Any nonnull character can be used as a delimiter (the **/** (slash) character is the delimiter in the previous format). Multiple **-s** flag expressions can be specified; the expressions are applied in the order specified, terminating with the first successful substitution. The optional trailing **g** character performs as in the **ed** command. The optional trailing **p** character causes successful substitutions to be written to the standard error. File-member or archive-member names that substitute to the empty string are ignored when reading and writing archives.

- t** Causes the access times of the archived files to be the same as they were before being read by the **pax** command.
- u** Ignores files that are older (having a less recent file modification time) than a preexisting file or archive member with the same name.

When extracting files (**-r** flag), an archive member with the same name as a file in the file system is extracted if the archive member is newer than the file.

When writing files to an archive file (**-w** flag), an archive member with the same name as a file in the file system is superseded if the file is newer than the archive member.

When copying files to a destination directory (**-rw** flags), the file in the destination hierarchy is replaced by the file in the source hierarchy or by a link to the file in the source hierarchy if the file in the source hierarchy is newer.

p

- v** Writes information about the process. If neither the **-r** or **-w** flags are specified, the **-v** flag produces a verbose table of contents that resembles the output of **ls -l**; otherwise, archive-member pathnames are written to standard error.
- w** Writes files to the standard output in the specified archive format.
- x format** Specifies the output archive format. The **pax** command recognizes the following formats:
  - cpio** Extended **cpio** interchange format. The default blocking value for this format for character special archive files is 5120. Blocking values from 512 to 32,256 in increments of 512 are supported.
  - ustar** Extended **tar** interchange format. This is the default output archive format. The default blocking value for this format for character special archive files is 10240. Blocking values from 512 to 32,256 in increments of 512 are supported.

Any attempt to append to an archive file in a format different from the existing archive format causes the **pax** command to exit immediately with a nonzero exit status.
- x** When traversing the file hierarchy specified by a pathname, the **pax** command does not descend into directories that have a different device ID.
- y** Prompts interactively for the disposition of each file. Substitutions specified by **-s** flags are performed before you are prompted for disposition. An EOF marker or an input line starting with the character **q** causes **pax** to exit. Otherwise, an input line starting with anything other than **y** causes the file to be ignored. This flag cannot be used in conjunction with the **-i** flag.

### Option Interaction and Processing Order

The flags that operate on the names of files or archive members (**-c**, **-i**, **-n**, **-s**, **-u**, and **-v**) interact as follows.

When extracting files (**-r** flag), archive members are selected, using the modified names, according to the user-specified pattern arguments as modified by the **-c**, **-n**, and **-u** flags. Then, any **-s** and **-i** flags modify, in that order, the names of the selected files. The **-v** flag writes the names resulting from these modifications.

When writing files to an archive file (**-w** flag), or when copying files, the files are selected according to the user-specified pathnames as modified by the **-n** and **-u** flags. Then, any **-s** and **-i** flags modify, in that order, the names resulting from these modifications. The **-v** flag writes the names resulting from these modifications.

If both the **-u** and **-n** flags are specified, the **pax** command does not consider a file selected unless it is newer than the file to which it is compared.

### Listing Member Files of Archived Files

When neither the **-r** nor the **-w** flags are specified, the **pax** command writes the names of the members of the archive file read from the standard input, with pathnames matching the specified patterns, to the standard output. If a named file is a directory, the file hierarchy contained in the directory is also written. You can specify the **pax** command without the **-r** or **-w** flags with the **-c**, **-d**, **-f**, **-n**, **-s**, and **-v** flags, and with the *pattern* argument.

If neither the **-r** or **-w** flags are included, **pax** lists the contents of the specified archive, one file per line. **pax** lists hard link pathnames as follows:

```
pathname ==linkname
```

**pax** lists symbolic link pathnames as follows:

```
pathname ->linkname
```

In both of the preceding cases, *pathname* is the name of the file that is being extracted, and *linkname* is the name of a file that appeared earlier in the archive.

If the **-v** flag is specified, the listing of hard link pathnames is output in the **ls -l** command format.

### Extracting Archive Files

When the **-r** flag is specified, but the **-w** flag is not, the **pax** command extracts the members of an archive file read from the standard input, and with pathnames matching the *pattern* argument if one is specified. If an extracted file is a directory, the file hierarchy contained in the directory is also extracted.

The extracted files are created relative to the current file hierarchy. The **-r** flag can be specified with the **-c**, **-d**, **-f**, **-n**, **-s**, and **-v** flags, and a *pattern* argument.

The access and modification times of the extracted files are the same as the archived files. The access permissions of the extracted files remain as archived unless affected by the user's default file creation mode. The **S\_ISUID** and **S\_ISGID** bits of the extracted files are cleared.

If intermediate directories are necessary to extract an archive member, the **pax** command creates the directories with access permissions set as the bitwise inclusive OR of the values of the **S\_IRWXU**, **S\_IRWXG**, and **S\_IRWXO** options.

If the selected archive format supports the specification of linked files (both the **tar** and **cpio** formats do), it is an error if these files cannot be linked when the archive is extracted. **pax** informs you of the error and continues processing.

### Writing Archive Files

When the **-w** flag is specified and the **-r** flag is not, the **pax** command writes the contents of the files specified by the *file* arguments to the standard output in an archive format. If no *file* arguments are specified, a list of files to copy, one per line, is read from the standard input. When the *file* argument specifies a directory, all of the files contained in the directory are written. The **-w** flag can be specified with the **-b**, **-d**, **-f**, **-i**, **-s**, **-t**, **-u**, **-v**, **-x**, and **-X** flags and with *file* arguments.

If **-w** is specified, but no files are specified, standard input is used. If neither **-f** or **-w** are specified, standard input must be an archive file.

### Copying Files

When both the **-r** and **-w** flags are specified, the **pax** command copies the files specified by the *file* arguments to the destination directory specified by the *directory* argument. If no file arguments are specified, a list of files to copy, one per line, is read from the standard input. If a specified file is a directory, the file hierarchy contained in the directory is also copied. The **-r** and **-w** flags can be specified with the **-d**, **-i**, **-k**, **-l**, **-p**, **-n**, **-s**, **-t**, **-u**, **-v**, and **-X** flags and with the *file* arguments. A *directory* argument must be specified.

Copied files are the same as if they were written to an archive file and subsequently extracted, except that there may be hard links between the original and the copied files.

### RETURN VALUE

The **pax** command returns a value of 0 (zero) if all files were successfully processed; otherwise, **pax** returns a value greater than 0 (zero).

### EXAMPLES

To copy the contents of the current directory to the tape drive, enter:

```
pax -w -f /dev/rmt/0m .
```

To copy the **olddir** directory hierarchy to **newdir** enter:

```
mkdir newdir
cd olddir
pax -rw olddir newdir
```

To read the archive **a.pax**, with all files rooted in the directory **/usr** in the archive extracted relative to the current directory, enter:

```
pax -r -s ' , /*usr/* , , ' -f a.pax
```

All of the preceding examples create archives in **tar** format.

The following pairs of commands demonstrate conversions from **cpio** and **tar** to **pax**. In all cases, the examples show comparable command-line usage rather than identical output formats. The **-x** flag can be specified to the **pax** commands shown here, producing archives to select specific output formats:

```
ls * | cpio -ocv
pax -wdv *

find /mydir -type f -print | cpio -oc
find /mydir -type f -print | pax -w

cpio -icdum < archive
pax -r < archive
```

```
(cd /fromdir; find . -print) | cpio -pdlum /todir
pax -rwl /fromdir /todir

tar cf archive *
pax -w -f archive *

tar xfv - < archive
pax -rv < archive

(cd /fromdir; tar cf - .) | (cd /todir; tar xf -)
pax -rw /fromdir /todir
```

**Notes**

When you use the **-i** flag (interactively renames files) on files to which there are hard links, **pax** does *not* create hard links to the renamed files.

**WARNINGS**

Because of industry standards and interoperability goals, **pax** does not support the archival of files larger than 8GB or files that have user/group IDs greater than 2048K. Files with user/group IDs greater than 2048K are archived and restored under the user/group ID of the current process, unless the `uname/gname` exists.

**AUTHOR**

**pax** was developed by Mark H. Colburn, OSF, and HP.

**SEE ALSO**

`ed(1)`, `tar(4)`.

**STANDARDS CONFORMANCE**

**pax**: XPG4, POSIX.2

This implementation of **pax** is based upon a *POSIX.2 draft* specification. HP intends to update **pax** to meet the final POSIX.2 Standard once it completes, and thus the **pax** implementation is likely to change in a future release of HP-UX, possibly in ways incompatible with the current implementation. HP recommends using the current implementation only if absolutely necessary.

**| p |**

**NAME**

pg - file perusal filter for soft-copy terminals

**SYNOPSIS**

**pg** [-*number*] [-*pstring*] [-*cefnrs*] [+*linenumber*] [+/*pattern*] [*file ...*]

**Remarks**

**pg** and **more** are both used in similar situations (see *more(1)*). Text highlighting features supported by **more** are not available from **pg**. However, **pg** has some useful features not provided by **more**.

**DESCRIPTION**

**pg** is a *text file* filter that allows the examination of *files* one screenful at a time on a soft-copy terminal. If **-** is used as a *file* argument, or **pg** detects NULL arguments in the command line, the standard input is used. Each screenful is followed by a prompt. To display a new page, press **Return**. Other possibilities are enumerated below.

This command is different from other paginators such as **more** in that it can back up for reviewing something that has already passed. The method for doing this is explained below.

In order to determine terminal attributes, **pg** scans the **terminfo** data base for the terminal type specified by the environment variable **TERM** (see *terminfo(4)*). If **TERM** is not defined, terminal type **dumb** is assumed.

**Options**

**pg** recognizes the following command line options:

- number**      *number* is an integer specifying the size (in lines) of the window that **pg** is to use instead of the default (on a terminal containing 24 lines, the default window size is 23).
- p string**    Causes **pg** to use *string* as the prompt. If the prompt string contains a **%d**, the first occurrence of **%d** in the prompt is replaced by the current page number when the prompt is issued. The default prompt string is a colon (**:**).
- c**            Home the cursor and clear the screen before displaying each page. This option is ignored if **clear\_screen** is not defined in the **terminfo** data base for this terminal type.
- e**            Causes **pg** to *not* pause at the end of each file.
- f**            Normally, **pg** splits lines longer than the screen width, but some sequences of characters in the text being displayed (such as escape sequences for underlining) generate undesirable results. The **-f** option inhibits **pg** from splitting lines.
- n**            Normally, commands must be terminated by a new-line character. This option causes an automatic end-of-command as soon as a command letter is entered.
- r**            Restricted mode. The shell escape is disallowed. **pg** will print an error message but does not exit.
- s**            Causes **pg** to print all messages and prompts in standout mode (usually inverse video).
- +linenumber**   Start display at *linenumber*.
- +/*pattern* /**   Start up at the first line containing text that matches the regular expression *pattern*.

**pg** looks in the environment variable **PG** to preset any flags desired. For example, if you prefer to view files using the **-c** mode of operation, the POSIX-shell command sequence **PG='-c' ; export PG** or the C-shell command **setenv PG -c** causes all invocations of **pg**, including invocations by programs such as **man** and **msgs**, to use this mode. The command sequence to set up the **PG** environment variable is normally placed in the user **.profile** or **.cshrc** file. No form of quoting is provided, so the string and pattern arguments are limited to single word.

The responses that can be typed when **pg** pauses can be divided into three categories: those causing further perusal, those that search, and those that modify the perusal environment.

Commands that cause further perusal normally take a preceding *address*, an optionally signed number indicating the point from which further text should be displayed. This *address* is interpreted either in

pages or lines, depending on the command. A signed *address* specifies a point relative to the current page or line; an unsigned *address* specifies an address relative to the beginning of the file. Each command has a default address that is used if none is provided.

Perusal commands and their defaults are as follows:

|                                            |                                                                                                                                                                                                               |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (+1)< <i>newline</i> > or < <i>blank</i> > | Displays one page. The address is specified in pages.                                                                                                                                                         |
| (+1) <b>l</b>                              | With a relative address, <b>pg</b> simulates scrolling the screen, forward or backward, the number of lines specified. With an absolute address <b>pg</b> prints a screenful beginning at the specified line. |
| (+1) <b>d</b> or <b>^D</b>                 | Simulates scrolling a half-screen forward or backward.                                                                                                                                                        |
| <b>i f</b>                                 | Skip <i>i</i> screens of text.                                                                                                                                                                                |
| <b>i z</b>                                 | Same as <i>newline</i> except that <i>i</i> , if present, becomes the new default number of lines per screenful.                                                                                              |

The following perusal commands take no *address*:

|                       |                                                                                     |
|-----------------------|-------------------------------------------------------------------------------------|
| <b>.</b> or <b>^L</b> | Typing a single period causes the current page of text to be redisplayed.           |
| <b>\$</b>             | Displays the last windowful in the file. Use with caution when the input is a pipe. |

The following commands are available for searching for text patterns in the text. The Basic Regular Expression syntax (see *regex(5)*) is supported. The terminal **/**, **^**, or **?** can be omitted from the pattern search commands. Regular expressions must always be terminated by a new-line character, even if the **-n** option is specified.

|                                            |                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>i /pattern /</b>                        | Search forward for the <i>i</i> th (default <i>i</i> =1) occurrence of <i>pattern</i> . Searching begins immediately after the current page and continues to the end of the current file, without wrap-around.                                                                                                             |
| <b>i ^pattern ^</b><br><b>i ?pattern ?</b> | Search backwards for the <i>i</i> th (default <i>i</i> =1) occurrence of <i>pattern</i> . Searching begins immediately before the current page and continues to the beginning of the current file, without wrap-around. The <b>^</b> notation is useful for Adds 100 terminals which cannot properly handle the <b>?</b> . |

After searching, **pg** normally displays the line found at the top of the screen. This can be modified by appending **m** or **b** to the search command to leave the line found in the middle or at the bottom of the window from now on. The suffix **t** can be used to restore the original situation.

**pg** users can modify the perusal environment with the following commands:

|                      |                                                                                                                                                                                                                                                                        |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>i n</b>           | Begin perusing the <i>i</i> th next file in the command line. The <i>i</i> is an unsigned number, default value is 1.                                                                                                                                                  |
| <b>i p</b>           | Begin perusing the <i>i</i> th previous file in the command line. <i>i</i> is an unsigned number, default is 1.                                                                                                                                                        |
| <b>i w</b>           | Display another window of text. If <i>i</i> is present, set the window size to <i>i</i> .                                                                                                                                                                              |
| <b>s filename</b>    | Save the input in the named file. Only the current file being perused is saved. The white space between the <b>s</b> and <i>filename</i> is optional. This command must always be terminated by a new-line character, even if the <b>-n</b> option is specified.       |
| <b>h</b>             | Help by displaying an abbreviated summary of available commands.                                                                                                                                                                                                       |
| <b>q</b> or <b>Q</b> | Quit <i>pg</i> .                                                                                                                                                                                                                                                       |
| <b>!command</b>      | <i>command</i> is passed to the shell, whose name is taken from the <b>SHELL</b> environment variable. If this is not available, the default shell is used. This command must always be terminated by a new-line character, even if the <b>-n</b> option is specified. |

At any time when the output is being sent to the terminal, the user can press the quit key (normally CTRL-**\**), the interrupt (break) key or the DEL key. This causes **pg** to stop sending output, and display the prompt. The user may then enter one of the 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.



If the standard output is not a terminal, **pg** is functionally equivalent to **cat** (see *cat(1)*), except that a header is printed before each file if more than one file is specified.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_COLLATE** determines the collating sequence used in evaluating regular expressions.

**LC\_CTYPE** determines the interpretation of text as single and/or multi-byte characters, and the characters matched by character class expressions in regular expressions.

**LANG** determines the language in which messages are displayed.

If **LC\_COLLATE** or **LC\_CTYPE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **pg** behaves as if all internationalization variables are set to "C". See *environ(5)*.

### International Code Set Support

Single- and multi-byte character code sets are supported.

## EXAMPLES

To use **pg** when reading system news:

```
news | pg -p "(Page %d):"
```

## WARNINGS

If terminal tabs are not set every eight positions, undesirable results may occur.

When using **pg** as a filter with another command that changes the terminal I/O options (such as *crypt(1)*), terminal settings may not be restored correctly.

While waiting for terminal input, **pg** responds to **BREAK**, **DEL**, and **^** by terminating execution. Between prompts, however, these signals interrupt *pg*'s current task and place the user in prompt mode. These should be used with caution when input is being read from a pipe, because an interrupt is likely to terminate the other commands in the pipeline.

Users of **more** will find that the **z** and **f** commands are available, and that the terminal **/**, **^**, or **?** can be omitted from the pattern search commands.

## FILES

|                                          |                                          |
|------------------------------------------|------------------------------------------|
| <code>/usr/share/lib/terminfo/?/*</code> | terminal information data base           |
| <code>/tmp/pg*</code>                    | temporary file when input is from a pipe |

## SEE ALSO

*crypt(1)*, *grep(1)*, *more(1)*, *terminfo(4)*, *environ(5)*, *lang(5)*, *regexp(5)*.

## STANDARDS CONFORMANCE

**pg**: SVID2, SVID3, XPG2, XPG3

**NAME**

pipcrm - remove a POSIX message queue or a POSIX named semaphore

**SYNOPSIS**

**pipcrm** [*option*]...

**DESCRIPTION**

The **pipcrm** command removes one or more specified POSIX message queue or POSIX named semaphore.

**Options**

The identifiers are specified by the following *options*:

- q** *msqname* Unlinks the queue name from the message queue specified by the argument, *msqname*. If there are no processes with existing open descriptors for the message queue, the queue is destroyed. If one or more processes have the message queue open, the removal of the queue is postponed until all descriptors for the queue have been closed.
- s** *semname* Unlinks the semaphore name from the semaphore specified by the argument, *semname*. If there are no processes with existing open descriptors for the semaphore, the semaphore is destroyed. If one or more processes have the semaphore open, the removal of the semaphore is postponed until all descriptors for the semaphore have been closed.

The details of the removals are described in *mq\_unlink(2)*, and *sem\_unlink(2)*. The names can be found by using **pipcs** (see *pipcs(1)*).

**SEE ALSO**

*pipcs(1)*, *mq\_unlink(2)*, *mq\_open(2)*, *mq\_send(2)*, *mq\_receive(2)*, *sem\_unlink(2)*, *sem\_open(2)*, *sem\_wait(2)*, *sem\_post(2)*.

**STANDARDS CONFORMANCE**

**pipcrm**: NONE

**| P |**

**NAME**

pipcs - report status of POSIX interprocess communication facilities

**SYNOPSIS**

**pipcs** [-qs] [-abo]

**DESCRIPTION**

**pipcs** displays certain information on active POSIX interprocess communication facilities. When no options are specified, **pipcs** displays information in short format for the POSIX message queues and POSIX named semaphores that are currently active in the system.

**Options**

The following options restrict the display to the corresponding facilities.

- (none) This is equivalent to **-qs**.
- q** Display information on active POSIX message queues.
- s** Display information on active POSIX named semaphores.

The following options add columns of data to the display. See "Column Description" below.

- (none) Display default columns: for all facilities: **T**, **MODE**, **OWNER**, **GROUP**, **REFCNT**, **NAME**.
- a** Display all columns, as appropriate. This is equivalent to **-bo**.
- b** Display largest-allowable-size information: for POSIX message queues: **QBYTES**.
- o** Display information on outstanding usage: for POSIX message queues: **QNUM**, **MNUM**; for POSIX named semaphores: **VAL**, **IVAL**.

**Column Descriptions**

The column headings and the meaning of the columns in an **pipcs** listing are given below. The columns are printed from left to right in the order shown below.

- T** Facility type:
  - q** POSIX message queue
  - s** POSIX named semaphore
- MODE** The facility access modes and flags: The mode consists of 9 characters that are interpreted as follows:
 

The 9 characters are interpreted as three sets of three characters each. The first set refers to the owner's permissions, the next to permissions of others in the group of the facility entry, and the last to all others.

Within each set, the first character indicates permission to read, the second character indicates permission to write or alter the facility entry, and the last character is currently unused.

  - r** Read permission is granted.
  - w** Write permission is granted.
  - a** Alter permission is granted.
  - The indicated permission is not granted.
- OWNER** The login name of the owner of the facility entry.
- GROUP** The group name of the group of the owner of the facility entry.
- REFCNT** The reference count value of the facility entry. Reference count of the POSIX message queue represents number of simultaneous open for the associated one. Similarly reference count of the POSIX named semaphore represents number of process referencing the associated one.
- NAME** The name of the facility entry.
- QNUM** The number of messages currently outstanding on the associated POSIX message queue.

**| p |**

|               |                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------|
| <b>MNUM</b>   | The maximum number of messages allowed on the associated POSIX message queue.                      |
| <b>QBYTES</b> | The maximum number of bytes allowed in messages outstanding on the associated POSIX message queue. |
| <b>VAL</b>    | The current value on the associated POSIX named semaphore.                                         |
| <b>IVAL</b>   | The initial value hold on the associated POSIX named semaphore.                                    |

**WARNINGS**

**pipcs** produces only an approximate indication of actual system status because system processes are continually changing while **pipcs** is acquiring the requested information.

Do not rely on the exact field widths and spacing of the output, as these will vary depending on the system, the release of HP-UX, and the data to be displayed.

**FILES**

|                    |             |
|--------------------|-------------|
| <b>/etc/group</b>  | Group names |
| <b>/etc/passwd</b> | User names  |

**SEE ALSO**

pipcrm(1), mq\_open(2), mq\_close(2), sem\_open(2), sem\_close(2), mq\_unlink(2), sem\_unlink(2), mq\_send(2), mq\_receive(2), sem\_wait(2), sem\_post(2).

**STANDARDS CONFORMANCE**

**pipcs**: NONE

**NAME**

pppd - PPP daemon

**SYNOPSIS****pppd** [*options...*]**DESCRIPTION**

**pppd** is a daemon process used in UNIX systems to manage connections to other hosts using PPP (Point to Point Protocol) or SLIP (Serial Line Internet Protocol). It uses the UNIX host's native serial ports. It communicates with the UNIX kernel's own TCP/IP implementation via the HP IP tunnel driver.

The functionality supplied by this daemon supersedes that provided by *pppl*(1) in HP-UX prior to Release 10.30. **pppl** only supported the SLIP and CSLIP (Compressed Header SLIP) protocols.

**Daemon Management Options**

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>auto</b>                      | Start in 'autocall' mode and detach from the controlling terminal to run as a daemon. Initiate a connection in response to a packet specified in the 'bringup' category in filter-file. Requires the remote address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>up</b>                        | When used with <b>auto</b> , bring the link up immediately rather than waiting for traffic. If the link goes down, attempt to restart it (after the call retry delay timer expires) without waiting for an outbound packet.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>dedicated</b>                 | Treat the connection as a dedicated line rather than a demand-dial connection. This option tells <b>pppd</b> to never give up on the connection; that is, if the peer tries to shut down the link, go ahead and do so, but then immediately try to reestablish the connection. Similarly, when first trying to connect, <b>pppd</b> will not give up after sending a fixed number of Configure-Request messages. Hangup events (LQM failures, loss of Carrier Detect) will still cause the device to be closed, just as with dial-up connections, and the <b>Systems</b> file will then be checked for alternate entries. If none are available, the connection will be reestablished after the call retry delay timer expires. Use a short call retry delay timer on dedicated circuits; something like <b>Any;5-30</b> should work well. Implies <b>up</b> . |
| <b>altdelay</b> <i>delay</i>     | Set the delay of <i>delay</i> seconds between dialing each alternate numbers in the <b>Systems</b> file for the same destination. The default value is 1 second.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>nodetach</b>                  | Don't detach from the controlling terminal in 'autocall' mode. When used with <b>log -</b> , this can be useful for watching the progress of the PPP session.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>log</b> <i>log-file</i>       | Append logging messages to log-file (default: <b>/var/adm/pppd.log</b> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>acct</b> <i>acct-file</i>     | Append session accounting messages to acct-file. If acct-file is the same as log-file, the session accounting messages will be interleaved with other logging information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>filter</b> <i>filter-file</i> | Look in filter-file for packet filtering and link management information (default: <b>/etc/ppp/Filter</b> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>debug</b> <i>debug-level</i>  | Set the log file verbosity to the following debug-level and each debugging verbosity level also provides the information of all the lower-numbered levels. <ul style="list-style-type: none"> <li>0 Daemon start messages</li> <li>1 Link status messages, calling attempts (the default)</li> <li>2 Chat script processing, input framing errors</li> <li>3 LCP, IPCP, PAP and CHAP negotiation</li> <li>4 LQM status summaries</li> <li>5 IP interface changes</li> <li>6 IP message summaries</li> <li>7 Full LQM reports</li> <li>8 All PPP messages (without framing)</li> </ul>                                                                                                                                                                                                                                                                          |

- 9 Characters read or written
- 10 Procedure call messages
- 11 Internal timers

**exec** *exec-cmd* Run ‘**exec-cmd up** *addr args*’ when the link comes up, and ‘**exec-cmd down** *addr args*’ when it goes down. *Addr* is the IP address of the peer, and *args* is the list of arguments given to **pppd**.

**nonice** Run at a normal user process priority, rather than using the nice() library routine to elevate **pppd** scheduling priority to -10.

### Communications Options

**asynctest** *asynctest* Set the desired Async Control Character Map to *asynctest*, expressed in C-style hexadecimal notation (default 0xA000).

**noasynctest** Disable LCP Async Control Character Map negotiation.

**escape** *odd-character* In addition to those characters specified in the PPP Async Control Character Map (which can include only 0x00 through 0x1F), also apply the escaping algorithm when transmitting *odd-character*. The value of *odd-character* must be between 0x00 and 0xFF, and cannot be any of 0x5E, 0x7D or 0x7E.

*Odd-character* can be specified as a decimal number, in C-style hexadecimal notation, or as an ASCII character with optional ‘^’ control-character notation. For example, the XON character could be specified as 17, 0x11, or ^Q.

If a character specified with the **escape** argument, when transformed into its escaped form, would be the same as a character contained in the peer’s negotiated Async Control Character Map, a warning will be printed in the log file and the character specified on the command line will not be escaped.

If a character specified with the **escape** argument, when transformed into its escaped form, would be the same as a character specified in another **escape** argument on the daemon’s command line, **pppd** will print an error message and exit.

**device** Communicate over the named device (default /**dev/tty**).

**comm-speed** Set communications rate to *comm-speed* bits per second.

**ignore-cd** Ignore the state of the CD (Carrier Detect, also called DCD, Data Carrier Detect) signal. This is useful for systems that don’t support CD but want to run PPP over a dedicated line.

**xonxoff** Set the line to use in-band (‘software’) flow control, using the characters DC3 (^S, XOFF, ASCII 0x13) to stop the flow and DC1 (^Q, XON, ASCII 0x11) to resume. (The default is to use no flow control.) For an outbound connection, this may be specified either in **Devices** or on the **pppd** command line.

**telnet** When used on an answering **pppd** command line, negotiate the telnet binary option and understand telnet escape processing. Not for use with **device** or **auto**.

### Link Management Options

**nooptions** Disable all LCP and IPCP options.

**noaccomp** Disable HDLC Address and Control Field compression.

**noptcomp** Disable LCP Protocol Field Compression.

**slip** Use RFC 1055 SLIP packet framing rather than PPP packet framing. Disables all option negotiation, and implies **noasynctest**, **noipaddress**, **vjslots 16**, **novjcid**, **nomagic**, **nomru**, and **mru 1006**. Implies **vjcomp** if peer sends a header-compressed TCP packet.

**extra-slip-end** When running in SLIP mode, prepend a SLIP packet framing character (0xC0) to each frame before transmission, even if this frame immediately follows the previous frame. By default, **pppd** transmits only one framing character between adjacent SLIP frames.

|                                                            |                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>extra-ppp-flag</b>                                      | When running in PPP mode, prepend a PPP packet framing character (0x7E) to each frame before transmission, even if this frame immediately follows the previous frame. By default, <b>pppd</b> transmits only one framing character between adjacent PPP frames.                                                                                                                               |
| <b>nomagic</b>                                             | Disable LCP Magic Number negotiation.                                                                                                                                                                                                                                                                                                                                                         |
| <b>mru</b> <i>mru-size</i>                                 | Set LCP Maximum Receive Unit value to <i>mru-size</i> for negotiation. The default is 1500 for PPP and 1006 for SLIP.                                                                                                                                                                                                                                                                         |
| <b>nomru</b>                                               | Disable LCP Maximum Receive Unit negotiation, and use 1500 for our interface.                                                                                                                                                                                                                                                                                                                 |
| <b>active</b>                                              | Begin LCP parameter negotiation immediately (the default).                                                                                                                                                                                                                                                                                                                                    |
| <b>passive</b>                                             | Do not send our first LCP packet until we receive an LCP packet from the peer.                                                                                                                                                                                                                                                                                                                |
| <b>timeout</b> <i>restart-time</i>                         | Set the LCP, IPCP, CCP, PAP, and CHAP option negotiation restart timers to <i>restart-time</i> (default 3 seconds).                                                                                                                                                                                                                                                                           |
| <b>lqrinterval</b> <i>time</i>                             | Send Link-Quality-Reports or Echo-Requests every <i>time</i> seconds (default 10 seconds). If the peer responds with a Protocol-Reject, send LCP Echo-Requests every <i>time</i> seconds instead, and use the received LCP Echo-Replies for link status policy decisions.                                                                                                                     |
| <b>lqthreshold</b> <i>min/per</i>                          | Set a minimum standard for link quality by considering the connection to have failed if fewer than <i>min</i> out of the last <i>per</i> LQRs we sent have been responded to by the peer (default 1/5).                                                                                                                                                                                       |
| <b>echolqm</b>                                             | Use LCP Echo-Requests rather than standard Link-Quality-Report messages for link quality assessment and policy decisions. The peer can override this if it actively tries to configure Link Quality Monitoring unless the <b>nolqm</b> parameter is also specified.                                                                                                                           |
| <b>nolqm</b>                                               | Don't send or recognize Link-Quality-Report messages. If <b>echolqm</b> is also specified, Echo-Request messages will be used to detect link failures.                                                                                                                                                                                                                                        |
| <b>idle</b> <i>idle-time</i> [ <i>/session-idle-time</i> ] | Shut down the link when <i>idle-time</i> seconds pass without receiving or transmitting a packet specified in the 'keepup' category in the filter file (default is to never consider the link idle).<br><br>If <i>session-idle-time</i> is specified and any TCP sessions are open, shut down the link when <i>session-idle-time</i> seconds pass without receiving or transmitting a packet. |
| <b>max-configure</b> <i>tries</i>                          | Set the PPP Max-Configure counter (the maximum number of Configure-Requests sent without a response) to <i>tries</i> .                                                                                                                                                                                                                                                                        |
| <b>max-terminate</b> <i>tries</i>                          | Set the PPP Max-Terminate counter (the maximum number of Terminate-Requests sent without a response) to <i>tries</i> .                                                                                                                                                                                                                                                                        |
| <b>max-failure</b> <i>tries</i>                            | Set the PPP Max-Failure counter (the maximum number of Configure-Naks sent without a positive response) to <i>tries</i> .                                                                                                                                                                                                                                                                     |

### IP Options

*local* : *remote*

The address of this machine, followed by the expected address for the remote machine. Can be specified either as symbolic names or as literal IP addresses, if their addresses cannot be discovered locally without using the PPP link.

Both addresses are optional, but a colon by itself is not valid, and the remote address is required when running as a daemon in 'autocall' mode. If only *local* : is specified when receiving an incoming call, the remote address will be discovered during IPCP IP-Address negotiations.

If either address is followed by a tilde character (~), or if the tilde appears alone, **pppd** accepts the IP address given by the peer during IPCP negotiations, whether for the local end or the peer's end of the link. (not available in SLIP mode)

| p |

Because SLIP cannot perform option negotiations, including IPCP, both addresses should normally be specified, and the tilde option is unavailable. To obtain a similar "feature", the peer must provide the IP address textually during the login process, and a new value must be obtained using the Systems file '\A' chat script feature (see *ppp.Systems* (4)).

|                                   |                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>netmask</b> <i>subnet-mask</i> | Set the subnet mask of the interface to <i>subnet-mask</i> , expressed either in C-style hexadecimal (e.g. 0xffff00) or in decimal dotted-quad notation (e.g. 255.255.255.0). The default subnet mask will be appropriate for the network (class A, B, or C), assuming no subnetting.                                                             |
| <b>need-ip-address</b>            | Ask the peer to assign us an IP address.                                                                                                                                                                                                                                                                                                          |
| <b>noipaddress</b>                | Disable IPCP IP-Address negotiation.                                                                                                                                                                                                                                                                                                              |
| <b>vjcomp</b>                     | Enable RFC 1144 'VJ' Van Jacobson TCP header compression negotiation with 16 slots and slot ID compression (this is the default with PPP framing). 'VJ' compression is enabled by default for async connections, and disabled by default for sync connections.                                                                                    |
| <b>novjcomp</b>                   | Disable RFC 1144 'VJ' Van Jacobson TCP header compression (this is the default with SLIP framing, until the peer sends a header-compressed TCP packet).                                                                                                                                                                                           |
| <b>vjslots</b> <i>vj-slots</i>    | Set the number of VJ compression slots (min 3, max 256, default 16).                                                                                                                                                                                                                                                                              |
| <b>novjcid</b>                    | Disable VJ compression slot ID compression (enabled by default).                                                                                                                                                                                                                                                                                  |
| <b>rfc1172-vj</b>                 | Backwards compatibility with older PPP implementations (4-byte VJ configuration option), but with the correct option negotiation value of 0x002d.                                                                                                                                                                                                 |
| <b>rfc1172-typo-vj</b>            | Backwards compatibility with older PPP implementations (4-byte VJ configuration option) that conform to the typographical error in RFC 1172 section 5.2 (Compression-Type value 0x0037).                                                                                                                                                          |
| <b>rfc1172-addresses</b>          | Backwards compatibility with older PPP implementations that conform to RFC 1172 section 5.1 (IP-Addresses, IPCP configuration option 1) and not with the newer RFC 1332 (IP-Address, IPCP configuration option 3), but that respond with something besides a Configure-Reject when they receive an IPCP Configure-Request containing an option 3. |

| P |

#### Authentication Options

|                               |                                                                                                                                                                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>requireauth</b>            | Require either PAP or CHAP authentication.                                                                                                                                                                                                          |
| <b>requirechap</b>            | Require CHAP authentication as described in RFC 1334.                                                                                                                                                                                               |
| <b>requiremschap</b>          | Require MS-CHAP authentication.                                                                                                                                                                                                                     |
| <b>requirepap</b>             | Require PAP authentication.                                                                                                                                                                                                                         |
| <b>rechap</b> <i>interval</i> | Demand that the peer re-authenticate itself (using CHAP) every <i>interval</i> seconds. If the peer fails the new challenge, the link is terminated.                                                                                                |
| <b>name</b> <i>identifier</i> | Provide the identifier used during PAP or CHAP negotiation. This option is necessary if the PPP peer requires authentication. The default value is the value returned by the <i>gethostname</i> (2) system call or the <i>hostname</i> (1) command. |

#### Microsoft Compatibility Options

|                               |                                                                                                                                                                                 |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ms-dns</b> <i>address</i>  | Set the MS DNS address to provide to the peer. First occurrence of this option on the command line sets the primary address; the second occurrence sets the secondary address.  |
| <b>ms-nbns</b> <i>address</i> | Set the MS NBNS address to provide to the peer. First occurrence of this option on the command line sets the primary address; the second occurrence sets the secondary address. |

#### Encryption Options

Encryption is not currently available in software exported from the USA. However, customer may contact sales@progressive-systems.com to obtain encryption functionality.



**Link Compression Options**

|                       |                                                                                                                                                          |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>compress</b>       | Offer all supported link compression types (currently only Predictor-1) when negotiating. The default is to propose and accept no link compression type. |
| <b>compress-pred1</b> | Accept any supported compression type, but prefer Predictor type 1 compression.                                                                          |
| <b>nopred1</b>        | Never use Predictor-1 compression.                                                                                                                       |

**LOG FILE**

Status information is recorded in the log file (`/var/adm/pppd.log` by default) by each copy of **pppd** running on a single machine. Each line in the file consists of a message preceded by the date, the time, and the process ID number of the daemon writing the message. The quantity and verbosity of messages are controlled with the **debug** option and with the **log** filter (see *ppp.Filter* (4)).

Each packet that brings up the link (at debug level 1 or more), each packet that matches the **log** filter (at any debug level), or any packet when the debug level is 7 or more writes a one-line description of the packet to the log file. The first item of the message is the protocol (**tcp**, **udp**, **icmp**, or a numeric protocol value). For ICMP packets, the keyword **icmp** is followed by the ICMP message type and sub code, separated by slashes. After the protocol comes an IP address and optionally a TCP or UDP port number, followed by an arrow indicating whether the packet was sent (->) or received (<-), followed by another address and port number, followed by the length of the packet in bytes before VJ TCP header compression, followed by zero or more keywords. For transmitted packets, the first IP address is the source address, while for received packets, the first IP address is the destination address. Well known TCP and UDP port numbers will be replaced by the name returned by the **getservbyport()** library function. The keywords and their meanings are:

|                    |                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------|
| <b>frag</b>        | The packet is a middle or later part of a fragmented IP frame.                            |
| <b>syn</b>         | The packet has the TCP SYN bit set.                                                       |
| <b>fin</b>         | The packet has the TCP FIN bit set.                                                       |
| <b>bringup</b>     | The transmitted packet matches the <b>bringup</b> filter and is bringing up the link.     |
| <b>!keepup</b>     | the packet has been rejected by the <b>keepup</b> filter.                                 |
| <b>!pass</b>       | The packet has been rejected by the <b>pass</b> filter.                                   |
| <i>dial failed</i> | The packet was dropped because <b>pppd</b> is waiting for the call retry timer to expire. |
| <b>(c)</b>         | The received packet is VJ TCP header compressed.                                          |
| <b>(u)</b>         | The received packet is VJ TCP header uncompressed.                                        |

For example, the following log file line

```
9/6-14:06:26-83 tcp 63.1.6.3/1050 -> 8.1.1.9/sntp 44 syn
```

indicates that at 2:06:26 PM on September 6, process ID 83 sent a 44-byte TCP packet with the SYN bit set from port 1050 on 63.1.6.3 to the SMTP port on 8.1.1.9.

**SIGNALS**

Upon reception of the following signals, **pppd** closes and reopens the log file, re-reads the filter and key files, then takes the indicated actions:

|                |                                                                                                                                                                                                                                                                                                                                         |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SIGKILL</b> | Don't use this. Never, never use this. Since <b>pppd</b> won't be able to shut down gracefully, it will leave your serial interfaces (whether <code>/dev/tty</code> ) and your IP tunnel driver in some unknown state. Use <b>SIGTERM</b> instead, so <b>pppd</b> will shut down cleanly, and leave the system in a well-defined state. |
| <b>SIGINT</b>  | Disconnect gracefully from an active session. If in 'autocall' mode, reset the call retry delay timer and call retry backoff interval. If <b>up</b> was specified, attempt to re-establish the link. Exit if not in 'autocall' mode.                                                                                                    |
| <b>SIGHUP</b>  | Disconnect abruptly from an active session. If <b>up</b> was specified, attempt to re-establish the link. Exit if not in 'autocall' mode.                                                                                                                                                                                               |
| <b>SIGTERM</b> | Disconnect gracefully from an active session, clean up the state of any serial and IP interfaces that are open, then exit.                                                                                                                                                                                                              |
| <b>SIGUSR1</b> | Increment the verbosity level for debugging information written to the log file.                                                                                                                                                                                                                                                        |

|                |                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------|
| <b>SIGUSR2</b> | Reset the debugging verbosity level to the base value (1 unless <b>debug 0</b> was supplied on the command line). |
| <b>SIGALRM</b> | Take no action except to re-read the filter and key files.                                                        |

## EXAMPLES

To run a pair of daemons on 'oursystem', one maintaining a constant link with 'backbonesystem' and the other prepared to initiate outbound calls to a neighboring machine named 'theirsystem', add the following to `/sbin/rc2.d/S522ppp`:

```
if [-f /etc/ppp/Autostart]; then
 /etc/ppp/Autostart
fi
```

Then make `/etc/ppp/Autostart` look like this:

```
#!/bin/sh

PATH=/usr/etc:/bin:/usr/bin

if [-f /var/adm/pppd.log]; then
 mv /var/adm/pppd.log /var/adm/OLDpppd.log
fi

echo -n "Starting PPP daemons:" >/dev/console

pppd oursystem:backbonesystem auto up
 (echo -n ' backbonesystem') >/dev/console
pppd oursystem:theirsystem auto idle 120
 (echo -n ' theirsystem') >/dev/console

echo '.' >/dev/console
```

To allow a PPP implementation running on 'theirsystem' to dial into 'oursystem', insert the following into `/etc/passwd` on 'oursystem':

```
Pthem?:?:105:20:Their PPP:/etc/ppp:/etc/ppp/Login
```

where group 20 is the gid of the ppp group which owns `/usr/etc/pppd`, and `/etc/ppp/Login` is an executable shell script that looks something like

```
#!/bin/sh
PATH=/usr/bin:/usr/etc:/bin
mesg n
stty -tostop
exec pppd `hostname`:
```

## RECOMMENDATIONS

Use host names when running `/etc/ppp/Autostart` from `/sbin/rc2.d/S522ppp` only if they are known locally. If a PPP connection to a DNS server would be required to resolve a host name, use its literal IP address instead.

## EXTERNAL INFLUENCES

### Environment Variables

The environment variable **PPPHOME**, if present, specifies the directory in which **pppd** looks for its configuration files (**Filter** and **Auth** for all connections, along with **Systems**, **Devices**, and **Dialers** if the connection is 'outbound'). You can specify **PPPHOME** either in the **Autostart** script or in an incoming connection's **Login** script. If **PPPHOME** is not present, **pppd** will expect to find its configuration files in `/etc/ppp/*`.

## SECURITY CONCERNS

**pppd** should be mode 4750, owned by root, and executable only by the members of the group containing all the incoming PPP login 'users'.

**AUTHOR**

**pppd** was developed by the Progressive Systems.

**SEE ALSO**

ppp.Auth(4), ppp.Devices(4), ppp.Dialers(4), ppp.Filter(4), ppp.Keys(4), ppp.Systems(4), RFC 1548, RFC 1549, RFC 1332, RFC 1333, RFC 1334, RFC 1172, RFC 1144, RFC 1055, [ds.internic.net:/internet-drafts/draft-ietf-pppext-compression-04.txt](https://ds.internic.net/internet-drafts/draft-ietf-pppext-compression-04.txt).

**STANDARDS CONFORMANCE**

HP PPP implements the IETF Proposed Standard Point-to-Point Protocol and many of its options and extensions, in conformance with RFCs 1548, 1549, 1332, 1333, 1334, and 1144. It can be configured to be conformant with earlier specifications of the PPP protocol, as described in RFCs 1134, 1171, and 1172. It implements the nonstandard SLIP protocol as described in RFCs 1055 and 1144.

**| p |**

**NAME**

pr - print files

**SYNOPSIS**

**pr** [*options*] [*files*]

**DESCRIPTION**

The **pr** command prints the named files on the standard output. If *file* is -, or if no files are specified, the standard input is assumed. By default, the listing is separated into pages, each headed by the page number, a date and time, and the name of the file.

By default, columns are of equal width, separated by at least one space; lines that do not fit are truncated. If the **-s** option is used, lines are not truncated and columns are separated by the separation character.

If the standard output is associated with a terminal, error messages are withheld until **pr** has completed printing.

**Options**

The following *options* can be used singly or combined in any order:

- +k**        Begin printing with page *k* (default is 1).
- k**        Produce *k*-column output (default is 1). This option should not be used with **-m**. The options **-e** and **-i** are assumed for multi-column output.
- c k**      Produce *k*-column output, same as **-k**.
- a**        Print multi-column output across the page. This option is appropriate only with the **-k** option.
- m**        Merge and print all files simultaneously, one per column (overrides the **-k** and **-a** options).
- d**        Double space the output.
- eck**     Expand *input* tabs to character positions *k*+1, 2×*k*+1, 3×*k*+1, etc. If *k* is 0 or is omitted, default tab settings at every eighth position are assumed. Tab characters in the input are expanded into the appropriate number of spaces. If *c* (any nondigit character) is given, it is treated as the input tab character (default for *c* is the tab character).
- ick**     In *output*, replace white space wherever possible by inserting tabs to character positions *k*+1, 2×*k*+1, 3×*k*+1, etc. If *k* is 0 or is omitted, default tab settings at every eighth position are assumed. If *c* (any nondigit character) is given, it is treated as the output tab character (default for *c* is the tab character).
- nck**     Provide *k*-digit line numbering (default for *k* is 5). The number occupies the first *k*+1 character positions of each column of normal output or each line of **-m** output. If *c* (any nondigit character) is given, it is appended to the line number to separate it from what ever follows (default for *c* is a tab).
- wk**      Set the width of a line to *k* character positions (default is 72 for equal-width, multi-column output; no limit otherwise). Width specifications are only effective for multi-columnar output.
- ok**      Offset each line by *k* character positions (default is 0). The number of character positions per line is the sum of the width and offset.
- lk**      Set the length of a page to *k* lines (default is 66). If *k* is less than what is needed for the page header and trailer, the **-t** option is in effect; that is, header and trailer lines are suppressed in order to make room for text.
- h**        Use the next argument as the header to be printed instead of the file name.
- p**        Pause before beginning each page if the output is directed to a terminal (**pr** rings the bell at the terminal and waits for a **Return**).
- F**        Use form-feed character for new pages (default is to use a sequence of line-feeds). Pause before beginning the first page if the standard output is associated with a terminal.
- f**        Same as **-F**. Provided for backwards compatibility.

| **p** |

- r** Print no diagnostic reports on failure to open files.
- t** Print neither the five-line identifying header nor the five-line trailer normally supplied for each page. Quit printing after the last line of each file without spacing to the end of the page.
- sc** Separate columns by the single character *c* instead of by the appropriate number of spaces (default for *c* is a tab).

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the interpretation of text and the arguments associated with the **-e**, **-i**, **-n**, and **-s** options as single-byte and/or multi-byte characters.

**LC\_TIME** determines the format and contents of date and time strings.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE**, **LC\_TIME**, or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of **C** (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **pr** behaves as if all internationalization variables are set to **C**. See *environ(5)*.

**International Code Set Support**

Single-byte and multi-byte character code sets are supported.

**RETURN VALUE**

The **pr** returns the following values upon completion:

- 0 Successful completion.
- >0 One or more of the input *files* do not exist or cannot be opened.

**EXAMPLES**

Print **file1** and **file2** as a double spaced, three column listing headed by “file list”:

```
pr -3dh "file list" file1 file2
```

Write **file1** on **file2**, expanding tabs to columns 10, 19, 28, 37, ... :

```
pr -e9 -t <file1 >file2
```

Print **file1** in default format with nonblank lines numbered down the left side:

```
nl file1 | pr
```

**FILES**

**/dev/tty**

**SEE ALSO**

**cat(1)**, **lp(1)**, **nl(1)**, **ul(1)**.

**STANDARDS CONFORMANCE**

**pr**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| p |**

**NAME**

praliases - print system-wide sendmail aliases

**SYNOPSIS**

```
praliases [-C file] [-f file] [key ...]
```

**DESCRIPTION**

**praliases** prints out the contents of the alias data base used by **sendmail** to resolve system-wide mail aliases. The alias data base is built with the command **newaliases** or **/usr/sbin/sendmail -bi**. See *sendmail*(1M).

The options are:

**-C file** Read the **sendmail** configuration file specified with this option instead of the default configuration file.

**-f** Accesses the alias database built from *file* with the command

```
newaliases -oA file
```

If this option is not used, **praliases** accesses the database built from the default alias file **/etc/mail/aliases**.

Note that **praliases** accesses the database, not the alias file itself. If the alias file has changed since the alias database was last built, naturally the output of **praliases** cannot match the contents of the alias file.

Each *key* argument, if any, is looked up in the alias database. **praliases** prints out the aliases to which each key expands in the form:

```
key: mailing list
```

where mailing list can be a comma-separated list of addresses to which the key resolves.

**Note**

**praliases** can be used by privileged users only.

**RETURN VALUE**

The **praliases** utility exits with 0 on success, and >0 if an error occurs.

**DIAGNOSTICS**

```
key: No such key
```

*key* was not found in the alias database.

**EXAMPLES**

```
$ praliases root postmaster no_user
root: jan_user
postmaster: joe_user
no_user: No such key
```

The output reveals that **root** is aliased to **jan\_user**, **postmaster** is aliased to **joe\_user**, and that there is no alias for the key **no\_user**.

**WARNINGS**

Because **sendmail** supports NIS aliases, some NIS key-words may appear in the **praliases** output. These key-words, which include **YP\_LAST\_MODIFIED** and **YP\_MASTER\_NAME**, may be safely ignored; they merely indicate that **sendmail** is properly updating the alias database.

**AUTHOR**

**praliases** was developed by the University of California, Berkeley.

**FILES**

|                             |                        |
|-----------------------------|------------------------|
| <b>/etc/mail/aliases</b>    | default alias file     |
| <b>/etc/mail/aliases.db</b> | default alias database |

**SEE ALSO**

*sendmail*(1M).

**NAME**

prealloc - preallocate disk storage

**SYNOPSIS**

**prealloc** *name size*

**DESCRIPTION**

**prealloc** preallocates at least *size* bytes of disk space for an ordinary file *name*, creating the file if *name* does not already exist. The space is allocated in an implementation-dependent fashion for fast sequential reads and writes of the file.

**prealloc** fails and no disk space is allocated if *name* already exists and is not an ordinary file of zero length, if insufficient space is left on disk, or if *size* exceeds the maximum file size or the file size limit of the process (see *ulimit(2)*). The file is zero-filled.

**DIAGNOSTICS**

**prealloc** returns one of the following values upon completion:

- 0** Successful completion.
- 1** *name* already exists and is not an ordinary file of zero length.
- 2** There is insufficient room on the disk.
- 3** *size* exceeds file size limits.

**EXAMPLES**

The following example preallocates 50 000 bytes for the file **myfile**:

```
prealloc myfile 50000
```

**WARNINGS**

Allocation of file space is highly dependent on current disk usage. A successful return does not indicate how fragmented the file actually might be if the disk is approaching its capacity.

**AUTHOR**

**prealloc** was developed by HP.

**SEE ALSO**

prealloc(2), ulimit(2).

**| p |**

**NAME**

printenv - print out the environment

**SYNOPSIS**

**printenv** [*name*]

**DESCRIPTION**

**printenv** prints out the values of the variables in the environment. If a *name* is specified, only its value is printed.

**RETURN VALUE**

If a *name* is specified and it is not defined in the environment, **printenv** returns 1; otherwise it returns zero.

**SEE ALSO**

sh(1), environ(5), csh(1).

| **P** |



**NAME**

printf - format and print arguments

**SYNOPSIS**

```
printf format [arg ...]
```

**DESCRIPTION**

**printf** writes formatted arguments to the standard output. The *arg* arguments are formatted under control of the *format* operand.

*format* is a character string patterned after the formatting conventions of *printf(3S)*, and contains the following types of objects:

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>characters</i>       | Characters that are not <i>escape sequences</i> or <i>conversion specifications</i> (as described below) are copied to standard output.                                                                                                                                                                                                                                                                                                                |
| <i>escape sequences</i> | These are interpreted as non-graphic characters: <ul style="list-style-type: none"> <li>\a alert</li> <li>\b backspace</li> <li>\f form-feed</li> <li>\n new-line</li> <li>\r carriage return</li> <li>\t tab</li> <li>\v vertical tab</li> <li>\' single quote character</li> <li>\\ backslash</li> <li>\n the 8-bit character whose ASCII code is the 1-, 2-, 3-, or 4-digit octal number <i>n</i>, whose first character must be a zero.</li> </ul> |

*conversion specification*

Specifies the output format of each argument ( see below).

Arguments following *format* are interpreted as strings if the corresponding format is either **c** or **s**; otherwise they are treated as constants.

**Conversion Specifications**

Each conversion specification is introduced by the percent character **%**. After the **%** character, the following can appear in the sequence indicated:

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>flags</i>       | Zero or more <i>flags</i> , in any order, which modify the meaning of the conversion specification. The flag characters and their meanings are: <ul style="list-style-type: none"> <li>- The result of the conversion is left-justified within the field.</li> <li>+ The result of a signed conversion always begins with a sign, + or -.</li> <li>&lt;space&gt; If the first character of a signed conversion is not a sign, a space character is prefixed to the result. This means that if the space flag and + flag both appear, the space flag is ignored.</li> <li># The value is to be converted to an "alternate form". For <b>c</b>, <b>d</b>, <b>i</b>, <b>u</b>, and <b>s</b> conversions, this flag has no effect. For <b>o</b> conversion, it increases the precision to force the first digit of the result to be a zero. For <b>x</b> or <b>X</b> conversion, a non-zero result has <b>0x</b> or <b>0X</b> prefixed to it. For <b>e</b>, <b>E</b>, <b>f</b>, <b>g</b>, and <b>G</b> conversions, the result always contains a radix character, even if no digits follow the radix character. For <b>g</b> and <b>G</b> conversions, trailing zeros are not removed from the result, contrary to usual behavior.</li> </ul> |
| <i>field width</i> | An optional string of decimal digits to specify a minimum <i>field width</i> . For an output field, if the converted value has fewer characters than the field width, it is padded on the left (or right, if the left-adjustment flag, - has been given) to the field width.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>precision</i>   | The <i>precision</i> specifies the minimum number of digits to appear for the <b>d</b> , <b>o</b> , <b>i</b> , <b>u</b> , <b>x</b> , or <b>X</b> conversions (the field is padded with leading zeros), the number of digits to appear after the radix character for the <b>e</b> and <b>f</b> conversions, the maximum number of significant digits for the <b>g</b> conversion, or the maximum number of characters to be printed from a string in <b>s</b> conversion. The precision takes the form of a period .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

followed by a decimal digit string. A null digit string is treated as a zero.

*conversion characters*

A *conversion character* indicates the type of conversion to be applied:

- d,i** The integer argument is printed a signed decimal (**d** or **i**), unsigned octal (**o**), unsigned decimal (**u**), or unsigned hexadecimal notation (**x** and **X**). The **x** conversion uses the numbers and letters **0123456789abcdef**, and the **X** conversion uses the numbers and letters **0123456789ABCDEF**. The *precision* component of the argument specifies the minimum number of digits to appear. If the value being converted can be represented in fewer digits than the specified minimum, it is expanded with leading zeroes. The default precision is 1. The result of converting a zero value with a precision of 0 is no characters.
- f** The floating-point number argument is printed in decimal notation in the style **[-]ddd rddd**, where the number of digits after the radix character, **r**, is equal to the *precision* specification. If the *precision* is omitted from the argument, six digits are output; if the *precision* is explicitly 0, no radix appears.
- e,E** The floating-point-number argument is printed in the style **[-]drddd e±dd**, where there is one digit before the radix character, and the number of digits after it is equal to the precision. When the precision is missing, six digits are produced; if the precision is 0, no radix character appears. The **E** conversion character produces a number with **E** introducing the exponent instead of **e**. The exponent always contains at least two digits. However, if the value to be printed requires an exponent greater than two digits, additional exponent digits are printed as necessary.
- g,G** The floating-point-number argument is printed in style **f** or **e** (or int style **E** in the case of a **G** conversion character), with the precision specifying the number of significant digits. The style used depends on the value converted; style **e** is used only if the exponent resulting from the conversion is less than **-h** or greater than or equal to the precision. Trailing zeros are removed from the result. A radix character appears only if it is followed by a digit.
- c** The first character of the argument is printed.
- s** The argument is taken to be a string, and characters from the string are printed until the end of the string or the number of characters indicated by the *precision* specification of the argument is reached. If the precision is omitted from the argument, it is interpreted as infinite and all characters up to the end of the string are printed.
- %** Print a **%** character; no argument is converted.
- b** Similar to the **s** conversion specifier, except that the string can contain backslash-escape sequences which are then converted to the characters they represent. **\c** will cause **printf** to ignore any remaining characters in the string operand containing it, any remaining string operands and any additional characters in the format operand.

In no case does a nonexistent or insufficient field width cause truncation of a field; if the result of a conversion is wider than the field width, the field is simply expanded to contain the conversion result.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_CTYPE** determines the interpretation of *arg* as single and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is

set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, *printf* behaves as if all internationalization variables are set to "C". See *environ(5)*.

### International Code Set Support

Single- and multi-byte character code sets are supported.

### RETURN VALUE

**printf** exits with one of the following values:

- 0** Successful completion;
- >0** Errors occurred. The exit value is increased by one for each error that occurred up to a maximum of 255.

### DIAGNOSTICS

If an argument cannot be converted into a form suitable for the corresponding conversion specification, or for any other reason cannot be correctly printed, a diagnostic message is printed to standard error, the argument is output as a string form as it was given on the command line, and the exit value is incremented.

### EXAMPLES

The following command prints the number 123 in octal, hexadecimal and floating point formats in their alternate form

```
printf "%#o, %#x, %#X, %#f, %#g, %#e\n" 123 123 123 123 123 123
```

resulting in the following output

```
0173, 0x7b, 0X7B, 123.000000, 123.000, 1.230000e+02
```

Print the outputs with their corresponding field widths and precision:

```
printf "%.6d, %10.6d, %.6f, %.6e, %.6s\n" 123 123 1.23 123.4 MoreThanSix
```

resulting in the following output

```
000123, 000123, 1.230000, 1.234000e+02, MoreTh
```

### SEE ALSO

*echo(1)*, *printf(3S)*.

### STANDARDS CONFORMANCE

**printf**: XPG4, POSIX.2

**NAME**

privatepw - Change WU-FTPD Group Access File Information

**SYNOPSIS**

```
/usr/bin/privatepw [-c] [-d accessgroup] [-f ftpgroups] [-g group accessgroup]
[-l] [-v]
```

**DESCRIPTION**

If a site supports the SITE GROUP and SITE GPASS commands, it is necessary to be able to add, delete and list enhanced access group information. The **privatepw** utility is used to update this information in the group access file (**/etc/ftpd/ftpgroups**). This is meant as an administrative tool and not to be run by the general user population. This command requires read/write permission to the appropriate ftpgroups file.

**Options**

In the options defined below, *accessgroup* is an alphanumeric name specified to a group. Please see the *Background* section for details.

- c** Create a new **/etc/ftpd/ftpgroups** file.
- d *accessgroup*** Delete the specified *accessgroup* information from the *ftpgroups* file.
- f *ftpgroups*** Use the specified *ftpgroups* file for all updates. If this option is not specified by default it takes the **/etc/ftpd/ftpgroups** file.
- g *group accessgroup*** Set the real system group (defined in **/etc/group**) to the *accessgroup* specified. This is valid only when adding a new *accessgroup* or changing the password of an existing *accessgroup*.
- l** List the contents of the appropriate ftpgroups file.
- v** This option causes the program to display copyright and version information, then terminate.

**Background**

After user logs in, the SITE GROUP and SITE GPASS commands may be used to specify an enhanced access group and associated password. If the group name and password are valid, the user becomes (via **setegid()**) a member of the group list in the group access file (**/etc/ftpd/ftpgroups**)

The format of the group access file is:

```
access_group_name : encrypted_password : real_group_name
```

where *access\_group\_name* is an arbitrary (alphanumeric + punctuation) string. *encrypted\_password* is the password encrypted via *crypt*(3), exactly like in **/etc/passwd**. *real\_group\_name* is the name of a valid group listed in **/etc/group**.

**AUTHOR**

**privatepw** was developed by the Washington University, St. Louis, Missouri.

**SEE ALSO**

ftpgroups(4), ftpaccess(4).

| P |

**NAME**

`prmail` - print out mail in the incoming mailbox file

**SYNOPSIS**

```
prmail [user ...]
```

**DESCRIPTION**

`prmail` prints the mail which waits for you or the specified user in the incoming mailbox file. The mailbox file is not disturbed.

`prmail` is functionally similar to the command:

```
cat /var/mail/mailfile | more
```

or

```
cat /var/mail/mailfile | pg
```

depending upon the setting of the user's **PAGER** environment variable

**AUTHOR**

`prmail` was developed by the University of California, Berkeley.

**FILES**

`/var/mail/*` incoming mailbox files

**SEE ALSO**

`from(1)`, `mail(1)`.

**NAME**

prof - display profile data

**SYNOPSIS**

**prof** [-tcan] [-ox] [-g] [-z] [-h] [-s] [-m *mdata*] [*prog*]

**DESCRIPTION**

**prof** interprets a profile file produced by **monitor()** (see *monitor(3C)*). The symbol table in the object file *prog* (**a.out** by default) is read and correlated with a profile file (**mon.out** by default). For each external text symbol, the percentage of time spent executing between the address of that symbol and the address of the next is printed, together with the number of times that function was called and the average number of milliseconds per call.

The mutually exclusive options **t**, **c**, **a**, and **n** determine the type of sorting of the output lines:

- t** Sort by decreasing percentage of total time (default).
- c** Sort by decreasing number of calls.
- a** Sort by increasing symbol address.
- n** Sort by symbol name in ascending collation order (see Environment Variables below).

The mutually exclusive options **o** and **x** specify the printing of the address of each symbol monitored:

- o** Print each symbol address (in octal) along with the symbol name.
- x** Print each symbol address (in hexadecimal) along with the symbol name.

The following options can be used in any combination:

- g** Include non-global symbols (static functions).
- z** Include all symbols in the profile range (see *monitor(3C)*), even if associated with zero number of calls and zero time.
- h** Suppress the heading normally printed on the report. (This is useful if the report is to be processed further.)
- s** Print a summary of several of the monitoring parameters and statistics on the standard error output.
- m *mdata*** Use file *mdata* instead of **mon.out** as the input profile file.

A program creates a profile file if it has been loaded using the **cc -p** option (see *cc\_bundled(1)*). This option to the **cc** command arranges for calls to **monitor()** at the beginning and end of execution (see *monitor(3C)*). It is the call to the **monitor** command at the end of execution that causes a profile file to be written. The number of calls to a function is tallied if the **-p** option was used when the file containing the function was compiled.

The name of the file created by a profiled program is controlled by the environment variable **PROFDIR**. If **PROFDIR** is not set, **mon.out** is produced in the directory current when the program terminates. If **PROFDIR=string**, **string/pid.progname** is produced, where *progname* consists of argv[0] with any path prefix removed, and *pid* is the program's process ID. If **PROFDIR** is set to a null string, no profiling output is produced.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_COLLATE** determines the collating order output by the **-n** option.

If **LC\_COLLATE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **prof** behaves as if all internationalization variables are set to "C" (see *environ(5)*).

**WARNINGS**

The times reported in successive identical runs may show variances of 20% or more, because of varying cache-hit ratios due to sharing of the cache with other processes. Even if a program seems to be the only one using the machine, hidden background or asynchronous processes may blur the data. In rare cases, the clock ticks initiating recording of the program counter may "beat" with loops in a program, grossly distorting measurements.

Call counts are always recorded precisely, however.

Only programs that call **exit()** (see *exit(2)*) or return from **main** cause a profile file to be produced, unless a final call to **monitor()** is explicitly coded.

The use of the **cc -p** option to invoke profiling imposes a limit of 600 functions that can have call counters established during program execution. For more counters, call **monitor()** directly. If this limit is exceeded, other data is overwritten and the **mon.out** file is corrupted. The number of call counters used is reported automatically by the **prof** command whenever the number exceeds 5/6 of the maximum.

**FILES**

|                |              |
|----------------|--------------|
| <b>mon.out</b> | for profile  |
| <b>a.out</b>   | for namelist |

**SEE ALSO**

*cc\_bundled(1)*, *exit(2)*, *profil(2)*, *crt0(3)*, *end(3C)*, *monitor(3C)*.

**STANDARDS CONFORMANCE**

**prof**: SVID2, SVID3, XPG2

**NAME**

prs - print and summarize an SCCS file

**SYNOPSIS**

prs [-d *dataspec*] [-r[*SID*]] [-e] [-l] [-c *cutoff*] [-a] *file* ...

**DESCRIPTION**

The **prs** command prints, on the standard output, parts or all of an SCCS file (see *scsfile*(4)) in a user-supplied format. If a directory is named, **prs** behaves as though each file in the directory were specified as a named file, except that non-SCCS files (last component of the path name does not begin with **s.**), and unreadable files are silently ignored. If a name of **-** is given, the standard input is read; each line of the standard input is taken to be the name of an SCCS file or directory to be processed; non-SCCS files and unreadable files are silently ignored. A **--** on the command line implies that all following arguments are file/directory names.

Arguments to **prs**, which can appear in any order, consist of options and file names.

**Options**

All of the described options apply independently to each named file:

- d *dataspec*** Used to specify the output data specification. *dataspec* is a string consisting of SCCS file *data keywords* (see Data Keywords below) interspersed with optional user-supplied text.
- r[*SID*]** Used to specify the *SCCS IDentification (SID)* string of a delta for which information is desired. If no *SID* is specified, the *SID* of the most recently created delta is assumed. If an *SID* is specified, it must agree exactly with an *SID* in the file (that is, the *SID* structure used by **get** does not work here — see *get*(1)).
- e** Requests information for all deltas created *earlier* than and including the delta designated via the **-r** option or the date given by the **-c** option.
- l** Requests information for all deltas created *later* than and including the delta designated via the **-r** option or the date given by the **-c** option.
- c *cutoff*** Cutoff date-time, in the form  
 YY[MM[DD[HH[MM[SS]]]]]  
 Units omitted from the date-time default to their maximum possible values. Thus, **-c7502** is equivalent to **-c750228235959**. One or more non-numeric characters can be used to separate the various 2-digit segments of the cutoff date (for example **-c77/2/2 9:22:25**).  
 For 2-digit year input (YY), the following interpretation is taken: [70-99, 00-69 (1970-1999, 2000-2069)].
- a** Requests printing of information for both removed, i.e., delta type = *R*, (see *rmDEL*(1)) and existing, that is, delta type = *D*, deltas. If the **-a** option is not specified, information is provided for existing deltas only.

If no option letters (or only **-a**) are given, **prs** prints the file name using the default *dataspec* and the **-e** option. This produces information on all deltas.

**Data Keywords**

Data keywords specify which parts of an SCCS file are to be retrieved and output. All parts of an SCCS file (see *scsfile*(4)) have an associated data keyword. There is no limit on the number of times a data keyword can appear in a *dataspec*.

The information printed by **prs** consists of: (1) the user-supplied text; and (2) appropriate values (extracted from the SCCS file) substituted for the recognized data keywords in the order of appearance in the *dataspec*. The format of a data keyword value is either **Simple** (S), in which keyword substitution is direct, or **Multi-line** (M), in which keyword substitution is followed by a carriage return.

User-supplied text is any text other than recognized data keywords. Escapes can be used as follows:



```
backslash \\ form feed \f
backspace \b new-line \n
carriage-return \r single quote \^
colon \: tab \t
```

The default *dataspec* is:

```
" :Dt: \t:DL: \nMRS: \n:MR: COMMENTS : \n:C: "
```

SCCS File Data Keywords

| Keyword | Data Item                       | File Section | Value                       | Fmt |
|---------|---------------------------------|--------------|-----------------------------|-----|
| :Dt:    | Delta information               | Delta Table  | See below*                  | S   |
| :DL:    | Delta line statistics           | "            | :Li: / :Ld: / :Lu:          | S   |
| :Li:    | Lines inserted by Delta         | "            | nnnnn                       | S   |
| :Ld:    | Lines deleted by Delta          | "            | nnnnn                       | S   |
| :Lu:    | Lines unchanged by Delta        | "            | nnnnn                       | S   |
| :DT:    | Delta type                      | "            | D or R                      | S   |
| :I:     | SCCS ID string (SID)            | "            | :R: . . :L: . . :B: . . :S: | S   |
| :R:     | Release number                  | "            | nnnn                        | S   |
| :L:     | Level number                    | "            | nnnn                        | S   |
| :B:     | Branch number                   | "            | nnnn                        | S   |
| :S:     | Sequence number                 | "            | nnnn                        | S   |
| :D:     | Date Delta created              | "            | :Dy: / :Dm: / :Dd:          | S   |
| :Dy:    | Year Delta created              | "            | nn                          | S   |
| :Dm:    | Month Delta created             | "            | nn                          | S   |
| :Dd:    | Day Delta created               | "            | nn                          | S   |
| :T:     | Time Delta created              | "            | :Th: . . :Tm: . . :Ts:      | S   |
| :Th:    | Hour Delta created              | "            | nn                          | S   |
| :Tm:    | Minutes Delta created           | "            | nn                          | S   |
| :Ts:    | Seconds Delta created           | "            | nn                          | S   |
| :P:     | Programmer who created Delta    | "            | logname                     | S   |
| :DS:    | Delta sequence number           | "            | nnnn                        | S   |
| :DP:    | Predecessor Delta seq-no.       | "            | nnnn                        | S   |
| :DI:    | Seq # of deltas incl, excl, ign | "            | :Dn: / :Dx: / :Dg:          | S   |
| :Dn:    | Deltas included (seq #)         | "            | :DS: :DS: . . .             | S   |
| :Dx:    | Deltas excluded (seq #)         | "            | :DS: :DS: . . .             | S   |
| :Dg:    | Deltas ignored (seq #)          | "            | :DS: :DS: . . .             | S   |
| :MR:    | MR numbers for delta            | "            | text                        | M   |
| :C:     | Comments for delta              | "            | text                        | M   |
| :UN:    | User names                      | User name    | text                        | M   |
| :FL:    | Flag list                       | Flags        | text                        | M   |
| :Y:     | Module type flag                | "            | text                        | S   |
| :MF:    | MR validation flag              | "            | yes or no                   | S   |
| :MP:    | MR validation pgm name          | "            | text                        | S   |
| :KF:    | Keyword error/warning flag      | "            | yes or no                   | S   |
| :KV:    | Keyword validation string       | "            | text                        | S   |
| :BF:    | Branch flag                     | "            | yes or no                   | S   |
| :J:     | Joint edit flag                 | "            | yes or no                   | S   |
| :LK:    | Locked releases                 | "            | :R: . . .                   | S   |
| :Q:     | User defined keyword            | "            | text                        | S   |
| :M:     | Module name                     | "            | text                        | S   |
| :FB:    | Floor boundary                  | "            | :R:                         | S   |
| :CB:    | Ceiling boundary                | "            | :R:                         | S   |
| :Ds:    | Default SID                     | "            | :I:                         | S   |
| :ND:    | Null delta flag                 | "            | yes or no                   | S   |
| :FD:    | File descriptive text           | Comments     | text                        | M   |
| :BD:    | Body                            | Body         | text                        | M   |
| :GB:    | Gotten body                     | "            | text                        | M   |

| p |

|             |                                  |     |                          |   |
|-------------|----------------------------------|-----|--------------------------|---|
| <b>:W:</b>  | A form of <i>what</i> (1) string | N/A | <b>:Z::M:\t:I:</b>       | S |
| <b>:A:</b>  | A form of <i>what</i> (1) string | N/A | <b>:Z::Y: :M: :I::Z:</b> | S |
| <b>:Z:</b>  | <i>what</i> (1) string delimiter | N/A | <b>@(#)</b>              | S |
| <b>:F:</b>  | SCCS file name                   | N/A | <i>text</i>              | S |
| <b>:PN:</b> | SCCS file path name              | N/A | <i>text</i>              | S |

\* **:Dt: = :DT: :I: :D: :T: :P: :DS: :DP:**

If no option letters (or only **-a**) are given, **pr<sup>s</sup>** prints the file name, using the default *dataspec*, and the **-e** option; thus, information on all deltas is produced.

**EXTERNAL INFLUENCES**

**Environment Variables**

**LC\_CTYPE** determines the interpretation of *dataspec* as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **pr<sup>s</sup>** behaves as if all internationalization variables are set to "C". See *environ*(5).

**International Code Set Support**

Single-byte and multi-byte character code sets are supported.

**DIAGNOSTICS**

Use *sc<sup>s</sup>help*(1) for explanations.

**EXAMPLES**

The entry

```
prs -d"Users and/or user IDs for :F: are :\n:UN:" s.file
```

may produce on the standard output:

```
Users and/or user IDs for s.file are:
xyz
131
abc
```

The entry

```
prs -d"Newest delta for pgm :M:: :I: Created :D: By :P:" -r s.file
```

may produce on the standard output:

```
Newest delta for pgm main.c: 3.7 Created 77/12/1 By cas
```

As a *special case* (when no specifications for selecting or printing are given)

```
prs s.file
```

may produce on the standard output:

```
D 1.1 77/12/1 00:00:00 cas 1 000000/00000/00000
MRs:
b178-12345
b179-54321
COMMENTS:
this is the comment line for s.file initial delta
```

for each delta table entry of the "D" type. The only option argument allowed to be used with the *special case* is the **-a** option.

**FILES**

/tmp/pr?????

**SEE ALSO**

admin(1), delta(1), get(1), sc<sup>s</sup>help(1), sc<sup>s</sup>file(4).

**prs(1)**

**prs(1)**

**STANDARDS CONFORMANCE**

**prs:** SVID2, SVID3, XPG2, XPG3, XPG4

**| p |**

**NAME**

ps - report process status

**SYNOPSIS**

**ps** [-a**deflPzx**] [-g *grplist*] [-p *proclist*] [-R *prmgrplist*] [-t *termlist*] [-u *uidlist*]  
[-Z *pset\_list*]

**XPG4 Synopsis**

**ps** [-a**AcdefHjlpzx**] [-C *cmdlist*] [-g *grplist*] [-G *gidlist*] [-n *namelist*] [-o *format*]  
[-p *proclist*] [-R *prmgrplist*] [-s *sidlist*] [-t *termlist*] [-u *uidlist*] [-U *uidlist*] [-Z *pset\_list*]

**DESCRIPTION**

**ps** prints information about selected processes. Use options to specify which processes to select and what information to print about them.

**Process Selection Options**

Use the following options to choose which processes should be selected.

NOTE: If an option is used in both the default (standard HP-UX) and XPG4 environments, the description provided here documents the default behavior. Refer to the **UNIX95** variable under **EXTERNAL INFLUENCES** for additional information on XPG4 behavior.

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (none)                      | Select those processes associated with the current terminal.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>-A</b>                   | (XPG4 Only.) Select all processes. (Synonym for <b>-e</b> .)                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>-a</b>                   | Select all processes except process group leaders and processes not associated with a terminal.                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>-C</b> <i>cmdlist</i>    | (XPG4 Only.) Select processes executing a command with a basename given in <i>cmdlist</i> .                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>-d</b>                   | Select all processes except process group leaders.                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>-e</b>                   | Select all processes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>-g</b> <i>grplist</i>    | Select processes whose process group leaders are given in <i>grplist</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-G</b> <i>gidlist</i>    | (XPG4 Only.) Select processes whose real group ID numbers or group names are given in <i>gidlist</i> .                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>-n</b> <i>namelist</i>   | (XPG4 Only.) This option is ignored; its presence is allowed for standards compliance.                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>-p</b> <i>proclist</i>   | Select processes whose process ID numbers are given in <i>proclist</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>-R</b> <i>prmgrplist</i> | Select processes belonging to PRM process resource groups whose names or ID numbers are given in <i>prmgrplist</i> . See <b>DEPENDENCIES</b> .                                                                                                                                                                                                                                                                                                                                                                 |
| <b>-s</b> <i>sidlist</i>    | (XPG4 Only.) Select processes whose session leaders are given in <i>sidlist</i> . (Synonym for <b>-g</b> .)                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>-t</b> <i>termlist</i>   | Select processes associated with the terminals given in <i>termlist</i> . Terminal identifiers can be specified in one of two forms: the device's file name (such as <b>tty04</b> ) or if the device's file name starts with <b>tty</b> , just the rest of it (such as <b>04</b> ). If the device's file is in a directory other than <b>/dev</b> or <b>/dev/pty</b> , the terminal identifier must include the name of the directory under <b>/dev</b> that contains the device file (such as <b>pts/5</b> ). |
| <b>-u</b> <i>uidlist</i>    | Select processes whose effective user ID numbers or login names are given in <i>uidlist</i> .                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>-U</b> <i>uidlist</i>    | (XPG4 Only.) Select processes whose real user ID numbers or login names are given in <i>uidlist</i> .                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>-Z</b> <i>pset_list</i>  | Select processes whose processor set ID's are given in <i>pset_list</i> . This option is supported only if the kernel supports processor sets functionality.                                                                                                                                                                                                                                                                                                                                                   |

If any of the **-a**, **-A**, **-d**, or **-e** options is specified, the **-C**, **-g**, **-G**, **-p**, **-R**, **-t**, **-u**, **-Z**, and **-U** options are ignored.

If more than one of **-a**, **-A**, **-d**, and **-e** are specified, the least restrictive option takes effect.

If more than one of the **-C**, **-g**, **-G**, **-p**, **-R**, **-t**, **-u**, **-Z**, and **-U** options are specified, processes will be selected if they match any of the options specified.

The lists used as arguments to the **-C**, **-g**, **-G**, **-p**, **-R**, **-t**, **-u**, **-Z**, and **-U** options can be specified in one of two forms:

- A list of identifiers separated from one another by a comma.
- A list of identifiers enclosed in quotation marks (") and separated from one another by a comma and/or one or more spaces.

### Output Format Options

Use the following options to control which columns of data are included in the output listing. The options are cumulative.

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (none)           | The default columns are: <b>pid</b> , <b>tty</b> , <b>time</b> , and <b>comm</b> , in that order.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-f</b>        | Show columns <b>user</b> , <b>pid</b> , <b>ppid</b> , <b>cpu</b> , <b>stime</b> , <b>tty</b> , <b>time</b> , and <b>args</b> , in that order.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>-l</b>        | Show columns <b>flags</b> , <b>state</b> , <b>uid</b> , <b>pid</b> , <b>ppid</b> , <b>cpu</b> , <b>intpri</b> , <b>nice</b> , <b>addr</b> , <b>sz</b> , <b>wchan</b> , <b>tty</b> , <b>time</b> , and <b>comm</b> , in that order.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-fl</b>       | Show columns <b>flags</b> , <b>state</b> , <b>user</b> , <b>pid</b> , <b>ppid</b> , <b>cpu</b> , <b>intpri</b> , <b>nice</b> , <b>addr</b> , <b>sz</b> , <b>wchan</b> , <b>stime</b> , <b>tty</b> , <b>time</b> , and <b>args</b> , in that order.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-c</b>        | (XPG4 Only.) Remove columns <b>cpu</b> and <b>nice</b> ; replace column <b>intpri</b> with columns <b>cls</b> and <b>pri</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>-j</b>        | (XPG4 Only.) Add columns <b>pgid</b> and <b>sid</b> after column <b>ppid</b> (or <b>pid</b> , if <b>ppid</b> is not being displayed).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>-z</b>        | Add column <b>pset</b> before column <b>prmgrp</b> / <b>prmid</b> . If <b>prmgrp</b> and <b>prmid</b> are not present, add column <b>pset</b> before column <b>pid</b> . (Note that <i>ps</i> (1) displays the kernel processor set id, where all kernel daemons run, as <b>KERN</b> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|                  | The <b>-z</b> option is supported only if the kernel supports processor sets functionality.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>-P</b>        | Add column <b>prmid</b> (for <b>-l</b> ) or <b>prmgrp</b> (for <b>-f</b> or <b>-fl</b> ) immediately before column <b>pid</b> . See DEPENDENCIES.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-o format</b> | (XPG4 Only.) <i>format</i> is a comma- or space-separated list of the columns to display, in the order they should be displayed. (Valid column names are listed below.) A column name can optionally be followed by an equals sign (=) and a string to use as the heading for that column. (Any commas or spaces after the equals sign will be taken as a part of the column heading; if more columns are desired, they must be specified with additional <b>-o</b> options.) The width of the column will be the greater of the width of the data to be displayed and the width of the column heading. If an empty column heading is specified for every heading, no heading line will be printed. This option overrides options <b>-c</b> , <b>-f</b> , <b>-j</b> , <b>-l</b> , <b>-z</b> , and <b>-P</b> ; if they are specified, they are ignored. |
| <b>-H</b>        | (XPG4 Only.) Shows the process hierarchy. Each process is displayed under its parent, and the contents of the <b>args</b> or <b>comm</b> column for that process is indented from that of its parent. Note that this option is expensive in both memory and speed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-x</b>        | Shows the command line in extended format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

The column names and their meanings are given below. Except where noted, the default heading for each column is the uppercase form of the column name.

|             |                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>addr</b> | The memory address of the process, if resident; otherwise, the disk address.                                                                                                                                                                                                                                                                                                                          |
| <b>args</b> | The command line given when the process was created. This column should be the last one specified, if it is desired. Only a subset of the command line is saved by the kernel; as much of the command line will be displayed as is available. The output in this column may contain spaces. The default heading for this column is <b>COMMAND</b> if <b>-o</b> is specified and <b>CMD</b> otherwise. |

|               |                                                                                                                                                                                                                                                                                                                                                          |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>cls</b>    | Process scheduling class, see <i>rtsched(1)</i> .                                                                                                                                                                                                                                                                                                        |
| <b>comm</b>   | The command name. The output in this column may contain spaces. The default heading for this column is <b>COMMAND</b> if <b>-o</b> is specified and <b>CMD</b> otherwise.                                                                                                                                                                                |
| <b>cpu</b>    | Processor utilization for scheduling. The default heading for this column is <b>C</b> .                                                                                                                                                                                                                                                                  |
| <b>etime</b>  | Elapsed time of the process. The default heading for this column is <b>ELAPSED</b> .                                                                                                                                                                                                                                                                     |
| <b>flags</b>  | Flags (octal and additive) associated with the process: <ul style="list-style-type: none"> <li>0 Swapped</li> <li>1 In core</li> <li>2 System process</li> <li>4 Locked in core (e.g., for physical I/O)</li> <li>10 Being traced by another process</li> <li>20 Another tracing flag</li> </ul> <p>The default heading for this column is <b>F</b>.</p> |
| <b>intpri</b> | The priority of the process as it is stored internally by the kernel. This column is provided for backward compatibility and its use is not encouraged.                                                                                                                                                                                                  |
| <b>gid</b>    | The group ID number of the effective process owner.                                                                                                                                                                                                                                                                                                      |
| <b>group</b>  | The group name of the effective process owner.                                                                                                                                                                                                                                                                                                           |
| <b>nice</b>   | Nice value; used in priority computation (see <i>nice(1)</i> ). The default heading for this column is <b>NI</b> .                                                                                                                                                                                                                                       |
| <b>pcpu</b>   | The percentage of CPU time used by this process during the last scheduling interval. The default heading for this column is <b>%CPU</b> .                                                                                                                                                                                                                |
| <b>pgid</b>   | The process group ID number of the process group to which this process belongs.                                                                                                                                                                                                                                                                          |
| <b>pid</b>    | The process ID number of the process.                                                                                                                                                                                                                                                                                                                    |
| <b>ppid</b>   | The process ID number of the parent process.                                                                                                                                                                                                                                                                                                             |
| <b>pri</b>    | The priority of the process. The meaning of the value depends on the process scheduling class; see <b>cls</b> , above, and <i>rtsched(1)</i> .                                                                                                                                                                                                           |
| <b>prmid</b>  | The PRM process resource group ID number.                                                                                                                                                                                                                                                                                                                |
| <b>prmgrp</b> | The PRM process resource group name.                                                                                                                                                                                                                                                                                                                     |
| <b>rgid</b>   | The group ID number of the real process owner.                                                                                                                                                                                                                                                                                                           |
| <b>rgroup</b> | The group name of the real process owner.                                                                                                                                                                                                                                                                                                                |
| <b>ruid</b>   | The user ID number of the real process owner.                                                                                                                                                                                                                                                                                                            |
| <b>pset</b>   | The processor set ID on which this process is running.                                                                                                                                                                                                                                                                                                   |
| <b>ruser</b>  | The login name of the real process owner.                                                                                                                                                                                                                                                                                                                |
| <b>sid</b>    | The session ID number of the session to which this process belongs.                                                                                                                                                                                                                                                                                      |
| <b>state</b>  | The state of the process: <ul style="list-style-type: none"> <li>0 Nonexistent</li> <li>S Sleeping</li> <li>W Waiting</li> <li>R Running</li> <li>I Intermediate</li> <li>Z Terminated</li> <li>T Stopped</li> <li>X Growing</li> </ul> <p>The default heading for this column is <b>S</b>.</p>                                                          |
| <b>stime</b>  | Starting time of the process. If the elapsed time is greater than 24 hours, the starting date is displayed instead.                                                                                                                                                                                                                                      |
| <b>sz</b>     | The size in physical pages of the core image of the process, including text, data, and stack space. Physical page size is defined by <b>_SC_PAGE_SIZE</b> in the header file <b>&lt;unistd.h&gt;</b> (see <i>sysconf(2)</i> and <i>unistd(5)</i> ).                                                                                                      |

| P |

|              |                                                                                                                                                |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>time</b>  | The cumulative execution time for the process.                                                                                                 |
| <b>tty</b>   | The controlling terminal for the process. The default heading for this column is <b>TT</b> if <b>-o</b> is specified and <b>TTY</b> otherwise. |
| <b>uid</b>   | The user ID number of the effective process owner.                                                                                             |
| <b>user</b>  | The login name of the effective process owner.                                                                                                 |
| <b>vsz</b>   | The size in kilobytes (1024 byte units) of the core image of the process. See column <b>sz</b> , above.                                        |
| <b>wchan</b> | The event for which the process is waiting or sleeping; if there is none, a hyphen (-) is displayed.                                           |

### Notes

**ps** prints the command name and arguments given at the time of the process was created. If the process changes its arguments while running (by writing to its *argv* array), these changes are not displayed by **ps**.

A process that has exited and has a parent, but has not yet been waited for by the parent, is marked **<defunct>** (see **zombie process** in *exit(2)*).

The time printed in the **stime** column, and used in computing the value for the **etime** column, is the time when the process was forked, *not* the time when it was modified by **exec()**.

To make the **ps** output safer to display and easier to read, all control characters in the **comm** and **args** columns are displayed as "visible" equivalents in the customary control character format,  $\hat{x}$ .

### EXTERNAL INFLUENCES

#### Environment Variables

**UNIX95** specifies to use the XPG4 behavior for this command. The changes for XPG4 include support for the entire option set specified above and include the following behavioral changes:

- The **TIME** column format changes from *mmm:ss* to *[dd-]hh:mm:ss*.
- When the **comm**, **args**, **user**, and **prmgrp** fields are included by default or the **-f** or **-l** flags are used, the column headings of those fields change to **CMD**, **CMD**, **USER**, and **PRMGRP**, respectively.
- **-a**, **-d**, and **-g** will select processes based on session rather than on process group.
- The uid or user column displayed by **-f** or **-l** will display effective user rather than real user.
- The **-u** option will select users based on effective UID rather than real UID.
- The **-C** and **-H** options, while they are not part of the XPG4 standard, are enabled.

**LC\_TIME** determines the format and contents of date and time strings. If it is not specified or is null, it defaults to the value of **LANG**.

If **LANG** is not specified or is null, it defaults to **C** (see *lang(5)*).

If any internationalization variable contains an invalid setting, all internationalization variables default to **C** (see *environ(5)*).

#### International Code Set Support

Single-byte character code sets are supported.

### EXAMPLES

Generate a full listing of all processes currently running on your machine:

```
ps -ef
```

To see if a certain process exists on the machine, such as the **cron** clock daemon, check the far right column for the command name, **cron**, or try

```
ps -f -C cron
```

### WARNINGS

Things can change while **ps** is running; the picture it gives is only a snapshot in time. Some data printed for defunct processes is irrelevant.

If two special files for terminals are located at the same select code, that terminal may be reported with either name. The user can select processes with that terminal using either name.

Users of **ps** must not rely on the exact field widths and spacing of its output, as these will vary depending on the system, the release of HP-UX, and the data to be displayed.

When non-standard options are mixed with standard options then the behavior may be non-standard.

## DEPENDENCIES

### HP Process Resource Manager

The **-P** and **-R** options require the optional HP Process Resource Manager (PRM) software to be installed and configured. See *prmconfig*(1) for a description of how to configure HP PRM, and *prmconf*(4) for the definition of "process resource group."

If HP PRM is not installed and configured and **-P** or **-R** is specified, a warning message is displayed and (for **-P**) hyphens (-) are displayed in the **prmid** and **prmgrp** columns.

## FILES

|                         |                                    |
|-------------------------|------------------------------------|
| <b>/dev</b>             | Directory of terminal device files |
| <b>/etc/passwd</b>      | User ID information                |
| <b>/var/adm/ps_data</b> | Internal data structure            |

## SEE ALSO

kill(1), nice(1), acctcom(1M), exec(2), exit(2), fork(2), sysconf(2), unistd(5).

HP Process Resource Manager: *prmconfig*(1), *prmconf*(4) in *HP Process Resource Manager User's Guide*.

## STANDARDS COMPLIANCE

**ps**: SVID2, XPG2, XPG3, XPG4

| **P** |



**NAME**

ptx - permuted index

**SYNOPSIS**

**ptx** [*options*] [*input*] [*output*]

**DESCRIPTION**

**ptx** generates the file *output* that can be processed with a text formatter to produce a permuted index of file *input* (standard input and output default). It has three phases: the first does the permutation, generating one line for each keyword in an input line. The keyword is rotated to the front. The permuted file is then sorted (see *sort*(1) and Environment Variables below). Finally, the sorted lines are rotated so the keyword comes at the middle of each line. **ptx** output is in the form:

```
.xx "tail" "before keyword" "keyword and after" "head"
```

where **.xx** is assumed to be an **nroff** or **troff** macro provided by the user, or provided by the **mptx** macro package (see NOTES below). The *before keyword* and *keyword and after* fields incorporate as much of the line as will fit around the keyword when it is printed. *tail* and *head*, at least one of which is always the empty string, are wrapped-around pieces small enough to fit in the unused space at the opposite end of the line.

The following *options* can be applied:

- f**           Fold uppercase and lowercase letters for sorting.
- t**           Prepare the output for the phototypesetter by using a line length of 100.
- w n**         Use the next argument, *n*, as the length of the output line. The default line length is 72 characters for **nroff** and 100 for **troff**.
- g n**         Use the next argument, *n*, as the number of characters that **ptx** will reserve in its calculations for each gap among the four parts of the line as finally printed. The default gap is 3.
- o only**      Use as keywords only the words given in the *only* file.
- i ignore**    Do not use as keywords any words given in the *ignore* file. If the **-i** and **-o** options are missing, use `/usr/lib/eign` as the *ignore* file.
- b break**     Use the characters in the *break* file to separate words. Tab, new-line, and space characters are *always* used as break characters. Punctuation characters are treated as part of the word in the absence of this option.
- r**           Take any leading non-blank characters of each input line to be a reference identifier (as to a page or chapter), separate from the text of the line. Attach that identifier as a 5th field on each output line.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_COLLATE** determines the order in which the output is sorted.

**LC\_CTYPE** determines the default break characters.

If **LC\_COLLATE** or **LC\_CTYPE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **ptx** behaves as if all internationalization variables are set to "C" (see *environ*(5)).

**International Code Set Support**

Single-byte character code sets are supported.

**WARNINGS**

Line length counts do not account for overstriking or proportional spacing.

Lines containing tildes (~) are botched because **ptx** uses that character internally.

**FILES**

`/usr/lib/eign`  
`/usr/bin/sort`

`/usr/share/lib/tmac/tmac.ptx`

**NOTES**

The **mptx** macro package is not provided as part of the HP-UX operating system. It is part of the Documenters Work Bench (DWB) software package originally developed by AT&T which has been ported to HP 9000 systems by various third-party software suppliers including Elan Computer Group, Inc. of Mountain View California and others.

Permuted indexes produced by using **ptx** usually have a 4-column format that some users prefer and others dislike greatly. The two-column format index provided in this manual is created by processing index entries that are hidden as comments at the end of each manual entry file.

**SEE ALSO**

nroff(1), mm(5).

| **P** |

**NAME**

`pwd` - working directory name

**SYNOPSIS**

`pwd`

**DESCRIPTION**

`pwd` prints the path name of the working (current) directory.

**EXTERNAL INFLUENCES****Environment Variables**

`LC_MESSAGES` determines the language in which messages are displayed.

If `LC_MESSAGES` is not specified in the environment or is set to the empty string, the value of `LANG` is used as a default for each unspecified or empty variable. If `LANG` is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of `LANG`.

If any internationalization variable contains an invalid setting, `pwd` behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**DIAGNOSTICS**

**Cannot open ..**

**Read error in ..**

Possible file system trouble; contact system administrator.

**pwd: cannot access parent directories**

Current directory has been removed (usually by a different process). Use `cd` command to move to a valid directory (see *cd(1)*).

**EXAMPLES**

This command lists the path of the current working directory. If your home directory were `/mnt/staff` and the command `cd camp/nevada` were executed from the home directory, typing `pwd` would produce the following:

```
/mnt/staff/camp/nevada
```

**| p |****AUTHOR**

`pwd` was developed by AT&T and HP.

**SEE ALSO**

*cd(1)*, *csh(1)*, *sh-posix(1)*, *sh(1)*.

**STANDARDS CONFORMANCE**

`pwd`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**NAME**

pwget, grget - get password and group information

**SYNOPSIS**

**pwget** [-n *name* | -u *uid* ]

**grget** [-n *name* | -g *gid* ]

**DESCRIPTION**

**pwget** and **grget** locate and display information from `/etc/passwd` and `/etc/group`.

The standard output of **pwget** contains lines of colon-separated password information whose format is the same as that used in the `/etc/passwd` file (see `passwd(4)`).

The standard output of **grget** contains lines of colon-separated group information whose format is the same as that used in the `/etc/group` file (see `group(4)`).

With no options, **pwget** and **grget** get all entries with `getpwent()` or `getgrent()` respectively, (see `getpwent(3C)` and `getgrent(3C)`), and output a line for each entry found.

**Options**

When an option is given, only a single entry is printed.

The options for **pwget** are:

-n *name*        Output the first entry that matches *name* using `getpwnam()` (see `getpwent(3C)`).

-u *uid*         Output the first entry that matches *uid* using `getpwuid()` (see `getpwent(3C)`).

The options for **grget** are:

-n *name*        Output the first entry that matches *name* using `getgrnam()` (see `getgrent(3C)`).

-g *gid*         Output the first entry that matches *gid* using `getgrgid()` (see `getgrent(3C)`).

**NETWORKING FEATURES****NFS**

If Network Information System (NIS) is in use, these commands provide password and group information based on the NIS version of the password and group databases in addition to the local password and group files.

**RETURN VALUE**

These commands return `0` upon success, `1` when a specific search fails, and `2` upon error.

**DEPENDENCIES****NFS:**

WARNING: If the Network Information System network database is in use and the NIS client daemon (`yplibind`) is not bound to a NIS server daemon (see `ypserv(1M)`), these utilities will wait until such a binding is established. These commands can be terminated in this condition by sending a `SIGINT` signal to the process (see `kill(1)`).

See `ypmatch(1)`, and `ypserv(1M)`.

**AUTHOR**

**pwget** and **grget** were developed by HP.

**FILES**

`/etc/group`        local group data file  
`/etc/passwd`      local password data file

**SEE ALSO**

`getgrent(3C)`, `getpwent(3C)`, `group(4)`, `passwd(4)`.

**NAME**

quota - display disk usage and limits

**SYNOPSIS**

**quota** [-v] [*user* ... ]

**DESCRIPTION**

The **quota** command displays the disk usage and limits for one or more *users*. Without the **-v** option, it displays information only when the usage exceeds the limits.

*user* is a user name or a numeric UID. The default is the login user name.

Only users with appropriate privileges can view the limits of other users.

**Options**

The **quota** command recognizes the following option:

**-v**                    Display the statistics whether they exceed limits or not. Note that no usage statistics exist if no quota is set.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **quota** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**AUTHOR**

Disk Quotas were developed by the University of California, Berkeley, Sun Microsystems, Inc., and HP.

**FILES**

*directory* /**quotas**    Quota statistics static storage for a file system, where *directory* is the root of the file system as specified to the **mount** command (see *mount(1M)*).

/etc/**mnttab**        List of currently mounted file systems

| q |

**SEE ALSO**

quota(5).

**NAME**

ranlib - regenerate archive symbol table

**SYNOPSIS**

**ranlib** *archive* ...

**DESCRIPTION**

**ranlib** regenerates the symbol tables of the specified archives. It is equivalent to executing **ar qs archive** on each of the archives. After using the **z** modifier of **ar**, the symbol table of an archive must be regenerated before it can be used.

**EXTERNAL INFLUENCES****Environment Variables**

The following internationalization variables affect the execution of **ranlib**:

**LANG**

Determines the locale category for native language, local customs and coded character set in the absence of **LC\_ALL** and other **LC\_\*** environment variables. If **LANG** is not specified or is set to the empty string, a default of **C** (see *lang*(5)) is used instead of **LANG**.

**LC\_ALL**

Determines the values for all locale categories and has precedence over **LANG** and other **LC\_\*** environment variables.

**LC\_CTYPE**

Determines the locale category for character handling functions.

**LC\_MESSAGES**

Determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

**LC\_NUMERIC**

Determines the locale category for numeric formatting.

**LC\_TIME**

Determines the format and contents of date and time formatting.

**NLSPATH**

Determines the location of message catalogues for the processing of **LC\_MESSAGES**.

If any internationalization variable contains an invalid setting, **ranlib** behaves as if all internationalization variables are set to **C**. See *environ*(5).

In addition, the following environment variable affects **ranlib**:

**TMPDIR**

Specifies a directory for temporary files (see *tmpnam*(3S)).

**SEE ALSO****System Tools:**

*ar*(1) create archived libraries

**Miscellaneous:**

*ar*(4) archive format  
*strip*(1) strip symbol and line number information from an object file

**NAME**

r`cp` - remote file copy

**SYNOPSIS****Copy Single File**

r`cp` [-**p**] [-**S** *size*] [-**R** *size*] *source\_file1* *dest\_file*

**Copy Multiple Files**

r`cp` [-**p**] [-**S** *size*] [-**R** *size*] *source\_file1* [*source\_file2*]... *dest\_dir*

**Copy One or More Directory Subtrees**

r`cp` [-**p**] [-**S** *size*] [-**R** *size*] -**r** *source\_dir1* [*source\_dir2*]... *dest\_dir*

**Copy Files and Directory Subtrees**

r`cp` [-**p**] [-**S** *size*] [-**R** *size*] -**r** *file\_or\_dir1* [*file\_or\_dir2*]... *dest\_dir*

**In Kerberos V5 Network Authentication Environments****Copy Single File**

r`cp` [-**k** *realm*] [-**P**] [-**p**] [-**S** *size*] [-**R** *size*] *source\_file1* *dest\_file*

**Copy Multiple Files**

r`cp` [-**k** *realm*] [-**P**] [-**p**] [-**S** *size*] [-**R** *size*] *source\_file1* [*source\_file2*]... *dest\_dir*

**Copy One or More Directory Subtrees**

r`cp` [-**k** *realm*] [-**P**] [-**p**] [-**S** *size*] [-**R** *size*] -**r** *source\_dir1* [*source\_dir2*]... *dest\_dir*

**Copy Files and Directory Subtrees**

r`cp` [-**k** *realm*] [-**P**] [-**p**] [-**S** *size*] [-**R** *size*] -**r** *file\_or\_dir1* [*file\_or\_dir2*]... *dest\_dir*

**DESCRIPTION**

The **r`cp`** command copies files, directory subtrees, or a combination of files and directory subtrees from one or more systems to another. In many respects, it is similar to the **cp** command (see `cp(1)`).

To use **r`cp`**, you must have read access to files being copied, and read and search (execute) permission on all directories in the directory path. Note that there are special requirements for third-party transfers, which are described in the **Third-Party Transfers** section below.

In a Kerberos V5 Network Authentication environment, **r`cp`** uses the Kerberos V5 protocol while initiating the connection to a remote host. The authorization mechanism is dependent on the command line options used to invoke **remshd** on the remote host (i.e., **-K**, **-R**, **-r**, or **-k**). Kerberos authentication and authorization rules are described in the Secure Internet Services man page, `sis(5)`.

Although Kerberos authentication and authorizations may apply, the Kerberos mechanism is **not** applied when copying files. The files are still transferred in clear text over the network.

The **fallback** option can be set in the `krb5.conf` file within *appdefaults Section*. Refer to the `krb5.conf(4)` manpage for more information on the *appdefaults Section*. If **fallback** is set to true and the kerberos authentication fails, **r`cp`** will use the non-secure mode of authentication.

**Note:** Command line options override the configuration file options.

**Options and Arguments**

**r`cp`** recognizes the following options and arguments:

*source\_file*, *source\_dir*

This option specifies the name of an existing file or directory on a local or remote machine that you want to be copied to a specified destination. The source file and directory names are constructed as follows:

*user\_name*@*hostname* :*pathname* /*filename*

or

*user\_name*@*hostname* :*pathname* /*dirname*

Component parts of file and directory names are described below. If multiple existing files and/or directory subtrees (*source\_file1*, *source\_file2*, ..., etc.) are specified, then the destination must be a directory. Shell file name expansion is allowed on

- both local and remote systems. Multiple files and directory subtrees can be copied from one or more systems to a single destination directory by using a single command.
- dest\_file* This option specifies the name of the destination file. If host name and path name are not specified, then the existing file is copied into a file named *dest\_file* in the current directory on the local system. If *dest\_file* already exists and is writable, then the existing file is overwritten. The destination file names are constructed in the same way as source files except that the usage of file name expansion characters is forbidden in the case of destination file names.
- dest\_dir* This option specifies the name of the destination directory. If host name and path names are not specified, then the existing file is copied into a directory named *dest\_dir* in the current directory on the local system. If *dest\_dir* already exists in the specified directory path (or current directory if not specified), then a new directory named *dest\_dir* is created underneath the existing directory named *dest\_dir*. The destination directory names are constructed the in same way as source directory tree names except that the usage of file name expansion characters is forbidden in the case of destination directory names.
- If the *source\_dir* has more than one file to be copied, the *dest\_dir* does not exist, and if the `-r` option is used for recursive copying, then `rcp` first creates the *dest\_dir* and later copies the files under the *source\_dir* to the *dest\_dir*.
- file\_or\_dir* If a combination of files and directories are specified for copying (either explicitly or by file name expansion), then only files are copied unless the `-r` option is specified. If the `-r` option is present, then all the files and directory subtrees whose names match the specified *file\_or\_dir* name are copied.
- `-k realm` This option is applicable only in a secure environment based on Kerberos V5. It can be used to obtain tickets from the remote host in the specified *realm* instead of the remote host's default realm as specified in the configuration file `krb.realms`.
- `-P` This option is applicable only in a secure environment based on Kerberos V5. It disables Kerberos authentication. If the remote host has been configured to prevent non-secure access, using this option would result in the generic error,
- `krcmd: connect: hostname: Connection refused`
- See *DIAGNOSTICS* in *remshd*(1M) for more details.
- `-p` This option can be used to preserve (duplicate) modification times and modes (permissions) of source files, ignoring the current setting of the `umask` file creation mode mask. If this option is specified, `rcp` preserves the sticky bit only if the target user is superuser.
- If the `-p` option is not specified, `rcp` preserves the mode and owner of *dest\_file* if it already exists; otherwise `rcp` uses the mode of the source file modified by the `umask` on the destination host. Modification and access times of the destination file are set to the time when the copy was made.
- `-s size` This option sets the size of the socket send buffer.
- `-R size` This option sets the size of the socket receive buffer.
- `-r` This option can be used to recursively copy directory subtrees rooted at the source directory name. If any directory subtrees are to be copied, `rcp` recursively copies each subtree rooted at the specified source directory name to directory *dest\_dir*. If *source\_dir* is being copied to an existing directory of the same name, `rcp` creates a new directory *source\_dir* within *dest\_dir* and copies the subtree rooted at *source\_dir* to *dest\_dir* / *source\_dir*. If *dest\_dir* does not exist, `rcp` first creates it and copies the subtree rooted at *source\_dir* to *dest\_dir* and the output will be similar irrespective of whether a wildcard character (*source\_dir*/\*) is used for copying or otherwise.

### Constructing File and Directory Names

As indicated above, file and directory names contain one, two, or four component parts:

*user\_name* Login name to be used for accessing directories and files on remote system.



|                 |                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------|
| <i>hostname</i> | Hostname of remote system where directories and files are located.                                                |
| <i>pathname</i> | Absolute directory path name or directory path name relative to the login directory of user <i>user_name</i> .    |
| <i>filename</i> | Actual name of source or destination file. File name expansion is allowed on source file names.                   |
| <i>dirname</i>  | Actual name of source or destination directory subtree. File name expansion is allowed on source directory names. |

Each *file* or *directory* argument is either a remote file name of the form *hostname:path*, or a local file name (with a slash (/) before any colon (:)). *hostname* can be either an official host name or an alias (see *hosts*(4)). If *hostname* is of the form *ruser@rhost*, *ruser* is used on the remote host instead of the current user name. An unspecified *path* (that is, *hostname:~*) refers to the remote user's login directory. If *path* does not begin with /, it is interpreted relative to the remote user's login directory on *hostname*. Shell metacharacters in remote *paths* can be quoted with backslash (\), single quotes (' '), or double quotes (" "), so that they will be interpreted remotely.

**rcp** does not prompt for passwords. In a non-secure or traditional environment, user authorization is checked by determining if the current local user name or any user name specified via *ruser* exists on *rhost*. In a Kerberos V5 Network Authentication or secure environment, the authorization method is dependent upon the command line options for **remshd** (see *remshd*(1M) for details). In either case, remote command execution via *remsh*(1) and *remd*(3N), or *rcmd\_af*(3N) in case of IPv6 systems, must be allowed and *remshd*(1M) must be executable on the remote host.

### Third-Party Transfers

Third-party transfers in the following form:

```
rcp ruser1@rhost1:path1 ruser2@rhost2:path2
```

are performed as:

```
remsh rhost1 -l ruser1 rcp path1 ruser2@rhost2:path2
```

Therefore, for a such a transfer to succeed, *ruser2* on *rhost2* must allow access by *ruser1* from *rhost1* (see *hosts.equiv*(4)).

### rcp With IPv6 Address

To invoke **rcp** with an IPv6 address, the IPv6 address must be enclosed in a pair of square brackets ([ and ]) as shown in the example below.

```
rcp source user@[IPv6_address]:dest
```

If the IPv6 address is not enclosed within square brackets, the first occurrence of a colon (:) is treated as the separator between the *hostname* and the *path*.

### WARNINGS

The **rcp** routine is confused by any output generated by commands in a **.cshrc** file on the remote host (see *cs*(1)).

Copying a file onto itself, for example:

```
rcp path `hostname`:path
```

may produce inconsistent results. The current HP-UX version of **rcp** simply copies the file over itself. However, some implementations of **rcp**, including some earlier HP-UX implementations, corrupt the file. In addition, the same file may be referred to in multiple ways, for example, via hard links, symbolic links, or NFS. It is not guaranteed that **rcp** will correctly copy a file over itself in all cases.

Implementations of **rcp** based on the 4.2BSD version (including the implementations of **rcp** prior to HP-UX 7.0) require that remote users be specified as *rhost.ruser*. If the first remote host specified in a third party transfer (*rhost1* in the example below) uses this older syntax, the command must have the form:

```
rcp ruser1@rhost1:path1 rhost2.ruser2:path2
```

since the target is interpreted by *rhost1*. A common problem is encountered when two remote files are to be copied to a remote target that specifies a remote user. If the two remote source systems, *rhost1* and *rhost2*, each expect a different form for the remote target, the command:

```
rcp rhost1:path1 rhost2:path2 rhost3.ruser3:path3
```

will certainly fail on one of the source systems. Perform such a transfer using two separate commands.

With the existing implementation of `rcp`, the remote copy may result in a system overwrite as described in the following example.

```
rcp -r path root@hostname: /
```

In this example, if you run `rcp` as root, and unintentionally type a space between the colon (`:`) and the slash (`/`), then `rcp` assumes both `path` and `root@hostname:` (the remote machine's root directory) as source. `rcp` always interprets the last argument as the destination. Therefore, the destination directory is the local machine's root directory (`/`). `rcp` copies the content of `path` to the root directory (`/`) first. It then does another copy with `root@hostname` as source to the root directory (`/`) again. This second copy overwrites the local system's root directory (`/`) with the remote system's root directory (`/`).

## DIAGNOSTICS

Diagnostics can occur from both the local and remote hosts. Those diagnostics that occur on the local host before the connection is completely established are written to standard error. Once the connection is established, any error messages from the remote host are written to standard output, like any other data.

**Error! could not retrieve authentication type.**

**Please notify sys admin.**

There are two authentication mechanisms used by `rcp`. One authentication mechanism is based on Kerberos and the other is not. The type of authentication mechanism is obtained from a system file which is updated by `inetsvcs_sec`(1M). If the system file does not contain known authentication types, the above error is displayed.

## AUTHOR

`rcp` was developed by the University of California, Berkeley.

## SEE ALSO

`cp`(1), `ftp`(1), `remsh`(1), `remshd`(1M), `inetsvcs_sec`(1M), `rcmd`(3N), `rcmd_af`(3N), `hosts`(4), `hosts.equiv`(4), `krb5.conf`(4), `sis`(5).

**NAME**

r`cs` - change RCS file attributes

**SYNOPSIS**

r`cs` [*options*] *file* ...

**DESCRIPTION**

r`cs` creates new RCS files or changes attributes of existing ones. An RCS file contains multiple revisions of text, an access list, a change log, descriptive text, and some control attributes. For r`cs` to work, the user's login name must be on the access list, except if the access list is empty, if the user is the owner of the file or the superuser, or if the `-i` option is present.

The user of the command must have read/write permission for the directory containing the RCS file and read permission for the RCS file itself. r`cs` creates a semaphore file in the same directory as the RCS file to prevent simultaneous update. For changes, r`cs` always creates a new file. On successful completion, r`cs` deletes the old one and renames the new one. This strategy makes links to RCS files useless.

Files ending in `,v` are RCS files; all others are working files. If a working file is given, r`cs` tries to find the corresponding RCS file first in directory `./RCS`, then in the current directory, as explained in *r`csintro`(5)*.

**Options**

r`cs` recognizes the following options:

- `-alogins` Appends the login names appearing in the comma-separated list *logins* to the access list of the RCS file.
- `-Aoldfile` Appends the access list of *oldfile* to the access list of the RCS file.
- `-c "string"` Sets the comment leader to *string*. The comment leader is printed before every log message line generated by the keyword `$Log$` during check out (see *co*(1)). This is useful for programming languages without multi-line comments. During r`cs` `-i` or initial `ci`, the comment leader is guessed from the suffix of the working file. Note, a comment leader is inserted at the beginning of each line of log information. The comment leader is determined by the suffix used with the file name, as in `foo.c`, or `foo.sh`, or `foo.p`. Note you can specify a different comment leader through the "r`cs`" command. The following table shows the comment leader associated with each file name suffix:

| SUFFIX | FILES          | Comment Character |
|--------|----------------|-------------------|
| c      | c              | ?*                |
| C      | C Header       | ?*                |
| sh     | shell          | '#'               |
| s      | Assembly       | '#'               |
| p      | pascal         | ?*                |
| r      | ratfor         | '#'               |
| e      | efl            | '#'               |
| l      | lex            | ?*                |
| y      | yacc           | ?*                |
| yr     | yacc-rarfor    | ?*                |
| ye     | yacc-efl       | ?*                |
| ml     | mocklisp       | ','               |
| mac    | macro          | ','               |
| f      | fortran        | 'c'               |
| ms     | ms-macros      | '\'               |
| me     | me-macros      | '\'               |
| ""     | empty suffix   | '#'               |
| nil    | unknown suffix | ""                |

`-e[logins]`

Erases the login names appearing in the comma-separated list *logins* from the access list of the RCS file. If *logins* is omitted, the entire access list is erased.

- i** Creates and initializes a new RCS file, but does not deposit any revision. If the RCS file has no path prefix, **r`cs`** tries to place it first into the subdirectory **./RCS**, then into the current directory. If the RCS file already exists, an error message is printed.
- l[*rev*]**  
Locks the revision with number *rev*. If a branch is given, the latest revision on that branch is locked. If *rev* is omitted, the latest revision on the trunk is locked. Locking prevents overlapping changes. A lock is removed with **ci** or **r`cs` -u** (see below).
- L** Sets locking to **strict**. Strict locking means that the owner of an RCS file is not exempt from locking for check in. This option should be used for files that are shared.
- nname[:*rev*]**  
Associates the symbolic name *name* with the branch or revision *rev*. **r`cs`** prints an error message if *name* is already associated with another number. If *rev* is omitted, the symbolic name is associated with the latest revision on the trunk. If *:rev* is omitted, the symbolic name is deleted.
- Nname[:*rev*]**  
Same as **-n**, except that it overrides a previous assignment of *name*.
- orange**  
Deletes ("obsoletes") the revisions given by *range*. A range consisting of a single revision number means that revision. A range consisting of a branch number means the latest revision on that branch. A range of the form *rev1–rev2* means revisions *rev1* to *rev2* on the same branch, **-rev** means from the beginning of the branch containing *rev* up to and including *rev*, and **rev-** means from revision *rev* to the head of the branch containing *rev*. None of the outdated revisions can have branches or locks.
- q** Quiet mode; diagnostics are not printed.
- sstate[:*rev*]**  
Sets the state attribute of the revision *rev* to *state*. If *rev* is omitted, the latest revision on the trunk is assumed. If *rev* is a branch number, the latest revision on that branch is assumed. Any identifier is acceptable for *state*. A useful set of states is **Exp** (for experimental), **Stab** (for stable), and **Rel** (for released). By default, **ci** sets the state of a revision to **Exp**.
- t[*txtfile*]**  
Writes descriptive text into the RCS file (deletes the existing text). If *txtfile* is omitted, **r`cs`** prompts the user for text supplied from the standard input, terminated with a line containing a single **.** or Ctrl-D. Otherwise, the descriptive text is copied from the file *txtfile*. If the **-i** option is present, descriptive text is requested even if **-t** is not given. The prompt is suppressed if the standard input is not a terminal.
- u[*rev*]**  
Unlocks the revision with number *rev*. If a branch is given, the latest revision on that branch is unlocked. If *rev* is omitted, the latest lock held by the user is removed. Normally, only the locker of a revision may unlock it. Somebody else unlocking a revision breaks the lock. This causes a mail message to be sent to the original locker. The message contains a commentary solicited from the breaker. The commentary is terminated with a line containing a single **.** or Control-D.
- U** Sets locking to non-strict. Non-strict locking means that the owner of a file need not lock a revision for check in. This option should *not* be used for files that are shared. The default (**-L** or **-U**) is determined by the system administrator.

### Access Control Lists (ACLs)

Do not add optional ACL entries to an RCS file, because they are deleted when the file is updated. The resulting access modes for the new file might not be as desired.

### DIAGNOSTICS

The RCS filename and the revisions outdated are written to the diagnostic output. The exit status always refers to the last RCS file operated upon, and is 0 if the operation was successful; 1 if unsuccessful.

### EXAMPLES

Add the names **jane**, **mary**, **dave**, and **jeff** to the access list of RCS file **vision,v**:

```
rcs -ajane,mary,dave,jeff vision
```

Set the comment leader to *tab\** for file **vision**:

```
rcs -c'tab*' vision
```

Associate the symbolic name **sso/6\_0** with revision **38.1** of file **vision**:

```
rcs -Nsso/6_0:38.1 vision
```

Lock revision **38.1** of file **vision,v** so that only the locker is permitted to check in (see *ci*(1)) the next revision of the file. This command prevents two or more people from simultaneously revising the same file and inadvertently overwriting each other's work.

```
rcs -l38.1 vision,v
```

#### **WARNINGS**

All **r**cs command options are available to anyone whose name appears in the file access list, including those to add and delete names in the access list, change strict locking, etc. If these options must be restricted, other security methods should be employed. Also see previous note regarding Access Control Lists.

#### **AUTHOR**

**r**cs was developed by Walter F. Tichy.

#### **SEE ALSO**

co(1), ci(1), rcsdiff(1), rcsmerge(1), rlog(1), rcsfile(4), acl(5), rcsintro(5).

**NAME**

rcsdiff - compare RCS revisions

**SYNOPSIS**

```
rcsdiff [-bcefh] [-rrev1] [-rrev2] file ...
```

**DESCRIPTION**

**rcsdiff** compares two revisions of each given RCS file and creates output very similar to **diff** (see *diff(1)*). A file name ending in **,v** is an RCS file name, otherwise it is a working file name. **rcsdiff** derives the working file name from the RCS file name and vice versa, as explained in *rcsintro(5)*. Pairs consisting of both an RCS and a working file name can also be specified.

**rcsdiff** recognizes the following options:

- b** Same as described in *diff(1)*;
- e** Same as described in *diff(1)*;
- f** Same as described in *diff(1)*;
- h** Same as described in *diff(1)*;
- n** Generate an edit script of the format used by RCS.
- c[n]**  
Generate a diff with lines of context. The default is to present 3 lines of context. To change, specify *n*; for example, **-c10** gives 10 lines of context.  
  
**-c** modifies the output format slightly from the normal *diff(1)* output. The “context” output begins with identification of the files involved and their creation dates, then each change is separated by a line with a dozen **\*** (asterisks). Lines removed from *file1* are marked with **-** (dashes); those added to *file2* with **+** (pluses). Lines that are changed from one file to the other are marked in both files with **!** (exclamation marks).

If both *rev1* and *rev2* are omitted, **rcsdiff** compares the latest revision on the trunk with the contents of the corresponding working file. This is useful for determining what was changed since the last check-in.

If *rev1* is given, but *rev2* is omitted, **rcsdiff** compares revision *rev1* of the RCS file with the contents of the corresponding working file.

If both *rev1* and *rev2* are given, **rcsdiff** compares revisions *rev1* and *rev2* of the RCS file.

Both *rev1* and *rev2* can be given numerically or symbolically.

| **r** |

**EXAMPLES**

Compare the latest trunk revision of RCS file **f.c,v** and the contents of working file **f.c**:

```
rcsdiff f.c
```

Compare the revisions 1.1 and 1.2 in the RCS file **foo.c,v**:

```
rcsdiff -r1.1 -r1.2 foo.c
```

**AUTHOR**

**rcsdiff** was developed by Walter F. Tichy.

**SEE ALSO**

ci(1), co(1), diff(1), ident(1), rcs(1), rcsmerge(1), rlog(1), rcsfile(4), rcsintro(5).

**NAME**

rscmerge - merge RCS revisions

**SYNOPSIS**

```
rscmerge -rrev1 [-rrev2] [-p] file
```

**DESCRIPTION**

**rscmerge** incorporates the changes between *rev1* and *rev2* of an RCS file into the corresponding working file. If **-p** is given, the result is printed on the standard output; otherwise the result overwrites the working file.

A file name ending in **,v** is an RCS file name; otherwise it is a working file name. **rscmerge** derives the working file name from the RCS file name and vice versa, as explained in *rscintro*(5). A pair consisting of both an RCS and a working file name can also be specified.

*rev1* cannot be omitted. If *rev2* is omitted, the latest revision on the trunk is assumed. Both *rev1* and *rev2* can be given numerically or symbolically.

**rscmerge** prints a warning if there are overlaps, and delimits the overlapping regions as explained for the **-j** option of *co*(1). The command is useful for incorporating changes into a checked-out revision.

**EXAMPLES**

Suppose you have released revision 2.8 of **f.c**. Assume furthermore that you just completed revision 3.4 when you receive updates to release 2.8 from someone else. To combine the updates to 2.8 and your changes between 2.8 and 3.4, put the updates to 2.8 into file **f.c** and execute:

```
rscmerge -p -r2.8 -r3.4 f.c >f.merged.c
```

Then examine **f.merged.c**. Alternatively, if you want to save the updates to 2.8 in the RCS file, check them in as revision 2.8.1.1 and execute **co -j**:

```
ci -r2.8.1.1 f.c
co -r3.4 -j2.8:2.8.1.1 f.c
```

As another example, the following command undoes the changes between revision 2.4 and 2.8 in your currently checked out revision in **f.c**:

```
rscmerge -r2.8 -r2.4 f.c
```

Note the order of the arguments, and that **f.c** is overwritten.

**WARNINGS**

**rscmerge** does not work for files that contain lines with a single **..**

**| r |****AUTHOR**

**rscmerge** was developed by Walter F. Tichy.

**SEE ALSO**

*ci*(1), *co*(1), *merge*(1), *ident*(1), *rsc*(1), *rscdiff*(1), *rlog*(1), *rscfile*(4).

**NAME**

rdist - remote file distribution program

**SYNOPSIS**

```
rdist [-bhinqvwyMR] [-f distfile] [-d var=value] [-m host] [label...]
rdist [-bhinqvwyMR] -c name... [login@]host[:dest]
```

**DESCRIPTION**

**rdist** facilitates the maintaining of identical copies of files over multiple hosts. It preserves the owner, group, mode, and modification time of files if possible and can update programs that are executing.

**-f *distfile*** Specify a *distfile* for **rdist** to execute. *distfile* contains a sequence of entries that specify the files to be copied, the destination hosts, and what operations to perform to do the updating. The format of *distfile* is described in detail later. If *distfile* is **-**, the standard input is used. If no **-f** option is present, the program looks first for a file called **distfile**, then **Distfile** in the local host's working directory to use as the input.

**-d *var=value*** Define *var* to have *value*. The **-d** option is used to define variable definitions in the *distfile*. *value* can be an empty string, one name, or a list of name separated by tabs and/or spaces and enclosed by a pair of parentheses. However, if the variable specified is already defined in the *distfile*, the **-d** option has no effect (because the *distfile* overrides the **-d** option).

**-m *host*** Limit which machines are to be updated. Multiple **-m** arguments can be given to limit updates to a subset of hosts that are listed in the *distfile*. For more information on the host format, refer to the section *destination\_list*.

*label* Label of a command to execute. The label must be defined in *distfile*.

**-c *name...*** The **-c** option forces **rdist** to interpret the remaining arguments as a small *distfile*. The equivalent *distfile* is as follows.

```
(name ...) -> [login@]host
 install [dest] ;
```

**Note:** In IPv6 enabled systems to use the **-c** option with an IPv6 address, the IPv6 address has to be enclosed in a square bracket pair ([ and ]). An example invocation of **rdist** with the **-c** option and an IPv6 address is as shown below:

```
rdist -c name user@[IPv6 address]:dest
```

If the IPv6 address is not enclosed within square brackets, the first occurrence of a colon (:) is treated as the separator between the *hostname* and the *path*.

- n** Print the commands without executing them. This option is useful for debugging *distfile*.
- q** Quiet mode. Files that are being modified are normally printed on standard output. The **-q** option suppresses this.
- R** Remove extraneous files. If a directory is being updated, any files that exist on the remote host that do not exist in the master directory are removed. This is useful for maintaining truly identical copies of directories.
- h** Follow symbolic links. Copy the file that the link points to rather than the link itself.
- i** Ignore unresolved links. **rdist** will normally try to maintain the link structure of files being transferred and warn the user if it cannot find all the links.
- v** Verify that the files are up to date on all the hosts. Any files that are out of date will be displayed but no files will be changed nor any mail sent.
- w** Whole mode. The whole file name is appended to the destination directory name. Normally, only the last component of a name is used when renaming files. This will preserve the directory structure of the files being copied instead of flattening the directory structure. For example, renaming a list of files such as (**dir1/f1** and **dir2/f2**) to **dir3** would create files **dir3/dir1/f1** and **dir3/dir2/f2** instead of **dir3/f1** and **dir3/f2**.
- y** Younger mode. Files are normally updated if their *mtime* and *size* (see *stat(2)*) disagree. The **-y** option causes **rdist** not to update files that are younger than the master copy. This can be used to prevent newer copies on other hosts from being replaced. A warning message is printed for files which are newer than the master copy.



- b** Binary comparison. Perform a binary comparison and update files if they differ rather than comparing dates and sizes.
- M** Check that mode, ownership, and group are the same in addition to any other form of comparison that is in effect. This option will cause files to be replaced but will only correct the problem with a directory and print a warning message.

The *distfile* used by **rdist** contains a sequence of entries that specify the files to be copied, the destination hosts, and what operations to perform to do the updating. Each entry has one of the following formats.

```
variable_name = name_list
[label:] source_list -> destination_list command_list
[label:] source_list :: time_stamp_file command_list
```

The first format is used for defining variables. The second format is used for distributing files to other hosts. The third format is used for making lists of files on the local host that have been changed since some given date. (See *EXAMPLES*.)

*variable\_name*

Specify the name of a variable.

*name\_list* List of names (such as list of hosts or lists of files) separated by tabs and/or spaces and enclosed by parentheses.

*source\_list*

Specify a list of files and/or directories on the local host to be used as the master copy for distribution. Each file in the *source\_list* is added to a list for changes, *if* the file is out of date on the host that is being updated (second format), or *if* the file is newer than the time stamp file (third format). *source\_list* may contain a single name, or multiple names separated by tabs and/or spaces and enclosed by parentheses.

*destination\_list*

List of hosts to which these files are to be copied. *destination\_list* may contain a single name, or multiple names separated by tabs and/or spaces and the whole list must be enclosed by parentheses. The host names in the *destination\_list* can also be in the form *login@host*. For example, **root@arpa**. In this case, the user **root** owns the files distributed at **arpa**.

*time\_stamp\_file*

Specify a given date to generate a list of files on the local host that were modified since that date.

*label:* Labels are optional. They are used to identify a command for partial updates.

*command\_list*

Specifies a list of commands to be performed.

The command list consists of zero or more commands of the following format.

```
install [options] opt_dest_name ;
notify name_list ;
except name_list ;
except_pat pattern_list ;
special name_list string ;
```

The **install** command is used to copy out-of-date files and/or directories. Each source file is copied to each host in the destination list. Directories are recursively copied in the same way. *opt\_dest\_name* is an optional parameter to rename files. If no **install** command appears in the command list or the destination name is not specified, *source\_list* is used. Directories in the path name will be created if they do not exist on the remote host. To help prevent disasters, a non-empty directory on a target will never be replaced with a regular file or a symbolic link. However, under the **-R** option a non-empty directory will be removed if the corresponding filename is completely absent on the master host. The *options* are **-b**, **-h**, **-i**, **-v**, **-w**, **-y**, **-M**, and **-R**, and have the same semantics as options on the command line, except that they only apply to the files in the specified *source\_list*. The login name used on the destination host is the same as on the local host, unless the destination name is of the form "login@host".

The **notify** command is used to mail the list of files updated (and any errors that may have occurred) to the listed names, in *name\_list*. If no **@** appears in the name, the destination host is appended to the name (e.g., name1@host, name2@host, ...).

The **except** command is used to update all of the files in the source list, *except* for the files listed in *name\_list*. This is usually used to copy everything in a directory except certain files.

The **except\_pat** command is like the **except** command except that *pattern\_list* is a list of regular expressions (see *ed*(1) for details). If one of the patterns matches some string within a file name, that file will be ignored. Note that since the backslash (\) is a quote character, it must be doubled to become part of the regular expression. Variables are expanded in *pattern\_list* but not shell file pattern matching characters. To include a \$, it must be escaped with the backslash.

The **special** command is used to specify *sh*(1) commands that are to be executed on the remote host after the file in *name\_list* is updated or installed. If the *name\_list* is omitted then the shell commands will be executed for every file updated or installed. The shell variable 'FILE' is set to the current filename before executing the commands in *string*. *string* starts and ends with double quotes (") and can cross multiple lines in *distfile*. Multiple commands to the shell should be separated by semi-colons (;). Commands are executed in the user's home directory on the host being updated. The **special** command can be used, for example, to rebuild private databases after a program has been updated. Shell variables cannot be used in the command because there is no escape mechanism for the \$ character.

Newlines, tabs, and blanks are only used as separators and are otherwise ignored. Comments begin with # and end with a newline.

A generalized way of dynamically building variable lists is provided by using a backquote syntax much like the shell. In this way, arbitrary commands that generate stdout with space-separated words may be used to build the list (see the use of **cat** command in the examples).

Variables to be expanded begin with \$ followed by the variable name enclosed in curly braces.

The shell meta-characters [, ], {, }, \*, and ? are recognized and expanded (on the local host only) in the same way as *cs*h(1). They can be escaped with a backslash. The ~ character is also expanded in the same way as *cs*h but is expanded separately on the local and destination hosts. When the **-w** option is used with a file name that begins with ~, everything except the home directory is appended to the destination name. File names which do not begin with / or ~ use the destination user's home directory as the root directory for the rest of the file name.

## DIAGNOSTICS

A complaint about mismatch of rdist version numbers may mean that an executable **rdist** is not in the shell's path on the remote system.

## EXAMPLES

The following is a small example.

```
HOSTS = (matisse root@arpa)

FILES = (/usr/lib /usr/bin /usr/local/games
 /usr/include/{*.h,{sys,rpc*,arpa}/*.h}
 /usr/man/man? `cat ./std-files`)

EXLIB = (Mail.rc aliases aliases.dir aliases.pag crontab dshrc
 sendmail.cf sendmail.fc sendmail.hf sendmail.st uucp vfont)

${FILES} -> ${HOSTS}
 install -R ;
 except /usr/lib/${EXLIB} ;
 except /usr/local/games/lib ;
 special /usr/sbin/sendmail " /usr/sbin/sendmail -bz" ;

srcs:
 /usr/local/src -> arpa
 except_pat (\\o$ /SCCS\$) ;

IMAGEN = (ips dviimp catdvi)

imagen:
 /usr/local/${IMAGEN} -> arpa
```

```
install /usr/local/lib ;
notify ralph ;

${FILES} :: stamp.cory
notify root@cory ;
```

**WARNINGS**

Source files must reside on the local host where **rdist** is executed.

There is no easy way to have a special command executed after all files in a directory have been updated.

Variable expansion only works for name lists and in the **special** command string; there should be a general macro facility.

**rdist** aborts on files that have a negative mtime (before Jan 1, 1970).

**rdist** does carry the atime when installing a file but will preserve it on an updated file.

There should be a 'force' option to allow replacement of non-empty directories by regular files or symlinks.

**AUTHOR**

**rdist** was developed by the University of California, Berkeley.

**rdist** appeared in the 4.3 Berkeley Software Distribution.

**FILES**

|                    |                                  |
|--------------------|----------------------------------|
| <b>distfile</b>    | Input command file.              |
| <b>/tmp/rdist*</b> | Temporary file for update lists. |

**SEE ALSO**

sh(1), csh(1), stat(2).

**| r |**

**NAME**

read - read a line from standard input

**SYNOPSIS**

```
read [-r] var ...
```

**DESCRIPTION**

**read** reads a single line from standard input. The line is split into fields as when processed by the shell (refer to shells in SEE ALSO); the first field is assigned to the first variable *var*, the second field to the second variable *var*, and so forth. If there are more fields than there are specified *var* operands, the remaining fields and their intervening separators are assigned to the last *var*. If there are more *vars* than fields, the remaining *vars* are set to empty strings.

The setting of variables specified by the *var* operands affect the current shell execution environment.

Standard input to **read** can be redirected from a text file.

Since **read** affects the current shell execution environment, it is usually provided as a normal shell special (built-in) command. Thus, if it is called in a subshell or separate utility execution environment similar to the following, it does not affect the shell variables in the caller's environment:

```
(read foo)
nohup read ...
find . -exec read ... ;
```

**Options and Arguments**

**read** recognizes the following options and command-line arguments:

|            |                                                                                                              |
|------------|--------------------------------------------------------------------------------------------------------------|
| <b>-r</b>  | Do not treat a backslash character in any special way. Consider each backslash to be part of the input line. |
| <i>var</i> | The name of an existing or non-existing shell variable.                                                      |

**EXTERNAL INFLUENCES****Environment Variables**

**IFS** determines the internal field separators used to delimit fields.

**RETURN VALUE**

**read** exits with one of the following values:

|              |                                                |
|--------------|------------------------------------------------|
| <b>0</b>     | Successful completion.                         |
| <b>&gt;0</b> | End-of-file was detected or an error occurred. |

**EXAMPLES**

Print a file with the first field of each line moved to the end of the line.

```
while read -r xx yy
do
 printf "%s %s \n" "$yy" "$xx"
done < input_file
```

**SEE ALSO**

csh(1), ksh(1), sh-posix(1), sh(1).

**STANDARDS CONFORMANCE**

**read**: SVID2, XPG2, XPG3, XPG4, POSIX.2 FIPS

**NAME**

readmail - read mail from a mail folder or incoming mailbox

**SYNOPSIS**

```
readmail [-ahnp] [-f folder] [number-list | pattern]
```

**DESCRIPTION**

The **readmail** program displays messages from your incoming mailbox or a specified mail folder.

Within the **elm** mail system (see *elm*(1) with no operands and optionally the **-h** or **-n** option, **readmail** displays the appropriate headers and the body of the current message.

With the *number-list* operand and no options, **readmail** displays the corresponding messages and a summary of the headers from your incoming mailbox.

With the *pattern* operand and no options, **readmail** displays the first message that matches the pattern and a summary of the headers from your incoming mailbox.

**Options**

**readmail** supports the following options.

- a** Print all messages that match *pattern*. If no pattern was specified, this option is ignored.
- f folder** Use file *folder* for the operations instead of the incoming mailbox.
- h** Include the entire header of the matched message or messages when displaying their text. The default is to display the **From:**, **Date:**, and **Subject:** lines only.
- n** Exclude all headers.
- p** Put form feeds (**Ctrl-L**) between message headers. This is useful when printing sets of messages.

**Operands**

**readmail** supports the following operands.

- number-list* A blank-separated list of the ordinal locations of messages in the mail file (i.e., their "message numbers"), up to 25 at a time. The character **\$** means the last message in the mail file. Similarly, **\*** represents every message in the file (i.e., **1 2 3 ... \$**)  
The message numbers are sorted into ascending order. Thus, **1 3 2** produces the same output as **1 2 3**.
- pattern* A string that is present in one of the messages. This pattern can be typed in directly (no quotes) if the words are separated by a single space in the actual message. The pattern matching is case sensitive, so **Hello** and **hello** are not equivalent. Leading digits (on the first word) are not permitted; however, you can precede them with a space and quote the entire string, if the space occurs in the message, as in **" 1st item of business" .**

| **r** |**EXAMPLES**

If you are using **vi** to reply to a message from within the **elm** mail system, you can insert the text of the current message with the command:

```
:r !readmail
```

If you define an alias similar to:

```
alias rd='readmail $ | page' (Korn or POSIX shell)
alias rd 'readmail $ | page' (C shell)
```

you can use it with a program such **newmail** to peruse mail as it arrives, without needing to start a mail system (see *newmail*(1)).

**AUTHOR**

**readmail** was developed by HP.

**FILES**

|                                   |                                     |
|-----------------------------------|-------------------------------------|
| <code>/var/mail/loginname</code>  | Incoming mailbox                    |
| <code>\$HOME/.elm/readmail</code> | Temporary file for <code>elm</code> |

**SEE ALSO**

`elm(1)`, `newmail(1)`, `vi(1)`.

**| r |**

**NAME**

remsh, rexec - execute from a remote shell

**SYNOPSIS**

```
remsh host [-l username] [-n] command
 host [-l username] [-n] command
```

```
rexec host [-l username] [-n] command
```

**In Kerberos V5 Network Authentication Environments**

```
remsh host [-l username] [-f|-F] [-k realm] [-P] [-n] command
 host [-l username] [-f|-F] [-k realm] [-P] [-n] command
```

**DESCRIPTION**

**remsh** connects to a specified *host* and executes a specified *command*. The host name can be either the official name or an alias as understood by **gethostbyname()** (see *gethostent(3N)* and *hosts(4)*). **remsh** copies its standard input (**stdin**) to the remote command, the standard output of the remote command to its standard output (**stdout**), and the standard error of the remote command to its standard error (**stderr**). Hangup, interrupt, quit, terminate, and broken pipe signals are propagated to the remote command. **remsh** exits when the sockets associated with **stdout** and **stderr** of the remote command are closed. This means that **remsh** normally terminates when the remote command does (see *remshd(1M)*).

By default, **remsh** uses the following path when executing the specified *command*:

```
/usr/bin:/usr/ccs/bin:/usr/bin/X11:/usr/contrib/bin:/usr/local/bin
```

**remsh** uses the default remote login shell with the **-c** option to execute the remote command. If the default remote shell is *csh*, *csh* sources the remote **.cshrc** file before the command. **remsh** cannot be used to run commands that require a terminal interface (such as **vi**) or commands that read their standard error (such as **more**). In such cases, use **rlogin** or **telnet** instead (see *rlogin(1)* and *telnet(1)*).

The remote account name used is the same as your local account name, unless you specify a different remote name with the **-l** option. This remote account name must be equivalent to the originating account. In addition, the remote host account name must also conform to other rules, which differ depending upon whether the remote host is operating in a Kerberos V5 Network Authentication, i.e., secure environment, or not.

In a non-secure, or traditional environment, the remote account name must be equivalent to the originating account; no provision is made for specifying a password with a command. For more details about **equivalent** hosts and how to specify them, see *hosts.equiv(4)*. The files inspected by **remshd** on the remote host are **/etc/hosts.equiv** and **\$HOME/.rhosts** (see *remshd(1M)*).

In a Kerberos V5 Network Authentication environment, the local host must be successfully authenticated before the remote account name is checked for proper authorization. The authorization mechanism is dependent on the command line options used to invoke **remshd** on the remote host (i.e., **-K**, **-R**, **-r**, or **-k**). For more information on Kerberos authentication and authorization see the Secure Internet Services man page, *sis(5)* and *remshd(1M)*.

Although Kerberos authentication and authorization may apply, the Kerberos mechanism is *not* applied to the *command* or to its response. All the information that is transferred between the local and remote host is still sent in cleartext over the network.

The default Kerberos options for the applications are set in the **krb5.conf** configuration file. Refer to the *appdefaults Section* in the *krb5.conf(4)* manpage for more information. The options **-f**, and **-F** described in the subsequent paragraphs, can be set in the **krb5.conf** file with the tag names **forward**, and **forwardable** respectively. Refer to the *krb5.conf(4)* manpage for more information on the *appdefaults Section*.

The **fallback** option can be set in the **krb5.conf** file within the *appdefaults Section*. If **fallback** is set to true and the kerberos authentication fails, **remsh** will use the non-secure mode of authentication.

**Note:** Command line options override the configuration file options.

In a secure or Kerberos V5-based environment, the following command line options are available:

```
-f Forward the ticket granting ticket (TGT) to the remote system. The TGT is not forwardable from that remote system.
```

- F** Forward the TGT to the remote system and have it forwardable from there to another remote system. The **-f** option and **-F** option are mutually exclusive.
- k realm** Obtain tickets from the remote host in the specified *realm* instead of the remote host's default realm as specified in the configuration file **krb.realms**.
- P** Disable Kerberos authentication.

If *command*, is not specified, instead of executing a single command, you will be logged in on the remote host using **rlogin** (see *rlogin(1)*). Any **rlogin** options typed in on the command line are transmitted to **rlogin**. If no *command* and the option **-P** are specified, **rlogin** will be invoked with **-P** to indicate that Kerberos authentication (or secure access) is not necessary. This means that if a password is requested, the password will be sent in cleartext. If *command* is specified, options specific to **rlogin** are ignored by **remsh**.

By default, **remsh** reads its standard input and sends it to the remote command because **remsh** has no way to determine whether the remote command requires input. The **-n** option redirects standard input to **remsh** from **/dev/null**. This is useful when running a shell script containing a **remsh** command, since otherwise **remsh** may use input not intended for it. The **-n** option is also useful when running **remsh** in the background from a job control shell, **/usr/bin/csh** or **/usr/bin/ksh**. Otherwise, **remsh** stops and waits for input from the terminal keyboard for the remote command. **/usr/bin/sh** automatically redirects its input from **/dev/null** when jobs are run in the background.

Host names for remote hosts can also be commands (linked to **remsh**) in the directory **/usr/hosts**. If this directory is specified in the **\$PATH** environment variable, you can omit **remsh**. For example, if **remotehost** is the name of a remote host, **/usr/hosts/remotehost** is linked to **remsh**, and if **/usr/hosts** is in your search path, the command

```
remotehost command
```

executes *command* on **remotehost**, and the command

```
remotehost
```

is equivalent to

```
rlogin remotehost
```

The **rexec** command works in the same way as **remsh** except that it uses the **rexec()** library routine and **rexecd** for command execution (see *rexec(3N)* and *rexecd(1M)*) and does not support Kerberos authentication. **rexec** prompts for a password before executing the command instead of using **hosts.equiv** for authentication. It should be used in instances where a password to a remote account is known but there are insufficient permissions for **remsh**.

## R

### EXAMPLES

Shell metacharacters that are not quoted are interpreted on the local host; quoted metacharacters are interpreted on the remote host. Thus the command line:

```
remsh otherhost cat remotefile >> localfile
```

appends the remote file **remotefile** to the local file **localfile**, while the command line

```
remsh otherhost cat remotefile ">>" otherremotefile
```

appends **remotefile** to the remote file **otherremotefile**.

If the remote shell is **/usr/bin/sh**, the following command line sets up the environment for the remote command before executing the remote command:

```
remsh otherhost . .profile 2>&- \; command
```

The **2>&-** throws away error messages generated by executing **.profile** when *stdin* and *stdout* are not a terminal.

The following command line runs **remsh** in the background on the local system, and the output of the remote command comes to your terminal asynchronously:

```
remsh otherhost -n command &
```

The background **remsh** completes when the remote command does.

The following command line causes **remsh** to return immediately without waiting for the remote command to complete:



```
remsh otherhost -n "command 1>&- 2>&- &"
```

(See *remshd*(1M) and *sh*(1)). If your login shell on the remote system is *cs**h*, use the following form instead:

```
remsh otherhost -n "sh -c \"command 1>&- 2>&- &\""
```

## RETURN VALUE

If **remsh** fails to set up the secondary socket connection, it returns 2. If it fails in some other way, it returns 1. If it fully succeeds in setting up a connection with **remshd**, it returns 0 once the remote command has completed. Note that the return value of **remsh** bears no relation to the return value of the remote command.

## DIAGNOSTICS

Besides the errors listed below, errors can also be generated by the library functions **rcmd()** and **rresvport()**. In the case of IPv6 systems, the library functions **rcmd()** and **rresvport()** are replaced by **rcmd\_af()** and **rresvport\_af()**; respectively, and can generate errors (see *rcmd*(3N) and *rcmd\_af*(3N)). These errors are preceded by the name of the library function that generated them. **remsh** can produce the following diagnostic messages:

**Error! could not retrieve authentication type.  
Please notify sys admin.**

There are two authentication mechanisms used by **remsh**. One authentication mechanism is based on Kerberos and the other is not. The type of authentication mechanism is obtained from a system file which is updated by **inetsvcs\_sec** (see *inetsvcs\_sec*(1M)). If the system file does not contain known authentication types, the above error is displayed.

**rlogin: ...**

Error in executing **rlogin** (**rlogin** is executed when the user does not specify any commands to be executed). This is followed by the error message specifying why the execution failed.

**shell/tcp: Unknown service**

The “shell” service specification is not present in the **/etc/services** file.

**Can't establish stderr**

**remsh** cannot establish secondary socket connection for **stderr**.

**<system call>: ...**

Error in executing system call. Appended to this error is a message specifying the cause of the failure.

**There is no entry for you (user ID *uid*) in /etc/passwd**

Check with the system administrator to see if your entry in the password file has been deleted by mistake.

Kerberos-specific errors are listed in *sis*(5).

## WARNINGS

For security reasons, the **/etc/hosts.equiv** and **.rhosts** files should exist, even if empty, and they should be readable and writable only by the owner. Note also that all information, including any passwords asked for, is passed unencrypted between the two hosts.

If **remsh** is run with an interactive command, it hangs.

## DEPENDENCIES

**remsh** is the same service as **rsh** on BSD systems. The name was changed due to a conflict with the existing System V command **rsh** (restricted shell).

## AUTHOR

**remsh** was developed by the University of California, Berkeley.

## FILES

**/usr/hosts/\*** for version of the command invoked only with hostname

## SEE ALSO

*rlogin*(1), *remshd*(1M), *rexc*(1M), *inetsvcs\_sec*(1M), *gethostent*(3N), *rcmd*(3N), *rcmd\_af*(3N), *rexc*(3N), *hosts*(4), *hosts.equiv*(4), *krb5.conf*(4).

**NAME**

rev - reverse lines of a file

**SYNOPSIS**

rev [*file*] ...

**DESCRIPTION**

**rev** copies the named files to the standard output, reversing the order of characters in every line. If no file is specified, the standard input is copied.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the interpretation of text as single- and/or multi-byte characters.

If **LC\_CTYPE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, *rev* behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

| **r** |

**NAME**

remsh, rexec - execute from a remote shell

**SYNOPSIS**

**remsh** *host* [-**l** *username*] [-**n**] *command*  
*host* [-**l** *username*] [-**n**] *command*

**rexec** *host* [-**l** *username*] [-**n**] *command*

**In Kerberos V5 Network Authentication Environments:**

**remsh** *host* [-**l** *username*] [-**f**|-**F**] [-**k** *realm*] [-**P**] [-**n**] *command*  
*host* [-**l** *username*] [-**f**|-**F**] [-**k** *realm*] [-**P**] [-**n**] *command*

**DESCRIPTION**

**remsh** connects to a specified *host* and executes a specified *command*. The host name can be either the official name or an alias as understood by **gethostbyname()** (see *gethostent*(3N) and *hosts*(4)). **remsh** copies its standard input (**stdin**) to the remote command, the standard output of the remote command to its standard output (**stdout**), and the standard error of the remote command to its standard error (**stderr**). Hangup, interrupt, quit, terminate, and broken pipe signals are propagated to the remote command. **remsh** exits when the sockets associated with **stdout** and **stderr** of the remote command are closed. This means that **remsh** normally terminates when the remote command does (see *remshd*(1M)).

By default, **remsh** uses the following path when executing the specified *command*:

**/usr/bin:/usr/ccs/bin:/usr/bin/X11:/usr/contrib/bin:/usr/local/bin**

**remsh** uses the default remote login shell with the **-c** option to execute the remote command. If the default remote shell is *csh*, *csh* sources the remote **.cshrc** file before the command. **remsh** cannot be used to run commands that require a terminal interface (such as **vi**) or commands that read their standard error (such as **more**). In such cases, use **rlogin** or **telnet** instead (see *rlogin*(1) and *telnet*(1)).

The remote account name used is the same as your local account name, unless you specify a different remote name with the **-l** option. This remote account name must be equivalent to the originating account. In addition, the remote host account name must also conform to other rules, which differ depending upon whether the remote host is operating in a Kerberos V5 Network Authentication, i.e., secure environment, or not.

In a non-secure, or traditional environment, the remote account name must be equivalent to the originating account; no provision is made for specifying a password with a command. For more details about **equivalent** hosts and how to specify them, see *hosts.equiv*(4). The files inspected by **remshd** on the remote host are **/etc/hosts.equiv** and **\$HOME/.rhosts** (see *remshd*(1M)).

In a Kerberos V5 Network Authentication environment, the local host must be successfully authenticated before the remote account name is checked for proper authorization. The authorization mechanism is dependent on the command line options used to invoke **remshd** on the remote host (i.e., **-K**, **-R**, **-r**, or **-k**). For more information on Kerberos authentication and authorization see the Secure Internet Services man page, *sis*(5) and *remshd*(1M).

Although Kerberos authentication and authorization may apply, the Kerberos mechanism is *not* applied to the *command* or to its response. All the information that is transferred between the local and remote host is still sent in cleartext over the network.

In a secure or Kerberos V5-based environment, the following command line options are available:

- f** Forward the ticket granting ticket (TGT) to the remote system. The TGT is not forwardable from that remote system.
- F** Forward the TGT to the remote system and have it forwardable from there to another remote system. The **-f** option and **-F** option are mutually exclusive.
- k realm** Obtain tickets from the remote host in the specified *realm* instead of the remote host's default realm as specified in the configuration file **krb.realms**.
- P** Disable Kerberos authentication.

**Note:** The Kerberos authentication and authorization mechanism and the above Kerberos specific options are not supported in *IPv6 enabled systems*.

If *command*, is not specified, instead of executing a single command, you will be logged in on the remote host using **rlogin** (see *rlogin*(1)). Any **rlogin** options typed in on the command line are transmitted

to **rlogin**. If no *command* and the option **-P** are specified, **rlogin** will be invoked with **-P** to indicate that Kerberos authentication (or secure access) is not necessary. This means that if a password is requested, the password will be sent in cleartext. If *command* is specified, options specific to **rlogin** are ignored by **remsh**.

By default, **remsh** reads its standard input and sends it to the remote command because **remsh** has no way to determine whether the remote command requires input. The **-n** option redirects standard input to **remsh** from **/dev/null**. This is useful when running a shell script containing a **remsh** command, since otherwise **remsh** may use input not intended for it. The **-n** option is also useful when running **remsh** in the background from a job control shell, **/usr/bin/csh** or **/usr/bin/ksh**. Otherwise, **remsh** stops and waits for input from the terminal keyboard for the remote command. **/usr/bin/sh** automatically redirects its input from **/dev/null** when jobs are run in the background.

Host names for remote hosts can also be commands (linked to **remsh**) in the directory **/usr/hosts**. If this directory is specified in the **\$PATH** environment variable, you can omit **remsh**. For example, if **remotehost** is the name of a remote host, **/usr/hosts/remotehost** is linked to **remsh**, and if **/usr/hosts** is in your search path, the command

```
remotehost command
```

executes *command* on **remotehost**, and the command

```
remotehost
```

is equivalent to

```
rlogin remotehost
```

The **rexec** command works in the same way as **remsh** except that it uses the **rexec()** library routine and **rexe**cd for command execution (see *rexec(3N)* and *rexe*cd(1M)) and does not support Kerberos authentication. **rexec** prompts for a password before executing the command instead of using **hosts.equiv** for authentication. It should be used in instances where a password to a remote account is known but there are insufficient permissions for **remsh**.

## EXAMPLES

Shell metacharacters that are not quoted are interpreted on the local host; quoted metacharacters are interpreted on the remote host. Thus the command line:

```
remsh otherhost cat remotefile >> localfile
```

appends the remote file **remotefile** to the local file **localfile**, while the command line

```
remsh otherhost cat remotefile ">>" otherremotefile
```

appends **remotefile** to the remote file **otherremotefile**.

If the remote shell is **/usr/bin/sh**, the following command line sets up the environment for the remote command before executing the remote command:

```
remsh otherhost . .profile 2>&- \; command
```

The **2>&-** throws away error messages generated by executing **.profile** when *stdin* and *stdout* are not a terminal.

The following command line runs **remsh** in the background on the local system, and the output of the remote command comes to your terminal asynchronously:

```
remsh otherhost -n command &
```

The background **remsh** completes when the remote command does.

The following command line causes **remsh** to return immediately without waiting for the remote command to complete:

```
remsh otherhost -n "command 1>&- 2>&- &"
```

(See *remshd(1M)* and *sh(1)*). If your login shell on the remote system is *csh*, use the following form instead:

```
remsh otherhost -n "sh -c \"command 1>&- 2>&- &\""
```

## RETURN VALUE

If **remsh** fails to set up the secondary socket connection, it returns 2. If it fails in some other way, it returns 1. If it fully succeeds in setting up a connection with **remshd**, it returns 0 once the remote

command has completed. Note that the return value of **remsh** bears no relation to the return value of the remote command.

## DIAGNOSTICS

Besides the errors listed below, errors can also be generated by the library functions **rcmd()** and **rresvport()**. In the case of IPv6 systems, the library functions **rcmd()** and **rresvport()** are replaced by **rcmd\_af()** and **rresvport\_af()**; respectively, and can generate errors (see *rcmd(3N)* and *rcmd\_af(3N)*). These errors are preceded by the name of the library function that generated them. **remsh** can produce the following diagnostic messages:

**Error! could not retrieve authentication type.  
Please notify sys admin.**

There are two authentication mechanisms used by **remsh**. One authentication mechanism is based on Kerberos and the other is not. The type of authentication mechanism is obtained from a system file which is updated by **inetsvcs\_sec** (see *inetsvcs\_sec(1M)*). If the system file does not contain known authentication types, the above error is displayed.

**rlogin: ...**

Error in executing **rlogin** (**rlogin** is executed when the user does not specify any commands to be executed). This is followed by the error message specifying why the execution failed.

**shell/tcp: Unknown service**

The “shell” service specification is not present in the **/etc/services** file.

**Can't establish stderr**

**remsh** cannot establish secondary socket connection for **stderr**.

**<system call>: ...**

Error in executing system call. Appended to this error is a message specifying the cause of the failure.

**There is no entry for you (user ID *uid*) in /etc/passwd**

Check with the system administrator to see if your entry in the password file has been deleted by mistake.

Kerberos-specific errors are listed in *sis(5)*.

## WARNINGS

For security reasons, the **/etc/hosts.equiv** and **.rhosts** files should exist, even if empty, and they should be readable and writable only by the owner. Note also that all information, including any passwords asked for, is passed unencrypted between the two hosts.

If **remsh** is run with an interactive command, it hangs.

| **r** |

## DEPENDENCIES

**remsh** is the same service as **rsh** on BSD systems. The name was changed due to a conflict with the existing System V command **rsh** (restricted shell).

## AUTHOR

**remsh** was developed by the University of California, Berkeley.

## FILES

**/usr/hosts/\*** for version of the command invoked only with hostname

## SEE ALSO

*rlogin(1)*, *remshd(1M)*, *rexecd(1M)*, *inetsvcs\_sec(1M)*, *gethostent(3N)*, *rcmd(3N)*, *rcmd\_af(3N)*, *rexec(3N)*, *hosts.equiv(4)*, *hosts(4)*.

**NAME**

rlog - print log messages and other information on RCS files

**SYNOPSIS**

rlog [*options*] *file* ...

**DESCRIPTION**

rlog prints information about RCS files. Files ending in *,v* are RCS files; all others are working files. If a working file is given, rlog tries to find the corresponding RCS file first in directory *./RCS*, then in the current directory, as explained in *rcsintr*(5).

rlog prints the following information for each RCS file: RCS file name, working file name, head (i.e., the number of the latest revision on the trunk), access list, locks, symbolic names, suffix, total number of revisions, number of revisions selected for printing, and descriptive text. This is followed by entries for the selected revisions in reverse chronological order for each branch. For each revision, rlog prints revision number, author, date/time, state, number of lines added/deleted (with respect to the previous revision), locker of the revision (if any), and log message. Without options, rlog prints complete information. The options below restrict this output.

**Options**

rlog recognizes the following options:

- d***dates* Print information about revisions whose check-in date and time fall within the ranges given by the semicolon-separated list of *dates*. A range of the form *d1<d2* or *d2>d1* selects the revisions that were deposited between *d1* and *d2* (inclusive). A range of the form *<d* or *d>* selects all revisions dated *d* or earlier. A range of the form *d<* or *>d* selects all revisions dated *d* or later. A range of the form *d* selects the single, latest revision dated *d* or earlier. The date/time strings *d*, *d1*, and *d2* are in the format explained in *co*(1). Quoting is normally necessary, especially for *<* and *>*. Note that the separator is a semicolon.
- h** Print only RCS file name, working file name, head, access list, locks, symbolic names, and suffix.
- l***lockers* Print information about locked revisions. If the comma-separated list *lockers* of login names is given, only the revisions locked by the given login names are printed. If the list is omitted, all locked revisions are printed.
- L** Ignore RCS files that have no locks set; convenient in combination with **-R**, **-h**, or **-l**.
- r***revisions* Print information about revisions given in the comma-separated list *revisions* of revisions and ranges. A range *rev1-rev2* means revisions *rev1* to *rev2* on the same branch, **-rev** means revisions from the beginning of the branch up to and including *rev*, and *rev-* means revisions starting with *rev* to the head of the branch containing *rev*. An argument that is a branch means all revisions on that branch. A range of branches means all revisions on the branches in that range.
- R** Print only the name of the RCS file; convenient for translating a working file name into an RCS file name.
- s***states* Print information about revisions whose state attributes match one of the states given in the comma-separated list *states*.
- t** Print the same as **-h**, plus the descriptive text.
- w***logins* Prints information about revisions checked in by users whose login names appearing in the comma-separated list *logins*. If *logins* is omitted, the user's login is assumed.

rlog prints the intersection of the revisions selected with the options **-d**, **-l**, **-s**, **-w**, and **-r**.

**EXAMPLES**

Print the names of all RCS files in the subdirectory named **RCS** that have locks:

```
rlog -L -R RCS/*,v
```

Print the headers of those files:

```
rlog -L -h RCS/*,v
```

Print the headers plus the log messages of the locked revisions:

```
rlog -L -l RCS/*,v
```

Print complete log information:

```
rlog RCS/*,v
```

Print the header and log messages of all revisions checked in after 1:00am on December 25th, 1991:

```
rlog -d">12/25/92, 1:00" RCS/*,v
```

Print the header and log messages of those revisions that were created between 10:00am and 2:00pm on July 4th, 1992:

```
rlog -d"07/04/92, 10:00 > 92/07/04, 14:00" RCS/*,v
```

**DIAGNOSTICS**

The exit status always refers to the last RCS file operated upon, and is 0 if the operation was successful, 1 if unsuccessful.

**AUTHOR**

**rlog** was developed by Walter F. Tichy.

**SEE ALSO**

ci(1), co(1), ident(1), rcs(1), rcsdiff(1), rcsmerge(1), rcsfile(4), rcsintro(5).

**| r |**

**NAME**

rlogin - remote login

**SYNOPSIS**

```
rlogin rhost [-7] [-8] [-ee] [-l username]
```

```
rhost [-7] [-8] [-ee] [-l username]
```

**In Kerberos V5 Network Authentication Environments**

```
rlogin rhost [-7] [-8] [-ee] [-f|-F] [-k realm] [-l username] [-P]
```

```
rhost [-7] [-8] [-ee] [-f|-F] [-k realm] [-l username] [-P]
```

**DESCRIPTION**

The **rlogin** command connects your terminal on the local host to the remote host (*rhost*). **rlogin** acts as a virtual terminal to the remote system. The host name *rhost* can be either the official name or an alias as listed in the file `/etc/hosts` (see *hosts*(4)).

In non-secure or traditional environment, **rlogin** allows a user to log in on an equivalent remote host, *rhost*, bypassing the normal login/password sequence, in a manner similar to the **remsh** command (see *remsh*(1)). For more information about equivalent hosts and how to specify them in the files `/etc/hosts.equiv` and `.rhosts`, see *hosts.equiv*(4). The searching of the files `/etc/hosts.equiv` and `.rhosts` occurs on the remote host, and the `.rhosts` file must be owned by the remote user account.

If the originating user account is not equivalent to the remote user account, the originating user is prompted for the password of the remote account. If this fails, a login name and password are prompted for, as when **login** is used (see *login*(1)).

In a Kerberos V5 Network Authentication environment, **rlogin** uses the Kerberos V5 protocol to authenticate the connection to a remote host. If the authentication is successful, user authorization will be performed according to the command line options selected for **rlogind** (i.e., **-K**, **-R**, **-r**, or **-k**). A password will not be required, so a password prompt will not be seen and a password will not be sent over the network where it can be observed. For further information on Kerberos authentication and authorization see the Secure Internet Services man page, *sis*(5) and *rlogind*(1M).

Although Kerberos authentication and authorization may apply, the Kerberos mechanism is **not** applied to the login session. All the information transferred between your host and the remote host is sent in cleartext over the network.

The terminal type specified by the current **TERM** environment variable is propagated across the network and used to set the initial value of your **TERM** environment variable on the remote host. Your terminal baud rate is also propagated to the remote host, and is required by some systems to set up the pseudo-terminal used by **rlogind** (see *rlogind*(1M)).

All echoing takes place at the remote site, so that (except for delays) the remote login is transparent.

If at any time **rlogin** is unable to read from or write to the socket connection on the remote host, the message **Connection closed** is printed on standard error and **rlogin** exits.

**Options**

**rlogin** recognizes the following options. Note that the options follow the *rhost* argument.

- 7** Set the character size to seven bits. The eighth bit of each byte sent is set to zero (space parity).
- 8** Use an eight-bit data path. This is the default HP-UX behavior.  
To use eight-bit characters, the terminal must be configured to generate either eight-bit characters with no parity, or seven bit characters with space parity. The HP-UX implementation of **rlogind** (see *rlogind*(1M)) interprets seven bit characters with even, odd, or mark parity as eight-bit non-USASCII characters. You may also need to reconfigure the remote host appropriately (see *stty*(1) and *tty*(7)). Some remote hosts may not provide the necessary support for eight-bit characters. In this case, or if it is not possible to disable parity generation by the local terminal, use the **-7** option.
- ee** Set the escape character to *e*. There is no space separating the option letter and the argument character. To start a line with the escape character, two of the escape characters must be entered. The default escape character is tilde (~). Some



characters may conflict with your terminal configuration, such as  $\hat{S}$ ,  $\hat{Q}$ , or back-space. Using one of these as the escape character may not be possible or may cause problems communicating with the remote host (see *stty*(1) and *tty*(7)).

- l** *username* This option can be used to set the user login name on the remote host to *username*. The default name is the current account name of the user invoking **rlogin**.

### Kerberos-specific Options

The default Kerberos options for the applications are set in the **krb5.conf** configuration file. Refer to the *appdefaults* Section in the *krb5.conf*(4) manpage for more information. The options **-f**, and **-F** described in the subsequent paragraphs, can be set in the **krb5.conf** file with the tag names **forward** and **forwardable** respectively. Refer to the *krb5.conf*(4) manpage for more information on the *appdefaults* Section.

The **fallback** option can be set in the **krb5.conf** file within *appdefaults* Section. If **fallback** is set to true and the kerberos authentication fails, **rlogin** will use the non-secure mode of authentication.

**Note:** Command line options override the configuration file options.

- f** This option is only applicable in a secure environment based on Kerberos V5. It can be used to forward the ticket granting ticket (TGT) to the remote system. The TGT is not forwardable from there.
- F** This option is only applicable in a secure environment based on Kerberos V5. It can be used to forward the TGT to the remote system and have it forwardable from there to another remote system. The **-f** option and **-F** option are mutually exclusive.
- k** *realm*  
This option is only applicable in a secure environment based on Kerberos V5. It can be used to obtain tickets from the remote host in the specified *realm* instead of the remote host's default realm as specified in the configuration file **krb.realms**.

### Escape Sequences

**rlogin** can be controlled with two-character escape sequences, in the form *ex*, where *e* is the escape character and *x* is a code character described below. Escape sequences are recognized only at the beginning of a line of input. The default escape character is tilde (~). It can be changed with the **-e** option.

The following escape sequences are recognized:

- ey* If *y* is NOT a code character described below, pass the escape character and *y* as characters to the remote host.
- ee* Pass the escape character as a character to the remote host.
- e.* Disconnect from the remote host.
- e!* Escape to a subshell on the local host. Use **exit** to return to the remote host.

If **rlogin** is run from a shell that supports job control (see *csch*(1), *ksh*(1), and *sh-posix*(1)), escape sequences can be used to suspend **rlogin**. The following escape sequences assume that  $\hat{Z}$  and  $\hat{Y}$  are set as the user's **susp** and **dsusp** characters, respectively (see *stty*(1) and *termio*(7)).

- e $\hat{Z}$*  Suspend the **rlogin** session and return the user to the shell that invoked **rlogin**. The **rlogin** job can be resumed with the **fg** command (see *csch*(1), *ksh*(1), and *sh-posix*(1)). *e $\hat{Z}$*  suspends both **rlogin** processes: the one transmitting user input to the remote login, and the one displaying output from the remote login.
- e $\hat{Y}$*  Suspend the **rlogin** session and return the user to the shell that invoked **rlogin**. The **rlogin** job can be resumed with the **fg** command (see *csch*(1), *ksh*(1), and *sh-posix*(1)). *e $\hat{Y}$*  suspends only the input process; output from the remote login continues to be displayed.

If you "daisy-chain" remote logins (for example, you **rlogin** from host A to host B and then **rlogin** from host B to host C) without setting unique escape characters, you can repeat the escape character until it reaches your chosen destination. For example, the first escape character, *e*, is seen as an escape character on host A; the second *e* is passed as a normal character by host A and seen as an escape character on host B; a third *e* is passed as a normal character by hosts A and B and accepted as a normal character by host C.

| **r** |

**Remote Host Name As Command**

The system administrator can arrange for more convenient access to a remote host (*rhost*) by linking **remsh** to **/usr/hosts/rhost**, allowing use of the remote host name (*rhost*) as a command (see *remsh*(1)). For example, if **remotehost** is the name of a remote host and **/usr/hosts/remotehost** is linked to **remsh**, and if **/usr/hosts** is in your search path, the command:

```
remotehost
```

is equivalent to:

```
rlogin remotehost
```

**RETURN VALUE**

**rlogin** sends an error message to standard error and returns a nonzero value if an error occurs before the connection to the remote host is completed. Otherwise, it returns a zero.

**DIAGNOSTICS**

Diagnostics can occur from both the local and remote hosts. Those diagnostics that occur on the local host before the connection is completely established are written to standard error. Once the connection is established, any error messages from the remote host are written to standard output, like any other data.

**Error! could not retrieve authentication type.**

**Please notify sys admin.**

There are two authentication mechanisms used by **rlogin**. One authentication mechanism is based on Kerberos and the other is not. The type of authentication mechanism is obtained from a system file which is updated by **inetsvcs\_sec** (see *inetsvcs\_sec*(1M)). If the system file does not contain known authentication types, the above error is displayed.

**login/tcp: Unknown service**

**rlogin** was unable to find the login service listed in the **/etc/services** database file.

**There is no entry for you (user ID *username*) in /etc/passwd**

**rlogin** was unable to find your user ID in the password file.

*Next Step:* Contact your system administrator.

*system call:*...

An error occurred when **rlogin** attempted the indicated system call. See the appropriate manual entry for information about the error.

**R****EXAMPLES**

Log in as the same user on the remote host **remote**:

```
rlogin remote
```

Set the escape character to a **!**, use a seven-bit data connection, and attempt a login as user **guest** on host **remhost**:

```
rlogin remhost -e! -7 -l guest
```

Assuming that your system administrator has set up the links in **/usr/hosts**, the following is equivalent to the previous command:

```
remhost -e! -7 -l guest
```

**WARNINGS**

For security purposes, the **/etc/hosts.equiv** and **.rhosts** files should exist, even if they are empty. These files should be readable and writable only by the owner. See *hosts.equiv*(4) for more information.

Note that all the information, including any passwords asked for, is passed unencrypted between the two hosts. In a Kerberos V5 Network Authentication environment, a password is not transmitted across the network, so it will be protected.

**rlogin** is unable to transmit the Break key as an interrupt signal to the remote system, regardless of whether the user has set **stty brkint** on the local system. The key assigned to **SIGINT** with the command **stty intr c** should be used instead (see *stty*(1)).

**AUTHOR**

**rlogin** was developed by the University of California, Berkeley.

**FILES**

|                         |                                         |
|-------------------------|-----------------------------------------|
| <b>\$HOME/.rhosts</b>   | User's private equivalence list         |
| <b>/etc/hosts.equiv</b> | List of equivalent hosts                |
| <b>/usr/hosts/*</b>     | For <i>rhost</i> version of the command |

**SEE ALSO**

csh(1), ksh(1), login(1), remsh(1), sh(1), sh-posix(1), stty(1), telnet(1), rlogind(1M), inetsvcs\_sec(1M), hosts(4), hosts.equiv(4), inetd.conf(4), krb5.conf(4), services(4), sis(5), termio(7), tty(7).

**NAME**

rm - remove files or directories

**SYNOPSIS**

```
rm [-f|-i] [-Rr] file ...
```

**DESCRIPTION**

The **rm** command removes the entries for one or more files from a directory. If an entry was the last link to the file, the file is destroyed. Removal of a file requires write and search (execute) permission in its directory, but no permissions on the file itself. However, if the sticky bit is set on the directory containing the file, only the owner of the file, the owner of the directory, or a user having appropriate privileges can remove the file.

If a user does not have write permission for a file to be removed and standard input is a terminal, a prompt containing the file name and its permissions is printed requesting that the removal of the file be confirmed (see Access Control Lists below). A line is then read from standard input. If that line begins with **y** the file is deleted; otherwise, the file remains. No questions are asked when the **-f** option is given or if standard input is not a terminal.

If *file* is of type directory, and the **-f** option is not specified, and either the permissions of *file* do not permit writing and standard input is a terminal or the **-i** option is specified, **rm** writes a prompt to standard error and reads a line from standard input. If the response does not begin with **y**, it does nothing more with the current file and goes on to any remaining files.

If *file* is a symbolic link, then only the symbolic link is removed. The file or directory pointed to by the symbolic link is not affected. If any of the intermediate path components of *file* happens to be a symbolic link, then **rm** follows the symbolic link and removes the *file*.

**Options**

**rm** recognizes the following options:

- f** Force each file or directory to be removed without prompting for confirmation, regardless of the permissions of the entry. This option also suppresses diagnostic messages regarding nonexistent operands.

This option does not suppress any diagnostic messages other than those regarding nonexistent operands. To suppress all error message and interactive prompts, the **-f** option should be used while redirecting standard error output to **/dev/null**.

This option ignores any previous occurrence of the **-i** option.

- i** Write a prompt to standard error requesting confirmation before removing each entry.

This option ignores any previous occurrence of the **-f** option.

- R** For each argument that is a directory, this option causes **rm** to recursively delete the entire contents of that directory before removing the directory itself. When used in conjunction with the **-i** option, **rm** asks whether to examine each directory before interactively removing files in that directory and again afterward to confirm removing the directory itself.

The **-R** option will descend to arbitrary depths in a file hierarchy and will not fail due to path length limitations unless the length of file name, **file** specified by the user exceeds system limitations.

- r** Equivalent to **-R**.

**Access Control Lists**

If a file has optional ACL entries, **rm** displays a plus sign (+) after the file's permissions. The permissions shown summarize the file's **st\_mode** value returned by **stat()** (see *stat(2)*). See also *acl(5)*.

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** provides a default value for the internationalization variables that are unset or null. If **LANG** is unset or null, the default value of "C" (see *lang(5)*) is used. If any of the internationalization variables contains an invalid setting, **rm** will behave as if all internationalization variables are set to "C". See *environ(5)*.

**LC\_ALL** If set to a non-empty string value, overrides the values of all the other internationalization variables.

**LC\_CTYPE** determines the interpretation of file names as single and/or multi-byte characters, the classification of characters as printable, and the characters matched by character class expressions in regular expressions.

**LC\_MESSAGES** determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

**NLSPATH** determines the location of message catalogues for the processing of **LC\_MESSAGES**.

### International Code Set Support

Single- and multibyte character code sets are supported.

### DIAGNOSTICS

Generally self-explanatory. Note that the **-f** option does not suppress all diagnostic messages.

It is forbidden to remove the file **..**, in order to avoid the consequences of using a command such as:

```
rm -r .*
```

If a designated file is a directory, an error comment is printed unless the **-R** or **-r** option is used.

### RETURN VALUE

**rm** exits with one of the following values:

- 0 If the **-f** option is not specified, 0 is returned only if all the named directory entries (the arguments specified in the **rm** command) are removed.
- If the **-f** option is specified, then all the existing named directory entries are removed. If any of the named directory entries are non-existent, **rm** still returns a zero.
- >0 An error occurred.

### EXAMPLES

Remove files with a prompt for verification:

```
rm -i file1 file2
```

Remove all the files in a directory:

```
rm -i mydirectory/*
```

Note that the previous command removes files only, and does not remove any directories in **mydirectory**.

Remove a file in the current directory whose name starts with **-** or **\*** or some other character that is special to the shell:

```
rm ./-filename
rm *filename
etc.
```

Remove a file in the current directory whose name starts with some strange (usually nonprinting, invisible) character or perhaps has spaces at the beginning or end of the filename, prompting for confirmation:

```
rm -i *filename*
```

If **\*filename\*** is not unique in the directory, enter **n** when each of the other files is prompted.

A powerful and dangerous command to remove a directory is:

```
rm -fR directoryname
```

or

```
rm -Rf directoryname
```

which removes all files and directories from **directoryname** without any prompting for verification to remove the files or the directories. This command should only be used when you are absolutely certain that all the files and directories in **directoryname** as well as **directoryname** itself are to be removed.

### DEPENDENCIES

**NFS**

**rm** does not display a plus sign (+) to indicate the existence of optional access control list entries when asking for confirmation before removing a networked file.

**SEE ALSO**

**rmdir**(1), **unlink**(2), **acl**(5).

**STANDARDS CONFORMANCE**

**rm**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| r |**

**NAME**

rmdel - remove a delta from an SCCS file

**SYNOPSIS**

**rmdel** -r *SID file* ...

**DESCRIPTION**

The **rmdel** command removes the delta specified by the *SID* from each named SCCS file. The delta to be removed must be the newest (most recent) delta in its branch in the delta chain of each named SCCS file. In addition, the *SID* specified must *not* be that of a version being edited for the purpose of making a delta (i.e., if a *p-file* (see *get*(1)) exists for the named SCCS file, the *SID* specified must *not* appear in any entry of the *p-file*).

If a directory is named, **rmdel** behaves as though each file in the directory were specified as a named file, except that non-SCCS files (last component of the path name does not begin with **s.**) and unreadable files are silently ignored. If a name of **-** is given, the standard input is read; each line of the standard input is taken to be the name of an SCCS file or directory to be processed; non-SCCS files and unreadable files are silently ignored. When **--** is specified on the command line, all following arguments are treated as file names.

The permissions to remove a delta are either (1) if you make a delta you can remove it; or (2) if you own the file and directory you can remove a delta.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the locale for the interpretation of text as single-byte and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

**LC\_MESSAGES** also determines the local language equivalent of the affirmative string ("yes").

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **rmdel** behaves as if all internationalization variables are set to "C". See *environ*(5).

**International Code Set Support**

Single-byte and multi-byte character code sets are supported.

**DIAGNOSTICS**

Use *sccshelp*(1) for explanations.

**FILES**

**x.file** See *delta*(1).

**z.file** See *delta*(1).

**SEE ALSO**

*delta*(1), *get*(1), *sccshelp*(1), *prs*(1), *sccsfile*(4).

**STANDARDS CONFORMANCE**

**rmdel**: SVID2, SVID3, XPG2, XPG3, XPG4

| **r** |

**NAME**

rmdir - remove directories

**SYNOPSIS**

**rmdir** [-f] [-i] [-p] *dir* ...

**DESCRIPTION**

**rmdir** removes the directory entry for each *dir* operand that refers to an empty directory.

Directories are removed in the order specified. Consequently, if a directory and a subdirectory of that directory are both specified as arguments, the subdirectory must be specified before the parent directory so that the parent directory will be empty when **rmdir** tries to remove it. Removal of a directory requires write and search (execute) permission in its parent directory, but no permissions on the directory itself; but if the sticky bit is set on the parent directory, only the owner of the directory, the owner of the parent directory, or a user having appropriate privileges can remove the directory.

**Options**

**rmdir** recognizes the following options:

- f** Force each directory to be removed without prompting for confirmation, regardless of the presence of the **-i** option. This option also suppresses diagnostic messages regarding non-existent operands.

This option does not suppress any diagnostic messages other than those regarding non-existent operands. To suppress all error message and interactive prompts, the **-f** option should be used while redirecting the standard error output to **/dev/null**.

This option ignores any previous occurrence of the **-i** option.

- i** Write a prompt to the standard error output requesting confirmation before removing each directory.

This option ignores any previous occurrence of the **-f** option.

- p** Path removal. If, after removing a directory with more than one pathname component, the parent directory of that directory is now empty, **rmdir** removes the empty parent directory. This continues until **rmdir** encounters a non-empty parent directory, or until all components of the original pathname have been removed.

When used in conjunction with the **-i** option, **rmdir** asks whether to remove each directory component of a path.

**R****EXTERNAL INFLUENCES****Environment Variables**

**LANG** provides a default value for the internationalization variables that are unset or null. If **LANG** is unset or null, the default value of "C" (see *lang(5)*) is used. If any of the internationalization variables contains an invalid setting, **rmdir** will behave as if all internationalization variables are set to "C". See *environ(5)*.

**LC\_ALL** If set to a non-empty string value, overrides the values of all the other internationalization variables.

**LC\_CTYPE** determines the interpretation of *dir* names as single and/or multi-byte characters, the classification of characters as printable, and the characters matched by character class expressions in regular expressions.

**LC\_MESSAGES** determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

**NLS\_PATH** determines the location of message catalogues for the processing of **LC\_MESSAGES**.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**DIAGNOSTICS**

Generally self-explanatory. Note that the **-f** option does not suppress all diagnostic messages.



**EXAMPLES**

To remove directories with a prompt for verification:

```
rmdir -i directories
```

To remove as much as possible of a path, type:

```
rmdir -p component1 /component2 /dir
```

**SEE ALSO**

*rm(1)*, *rmdir(2)*, *stat(2)*.

**STANDARDS CONFORMANCE**

*rmdir*: SVID2, XPG2, XPG3, XPG4

**NAME**

rmnl - remove extra new-line characters from file

**SYNOPSIS**

**rmnl**

**DESCRIPTION**

**rmnl** removes all blank lines from a file (except at beginning of file as explained below), and is useful for removing excess white space from files for display on a CRT terminal. Groups of two or more successive \n (new-line) characters are reduced to a single \n character, effectively eliminating all blank lines in the file *except* that one or more blank lines at the beginning of a file remain as a single blank line.

To remove redundant blank lines rather than all blank lines, use *ssp(1)*.

To remove all blank lines from a file including beginning of file, use **rmnl** piped to **ssp**, or **ssp** piped to **rmnl**.

**EXTERNAL INFLUENCES****International Code Set Support**

Single- and multi-byte character code sets are supported.

**SEE ALSO**

man(1), ssp(1).

| **r** |

**NAME**

rncd-confgen - rncd key generation tool

**SYNOPSIS**

```
rncd-confgen [-a] [-b keysize] [-c keyfile] [-h] [-k keyname] [-p port] [-r randomfile]
[-s address] [-t chrootdir] [-u user]
```

**DESCRIPTION**

**rncd-confgen** can be used to generate **rncd.conf**, the configuration file for **rncd**. Alternatively, it can be run with the **-a** option to set up a **rncd.key** file and avoid the need for a **rncd.conf** file and a **controls** statement altogether.

**Options**

- a** This option is used to configure **rncd** automatically. This creates a file **rncd.key** in **/etc** (or whatever **sysconfdir** was specified when BIND was built) that is read by both **rncd** and **named** on startup. The **rncd.key** file defines a default command channel and authentication key allowing **rncd** to communicate with **named** with no further configuration. Running **rncd-confgen -a** allows BIND 9 and **rncd** to be used as drop-in replacements for BIND 8 and **ndc**, with no changes to the existing BIND 8 **named.conf** file.
- b *keysize*** This option is used to specify the size of the authentication key in bits. The value must range between 1 and 512 bits. Default is 128 bits.
- c *keyfile*** This option is used with the **-a** option to specify an alternate location for **rncd.key**.
- h** This option is used to print a short summary of the options and arguments to **rncd-confgen**.
- k *keyname*** This option is used to specify the key name of the **rncd** authentication key. This must be a valid domain name. Default is **rncd-key**.
- p *port*** This option is used to specify the command channel port where **named** listens for connections from **rncd**. Default is 953.
- r *randomfile*** This option is used to specify a source file of random data for generating the authorization. If the operating system does not provide a **/dev/random** or equivalent device, the default source of randomness is keyboard input. *randomdev* specifies the name of a character device or a file containing random data to be used instead of the default. The special value **keyboard** indicates that keyboard input needs to be used.
- s *address*** This option is used to specify the IP address where **named** listens for command channel connections from **rncd**. Default is the loopback address **127.0.0.1**.
- t *chrootdir*** This option is used with the **-a** option to specify a directory where **named** will run **chrooted**. An additional copy of the **rncd.key** will be written relative to this directory so that it will be found by the **chrooted named**.
- u *user*** This option is used with the **-a** option to set the owner of the **rncd.key** file generated. If **-t** is also specified, only the file in the chroot area has its owner changed.

**EXAMPLES**

To allow **rncd** to be used with no manual configuration, run:

```
rncd-confgen -a
```

To print a sample **rncd.conf** file and corresponding **controls** and **key** statements to be manually inserted into **named.conf**, run:

```
rncd-confgen
```

**AUTHOR**

**rndc-confgen** was developed by the Hewlett-Packard Company.

**SEE ALSO**

**rndc(1)**, **named(1M)**, **rndc.conf(4)**, and *BIND 9 Administrator Reference Manual*.

**| r |**

**NAME**

rncd - name server control utility

**SYNOPSIS**

**rncd** [-c *config-file*] [-k *keyname*] [-m] [-p *port#*] [-s *server*] [-v] [-y *key\_id*] *command* [*command...*]

**DESCRIPTION**

This command allows the system administrator to control the operation of a name server. If **rncd** is invoked without any command line options or arguments, it prints a short summary of the supported commands and the available options and their arguments.

**rncd** communicates with the name server over a TCP connection, sending commands authenticated with digital signatures. In the current versions of **rncd**, the only supported encryption algorithm is HMAC-MD5, which uses a shared secret on each end of the connection. This provides TSIG-style authentication for the command request and the name server's response. All commands sent over the channel must be signed by a *key\_id* known to the server.

**rncd** reads its default configuration file, **/etc/rncd.conf** to determine how to contact the name server and decide what algorithm and keys it should use.

**Options**

- |                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c</b> <i>config-file</i> | This option can be used to specify an alternate configuration file. The default configuration file is <b>/etc/rncd.conf</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>-k</b> <i>keyname</i>     | This option can be used to specify the keyname of the <b>rncd</b> authentication key. This must be a valid domain name. Default is <b>rncd-key</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>-m</b>                    | Provides debugging information to the developers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>-p</b> <i>port#</i>       | This option specifies that <b>rncd</b> should send commands to TCP port number <i>port#</i> on the system running the name server instead of BIND 9.1.3's default control channel port, 953.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>-s</b> <i>server</i>      | This option is used to specify the server on which this command is run. <i>server</i> is the name or address of the server which matches a <b>server</b> statement in the configuration file for <b>rncd</b> . If no <i>server</i> is supplied on the command line, the host named by the <i>default-server</i> clause in the <b>options</b> statement of the configuration file, <b>rncd.conf</b> will be used.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>-v</b>                    | Provides debugging information and is primarily of interest only to the BIND 9 developers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-y</b> <i>key_id</i>      | This option identifies the <i>key_id</i> to use from the configuration file. <i>key_id</i> must be known to <b>named</b> with the same algorithm and secret string for control message validation to succeed. If the <b>-y</b> option is not specified, <b>rncd</b> will first look for a <i>key</i> clause in the <i>server</i> statement of <b>rncd.conf</b> file. If no <i>server</i> statement is present for that host, then the <i>default-key</i> clause of the <b>options</b> statement of the configuration file, <b>rncd.conf</b> will be used.                                                                                                                                                                                                                                                                                                                      |
| <i>command</i>               | <i>command</i> is one of the following:<br><b>reload</b> Reload configuration file and zones.<br><b>reload zone</b> Reload the given zone.<br><b>refresh zone</b> Schedule zone maintenance for the given zone.<br><b>stats</b> Write server statistics to the statistics file as specified by the <b>statistics-file</b> directive of the <b>options</b> statement in the <b>named.conf</b> configuration file. If the <b>statistics-file</b> directive is not specified, the statistics is dumped to the <b>named.stats</b> file in the directory specified by the <b>directory</b> directive of the <b>options</b> statement in the <b>named.conf</b> configuration file.<br><b>querylog</b> Toggle query logging.<br><b>dumpdb</b> Dump the current contents of the cache into the file specified by the <b>dump-file</b> directive of the <b>options</b> statement in the |

| r |

|                            |                                                                                                                                                                                                                                                                                                 |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            | configuration file, <b>named.conf</b> . If the <b>dump-file</b> directive is not specified, the cache data is dumped to the <b>named_dump.db</b> file in the directory specified by the <b>directory</b> directive of the <b>options</b> statement in the <b>named.conf</b> configuration file. |
| <b>stop</b>                | Stop the server. Before stopping the server, any recent changes made through dynamic update or IXFR will be saved to the master files of the updated zones.                                                                                                                                     |
| <b>halt</b>                | Halt the server immediately. Any recent changes made through dynamic update or IXFR will not be saved to the master files. They are rolled forward from the journal files when the server is restarted.                                                                                         |
| <b>reconfig</b>            | Reload configuration file and new zones only.                                                                                                                                                                                                                                                   |
| <b>trace</b>               | Increment debugging level by 1.                                                                                                                                                                                                                                                                 |
| <b>trace level</b>         | Change the debugging level.                                                                                                                                                                                                                                                                     |
| <b>notrace</b>             | Set debugging level to 1.                                                                                                                                                                                                                                                                       |
| <b>flush</b>               | Flush all the server's caches.                                                                                                                                                                                                                                                                  |
| <b>flush</b> <i>[view]</i> | Flush the server's cache for a view.                                                                                                                                                                                                                                                            |
| <b>status</b>              | Display the status of the server.                                                                                                                                                                                                                                                               |

#### LIMITATIONS

Note that the configuration file for **rncd** contains shared secrets which are used to send authenticated control commands to name servers. It should therefore not have general read or write access.

There is currently no way to provide the shared secret for a *key\_id* without using the configuration file.

#### AUTHOR

**rncd** was developed by ISC (Internet Software Consortium).

#### SEE ALSO

dnssec-keygen(1) named(1M), rncd.conf(4), RFC2845.

**NAME**

rpcgen - an RPC protocol compiler

**SYNOPSIS**

rpcgen *infile*

rpcgen [ **-a** ] [ **-b** ] [ **-C** ] [ **-D** *name* [ = *value* ] ] [ **-i** *size* ] [ **-I** [ **-K** *seconds* ] ]  
 [ **-L** ] [ **-M** ] [ **-N** ] [ **-T** ] [ **-u** ] [ **-Y** *pathname* ] *infile*

rpcgen [ **-c** | **-h** | **-l** | **-m** | **-t** | **-Sc** | **-Ss** | **-Sm** ] [ **-o** *outfile* ] [ *infile* ]

rpcgen [ **-s** *nettype* ] [ **-u** ] [ **-o** *outfile* ] [ *infile* ]

rpcgen [ **-n** *netid* ] [ **-u** ] [ **-o** *outfile* ] [ *infile* ]

**DESCRIPTION**

**rpcgen** is a tool that generates C code to implement an RPC protocol. The input to **rpcgen** is a language similar to C known as RPC Language (Remote Procedure Call Language).

**rpcgen** is normally used as in the first synopsis where it takes an input file and generates three output files. If the *infile* is named **proto.x**, then **rpcgen** generates a header in **proto.h**, XDR routines in **proto\_xdr.c**, server-side stubs in **proto\_svc.c**, and client-side stubs in **proto\_clnt.c**. With the **-T** option, it also generates the RPC dispatch table in **proto\_tbl.i**.

**rpcgen** can also generate sample client and server files that can be customized to suit a particular application. The **-Sc**, **-Ss** and **-Sm** options generate sample client, server and makefile, respectively. The **-a** option generates all files, including sample files. If the *infile* is **proto.x**, then the client side sample file is written to **proto\_client.c**, the server side sample file to **proto\_server.c** and the sample makefile to **makefile.proto**.

The server created can be started both by the port monitors (for example, **inetd** or **listen**) or by itself. When it is started by a port monitor, it creates servers only for the transport for which the file descriptor 0 was passed. The name of the transport must be specified by setting up the environment variable **PM\_TRANSPORT**. When the server generated by **rpcgen** is executed, it creates server handles for all the transports specified in **NETPATH** environment variable, or if it is unset, it creates server handles for all the visible transports from **/etc/netconfig** file. Note: the transports are chosen at run time and not at compile time. When the server is self-started, it backgrounds itself by default. A special define symbol **RPC\_SVC\_FG** can be used to run the server process in foreground.

The second synopsis provides special features which allow for the creation of more sophisticated RPC servers. These features include support for user provided **#defines** and RPC dispatch tables. The entries in the RPC dispatch table contain:

- pointers to the service routine corresponding to that procedure,
- a pointer to the input and output arguments
- the size of these routines

A server can use the dispatch table to check authorization and then to execute the service routine; a client library may use it to deal with the details of storage management and XDR data conversion.

The other three synopses shown above are used when one does not want to generate all the output files, but only a particular one. See the **EXAMPLES** section below for examples of **rpcgen** usage. When **rpcgen** is executed with the **-s** option, it creates servers for that particular class of transports. When executed with the **-n** option, it creates a server for the transport specified by *netid*. If *infile* is not specified, **rpcgen** accepts the standard input.

The C preprocessor, **cc -E** is run on the input file before it is actually interpreted by **rpcgen**. For each type of output file, **rpcgen** defines a special preprocessor symbol for use by the **rpcgen** programmer:

|                 |                                                 |
|-----------------|-------------------------------------------------|
| <b>RPC_HDR</b>  | defined when compiling into headers             |
| <b>RPC_XDR</b>  | defined when compiling into XDR routines        |
| <b>RPC_SVC</b>  | defined when compiling into server-side stubs   |
| <b>RPC_CLNT</b> | defined when compiling into client-side stubs   |
| <b>RPC_TBL</b>  | defined when compiling into RPC dispatch tables |

Any line beginning with “%” is passed directly into the output file, uninterpreted by **rpcgen**. To specify the path name of the C preprocessor use **-Y** flag.

For every data type referred to in *infile*, **rpcgen** assumes that there exists a routine with the string **xdr\_** prepended to the name of the data type. If this routine does not exist in the RPC/XDR library, it

| **r** |

must be provided. Providing an undefined data type allows customization of XDR routines.

### Options

- a** Generate all files, including sample files.
- b** Backward compatibility mode. Generate transport specific RPC code for older versions of the operating system.
- c** Compile into XDR routines.
- C** Generate header and stub files which can be used with ANSI C compilers. Headers generated with this flag can also be used with C++ programs.
- Dname[=value]** Define a symbol *name*. Equivalent to the **#define** directive in the source. If no *value* is given, *value* is defined as **1**. This option may be specified more than once.
- h** Compile into **C** data-definitions (a header). **-T** option can be used in conjunction to produce a header which supports RPC dispatch tables.
- i size** Size at which to start generating inline code. This option is useful for optimization. The default size is 5.
- I** Compile support for *inetd*(1M) in the server side stubs. Such servers can be self-started or can be started by *inetd*. When the server is self-started, it backgrounds itself by default. A special define symbol **RPC\_SVC\_FG** can be used to run the server process in foreground, or the user may simply compile without the **-I** option.  
  
If there are no pending client requests, the *inetd* servers exit after 120 seconds (default). The default can be changed with the **-K** option. All of the error messages for *inetd* servers are always logged with *syslog*(3C).  
  
Note: This option is supported for backward compatibility only. It should always be used in conjunction with the **-b** option which generates backward compatibility code. By default (i.e., when **-b** is not specified), **rpcgen** generates servers that can be invoked through portmonitors.
- K seconds** By default, services created using **rpcgen** and invoked through port monitors wait **120** seconds after servicing a request before exiting. That interval can be changed using the **-K** flag. To create a server that exits immediately upon servicing a request, use **-K 0**. To create a server that never exits, the appropriate argument is **-K -1**.  
  
When monitoring for a server, some portmonitors, like *listen*, *always* spawn a new process in response to a service request. If it is known that a server will be used with such a monitor, the server should exit immediately on completion. For such servers, **rpcgen** should be used with **-K 0**.
- l** Compile into client-side stubs.
- L** When the servers are started in foreground, use *syslog*(3C) to log the server errors instead of printing them on the standard error.
- m** Compile into server-side stubs, but do not generate a main routine. This option is useful for doing callback-routines and for users who need to write their own main routine to do initialization.
- M** Generate multithread-safe stubs for passing arguments and results between **rpcgen** generated code and user written code. This option is useful for users who want to use threads in their code.
- N** This option allows procedures to have multiple arguments. It also uses the style of parameter passing that closely resembles C. So, when passing an argument to a remote procedure, you do not have to pass a pointer to the argument, but can pass the argument itself. This behavior is different from the old style of **rpcgen** generated code. To maintain backward compatibility, this option is not the default.
- n netid** Compile into server-side stubs for the transport specified by *netid*. There should be an entry for *netid* in the **netconfig** database. This option may be specified more than once, so as to compile a server that serves multiple transports.

| **r** |



- o** *outfile* Specify the name of the output file. If none is specified, standard output is used (**-c**, **-h**, **-l**, **-m**, **-n**, **-s**, **-Sc**, **-Sm**, **-Ss**, and **-t** modes only).
- s** *nettype* Compile into server-side stubs for all the transports belonging to the class *nettype*. The supported classes are **netpath**, **visible**, **circuit\_n**, **circuit\_v**, **datagram\_n**, **datagram\_v**, **tcp**, and **udp** (see *rpc(3N)* for the meanings associated with these classes). This option may be specified more than once. Note: the transports are chosen at run time and not at compile time.
- Sc** Generate sample client code that uses remote procedure calls.
- Sm** Generate a sample Makefile which can be used for compiling the application.
- Ss** Generate sample server code that uses remote procedure calls.
- t** Compile into RPC dispatch table.
- T** Generate the code to support RPC dispatch tables.  
  
The options **-c**, **-h**, **-l**, **-m**, **-s**, **-Sc**, **-Sm**, **-Ss**, and **-t** are used exclusively to generate a particular type of file, while the options **-D** and **-T** are global and can be used with the other options.
- u** When the server-side stub is produced, additional code to handle signals is generated. On reception of a signal, this signal handler code unmaps the server program from the port mapper before the server terminates. This code is added only if a **main()** routine is produced in the server-side stub. The **-u** option must not be specified with the **-c**, **-h**, **-l**, **-m**, **-Sc**, **-Sm**, **-Ss** options. The following signals are trapped: **SIGHUP**, **SIGINT**, **SIGQUIT**, and **SIGTERM**.
- Y** *pathname* Give the name of the directory where **rpcgen** will start looking for the C-preprocessor.

**EXAMPLES**

The following example:

```
example% rpcgen -T prot.x
```

generates all the five files: **prot.h**, **prot\_clnt.c**, **prot\_svc.c**, **prot\_xdr.c** and **prot\_tbl.i**.

The following example sends the C data-definitions (header) to the standard output.

```
example% rpcgen -h prot.x
```

To send the test version of the **-DTEST**, server side stubs for all the transport belonging to the class **datagram\_n** to standard output, use:

```
example% rpcgen -s datagram_n -DTEST prot.x
```

To create the server side stubs for the transport indicated by *netid* **tcp**, use:

```
example% rpcgen -n tcp -o prot_svc.c prot.x
```

**AUTHOR**

**rpcgen** was developed by Sun Microsystems, Inc.

**SEE ALSO**

*cc\_bundled(1)*, *inetd(1M)*, *syslog(3C)*, *rpc(3N)*, *rpc\_svc\_calls(3N)*.

| **r** |

**NAME**

rtprio - execute process with real-time priority

**SYNOPSIS**

```
rtprio priority command [arguments]
```

```
rtprio priority -pid
```

```
rtprio -t command [arguments]
```

```
rtprio -t -pid
```

**DESCRIPTION**

**rtprio** executes *command* with a real-time priority, or changes the real-time priority of currently executing process *pid*. Real-time priorities range from zero (highest) to 127 (lowest). Real-time processes are not subject to priority degradation, and are all of greater (scheduling) importance than non-real-time processes. See *rtprio(2)* for more details.

If **-t** is specified instead of a real-time *priority*, **rtprio** executes *command* with a timeshare (non-real-time) priority, or changes the currently executing process *pid* from a possibly real-time priority to a timeshare priority. The former is useful to spawn a timeshare priority command from a real-time priority shell.

If **-t** is not specified, *command* is not scheduled, or *pid*'s real-time priority is not changed, if the user is not a member of a group having **PRIV\_RTPRIO** access and is not the user with appropriate privileges. When changing the real-time priority of a currently executing process, the effective user ID of the calling process must be the user with appropriate privileges, or the real or effective user ID must match the real or saved user ID of the process to be modified.

**RETURN VALUE**

**rtprio** returns exit status 0 if *command* is successfully scheduled or if *pid*'s real-time priority is successfully changed, 1 if *command* is not executable or *pid* does not exist, and 2 if *command* (*pid*) lacks real-time capability, or the invoker's effective user ID is not a user who has appropriate privileges, or the real or effective user or the real or effective user ID does not match the real or saved user ID of the process being changed.

**EXAMPLES**

Execute file **a.out** at a real-time priority of 100:

```
rtprio 100 a.out
```

Set the currently running process pid 24217 to a real-time priority of 40:

```
rtprio 40 -24217
```

**AUTHOR**

**rtprio** was developed by HP.

**SEE ALSO**

setprivgrp(1M), getprivgrp(2), rtprio(2).

| **r** |

**NAME**

rtsched - execute process with real-time priority

**SYNOPSIS**

**rtsched** -s *scheduler* -p *priority* *command* [*arguments* ]  
**rtsched** [ -s *scheduler* ] -p *priority* -P *pid*

**DESCRIPTION**

**Rtsched** executes *command* with POSIX or HP-UX real-time priority, or changes the real-time priority of currently executing process *pid*.

All POSIX real-time priority processes are of greater scheduling importance than processes with HP-UX real-time or HP-UX timeshare priority. All HP-UX real-time priority processes are of greater scheduling importance than HP-UX timeshare priority processes, but are of lesser importance than POSIX real-time processes. Neither POSIX nor HP-UX real-time processes are subject to degradation. POSIX real-time processes may be scheduled with one of three different POSIX real-time schedulers: SCHED\_FIFO, SCHED\_RR, or SCHED\_RR2. See *rtsched*(2) for details.

**Rtsched** is a superset of **rtprio**. See *rtprio*(1).

**Options**

-s *scheduler* Specify the desired scheduler:

POSIX real-time schedulers: SCHED\_FIFO  
 SCHED\_RR  
 SCHED\_RR2

HP-UX real-time scheduler: SCHED\_RTPRIO

HP-UX timeshare scheduler: SCHED\_HPUX  
 SCHED\_NOAGE

-p *priority* Specify priority range; any integer within the inclusive priority range of the corresponding scheduler. -p *priority* is required for all schedulers except SCHED\_HPUX. If scheduler is SCHED\_HPUX, the *priority* argument is ignored. The default *priority* range of each scheduler is as follows:

| scheduler    | highest priority | lowest priority |
|--------------|------------------|-----------------|
| SCHED_FIFO   | 31               | 0               |
| SCHED_RR     | 31               | 0               |
| SCHED_RR2    | 31               | 0               |
| SCHED_RTPRIO | 0                | 127             |
| SCHED_NOAGE  | 178              | 255             |
| SCHED_HPUX   | N/A              | N/A             |

*Note:* Higher numerical values for the *priority* represent higher priorities under POSIX real-time schedulers, whereas lower numerical values for the *priority* represent higher priorities under HP-UX real-time and timeshare schedulers.

-P Specify an already executing process ID (*pid*).

*Command* is not scheduled, or *pid*'s real-time priority is not changed, if the user is not a member of a group having **PRIV\_RTSCHED** access and is not the user with appropriate privileges. When changing the real-time priority of a currently executing process, the effective user ID of the calling process must be the user with appropriate privileges, or the real or effective user ID must match the real or saved user ID of the process to be modified.

In presence of processor sets (See *pset\_create*(2) for details), the application execution is restricted to processors in the application's processor set. The threads in different processor sets do not compete with one another for processors based on their scheduling policy and priority values. The scheduler looks only at threads assigned to a processor's processor set to choose the next thread to run.

**RETURN VALUE**

**rtsched** returns exit status:



- 0** if *command* is successfully scheduled or if *pid*'s real-time priority is successfully changed;
- 1** if *command* is not executable, *pid* does not exist, or *priority* is not within the priority range for the corresponding scheduler;
- 2** if *command* (*pid*) lacks real-time capability, or the invoker's effective user ID is not a user who has appropriate privileges, or the real or effective user or the real or effective user ID does not match the real or saved user ID of the process being changed; or
- 5** if rtsched encountered an internal error or if rtsched is not supported by this release.

**EXAMPLES**

Execute file **a.out** with SCHED\_FIFO at a priority of 10:

```
rtsched -s SCHED_FIFO -p 10 a.out
```

Execute file **a.out** with SCHED\_RTPRIO at a priority of 127 (this is synonymous to **rtprio 127 a.out**):

```
rtsched -s SCHED_RTPRIO -p 127 a.out
```

Execute file **a.out** with the SCHED\_HPUX scheduler:

```
rtsched -s SCHED_HPUX a.out
```

This is useful to spawn a timeshare priority command from a real-time priority shell.

Set the currently running process pid 24217 to execute with SCHED\_RR2 at a priority of 20:

```
rtsched -s SCHED_RR2 -p 20 -P 24217
```

Now change its priority to 10 using the same scheduler:

```
rtsched -p 10 -P 24217
```

**WARNINGS**

*Priority* values used by **rtsched** may differ from those used by other commands. For example, *ps(1)* displays the internal representation of priority values.

**AUTHOR**

**rtsched** was developed by HP.

**SEE ALSO**

*rtprio(1)*, *setprivgrp(1M)*, *getprivgrp(2)*, *pset\_create(2)*, *rtprio(2)*, *rtsched(2)*.

**NAME**

rup - show host status of local machines (RPC version)

**SYNOPSIS**

**rup** [-h] [-l] [-t] [*host* ...]

**DESCRIPTION**

**rup** gives a status similar to **uptime** for remote machines. It broadcasts on the local network and displays the responses it receives. Though the listing is normally in the order responses are received, the order can be changed by using command-line options. The broadcast process takes about two minutes.

When *host* arguments are given, instead of broadcasting, **rup** only queries the list of specified hosts.

A remote host only responds if it is running the **rstatd** daemon (see *rstatd* (1M)).

**Options**

**rup** recognizes the following command-line options:

- h Sort the display alphabetically by host name.
- l Sort the display by load average.
- t Sort the display by up time.

**WARNINGS**

Broadcasting does not work through gateways; therefore, **rup** does not report information about machines that are reachable only through gateways.

**DIAGNOSTICS**

The exit code of **rup** is the number of errors (eg. bad host names) encountered; zero for success.

**AUTHOR**

**rup** was developed by Sun Microsystems, Inc.

**FILES**

**/etc/inetd.conf**

**SEE ALSO**

ruptime(1), inetd(1M), rstatd(1M), services(4).

**NAME**

ruptime - show status of local machines

**SYNOPSIS**

**ruptime** [-a] [-r] [-l] [-t] [-u]

**DESCRIPTION**

**ruptime** outputs a status line for each machine on the local network that is running the **rwwho** daemon. **ruptime**'s status lines are formed from packets broadcast once every 3 minutes between **rwwho** daemons (see *rwwho(1M)*) on each host on the network. Each status line has a field for the name of the machine, the status of the machine (up or down), how long the machine has been up or down, the number of users logged into the machine, and the 1-, 5- and 15-minute load averages for the machine when the packet was sent.

The status of the machine is reported as "up" unless no report has been received from the machine for 11 minutes or more.

The length of time that the machine has been up is shown as:

*days+hours:minutes*

Load averages are the average number of jobs in the run queue over the last 1-, 5- and 15-minute intervals when the packet was sent.

An example status line output by **ruptime** might be:

**machine1 up 1+5:15, 7 users, load 1.47, 1.16, 0.80**

The above status line would be interpreted as follows:

**machine1** is presently "up" and has been up for 1 day, 5 hours and 15 minutes. It currently has 7 users logged in. Over the last 1-minute interval, an average of 1.47 jobs were in the run queue. Over the last 5-minute interval, an average of 1.16 jobs were in the run queue. Over the last 15-minute interval, an average of 0.80 jobs were in the run queue.

If a user has not used the system for an hour or more, the user is considered idle. Idle users are not shown unless the **-a** option is specified.

**Options**

If no options are specified, the listing is sorted by host name. Options change sorting order as follows:

- l** Sort by load average.
- t** Sort by up time.
- u** Sort by the number of users.
- r** Reverse the sort order.

**DIAGNOSTICS**

**no hosts!?!**

No status report files in **/var/spool/rwho**. *Next Step:* Ask the system administrator to check whether the **rwwho** daemon is running.

**AUTHOR**

**ruptime** was developed by the University of California, Berkeley.

**FILES**

**/var/spool/rwho/whod.\*** Data files

**SEE ALSO**

*rwwho(1)*, *rwwho(1M)*.

| **r** |

**NAME**

**rusers** - determine who is logged in on machines on local network

**SYNOPSIS**

**rusers** [-a] [-h] [-i] [-l] [-u] [*host ...*]

**DESCRIPTION**

**rusers** produces output similar to the "quick" option of *who(1)*, but for remote machines. It broadcasts on the local network and prints the responses it receives. Though the listing is normally in the order that responses are received, the order can be changed by specifying a command-line option. The broadcast process takes about two minutes.

When *host* arguments are given, instead of broadcasting, **rusers** only queries the list of specified hosts.

For each machine, the default is to print a line listing the host name and all users on that host. When the **-l** option is given, **rusers** uses an output format similar to *rwho(1)*. If a user has not typed on the system for a minute or more, the idle time is reported.

A remote host only responds if it is running the *rusersd(1M)* daemon.

**Options**

**rusers** recognizes the following command-line options:

- a** Give a report for a machine even if no users are logged in on it.
- h** Sort alphabetically by host name.
- i** Sort by idle time.
- l** Give a longer listing in the style of **who-R** (see *who(1)*).
- u** Sort by number of users.

**RETURN VALUE**

**rusers** returns exit code zero if no errors are encountered; otherwise it returns the number of errors found.

**AUTHOR**

**rusers** was developed by Sun Microsystems, Inc.

**WARNINGS**

Broadcasting does not work through gateways; therefore, **rusers** does not report information about machines that are reached only through gateways.

**| r |****FILES**

*/etc/inetd.conf*

**SEE ALSO**

*rwho(1)*, *who(1)*, *inetd(1M)*, *rusersd(1M)*.

**NAME**

**rwho** - show who is logged in on local machines

**SYNOPSIS**

**rwho** [-a]

**DESCRIPTION**

**rwho** produces output similar to the output of the HP-UX **who** command for all machines on the local network that are running the **rwho** daemon (see **who(1)** and **rwhod(1M)**). If **rwhod** has not received a report from a machine for 11 minutes, **rwho** assumes the machine is down and **rwho** does not report users last known to be logged into that machine.

**rwho**'s output line has fields for the name of the user, the name of the machine, the user's terminal line, the time the user logged in, and the amount of time the user has been idle. Idle time is shown as:

*hours :minutes*

If a user has not typed to the system for a minute or more, **rwho** reports this as idle time. If a user has not typed to the system for an hour or more, the user is omitted from **rwho**'s output unless the **-a** flag is given.

An example output line from **rwho** would look similar to:

```
joe_user machine1:tty0p1 Sep 12 13:28 :11
```

This output line could be interpreted as **joe\_user** is logged into **machine1** and his terminal line is **tty0p1**. **joe\_user** has been logged on since September 12 at 13:28 (1:28 p.m.). **joe\_user** has not typed anything into **machine1** for 11 minutes.

**WARNINGS**

**rwho**'s output becomes unwieldy when the number of users for each machine on the local network running **rwhod** becomes large. One line of output occurs for each user on each machine on the local network that is running **rwhod**.

**AUTHOR**

**rwho** was developed by the University of California, Berkeley.

**FILES**

**/var/spool/rwho/whod.\*** Information about other machines.

**SEE ALSO**

**ruptime(1)**, **rusers(1)**, **rwhod(1M)**.

| **r** |



**NAME**

sact - print current SCCS file editing activity

**SYNOPSIS**

**sact** *file* ...

**DESCRIPTION**

The **sact** command informs the user of any impending deltas to a named SCCS file. This situation occurs when **get -e** has been previously executed without a subsequent execution of **delta** (see *delta(1)* and *get(1)*). If a directory is named on the command line, **sact** behaves as though each file in the directory were specified as a named file, except that non-SCCS files (last component of path name does not begin with **s**.) and unreadable files are silently ignored. If a name of **-** is given, the standard input is read with each line being taken as the name of an SCCS file to be processed.

The output for each named file consists of five fields separated by spaces.

|         |                                                                                                            |
|---------|------------------------------------------------------------------------------------------------------------|
| Field 1 | SID of a delta that currently exists in the SCCS file to which changes will be made to make the new delta. |
| Field 2 | SID for the new delta to be created.                                                                       |
| Field 3 | Logname of the user making the delta (i.e., executed a <b>get</b> for editing).                            |
| Field 4 | Date when <b>get -e</b> was executed.                                                                      |
| Field 5 | Time when <b>get -e</b> was executed.                                                                      |

**DIAGNOSTICS**

Use *sccshelp(1)* for explanations.

**SEE ALSO**

*delta(1)*, *get(1)*, *sccshelp(1)*, *unget(1)*.

**STANDARDS CONFORMANCE**

**sact**: SVID2, SVID3, XPG2, XPG3, XPG4

**| s |**

**NAME**

samlog\_viewer - a tool for viewing and saving the SAM logfile

**SYNOPSIS**

```
/usr/sam/bin/samlog_viewer [-s mddhhmm[[cc]yy]] [-e mddhhmm[[cc]yy]] [-l SDVC]
[-u user] [-o ofile] [-t] [-n] [file]
```

**DESCRIPTION**

The **samlog\_viewer** command enables the viewing of part or all of the SAM logfile (or another file containing data in the same format) at varying levels of detail. This tool is run by SAM whenever the **View SAM Log** option is chosen. It can also be run independently of SAM, in either interactive or noninteractive mode.

The **samlog\_viewer** command executes in either interactive or noninteractive mode, depending on the options given. In noninteractive mode, **samlog\_viewer** filters the source file and writes the resulting data either to stdout or to a destination file. In interactive mode, **samlog\_viewer** displays a graphical user interface enabling you to try different combinations of filtering, save one or more versions of the source file to other files, scroll back and forth among the logfile entries, etc.

Under no circumstances is **samlog\_viewer** destructive to the contents of the SAM logfile (or whatever source file is specified by *file*). The contents of the source file are filtered and displayed (or output) according to the settings of the available filters. Multiple instances of **samlog\_viewer** can be run simultaneously without harmful effects.

**Filters**

**samlog\_viewer** supports three types of filters: level of detail, date/time, and user filters. These filters can be used in combination to provide highly selective logfile viewing.

The level of detail filters control how much detail is displayed. The SAM logfile may contain entries of many different types. The entry types currently supported are: summary, detail, verbose, error, and note. The level of detail filters display some or all of these entry types, depending on which filter is chosen. The level of detail filters are:

|                      |                                                                                                                                                                           |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Summary</b>       | Displays only the higher level messages. These include <i>summary</i> , <i>error</i> , and <i>note</i> (warnings, other entries worthy of special attention) entry types. |
| <b>Detail</b>        | Includes <b>Summary</b> level of detail, and adds <i>detail</i> log entries. If no level of detail is specified, this is the default.                                     |
| <b>Verbose</b>       | Includes <b>Detail</b> level of detail, and adds <i>verbose</i> log entries.                                                                                              |
| <b>Commands Only</b> | Displays only the literal commands that were executed. These commands may include HP-UX commands as well as SAM commands and scripts.                                     |

The date/time filters are used to ask for entries written since a specific date/time, before a specific date/time, or both.

The user filters are for viewing only those entries written by a particular user. If invoked with the **-u** option, its argument is used as the user whose entries should be shown. If **-u** is not specified, then the value of the environment variable **SAM\_RESTRICTED\_USER** is used as the name of the user. If neither **-u** or **SAM\_RESTRICTED\_USER** is present, then no user filtering is done and entries from all users are shown. If both **-u** and **SAM\_RESTRICTED\_USER** are in effect, the **-u** option overrides the setting of the environment variable.

If user filtering is selected, either by the **SAM\_RESTRICTED\_USER** environment variable or by the **-u** option, **samlog\_viewer** displays only entries made by that user and disallows any changes to the user filtering. This is useful for allowing nonprivileged users to run **samlog\_viewer** and see only those entries that pertain to them. Otherwise, **samlog\_viewer** allows the user filtering to be changed, or completely disabled, from its interactive filtering screen.

**Options**

The following options enable you to set up filtering and other attributes. If **samlog\_viewer** runs interactively, these attributes may also be set and modified in the various supported menus and displays. The available options are:

|                           |                                                                                                                                                                                                     |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-s mddhhmm[[cc]yy]</b> | The <b>-s</b> option sets the start date/time filter to the date/time given by its argument. The date/time is specified in the same way as it is for the <b>date</b> command (see <i>date</i> (1)): |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

*mm* Month specified as a two digit number (e.g., **08**).

*dd* Day specified as a two digit number.

*hh* Hour specified as a two digit number (24-hour clock form).

*mm* Minute specified as a two digit number.

*cc* Century minus one. *cc* is optional when specifying the year.

*yy* The last two digits of the desired year. If this is not specified, the current year is used.

**Note:** To specify a year in two digits, the valid ranges are 70-99 (20th century) and 00-37 (21st century).

To specify a year in four digits, the valid range is 1970-2037.

If no start time is given, the beginning of the log is used as the start time.

**-e** *mmddhhmm*[[*cc*]*yy*] The **-e** option sets the end date/time filter to the date/time given by its argument. The date and time is specified as described above for the **-s** option. If no end time is given, then an end date/time of infinity (no end time) is used.

**-l** *SDVC* The **-l** option sets the desired level of detail. One of the letters **SDVC** must be specified as the required argument. The level of detail is set as follows:

**S** = Summary  
**D** = Detail  
**V** = Verbose  
**C** = Commands Only

If the **-l** option is not specified, the **Detail** level of detail is used by default.

**-u** *user* The **-u** option sets the user filter to the user name or user ID specified by *user*. Only entries logged by this user are displayed. If the **-u** option is omitted, entries logged by all users are displayed by default.

**-o** *ofile* The **-o** option causes the filtered output to be written to the output file *ofile*. The **-o** option implies the **-n** option described below. If *ofile* is **-**, the output is written to stdout. If **-o** is omitted, the output is written to either stdout (if **-n** is specified) or to the interactive **samlog\_viewer** display (if **-n** is omitted).

**-t** The **-t** option enables automatic timestamping. If specified, each log entry is tagged with the time of day at which it was written. Timestamping is disabled by default.

**-n** The **-n** option forces noninteractive behavior. If specified, **samlog\_viewer** runs noninteractively, using the default or specified values for all supported options and source/destination files.

*file* Specifies the name of the file from which log data is read. The format of the data in the specified file must be the same as that used for raw SAM logfile data. If omitted, the SAM logfile is read. If *file* is **-**, stdin is read and **samlog\_viewer** runs noninteractively. If given, *file* must be the last argument specified on the command line.

| **S** |

## EXAMPLES

Capture the current contents of the SAM logfile using default filtering, and put into the file **sam.out**:

```
samlog_viewer -n >sam.out
```

The following example does the same thing:

```
samlog_viewer -o sam.out
```

View only the commands executed by SAM on behalf of user **tom**, between 8am June 5, 1994 and 10pm August 14, 1994, and view the data interactively:

```
 samlog_viewer -s 060508001994 -e 081422001994 -lC -u tom
```

Noninteractively read data from stdin, timestamp it, and save the result in a file called **stdin.out**:

```
 cat datafile | samlog_viewer -t -o stdin.out -
```

Do the same as above, but instead have the data appear on stdout:

```
 cat datafile | samlog_viewer -t -o - -
```

or

```
 cat datafile | samlog_viewer -tn -
```

#### FILES

|                                      |                                                                                                                                                                                                           |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/var/sam/log/samlog</code>     | SAM logfile.                                                                                                                                                                                              |
| <code>/var/sam/log/samlog.old</code> | Archived version of <b>samlog</b> , created when the logfile is automatically trimmed by SAM when its size becomes too large. Its contents are included in the log entries read by <b>samlog_viewer</b> . |
| <code>/tmp/LFV_pid</code>            | Temporary files used by <b>samlog_viewer</b> .                                                                                                                                                            |
| <code>/tmp/LFV_RUNpid</code>         |                                                                                                                                                                                                           |

| **s** |

**NAME**

sccs - front-end utility program for SCCS commands

**SYNOPSIS**

**sccs** [-r] [-d *rootpath*] [-p *dirpath*] *command* [*options*] [*file* ...]

**DESCRIPTION**

The **sccs** command is a straightforward front end to the various programs comprising the Source Code Control System. It includes the capability of running set-user-id to another user to allow shared access to the SCCS files. **sccs** reduces the need to explicitly reference the SCCS filenames. The SCCS filenames are generated by prepending the string **SCCS/s.** to the working *files* specified. The default SCCS sub-directory name can be overridden with the **-p dirpath** option.

The *command* supplied to the **sccs** command can either be an SCCS program or a pseudo command. The SCCS programs that **sccs** handles include **admin**, **cdc**, **comb**, **delta**, **get**, **help**, **prs**, **rmDEL**, **sact**, **unget**, **val**, **what** and **sccsdiff**. The pseudo commands are:

- |                |                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>check</b>   | Prints a list of all files being edited. Returns a non-zero exit status if a file is being edited. The intent is to allow an 'install' entry in a makefile to verify that everything is included in the SCCS file before a version is installed. See the <b>info</b> pseudo command for a description of the <b>-b</b> , <b>-u user</b> and <b>-U</b> options.                               |
| <b>clean</b>   | Removes all files from the current directory or the named directory that can be recreated from the SCCS files. Does not remove files that are in the process of being edited. If <b>-b</b> is given, branches (i.e. SID's with three or more components) are ignored in determining which files are being edited. Therefore, any edits on branches can be lost.                              |
| <b>create</b>  | Creates the initial SCCS file, taking the contents from <i>file</i> . Any options to <b>admin</b> are accepted. If the files are created successfully, the original files are renamed with a <b>,</b> (comma) on the front. Read-only copies are retrieved with <b>get</b> . The renamed files should be removed after you have verified that the SCCS files have been created successfully. |
| <b>delget</b>  | Runs <b>delta</b> on the named files and then <b>get</b> the new versions. The new versions of the files have expanded identification keywords, and cannot be edited. The <b>[-mprsy]</b> options are passed to <b>delta</b> , and the <b>[-bceiklsx]</b> options are passed to <b>get</b> .                                                                                                 |
| <b>deledit</b> | Equivalent to <b>delget</b> , except that the <b>get</b> phase includes the <b>-e</b> option.                                                                                                                                                                                                                                                                                                |
| <b>diffs</b>   | Gives a <b>diff</b> listing between the current version of the files being edited and the versions in SCCS format. The <b>[-rcixt]</b> options are passed to <b>get</b> . The <b>[-lsefhb]</b> options are passed to <b>diff</b> . The <b>-C</b> option is passed to <b>diff</b> as <b>-c</b> .                                                                                              |
| <b>edit</b>    | Equivalent to <b>get -e</b> .                                                                                                                                                                                                                                                                                                                                                                |
| <b>enter</b>   | Equivalent to <b>create</b> , except <b>get</b> is omitted. This pseudo command is useful when you want to run the <b>edit</b> command immediately after creating the SCCS file.                                                                                                                                                                                                             |
| <b>fix</b>     | Removes a named delta, but leaves a copy of the delta in the current directory. The <b>-r SID</b> option is required and must point to a leaf in the source tree. Since a record of the changes is not preserved, <b>fix</b> should be used carefully.                                                                                                                                       |
| <b>info</b>    | Lists all the files being edited. The <b>-b</b> option ignores branches in determining which files are being edited. The <b>-u user</b> option lists only the files being edited by <i>user</i> . The <b>-U</b> option is equivalent to <b>-u current_user</b> .                                                                                                                             |
| <b>print</b>   | Prints information about named files. Equivalent to <b>prs -a</b> followed by <b>get -p -m -s</b> .                                                                                                                                                                                                                                                                                          |
| <b>tell</b>    | Lists all the files being edited, with a newline after each entry. See the <b>info</b> section for a description of the <b>-b</b> , <b>-u user</b> and <b>-U</b> options.                                                                                                                                                                                                                    |
| <b>unedit</b>  | Equivalent to <b>unget</b> . Any changes made since the last <b>get</b> are lost. Use with caution.                                                                                                                                                                                                                                                                                          |

Certain commands, **admin**, **cdc**, **check**, **clean**, **diffs**, **info**, **rmDEL**, **sccsdiff**, and **tell** cannot use the set-user-id feature, as this would allow anyone to change the authorizations. These commands are always run as the real user.

### Options

The *options* supplied to the SCCS commands are documented in the corresponding SCCS man pages. The *options* supplied to the pseudo commands are documented in the above section. All other options preceding *command* are documented as follows:

- r** Runs **sccs** as the real user rather than the effective user **sccs** is set-user-id to.
- d *rootpath*** Gives the pathname to be used as the root directory for the SCCS files. *rootpath* defaults to the current directory. This flag takes precedence over the PROJECTDIR environment variable.
- p *dirpath*** Specifies the pathname for the SCCS files. The default is the SCCS directory. *dirpath* is appended to *rootpath* and is inserted before the final component of the path-name.

The command **sccs -d /usr -p cmd get src/b** converts to **get/usr/src/cmd/s.b**. This can be used to create aliases. For example, the command **alias syssccs="sccs -p /usr/src/cmd"** makes **syssccs** an alias that can be used in commands like **syssccsgetb**.

### EXTERNAL INFLUENCES

#### Environment Variables

If the **PROJECTDIR** environment variable is set, its value is used to determine the **-d *rootpath*** option value for *rootpath*. If **PROJECTDIR** begins with a / (slash), the value is used directly; otherwise, the value is assumed to be a login name and the home directory corresponding to login name is examined for a subdirectory named **src** or **source**. If found, this directory path is used. Otherwise, the value is used as a relative path name.

**LC\_CTYPE** determines the interpretation of text within file as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **sccs** behaves as if all internationalization variables are set to "C". See *environ(5)*.

#### International Code Set Support

Single-byte and multi-byte character code sets are supported.

### EXAMPLES

To create a new SCCS file:

```
sccs create file
```

To get a file for editing, edit it, create a new delta and get file for editing:

```
sccs edit file.c
ex file.c
sccs deledit file.c
```

To get a file from another directory (**/usr/src/cmd/SCCS/s.cc.c**):

```
sccs -d /usr/src get cmd/cc.c
```

To make a delta of a large number of files in the current directory, enter:

```
sccs delta *.c
```

To get a list of files being edited that are not on branches, enter:

```
sccs info -b
```

To get a list of files being edited by you, enter:

```
sccs tell -u
```

In a makefile, to get source files from an SCCS file if it does not already exist, enter:

```
SRCS = <list of source files>
$(SRCS) :
 sccs get $(REL) $@
```

**RETURN VALUE**

A successful completion returns 0. On error, **sccs** exists with a value from <sys-exits.h> or the exit value from the *command* that was invoked. The only exception is the **check** pseudo command which returns a non-zero exit status if a file is being edited.

**SEE ALSO**

admin(1), cdc(1), comb(1), delta(1), get(1), prs(1), rmdel(1), sact(1), sccsdiff(1), sccshelp(1), unget(1), val(1), vc(1), what(1), sccsfile(4).

**STANDARDS CONFORMANCE**

**sccs**: XPG4

**NAME**

sccsdiff - compare two versions of an SCCS file

**SYNOPSIS**

**sccsdiff** **-rSID1 -rSID2** [-p] [-sn] *file* ...

**DESCRIPTION**

The **sccsdiff** command compares two versions of an SCCS file, and generates the differences between the two versions. Any number of SCCS files may be specified, but arguments apply to all files.

- rSID?** *SID1* and *SID2* specify the deltas of an SCCS file that are to be compared. Versions are passed to **bdiff** in the order given (see *bdiff(1)*). The SIDs accepted, and the corresponding version retrieved for the comparison are the same as for **get** (see *get(1)*).
- p** Pipe output for each file through **pr** (see *pr(1)*).
- sn** *n* is the file segment size that **bdiff** passes to **diff** (see *diff(1)*). This is useful when **diff** fails due to a high system load.

**EXTERNAL INFLUENCES****International Code Set Support**

Single- and multi-byte character code sets are supported with the exception that multi-byte-character file names are not supported.

**DIAGNOSTICS**

*file*: **No differences**

The two versions are identical.

Use *sccshelp(1)* for explanations.

**FILES**

*/tmp/get?????* Temporary files

**SEE ALSO**

*bdiff(1)*, *diff(1)*, *get(1)*, *pr(1)*, *sccshelp(1)*.

| **S** |



**NAME**

sccshelp - ask for help on SCCS commands

**SYNOPSIS**

**sccshelp** [*arg*] ...

**DESCRIPTION**

The **sccshelp** command finds information to explain a message from an SCCS command or to explain the use of a SCCS command. Zero or more arguments can be supplied. If no arguments are given, **sccshelp** prompts for one:

**What is the message number or SCCS command name?**

The arguments can be either message numbers (which normally appear in parentheses following messages) or command names, of one of the following types:

- type 1 Begins with nonnumerics, ends in numerics. The nonnumeric prefix is usually an abbreviation for the program or set of routines which produced the message (e.g., **ge6**, for message 6 from the **get** command).
- type 2 Does not contain numerics (as a command, such as **get**).
- type 3 Is all numeric (e.g., **212**).

The response of the program is the explanatory information related to the argument, if there is any.

You can use **sccshelp** to support other commands by means of the **helploc** file. To do this, create help files in the appropriate format and add the location of the helpfiles to **/usr/share/lib/sccshelp/helploc**.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the interpretation of text as single- and/or multibyte characters.

**LANG** determines the language in which messages are displayed.

If **LC\_CTYPE** is not specified or is null, it defaults to the value of **LANG**. If **LANG** is not specified or is null, it defaults to "C" (see *lang(5)*). If any internationalization variable contains an invalid setting, all internationalization variables default to "C". See *environ(5)*.

**International Code Set Support**

Single- and multibyte character code sets are supported.

**DIAGNOSTICS**

**There is not a place to look for help on the subject of *arg***

**There is not any help available on the subject of *arg***

When all else fails, try **sccshelp stuck**.

| S |

**EXAMPLES**

If you enter the SCCS **get** command without parameters, you would get the message:

```
ERROR: missing file arg (cm3)
```

If you request help for the command:

```
sccshelp get
```

it displays:

```
get:
get [-r<SID>] [-c<cutoff>] [-i<list>] [-x<list>] [-a<seq no.>]
 [-k] [-e] [-l<p>] [-p] [-m] [-n] [-s] [-b] [-g] [-t]
 [-w<string>] file ...
```

If you request help for the error number:

```
sccshelp cm3
```

it displays:

**cm3:**  
"missing file arg"  
You left off the name of the file to be processed.

**WARNINGS**

Only SCCS commands currently use **sccshelp**.

**FILES**

|                                        |                                                                                                          |
|----------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>/usr/share/lib/sccshelp</b>         | Directory containing files of message text                                                               |
| <b>/usr/share/lib/sccshelp/cmds</b>    | List of commands supported by sccshelp                                                                   |
| <b>/usr/share/lib/sccshelp/helploc</b> | File containing the locations of help files that are not in the <b>/usr/share/lib/sccshelp</b> directory |

| **s** |

**NAME**

script - make typescript of terminal session

**SYNOPSIS**

**script** [-a] [*file*]

**DESCRIPTION**

**script** makes a typescript of everything printed on your terminal. It starts a shell named by the **SHELL** environment variable, or by default **/usr/bin/sh**, and silently records a copy of output to your terminal from that shell or its descendents, using a pseudo-terminal device (see *pty(7)*).

All output is written to *file*, or appended to *file* if the **-a** option is given. If no file name is given, the output is saved in a file named **typescript**. The recording can be sent to a line printer later with *lp(1)*, or reviewed safely with the **-v** option of *cat(1)*.

The recording ends when the forked shell exits (or the user ends the session by typing "exit") or the shell and all its descendents close the pseudo-terminal device.

This program is useful when operating a CRT display and a hard-copy record of the dialog is desired. It can also be used for a simple form of session auditing.

**script** respects the convention for login shells as described in *su(1)*, *sh(1)*, and *ksh(1)*. Thus, if it is invoked with a command name beginning with a hyphen (-) (that is, **-script**), **script** passes a basename to the shell that is also preceded by a hyphen.

The input flow control can be enabled by setting environmental variable **SCRIPT\_USE\_IXOFF**, before running **script**. Please see **WARNINGS** section for details on using this environment variable.

**EXAMPLES**

Save everything printed on the user's screen into file **scott**:

```
script scott
```

Append a copy of everything printed to the user's screen to file **temp**:

```
script -a temp
```

**WARNINGS**

A command such as **cat scott**, which displays the contents of the destination file, should not be issued while executing **script** because it would cause **script** to log the output of the **cat** command to itself until all available disk space is filled. Other commands, such as *more(1)*, can cause the same problem but to a lesser degree.

**script** records all received output in the *file*, including typing errors, backspaces, and cursor motions. Note that it does not record typed characters; only echoed characters. Thus passwords are not recorded in the *file*. Responses other than simple echoes (such as output from screen-oriented editors and **ksh** command editing) are recorded as they appeared in the original session.

When there is no input flow control (**SCRIPT\_USE\_IXOFF** is not set), there can be some data loss while using **script**. However, *script(1)* can behave unexpectedly, if **SCRIPT\_USE\_IXOFF** is set and **IXANY** is not set.

**AUTHOR**

**script** was developed by the University of California, Berkeley and HP.

| S |

**NAME**

sdiff - side-by-side difference program

**SYNOPSIS**

**sdiff** [*options ...*] *file1 file2*

**DESCRIPTION**

**sdiff** uses the output of *diff*(1) with the **-b** option, which ignores trailing blanks (spaces and tabs) and treats other strings of blanks as equal, to produce a side-by-side listing of two files, indicating those lines that are different. Each line of the two files is printed with a blank gutter between them if the lines are identical, a **<** in the gutter if the line only exists in *file1*, a **>** in the gutter if the line only exists in *file2*, and a **|** for lines that are different.

For example:

```

abc | xyz
abc | abc
bca <
cba <
dcb | dcb
 > cde

```

**Options**

**sdiff** recognizes the following options:

- w** *n*        Use the next argument, *n*, as the width of the output line. The maximum value of *n* is 2048 (LINE\_MAX). The default line length is 130 characters.
- l**            Only print on the left side when lines are identical.
- s**            Do not print identical lines.
- o** *output*   Use the next argument, *output*, as the name of a third file that is created as a user-controlled merging of *file1* and *file2*. Identical lines of *file1* and *file2* are copied to *output*. Sets of differences, as produced by *diff*(1), are printed; where a set of differences share a common gutter character. After printing each set of differences, **sdiff** prompts the user with a **%** and waits for one of the following user-typed commands:
  - l**    append the left column to the output file
  - r**    append the right column to the output file
  - s**    turn on silent mode; do not print identical lines
  - v**    turn off silent mode
  - e l** call the editor with the left column
  - e r** call the editor with the right column
  - e b** call the editor with the concatenation of left and right
  - e**    call the editor with a zero length file
  - q**    exit from the program

On exit from the editor, the resulting file is concatenated on the end of the *output* file.

**EXAMPLES**

Print a side-by-side diff of two versions of a file on a printer capable of printing 132 columns:

```
sdiff -w132 prog.c.old prog.c | lp -dlineprinter
```

Retrieve the most recently checked in version of a file from RCS and compare it with the version currently checked out:

```
co -p prog.c > /tmp/$$; sdiff /tmp/$$ prog.c | more; rm /tmp/$$
```

**SEE ALSO**

*diff*(1), *ed*(1).

**NAME**

sed - stream text editor

**SYNOPSIS**

**sed** [-n] *script* [*file* ...]

**sed** [-n] [-e *script*] ... [-f *script\_file*] ... [*file* ...]

**DESCRIPTION**

**sed** copies the named text *files* (standard input default) to the standard output, edited according to a script containing up to 100 commands. Only complete input lines are processed. Any input text at the end of a file that is not terminated by a new-line character is ignored.

**Options**

**sed** recognizes the following options:

**-f** *script\_file* Take script from file *script\_file*.

**-e** *script* Edit according to *script*. If there is just one **-e** option and no **-f** options, the flag **-e** can be omitted.

**-n** Suppress the default output.

**sed** interprets all **-e***script* and **-f***script\_file* arguments in the order given. Use caution, if mixing **-e** and **-f** options, to avoid unpredictable or incorrect results.

**Command Scripts**

A script consists of editor commands, one per line, of the following form:

```
[address [, address]] function [arguments]
```

In normal operation, **sed** cyclically copies a line of input into a *pattern space* (unless there is something left after a **D** command), applies in sequence all commands whose *addresses* select that pattern space, and, at the end of the script, copies the pattern space to the standard output (except under **-n**) and deletes the pattern space.

Some of the commands use a *hold space* to save all or part of the *pattern space* for subsequent retrieval.

**Command Addresses**

An *address* is either a decimal number that counts input lines cumulatively across files, a **\$** which addresses the last line of input, or a context address; that is, a */regular expression /* in the style of *ed*(1) modified thus:

- In a context address, the construction *\?regular expression?*, where *?* is any character, is identical to */regular expression /*. Note that in the context address *\xabc\xdefx*, the second **x** stands for itself, so that the regular expression is **abcxdef**.
- The escape sequence **\n** matches a new-line character embedded in the pattern space.
- A period (**.**) matches any character except the terminal new-line of the pattern space.
- A command line with no addresses selects every pattern space.
- A command line with one address selects each pattern space that matches the address.
- A command line with two addresses selects the inclusive range from the first pattern space that matches the first address through the next pattern space that matches the second (if the second address is a number less than or equal to the line number first selected, only one line is selected). Thereafter the process is repeated, looking again for the first address.

**sed** supports Basic Regular Expression syntax (see *regexp*(5)).

Editing commands can also be applied to only non-selected pattern spaces by use of the negation function **!** (described below).

**Command Functions**

In the following list of functions, the maximum number of permissible addresses for each function is indicated in parentheses. Other function elements are interpreted as follows:

*text* One or more lines, all but the last of which end with **\** to hide the new-line. Backslashes in *text* are treated like backslashes in the replacement string of an **s**

command, and can be used to protect initial blanks and tabs against the stripping that is done on every script line.

- rfile* Must terminate the command line, and must be preceded by exactly one blank.
- wfile* Must terminate the command line, and must be preceded by exactly one blank. Each *wfile* is created before processing begins. There can be at most 10 distinct *wfile* arguments.

**sed** recognizes the following functions:

- (1) **a** \ *text* Append. Place *text* on the output before reading next input line.
- (2) **b** *label* Branch to the **:** command bearing *label*. If no *label* is specified, branch to the end of the script.
- (2) **c** \ *text* Change. Delete the pattern space. With 0 or 1 address or at the end of a 2-address range, place *text* on the output. Start the next cycle.
- (2) **d** Delete pattern space and start the next cycle.
- (2) **D** Delete initial segment of pattern space through first new-line and start the next cycle.
- (2) **g** Replace contents of the pattern space with contents of the hold space.
- (2) **G** Append contents of hold space to the pattern space.
- (2) **h** Replace contents of the hold space with contents of the pattern space.
- (2) **H** Append the contents of the pattern space to the hold space.
- (1) **i** \ *text* Insert. Place *text* on the standard output.
- (2) **l** List the pattern space on the standard output in an unambiguous form. Non-printing characters are spelled in three-digit octal number format (with a preceding backslash), and long lines are folded.
- (2) **n** Copy the pattern space to the standard output if the default output has not been suppressed (by the **-n** option on the command line or the **#n** command in the *script* file). Replace the pattern space with the next line of input.
- (2) **N** Append the next line of input to the pattern space with an embedded new-line. (The current line number changes.)
- (2) **p** Print. Copy the pattern space to the standard output.
- (2) **P** Copy the initial segment of the pattern space through the first new-line to the standard output.
- (1) **q** Quit. Branch to the end of the script. Do not start a new cycle.
- (1) **r** *rfile* Read contents of *rfile* and place on output before reading the next input line.
- (2) **s** /*regular expression* /*replacement* /*flags*  
Substitute *replacement* string for instances of *regular expression* in the pattern space. Any character can be used instead of **/**. For a fuller description see *ed*(1). *flags* is zero or more of:
- n* *n* = 1-2048 (**LINE\_MAX**). Substitute for just the *n*th occurrence of *regular expression* in the pattern space.
- g** Global. Substitute for all non-overlapping instances of *regular expression* rather than just the first one.
- p** Print the pattern space if a replacement was made and the default output has been suppressed (by the **-n** option on the command line or the **#n** command in the *script* file).
- w** *wfile* Write. Append the pattern space to *wfile* if a replacement was made.
- (2) **t** *label* Test. Branch to the **:** command bearing the *label* if any substitutions have been made since the most recent reading of an input line or execution of a **t**. If *label* is empty, branch to the end of the script.

| S |

- (2) **w** *wfile* Write. Append the pattern space to *wfile*.
- (2) **x** Exchange the contents of the pattern and hold spaces.
- (2) **Y/string1/string2/**  
Transform. Replace all occurrences of characters in *string1* with the corresponding character in *string2*. The lengths of *string1* and *string2* must be equal.
- (2) **!** *function*  
Don't. Apply the *function* (or group, if *function* is { }) only to lines *not* selected by the address or addresses.
- (0) **:** *label* This command does nothing; it bears a *label* for **b** and **t** commands to branch to.
- (1) **=** Place the current line number on the standard output as a line.
- (2) **{** Execute the following commands through a matching **}** only when the pattern space is selected. The syntax is:
 

```

 { cmd1
 cmd2
 cmd3
 .
 .
 .
 }
```
- (0) An empty command is ignored.
- (0) **#** If a **#** appears as the first character on the first line of a script file, that entire line is treated as a comment with one exception: If the character after the **#** is an **n**, the default output is suppressed. The rest of the line after **#n** is also ignored. A script file must contain at least one non-comment line.

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** provides a default value for the internationalization variables that are unset or null. If **LANG** is unset or null, the default value of "C" (see *lang(5)*) is used. If any of the internationalization variables contains an invalid setting, **sed** will behave as if all internationalization variables are set to "C". See *environ(5)*.

**LC\_ALL** If set to a non-empty string value, overrides the values of all the other internationalization variables.

**LC\_CTYPE** determines the interpretation of text as single and/or multi-byte characters, the classification of characters as printable, and the characters matched by character class expressions in regular expressions.

**LC\_MESSAGES** determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

**NLSPATH** determines the location of message catalogues for the processing of **LC\_MESSAGES**.

### International Code Set Support

Single- and multi-byte character code sets are supported.

## EXAMPLES

Make a simple substitution in a file from the command line or from a shell script, changing **abc** to **xyz**:

```
sed 's/abc/xyz/' file1 >file1.out
```

Same as above but use shell or environment variables **var1** and **var2** in search and replacement strings:

```
sed "s/$var1/$var2/" file1 >file1.out
```

or

```
sed 's/'$var1'/'$var2'' file1 >file1.out
```

Multiple substitutions in a single command:

```
sed -e 's/abc/xyz/' -e 's/lmn/rst/' file1 >file1.out
```

or

```
sed -e 's/abc/xyz/' \
-e 's/lmn/rst/' \
file1 >file1.out
```

#### WARNINGS

**sed** limits command scripts to a total of not more than 100 commands.

The hold space is limited to 8192 characters.

**sed** processes only text files. See the glossary for a definition of text files and their limitations.

#### AUTHOR

**sed** was developed by OSF and HP.

#### SEE ALSO

awk(1), ed(1), grep(1), environ(5), lang(5), regexp(5).

**sed**: *A Non-Interactive Streaming Editor* tutorial in the *Text Processing Users Guide*.

#### STANDARDS CONFORMANCE

**sed**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| **s** |



**NAME**

send\_sound - play an audio file

**SYNOPSIS**

```
/opt/audio/bin/send_sound [-format_switch] [-server system] [-loop number]
[-pri priority] [-srate rate] [-prate rate] [-stereo]
```

**DESCRIPTION**

This command plays an audio file. **send\_sound** is the command used when you double-click an audio file from the HP VUE File Manager. The file begins playing, according to the settings of the Audio Control Panel.

**-format\_switch**

is one these formats:

|            |                                       |
|------------|---------------------------------------|
| <b>au</b>  | Sun file format                       |
| <b>snd</b> | NeXT file format                      |
| <b>wav</b> | Microsoft RIFF Waveform file format   |
| <b>u</b>   | MuLaw format                          |
| <b>a1</b>  | ALaw                                  |
| <b>116</b> | linear 16-bit format                  |
| <b>108</b> | offset (unsigned) linear 8-bit format |
| <b>18</b>  | linear 8-bit format                   |

If you omit the filename with this option, **send\_sound** plays the audio data from **stdin**.

**-server system**

plays the file on the output of an audio server identified by *system* which is either a system name or a TCP/IP address.

**-loop number**

plays the file the *number* of times you supply. Note that you cannot use this option if the source is **stdin**; a filename is required.

**-pri priority** plays file with the *priority* you supply, either **urgent**, **hi**, **normal**, or **lo**.

**-srate rate** sets the sample *rate* of the source audio.

**-prate rate** plays the audio file at the sampling *rate* you enter.

**-stereo** plays a stereo file. This option is needed only for a raw data file with stereo data.

| S |

**AUTHOR**

**send\_sound** was developed by HP.

NeXT is a trademark of NeXT Computers, Inc.

Microsoft is a trademark of Microsoft Corporation.

**SEE ALSO**

audio(5), asecure(1M), aserver(1M), attributes(1), convert(1).

**NAME**

serialize - force target process to run serially with other processes

**SYNOPSIS**

```
serialize command [command_args]
```

```
serialize [-t] [-p pid]
```

**DESCRIPTION**

The **serialize** command is used to force the target process to run serially with other processes also marked by this command. The target process can be referred to by *pid* value, or it can be invoked directly on the *command*. Once a process has been marked by **serialize**, the process stays marked until process completion unless **serialize** is reissued on the serialized process with the **-t** option. The **-t** option causes the *pid* specified with the **-p** option to return to normal timeshare scheduling algorithms.

This call is used to improve process throughput, since process throughput usually increases for large processes when they are executed serially instead of allowing each program to run for only a short period of time. By running large processes one at a time, the system makes more efficient use of the CPU as well as system memory, since each process does not end up constantly faulting in its working set, to only have the pages stolen when another process starts running. As long as there is enough memory in the system, processes marked by **serialize** behave no differently from other processes in the system. However, once memory becomes tight, processes marked by **serialize** are run one at a time with the highest priority processes being run first. Each process will run for a finite interval of time before another serialized process is allowed to run.

**Options**

**serialize** supports the following options:

- t**        Indicates the process specified by *pid* should be returned to timeshare scheduling.
- p**        Indicates the *pid* of the target process.

If neither option is specified, **serialize** is invoked on the command line passed in.

**RETURN VALUE**

**serialize** returns the following value:

- 0        Successful completion.
- 1        Invalid *pid* specification, nonnumeric entry, or <= PID\_MAXSYS.
- 2        Could not execute the specified command.
- 3        No such process.
- 4        Must be root or a member of a group having SERIALIZE privilege to execute **serialize**.

**ERRORS**

**serialize** fails under the following condition and sets **errno** (see *errno(2)*) to the following value:

- [ESRCH]        The *pid* passed in does not exist.

**EXAMPLES**

Use **serialize** to force a database application to run serially with other processes marked for serialization:

```
serialize database_app
```

Force a currently running process with a *pid* value of 215 to run serially with other processes marked for serialization:

```
serialize -p 215
```

Return a process previously marked for serialization to normal timeshare scheduling. The *pid* of the target process for this example is 174:

```
serialize -t -p 174
```

**WARNINGS**

The user has no way of forcing an execution order on serialized processes.

**AUTHOR**

**serialize** was developed by HP.

**SEE ALSO**

setprivgrp(1M), getprivgrp(2), serialize(2).

| **s** |

**NAME**

setacl - modify access control lists (ACLs) for files (JFS File Systems only)

**SYNOPSIS**

```
setacl [-n] -s acl_entries file...
setacl [-n] -m|-d acl_entries [-m|-d acl_entries]... file...
setacl [-n] -f acl_file file...
```

**DESCRIPTION**

For each *file* specified, **setacl** will either replace its entire ACL, including the default ACL on a directory, or it will add, modify, or delete one or more ACL entries, including default entries on directories.

The **-s** option will set the ACL to the entries specified on the command line. The **-f** option will set the ACL to the entries contained within the file *acl\_file*. The **-d** option will delete one or more specified entries from the file's ACL. The **-m** option will add or modify one or more specified ACL entries.

One of the options **-s**, **-m**, **-d**, or **-f** must be specified. If **-s** or **-f** are specified, other options are invalid. The **-m** and **-d** options may be combined, and multiple **-m** and **-d** options may be specified.

For the **-m** and **-s** options, *acl\_entries* are one or more comma separated ACL entries selected from the following list. For the **-f** option, *acl\_file* must contain ACL entries, one to a line, selected from the same list. Default entries may only be specified for directories. **Bold face** indicates that characters must be typed as specified, brackets denote optional characters, and *italicized* characters are to be specified by the user. Choices, of which exactly one must be selected, are separated by vertical bars.

```
u[ser]:operm |perm
u[ser]:uid:operm |perm
g[roup]:operm |perm
g[roup]:gid:operm |perm
c[lass]:operm |perm
o[ther]:operm |perm
d[efault]:u[ser]:operm |perm
d[efault]:u[ser]:uid:operm |perm
d[efault]:g[roup]:operm |perm
d[efault]:g[roup]:gid:operm |perm
d[efault]:c[lass]:operm |perm
d[efault]:o[ther]:operm |perm
```

For the **-d** option, *acl\_entries* are one or more comma separated ACL entries without permissions, selected from the following list. Note that the entries for file owner, owning group, and others may not be deleted.

```
u[ser]:uid
g[roup]:gid
d[efault]:u[ser]:
d[efault]:u[ser]:uid
d[efault]:g[roup]:
d[efault]:g[roup]:gid
d[efault]:c[lass]:
d[efault]:o[ther]:
```

In the above lists, the user specifies the following:

*perm* is a permissions string composed of the characters **r** (read), **w** (write), and **x** (execute), each of which may appear at most one time, in any order. The character **-** may be specified as a placeholder.

*operm* is the octal representation of the above permissions, with 7 representing all permissions, or **rwax**, and 0 representing no permissions, or **---**.

*uid* is a login name or user ID.

*gid* is a group name or group ID.

The options have the following meanings:

**-n** Normally, **setacl** recalculates the group class entry so as to ensure that permissions granted in the additional ACL entries will actually be granted, and the value specified in the **class** entry is

ignored. If the **-n** option is specified, the recalculation is not performed, and the value specified in the **class** entry is used.

- s** Set a file's ACL. All old ACL entries are removed, and replaced with the newly specified ACL. There must be exactly one **user** entry specified for the owner of the file, exactly one **group** entry specified for the owning group of the file, and exactly one **other** entry specified. If the **-n** option is not specified there must also be exactly one **class** entry specified. There may be additional **user** ACL entries and additional **group** ACL entries specified, but there may not be duplicate additional **user** ACL entries with the same *uid*, or duplicate additional **group** ACL entries with the same *gid*.

If the file is a directory, default ACL entries may be specified. There may be at most one **default:user** entry for the owner of the file, at most one **default:group** entry for the owning group of the file, at most one **default:class** entry for the file group class, and at most one **default:other** entry for other users. There may be additional **default:user** entries and additional **default:group** entries specified, but there may not be duplicate additional **default:user** entries with the same *uid*, or duplicate additional **default:group** entries with the same *gid*.

**setacl** never recalculates the **default:class** entry, regardless of whether or not the **-n** option was specified.

An entry with no permissions will result in the specified *uid* or *gid* being denied access to the file.

The entries need not be in order. They will be sorted by the command before being applied to the file.

- m** Add one or more new ACL entries to the file, and/or change one or more existing ACL entries on the file. If an entry already exists for a specified *uid* or *gid*, the specified permissions will replace the current permissions. If an entry does not exist for the specified *uid* or *gid*, an entry will be created.
- d** Delete one or more existing ACL entries from the file. The entries for the file owner, the owning group, and others may not be deleted from the ACL. Note that deleting an entry does not necessarily have the same effect as removing all permissions from the entry. Specifically, deleting an entry for a specific user would cause that user's permissions to be determined by the **other** entry (or the owning **group** entry, if the user is in that group).
- f** Set a file's ACL with the ACL entries contained in the file named *acl\_file*. The same constraints on specified entries hold as with the **-s** option. The entries are not required to be in any specific order in the file specified as *acl\_file*. The character **"#"** in *acl\_file* may be used to indicate a comment. All characters, starting with the **"#"**, until the end of the line, will be ignored. Note that if the *acl\_file* has been created as the output of the **getacl** command, any effective permissions, which will have been written with a preceding **"#"**, will also be ignored.

When the **setacl** command is used, it may result in changes to the file permission bits. When the **user** ACL entry for the file owner is changed, the file owner permission bits will be modified. When the **other** ACL entry is changed, the file other permission bits will be modified. When additional **user** ACL entries and/or any **group** ACL entries are set or modified, the file group permission bits will be modified to reflect the maximum permissions allowed by the additional user entries and all the group entries.

If an ACL contains no additional **user** or additional **group** entries, the permissions in the **group** entry for the object owning group and the **class** entry must be the same. Therefore, if the **-d** option is specified and results in no additional **user** entries and no additional **group** entries, the **class** entry permissions will be set equal to the permissions of the owning group entry. This happens regardless of whether or not the **-n** option was specified.

A directory may contain **default** ACL entries. If a file is created in a directory which contains **default** ACL entries, the entries will be added to the newly created file. Note that the default permissions specified for the file owner, file owning group, and others, will be constrained by the umask and the mode specified in the file creation call.

If an ACL contains no additional **default:user** or additional **default:group** entries and a **default:group** entry is specified for the object owning group, then a **default:class** entry must also be specified, and the permissions in the **default:group** entry for the object owning group and the permissions for the **default:class** entry must be the same.

This command may be executed on a file system that does not support ACLs, to set the permissions for the three base entries for the file owner, file owning group, and others. Additional entries and **default** entries will not be allowed in this case.

| **S** |

**EXAMPLES**

To add one ACL entry to file **filea**, giving user **archer** read permission only, type:

```
setacl -m user:archer:r-- filea
```

If an entry for user **archer** already exists, this command will set the permissions in that entry to **r--**.

To replace the entire ACL for file **filea**, adding entries for users **archer** and **fletcher**, allowing read/write access, an entry for the file owner allowing all access, an entry for the file group allowing read access only, and an entry for others disallowing all access, type:

```
setacl -s user::rwx,user:archer:rw-,user:fletcher:rw-,\
group::r--,other:--- filea
```

Note that following this command, the file permission bits would be set to **-rwxrw----**. Even though the file owning group has only read permission, the maximum permissions available to all additional **user** ACL entries, and all **group** ACL entries, are read and write, since the two additional **user** entries both specify these permissions.

To set the same ACL on file **filea** as in the above example, using the **-f** option, type:

```
setacl -f filea.acl filea
```

with file **filea.acl** edited to contain:

```
user::rwx
user:archer:rw-
user:fletcher:rw-
group::r--
other:---
```

Because the **-n** option was not specified, no **class** entry was needed. If a **class** entry had been present it would have been ignored.

**FILES**

```
/etc/passwd user IDs
/etc/group group IDs
```

**SEE ALSO**

acl(2), aclsort(3C), chmod(1), getacl(1), ls(1).

| **S** |

**NAME**

sh-posix: sh, rsh - standard and restricted POSIX.2-conformant command shells

**SYNOPSIS**

**sh** *[-|+]{aefhikmnp~~r~~stuvx}* *[-|+]{o option}...* *[-c string]* *[arg]...*

**rsh** *[-|+]{aefhikmnp~~r~~stuvx}* *[-|+]{o option}...* *[-c string]* *[arg]...*

**Remarks**

This shell is intended to conform to the shell specification of the POSIX.2 *Shell and Utility* standards. Check any standards conformance documents shipped with your system for information on the conformance of this shell to any other standards.

**List of Subheadings in DESCRIPTION**

|                   |                         |                          |
|-------------------|-------------------------|--------------------------|
| Shell Invocation  | Tilde Substitution      | Environment              |
| Options           | Command Substitution    | Functions                |
| rsh Restrictions  | Parameter Substitution  | Jobs                     |
| Definitions       | Blank Interpretation    | Signals                  |
| Commands          | File Name Generation    | Execution                |
| Simple Commands   | Quoting                 | Command Reentry          |
| Compound Commands | Arithmetic Evaluation   | Command Line Editing     |
| Special Commands  | Prompting               | emacs/gmacs Editing Mode |
| Comments          | Conditional Expressions | vi Editing Mode          |
| Aliasing          | Input/Output            |                          |

**DESCRIPTION**

**sh** is a command programming language that executes commands read from a terminal or a file.

**rsh** is a restricted version of **sh**. See the *rsh Restrictions* subsection below.

**Shell Invocation**

If the shell is invoked by an **exec\*( )** system call and the first character of argument zero (shell parameter 0) is dash (-), the shell is assumed to be a login shell and commands are read first from **/etc/profile**, then from either **.profile** in the current directory or **\$HOME/.profile** if either file exists, and finally from the file named by performing parameter substitution on the value of the environment parameter **ENV**, if the file exists. If the **-s** option is not present and an *arg* is, a path search is performed on the first *arg* to determine the name of the script to execute. When running **sh** with *arg*, the script *arg* must have read permission and any **setuid** and **setgid** settings will be ignored. Commands are read as described below.

Shell output, except for the output of some of the commands listed in the *Special Commands* subsection, is written to standard error (file descriptor 2).

**Options**

The following options are interpreted by the shell when it is invoked.

- c** *string*      Read commands from *string*.
- i**              If **-i** is present or if the shell input and output are attached to a terminal (as reported by **tty()**), the shell is interactive. In this case **SIGTERM** is ignored and **SIGINT** is caught and ignored (so that **wait** is interruptible). In all cases, **SIGQUIT** is ignored by the shell. See *signal(5)*.
- r**              The shell is a restricted shell.
- s**              If **-s** is present or if no arguments remain, commands are read from the standard input.

The remaining options and operands are described under the **set** command in the *Special Commands* subsection.

**rsh Restrictions**

**rsh** is used to set up login names and execution environments where capabilities are more controlled than those of the standard shell. The actions of **rsh** are identical to those of **sh**, except that the following are forbidden:

- Changing directory (see the **cd** special command and *cd(1)*)

- Setting the value of **SHELL**, **ENV**, or **PATH**
- Specifying path or command names containing /
- Redirecting output (>, >|, <>, and >>)

The restrictions above are enforced after the **.profile** and **ENV** files are interpreted.

When a command to be executed is found to be a shell procedure, **rsh** invokes **sh** to execute it. Thus, the end-user is provided with shell procedures accessible to the full power of the standard shell, while being restricted to a limited menu of commands. This scheme assumes that the end-user does not have write and execute permissions in the same directory.

These rules effectively give the writer of the **.profile** file complete control over user actions, by performing guaranteed set-up actions and leaving the user in an appropriate directory (probably not the login directory).

The system administrator often sets up a directory of commands (usually **/usr/rbin**) that can be safely invoked by **rsh**. HP-UX systems provide a restricted editor **red** (see *ed(1)*), suitable for restricted users.

### Definitions

|                             |                                                                                                                                                                                     |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>metacharacter</b>        | One of the following characters:<br><b> ; &amp; ( )   &lt; &gt; </b> newline space tab                                                                                              |
| <b>blank</b>                | A tab or a space.                                                                                                                                                                   |
| <b>identifier</b>           | A sequence of letters, digits, or underscores starting with a letter or underscore. Identifiers are used as names for <b>functions</b> and <b>named parameters</b> .                |
| <b>word</b>                 | A sequence of <b>characters</b> separated by one or more nonquoted <b>metacharacters</b> .                                                                                          |
| <b>command</b>              | 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.   |
| <b>special command</b>      | A command that is carried out by the shell without creating a separate process. Except for documented side effects, most special commands can be implemented as separate utilities. |
| <b>#</b>                    | Comment delimiter. A word beginning with <b>#</b> and all following characters up to a newline are ignored.                                                                         |
| <b>parameter</b>            | An <b>identifier</b> , a decimal number, or one of the characters <b>!, #, \$, *, -, ?, @, and _</b> . See the <i>Parameter Substitution</i> subsection.                            |
| <b>named parameter</b>      | A <b>parameter</b> that can be assigned a value. See the <i>Parameter Substitution</i> subsection.                                                                                  |
| <b>variable</b>             | A <b>parameter</b> .                                                                                                                                                                |
| <b>environment variable</b> | A <b>parameter</b> that is known outside the local shell, usually by means of the <b>export</b> special command.                                                                    |

| S |

### Commands

A command can be a simple command that executes an executable file, a special command that executes within the shell, or a compound command that provides flow of control for groups of simple, special, and compound commands.

### Simple Commands

A simple command is a sequence of blank-separated words that may be preceded by a parameter assignment list. (See the *Environment* subsection). 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 **128+errorstatus** if it terminates abnormally (see *signal(5)* for a list of *errorstatus* values).

A **pipeline** is a sequence of one or more commands separated by a bar (|) and optionally preceded by an exclamation mark (!). The standard output of each command but the last is connected by a pipe (see *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. If ! does not precede the pipeline, the exit status of the pipeline is the exit status of the last command in the pipeline. Otherwise, the exit status of the pipeline is



the logical negation of the exit status of the last command in the pipeline.

A *list* is a sequence of one or more pipelines separated by `;`, `&`, `&&`, or `|`, and optionally terminated by `;`, `&`, or `|&`.

- `;` Causes sequential execution of the preceding pipeline. An arbitrary number of newlines can appear in a *list*, instead of semicolons, to delimit commands.
- `&` Causes asynchronous execution of the preceding pipeline (that is, the shell does not wait for that pipeline to finish).
- `|&` Causes asynchronous execution of the preceding command or pipeline with a two-way pipe established to the parent shell. The standard input and output of the spawned command can be written to and read from by the parent shell using the `-p` option of the special commands `read` and `print`.
- `&&` Causes the *list* following it to be executed only if the preceding pipeline returns a zero value.
- `||` Causes the *list* following it to be executed only if the preceding pipeline returns a nonzero value.

Of these five symbols, `;`, `&`, and `|&` have equal precedence, which is lower than that of `&&` and `||`. The symbols `&&` and `||` also have equal precedence.

### Compound Commands

Unless otherwise stated, the value returned by a compound command is that of the last simple command executed in the compound command. The `;` segment separator can be replaced by one or more newlines.

The following keywords are recognized only as the first word of a command and when not quoted:

|                 |                   |                   |                       |                    |
|-----------------|-------------------|-------------------|-----------------------|--------------------|
| <code>!</code>  | <code>}</code>    | <code>elif</code> | <code>for</code>      | <code>then</code>  |
| <code>[[</code> | <code>case</code> | <code>else</code> | <code>function</code> | <code>time</code>  |
| <code>]]</code> | <code>do</code>   | <code>esac</code> | <code>if</code>       | <code>until</code> |
| <code>{</code>  | <code>done</code> | <code>fi</code>   | <code>select</code>   | <code>while</code> |

A compound command is one of the following.

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

Execute the *list* associated with the first *pattern* that matches *word*. The form of the patterns is identical to that used for file name generation (see the *File Name Generation* subsection). The `;;` case terminator cannot be replaced by newlines.

**for** *identifier* [**in** *word* ...] **;** **do** *list* **;** **done**

Set *identifier* to each *word* in sequence and execute the **do** *list*. If **in** *word* ... is omitted, set *identifier* to each set positional parameter instead. See the *Parameter Substitution* subsection. Execution ends when there are no more positional parameters or words in the list.

**function** *identifier* { *list* **;** }  
*identifier* ( ) { *list* **;** }

Define a function named by *identifier*. A function is called by executing its identifier as a command. The body of the function is the *list* of commands between { and }. See the *Functions* subsection.

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

Execute the **if** *list* and, if its exit status is zero, execute the first **then** *list*. Otherwise, execute the **elif** *list* (if any) and, if its exit status is zero, execute the next **then** *list*. Failing that, execute the **else** *list* (if any). If no **else** *list* or **then** *list* is executed, **if** returns a zero exit status.

**select** *identifier* [**in** *word* ...] **;** **do** *list* **;** **done**

Print the set of *words* on standard error (file descriptor 2), each preceded by a number. If **in** *word* ... is omitted, print the positional parameters instead (see the *Parameter Substitution* subsection). Print the **PS3** prompt and read a line from standard input into the parameter **REPLY**. If this line consists of the number of one of the listed *words*, set *identifier* to the corresponding *word*, execute *list*, and repeat the **PS3** prompt. If the line is empty, print the selection list again, and repeat the **PS3** prompt. Otherwise, set *identifier* to null, execute *list*, and repeat the **PS3** prompt. The select loop repeats until a **break** special command or end-of-file is encountered.

| S |

**time** *pipeline*

Execute the *pipeline* and print the elapsed time, the user time, and the system time on standard error. Note that the **time** keyword can appear anywhere in the *pipeline* to time the entire *pipeline*. To time a particular command in a *pipeline*, see *time*(1).

**until** *list* ; **do** *list* ; **done**

Execute the **until** *list*. If the exit status of the last command in the list is nonzero, execute the **do** *list* and execute the **until** *list* again. When the exit status of the last command in the **until** *list* is zero, terminate the loop. If no commands in the **do** *list* are executed, **until** returns a zero exit status.

**while** *list* ; **do** *list* ; **done**

Execute the **while** *list*. If the exit status of the last command in the list is zero, execute the **do** *list* and execute the **while** *list* again. When the exit status of the last command in the **while** *list* is nonzero, terminate the loop. If no commands in the **do** *list* are executed, **while** returns a nonzero exit status.

**(** *list* **)**

Execute *list* in a separate environment. If two adjacent open parentheses are needed for nesting, a space must be inserted between them to avoid arithmetic evaluation.

**{** *list* ; **}**

Execute *list*, but not in a separate environment. Note that **{** is a keyword and requires a trailing blank to be recognized.

**[** *expression* **]**

Evaluate *expression* and return a zero exit status when *expression* is true. See the *Conditional Expressions* subsection for a description of *expression*. Note that **[** and **]** are keywords and require blanks between them and *expression*.

**Special Commands**

Special commands are simple commands that are executed in the shell process. They permit input/output redirection. Unless otherwise indicated, file descriptor 1 (standard output) is the default output location and the exit status, when there are no syntax errors, is zero.

Commands that are marked with "%" 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. Certain errors cause a script that contains them to abort.

| **S** |

Words following commands marked with "&" 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 word-splitting and file-name generation are not performed.

**% :** [*arg*]...

(colon) Only expand parameters. A zero exit status is returned.

**% .** *file* [*arg*]...

(period) Read and execute commands from *file* and return. The commands are executed in the current shell environment. The search path specified by **PATH** is used to find the directory containing *file*. If any arguments *arg* are given, they become the positional parameters. Otherwise, the positional parameters are unchanged. The exit status is the exit status of the last command executed.

**& alias** [**-tx**] [*name*[=*value*] ]...

With *name*=*value* specified, define *name* as an alias and assign it the value *value*. A trailing space in *value* causes the next word to be checked for alias substitution.

With *name*=*value* omitted, print the list of aliases in the form *name*=*value* on standard output.

With *name* specified without =*value*, print the specified alias.

With **-t**, set tracked aliases. The value of a tracked alias is the full path name corresponding to the given *name*. The value of a tracked alias becomes undefined when the value of **PATH** is reset, but

the alias remains tracked. With *name=value* omitted, print the list of tracked aliases in the form *name=pathname* on standard output.

With **-x**, set exported aliases. An exported alias is defined across subshell environments. With *name=value* omitted, print the list of exported aliases in the form *name=value* on standard output.

Alias returns true unless a *name* is given for which no alias has been defined.

See also the **unalias** special command.

**bg** [*job*]...

Put the specified *jobs* into the background. The current job is put in the background if *job* is unspecified. See the *Jobs* subsection for a description of the format of *job*. See also the **fg** special command.

% **break** [*n*]

Exit from the enclosing **for**, **select**, **until**, or **while** loop, if any. If *n* is specified, exit from *n* levels.

**cd** [-L|-P] [*arg*]

**cd** *old new*

In the first form, change the current working directory (**PWD**) to *arg*. If *arg* is **-**, the directory is changed to the previous directory (**OLDPWD**). The shell parameter **HOME** is the default *arg*. After the **cd**, the **PWD** and **OLDPWD** environment variables are set to the new current directory and the former directory respectively.

With **-L** (default), preserve logical naming when treating symbolic links. **cd -L ..** moves the current directory one path component closer to the root directory.

With **-P**, preserve the physical path when treating symbolic links. **cd -P ..** changes the working directory to the actual parent directory of the current directory.

The shell parameter **CDPATH** defines the search path for the directory containing *arg*. Alternative directory names are separated by a colon (:). If **CDPATH** is null or undefined, the default value is 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 /, the search path is not used. Otherwise, each directory in the path is searched for *arg*. See also *cd*(1).

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.

**command** [*arg*]...

Treat *arg* as a command, but disable function lookup on *arg*. See *command*(1) for usage and description.

% **continue** [*n*]

Resume the next iteration of the enclosing **for**, **select**, **until**, or **while** loop. If *n* is specified, resume at the *n*th enclosing loop.

**echo** [*arg*]...

Print *arg* on standard output. See *echo*(1) for usage and description. See also the **print** special command.

% **eval** [*arg*]...

Read the arguments as input to the shell and execute the resulting commands. Allows parameter substitution for keywords and characters that would otherwise be unrecognized in the resulting commands.

% **exec** [*arg*]...

Parameter assignments remain in effect after the command completes. If *arg* is given, execute the command specified by the arguments in place of this shell without creating a new process. Input/output arguments may appear and affect the current process. If no arguments are given, 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 another program is invoked.

| S |

% **exit** [*n*]

Exit from the shell with the exit status specified by *n*. If *n* is omitted, the exit status is that of the last command executed. An end-of-file also causes the shell to exit, except when a shell has the **ignoreeof** option set. (See the **set** special command.)

%& **export** [*name*[=*value*]]...

%& **export -p**

Mark the given variable *names* for automatic export to the environment of subsequently executed commands. Optionally, assign values to the variables.

With no arguments, write the names and values of all exported variables to standard output,

With **-p**, write the names and values of all exported variables to standard output, in a format with the proper use of quoting, so that it is suitable for re-input to the shell as commands that achieve the same exporting results.

**fc** [**-r**] [**-e** *ename*] [*first* [*last*]]

**fc -l** [**-nr**] [*first* [*last*]]

**fc -s** [*old=new*] [*first*]

**fc -e -** [*old=new*] [*command*]

List, or edit and reexecute, commands previously entered to an interactive shell. A range of commands from *first* to *last* is selected from the last **HISTSIZE** commands typed at the terminal. The arguments *first* and *last* can be specified as a number or string. A given string is used to locate the most recent command. A negative number is used to offset the current command number.

With **-l**, list the commands on standard output. Without **-l**, invoke the editor program *ename* on a file containing these keyboard commands. If *ename* is not supplied, the value of the parameter **FCEDIT** (default **/usr/bin/ed**) is used as the editor. Once editing has ended, the commands (if any) are executed. If *last* is omitted, only the command specified by *first* is used. If *first* is not specified, the default is the previous command for editing and **-16** for listing.

With **-r**, reverse the order of the commands.

With **-n**, suppress command numbers when listing.

With **-s**, reexecute the command without invoking an editor.

The *old=new* argument replaces the first occurrence of string *old* in the command to be reexecuted by the string *new*.

**fg** [*job*]...

Bring each *job* into the foreground in the order specified. If no *job* is specified, bring the current job into the foreground. See the *Jobs* subsection for a description of the format of *job*. See also the **bg** special command.

**getopts** *optstring name* [*arg*]...

Parse the argument list, or the positional parameters if no arguments, for valid options. On each execution, return the next option in *name*. See **getopts**(1) for usage and description.

An option begins with a **+** or a **-**. An argument not beginning with **+** or **-**, or the argument **--**, ends the options. *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.

For an option specified as **-letter**, *name* is set to *letter*. For an option specified as **+letter**, *name* is set to **+letter**. The index of the next *arg* is stored in **OPTIND**. The option argument, if any, is stored in **OPTARG**. If no option argument is found, or the option found does not take an argument, **OPTARG** is unset.

A leading **:** in *optstring* causes **getopts** to store the letter of an invalid option in **OPTARG**, and to set *name* to **?** for an unknown option and to **:** when a required option argument is missing. Otherwise, **getopts** prints an error message. The exit status is nonzero when there are no more options.

& **hash** [*utility*]...

& **hash -r**

Affect the way the current shell environment remembers the locations of utilities. With *utility*, add utility locations to a list of remembered locations. With no arguments, print the contents of the list.

| S |

With **-r**, forget all previously remembered utility locations.

**jobs** [-lnp] [job]...

List information about each given job, or all active jobs if *job* is not specified. With **-l**, list process IDs in addition to the normal information. With **-n**, display only jobs that have stopped or exited since last notified. With **-p**, list only the process group. See the *Jobs* subsection for a description of the format of *job*.

**kill** [-s *signal*] *process* ...

**kill** -l

**kill** [-*signal*] *process* ...

Send either signal 15 (**SIGTERM**, terminate) or the specified *signal* to the specified jobs or processes. If the signal being sent is **TERM** (terminate) or **HUP** (hangup), the job or process is sent a **CONT** (continue) signal when stopped. See *kill(1)* for usage and description.

With **-l**, list the signal names and numbers.

**let** *arg* ...

(( *arg* ... ))

Evaluate each *arg* as a separate arithmetic expression. See the *Arithmetic Evaluation* subsection for a description of arithmetic expression evaluation. The exit status is 0 if the value of the last expression is nonzero, and 1 otherwise.

% **newgrp** [-] [*group*]

Replace the current shell with a new one having *group* as the user's group. The default group is the user's login group. With **-**, also execute the user's **.profile** and **\$ENV** files. See *newgrp(1)* for usage and description. Equivalent to **exec newgrp arg ...**

**print** [-nprRsu[n]] [*arg*]...

The shell output mechanism. With no options or with option **-** or **--**, print the arguments on standard output as described in *echo(1)*. See also *printf(1)*.

With **-n**, do not add a newline character to the output.

With **-p**, write the arguments onto the pipe of the process spawned with **|&** instead of standard output.

With **-R** or **-r** (raw mode), ignore the escape conventions of **echo**. With **-R**, print all subsequent arguments and options other than **-n**.

With **-s**, write the arguments into the history file instead of to standard output.

With **-u**, specify a one-digit file descriptor unit number *n* on which the output will be placed. The default is **1** (standard output).

**pwd** [-L|-P]

Print the name of the current working directory (equivalent to **print -r - \$PWD**). With **-L** (the default), preserve the logical meaning of the current directory. With **-P**, preserve the physical meaning of the current directory if it is a symbolic link. See also the **cd** special command, *cd(1)*, *ln(1)*, and *pwd(1)*.

**read** [-prsu[n]] [*name?prompt*] [*name*]...

The shell input mechanism. Read one line (by default, from standard input) and break it up into words using the characters in **IFS** as separators. The first word is assigned to the first *name*, the second word to the second *name*, and so on; the remaining words are assigned to the last *name*. See also *read(1)*. The return code is **0**, unless an end-of-file is encountered.

With **-p**, take the input line from the input pipe of a process spawned by the shell using **|&**. An end-of-file with **-p** causes cleanup for this process so that another process can be spawned.

With **-r** (raw mode), a **\** at the end of a line does not signify line continuation.

With **-s**, save the input as a command in the history file.

With **-u**, specify a one-digit file descriptor unit to read from. The file descriptor can be opened with the **exec** special command. The default value of *n* is **0** (standard input). If *name* is omitted, **REPLY** is used as the default *name*.

If the first argument contains a `?`, the remainder of the argument is used as a *prompt* when the shell is interactive.

If the given file descriptor is open for writing and is a terminal device, the prompt is placed on that unit. Otherwise, the prompt is issued on file descriptor 2 (standard error).

```
%& readonly [name[=value]]...
```

```
%& readonly -p
```

Mark the given *names* read only. These names cannot be changed by subsequent assignment.

With `-p`, write the names and values of all read-only variables to standard output in a format with the proper use of quoting so that it is suitable for re-input to the shell as commands that achieve the same attribute-setting results.

```
% return [n]
```

Cause a shell function to return to the invoking script with the return status specified by *n*. If *n* is omitted, the return status is that of the last command executed. Only the low 8 bits of *n* (decimal 0 to 255) are passed back to the caller. If `return` is invoked while not in a function or a `.` script (see the `.` special command), it has the same effect as an `exit` command.

```
% set [{"-|+"}abCefhkmnopstuvx] [{"-|+"}o option]... [{"-|+"}A name] [arg]...
```

Set (`-`) or clear (`+`) execution options or perform array assignments (`-A`, `+A`). All options except `-A` and `+A` can be supplied in a shell invocation (see the *SYNOPSIS* section and the *Shell Invocation* subsection).

Using `+` instead of `-` before an option causes the option to be turned off. These options can also be used when invoking the shell. The current list of set single-letter options is contained in the shell variable `-`. It can be examined with the command `echo $-`.

The `-` and `+` options can be intermixed in the same command, except that there can be only one `-A` or `+A` option.

Unless `-A` or `+A` is specified, the remaining *arg* arguments are assigned consecutively to the positional parameters `1`, `2`, ....

The `set` command with neither arguments nor options displays the names and values of all shell parameters on standard output. See also *env*(1).

The options are defined as follows.

- `-A` Array assignment. Unset the variable *name* and assign values sequentially from the list *arg*. With `+A`, do not unset the variable *name* first.
- `-a` Automatically export subsequently defined parameters.
- `-b` Cause the shell to notify the user asynchronously of background jobs as they are completed. When the shell notifies the user that a job has been completed, it can remove the job's process ID from the list of those known in the current shell execution environment.
- `-C` Prevent redirection `>` from truncating existing files. Requires `>|` to truncate a file when turned on.
- `-e` Execute the `ERR` trap, if set, and exit *if* a command has a nonzero exit status, and is not part of the compound list following a `if`, `until`, or `while` keyword, and is not part of an AND or OR list, and is not a pipeline preceded by the `!` reserved word. This mode is disabled while reading profiles.
- `-f` Disable file name generation.
- `-h` Specify that each command whose name is an *identifier* becomes a tracked alias when first encountered.
- `-k` Place all parameter assignment arguments (not just those that precede the command name) into the environment for a command.
- `-m` Run background jobs in a separate process group and print a line upon completion. The exit status of background jobs is reported in a completion message. This option is turned on automatically for interactive shells.
- `-n` Read commands and check them for syntax errors, but do not execute them. The `-n` option is ignored for interactive shells.

| S |

- o Set an *option* argument from the following list. Repeat the -o option to specify additional *option* arguments.
  - allexport** Same as -a.
  - bgnice** Run all background jobs at a lower priority.
  - emacs** Use a **emacs**-style inline editor for command entry.
  - errexit** Same as -e.
  - gmacs** Use a **gmacs**-style inline editor for command entry.
  - ignoreeof** Do not exit from the shell on end-of-file (*eof*, as defined by **stty**; default is ^D). The **exit** special command must be used.
  - keyword** Same as -k.
  - markdirs** Append a trailing / to all directory names resulting from file name generation.
  - monitor** Same as -m.
  - noclobber** Same as -C.
  - noexec** Same as -n.
  - noglob** Same as -f.
  - nolog** Do not save function definitions in history file.
  - notify** Same as -b.
  - nounset** Same as -u.
  - privileged** Same as -p.
  - trackall** Same as -h.
  - verbose** Same as -v.
  - vi** Use a **vi**-style inline editor for command entry.
  - viraw** Process each character as it is typed in **vi** mode (always on).
  - xtrace** Same as -x.
- p Disable 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 user ID (group ID) is not equal to the real user ID (group ID). Turning this off causes the effective user ID and group ID to be set to the real user ID and group ID.
- s Sort the positional parameters.
- t 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.
- Turn off -x and -v options and stop examining arguments for options.
- Do not change any of the options; useful in setting parameter 1 to a value beginning with -. If no arguments follow this option, the positional parameters are unset.

% **shift** [*n*]

Rename the positional parameters from *n+1* ... to 1 .... The default value of *n* is 1. *n* can be any arithmetic expression that evaluates to a nonnegative number less than or equal to \$#.

**test** [*expr*]

Evaluate conditional expression *expr*. See *test*(1) for usage and description. See also the *Conditional Expressions* subsection.

The arithmetic comparison operators are not restricted to integers. They allow any arithmetic expression. The following additional primitive expressions are allowed:

- L** *file* True if *file* is a symbolic link.
- e** *file* True if *file* exists.
- file1* **-nt** *file2* True if *file1* is newer than *file2*.
- file1* **-ot** *file2* True if *file1* is older than *file2*.
- file1* **-ef** *file2* True if *file1* has the same device and i-node number as *file2*.

% **times**

Print the accumulated user and system times for the shell and for processes run from the shell.

| S |

% **trap** [*arg*] [*sig*]...

Set *arg* as a command that is read and executed when the shell receives a *sig* signal. (Note that *arg* is scanned once when the trap is set and once when the trap is taken.) Each *sig* can be given as the number or name of a signal. Letter case is ignored. For example, **3**, **QUIT**, **quit**, and **SIGQUIT** all specify the same signal. Use **kill -l** to get a list of signals.

Trap commands are executed in signal number order. Any attempt to set a trap on a signal that was ignored upon entering the current shell is ineffective. Traps remain in effect for a given shell until explicitly changed with another **trap** command; that is, a trap set within a function will remain in effect even after the function returns.

If *arg* is - (or if *arg* is omitted and the first *sig* is numeric), reset all traps for each *sig* to their original values.

If *arg* is the null string ('' or ""), each *sig* is ignored by the shell and by the commands it invokes.

If *sig* is **DEBUG**, then *arg* is executed after each command. If *sig* is **ERR**, *arg* is executed whenever a command has a nonzero exit code. If *sig* is **0** or **EXIT**, the command *arg* is executed on exit from the shell.

With no arguments, print a list of commands associated with each signal name.

& **typeset** [[-|+]**LRZfilrtux**[*n*]] [*name*[=*value*]]... *name*=*value* [*name*=*value*]...

Assign types and a value to a local named parameter *name*. See also the **export** special command. Parameter assignments remain in effect after the command completes. When invoked inside a function, create a new instance of the parameter *name*. The parameter value and type are restored when the function completes.

The following list of attributes can be specified. Use + instead of - to turn the options off.

- L** Left justify and remove leading blanks from *value*. If *n* is nonzero, it defines the width of the field; otherwise, it is determined by the width of the value of first assignment. When *name* is assigned, the value is filled on the right with blanks or truncated, if necessary, to fit into the field. Leading zeros are removed if the **-Z** option is also set. The **-R** option is turned off. Flagged as **leftjust** *n*.
- R** Right justify and fill with leading blanks. If *n* is nonzero, 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 parameter is reassigned. The **-L** option is turned off. Flagged as **rightjust** *n*.
- Z** Right justify and fill with leading zeros if the first nonblank character is a digit and the **-L** option has not been set. If *n* is nonzero it defines the width of the field; otherwise, it is determined by the width of the value of first assignment. Flagged as **zerofill** *n* plus the flag for **-L** or **-R**.
- f** Cause *name* to refer to function names rather than parameter names. No assignments can be made to the *name* declared with the **typeset** statement. The only other valid options are **-t** (which turns on execution tracing for this function) and **-x** (which allows the function to remain in effect across shell procedures executed in the same process environment). Flagged as **function**.
- i** Parameter is an integer. This makes arithmetic faster. If *n* is nonzero it defines the output arithmetic base; otherwise, the first assignment determines the output base. Flagged as **integer** [**base** *n*].
- l** Convert all uppercase characters to lowercase. The uppercase **-u** option is turned off. Flagged as **lowercase**.
- r** Mark any given *name* as "read only". The name cannot be changed by subsequent assignment. Flagged as **readonly**.
- t** Tag the named parameters. Tags are user-definable and have no special meaning to the shell. Flagged as **tagged**.
- u** Convert all lowercase characters to uppercase characters. The lowercase **-l** option is turned off. Flagged as **uppercase**.
- x** Mark any given *name* for automatic export to the environment of subsequently executed commands. Flagged as **export**.

| S |



**typeset** alone displays a list of parameter names, prefixed by any flags specified above.

**typeset** - displays the parameter names followed by their values. Specify one or more of the option letters to restrict the list. Some options are incompatible with others.

**typeset** + displays the parameter names alone. Specify one or more of the option letters to restrict the list. Some options are incompatible with others.

**ulimit** [-HSacdfnst] [*limit*]

Set or display a resource limit. The limit for a specified resource is set when *limit* is specified. The value of *limit* can be a number in the unit specified with each resource, or the keyword **unlimited**.

The **-H** and **-S** flags specify whether the hard limit or the soft limit is set for the given resource. A hard limit cannot be increased once it is set. A soft limit can be increased up to the hard limit. If neither **-H** nor **-S** 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, the limit name and unit are printed before the value.

If no option is given, **-f** is assumed.

- a** List all of the current resource limits.
- c** The number of 512-byte blocks in the size of core dumps.
- d** The number of kilobytes in the size of the data area.
- f** The number of 512-byte blocks in files written by child processes (files of any size can be read).
- n** The number of file descriptors.
- s** The number of kilobytes in the size of the stack area.
- t** The number of seconds to be used by each process.

**umask** [-S] [*mask*]

Set the user file-creation mask *mask*. *mask* can be either an octal number or a symbolic value as described in *umask*(1). A symbolic value shows permissions that are unmasked. An octal value shows permissions that are masked off.

Without *mask*, print the current value of the mask. With **-S**, print the value in symbolic format. Without **-S**, print the value as an octal number. The output from either form can be used as the *mask* of a subsequent invocation of **umask**.

**unalias** *name* ...

**unalias** -a

Remove each *name* from the alias list. With **-a**, remove all **alias** definitions from the current shell execution environment. See also the **alias** special command.

% **unset** [-fv] *name* ...

Remove the named shell parameters from the parameter list. Their values and attributes are erased. Read-only variables cannot be unset. With **-f**, *names* refer to function names. With **-v**, *names* refer to variable names. Unsetting **\_**, **ERRNO**, **LINENO**, **MAILCHECK**, **OPTARG**, **OPTIND**, **RANDOM**, **SECONDS**, and **TMOUT** removes their special meaning, even if they are subsequently assigned to.

**wait** [*job*]

Wait for the specified *job* to terminate or stop, and report its status. This status becomes the return code for the **wait** command. Without *job*, wait for all currently active child processes to terminate or stop. The termination status returned is that of the last process. See the *Jobs* subsection for a description of the format of *job*.

**whence** [-pv] *name* ...

For each *name*, indicate how it would be interpreted if used as a command name. With **-v**, produce a more verbose report. With **-p** do a path search for *name*, disregarding any use as an alias, a function, or a reserved word.

## Comments

A **word** beginning with **#** causes that word and all the following characters up to a newline 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 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 additional aliases. If the last character of the alias value is a **blank**, the word following the alias is also checked for alias substitution. Aliases can be used to redefine special commands, but cannot be used to redefine the keywords listed in the *Compound Commands* subsection. Aliases can be created, listed, and exported with the **alias** command and can be removed with the **unalias** command. Exported aliases remain in effect for subshells but must be reinitialized for separate invocations of the shell (see the *Shell Invocation* subsection).

Aliasing is performed when scripts are read, not while they are executed. Therefore, for it to take effect, an **alias** must be executed before the command referring to the alias is read.

Aliases are frequently used as a shorthand for full path names. An option to the aliasing facility allows the value of the alias to be automatically set to the full path name of the corresponding command. These aliases are called **tracked** aliases. The value of a tracked alias is defined the first time the identifier is read and becomes undefined each time the **PATH** variable is reset. These aliases remain tracked so that the next reference will redefine the value. Several tracked aliases are compiled into the shell. The **-h** option of the **set** command converts each command name that is an **identifier** into a tracked alias.

The following **exported aliases** are compiled into the shell but can be unset or redefined:

```
autoload='typeset -fu'
command='command '
functions='typeset -f'
history='fc -l'
integer='typeset -i'
local=typeset
nohup='nohup '
r='fc -e -'
stop='kill -STOP'
suspend='kill -STOP $$'
type='whence -v'
```

## Tilde Substitution

After alias substitution is performed, each word is checked to see if it begins with an unquoted tilde (~). If it does, the word up to a / is checked to see if it matches a user name in the **/etc/passwd** file. 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 ~ alone or before a / is replaced by the value of the **HOME** parameter. A ~ followed by a + or - is replaced by the value of the parameter **PWD** and **OLDPWD**, respectively. In addition, tilde substitution is attempted when the value of a parameter assignment begins with a ~.

## Command Substitution

The standard output from a command enclosed in parenthesis preceded by a dollar sign (**\$(...)**) or a pair of grave accents (**`...`**) can be used as part or all of a word; trailing newlines are removed. In the second (archaic) form, the string between the accents is processed for special quoting characters before the command is executed. See the *Quoting* subsection. The command substitution **\$(cat file)** can be replaced by the equivalent but faster **\$(<file)**. Command substitution of most special commands that do not perform input/output redirection are carried out without creating a separate process.

An arithmetic expression enclosed in double parenthesis preceded by a dollar sign (**\$( (... ))**) is replaced by the value of the arithmetic expression within the double parenthesis. See the *Arithmetic Evaluation* subsection for a description of arithmetic expressions.

## Parameter Substitution

A **parameter** is an identifier, one or more decimal digits, or one of the characters **!, #, \$, \*, -, ?, @,** and **\_**. A **named parameter** (a parameter denoted by an identifier) has a value and zero or more attributes. Named parameters can be assigned values and attributes with the **typeset** special command. Exported parameters pass values and attributes to the environment.

The shell supports a limited one-dimensional array facility. An element of an array parameter is referenced by a subscript. A subscript is denoted by a `[`, followed by an arithmetic expression, followed by a `]`. See the *Arithmetic Evaluation* subsection. To assign values to an array, use `set -A name value ...`. The value of all subscripts must be in the range of 0 through 1023. Arrays need not be declared. Any reference to a named parameter with a valid subscript is legal and an array is created if necessary. Referencing an array parameter without a subscript is equivalent to referencing the first element.

If the `-i` integer attribute is set for *name*, the *value* is subject to arithmetic evaluation.

Positional parameters, parameters denoted by a number, can be assigned values with the `set` special command. Parameter 0 is set from argument zero when the shell is invoked.

Use the prefix character `$` to specify the value of a parameter for substitution.

`$parameter`

`${parameter}`

`${parameter[subscript]}`

Substitute the value of the parameter, if any. Braces are required when *parameter* is followed by a letter, digit, or underscore that should not be interpreted as part of its name or when a named parameter is subscripted. If *parameter* is one or more digits, it is a positional parameter. A positional parameter of more than one digit must be enclosed in braces. The shell reads all the characters from `${` to the matching `}` as part of the same word, even if it contains braces or metacharacters.

If *parameter* is `*` or `@`, all the positional parameters, starting with 1, are substituted (separated by a field separator character). See the *Quoting* subsection.

If an array parameter with subscript `*` or `@` is used, the value for each element is substituted (separated by a field separator character).

`${#parameter}`

If *parameter* is `*` or `@`, the number of positional parameters is substituted. Otherwise, the length of the value of the *parameter* is substituted.

`${#parameter[*]}`

Substitute the number of elements in the array.

`${parameter:-word}`

If *parameter* is set and is nonnull, substitute its value; otherwise, substitute *word*.

`${parameter:=word}`

If *parameter* is not set or is null, set it to *word*; then substitute the value of the parameter. Positional parameters may not be assigned in this way.

`${parameter:?word}`

If *parameter* is set and is nonnull, substitute its value; otherwise, print *word* and exit from the shell. If *word* is omitted, a standard message is printed.

`${parameter:+word}`

If *parameter* is set and is nonnull, substitute *word*; otherwise, substitute nothing.

`${parameter#pattern}`

`${parameter##pattern}`

If the shell *pattern* matches the beginning of the value of *parameter*, the value of this substitution is the value of the *parameter* with the matched portion deleted; otherwise, the value of this *parameter* is substituted. In the former case, the smallest matching pattern is deleted; in the latter case, the largest matching pattern is deleted. These characters `#` or `%` should be escaped by a backslash (`\`) or quotes (`"`).

`${parameter%pattern}`

`${parameter%%pattern}`

If the shell *pattern* matches the end of the value of *parameter*, the value of *parameter* with the matched part is deleted; otherwise, substitute the value of *parameter*. In the former, the smallest matching pattern is deleted; in the latter, the largest matching pattern is deleted. These characters `#` or `%` should be escaped by a backslash (`\`) or quotes (`"`).

In the above, *word* is not evaluated unless it is used as the substituted string. Thus, in the following example, `pwd` is executed only if `d` is not set or is null:

```
echo ${d:-$(pwd)}
```

If the colon (:) is omitted from the above expressions, the shell only checks to determine whether or not *parameter* is set.

- The following parameters are set automatically by the shell:

|           |                                                                                                                                                                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0         | The string used to call the command or script, set from invocation argument zero.                                                                                                                                                                                                                                                                   |
| 1, 2, ... | The positional parameters.                                                                                                                                                                                                                                                                                                                          |
| *, @      | All the set positional parameters, separated by a field separator character. See the <i>Quoting</i> subsection.                                                                                                                                                                                                                                     |
| #         | The number of set positional parameters in decimal.                                                                                                                                                                                                                                                                                                 |
| -         | Flags supplied to the shell on invocation or by the <b>set</b> command.                                                                                                                                                                                                                                                                             |
| ?         | The decimal exit status returned by the last executed command.                                                                                                                                                                                                                                                                                      |
| \$        | The process number of this shell.                                                                                                                                                                                                                                                                                                                   |
| _         | Initially, the absolute path name 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 <b>MAIL</b> file when checking for mail. |
| !         | The process number of the last background command invoked.                                                                                                                                                                                                                                                                                          |
| ERRNO     | The value of <b>errno</b> as set by the most recently failed system call. This value is system-dependent and is intended for debugging purposes.                                                                                                                                                                                                    |
| LINENO    | The line number of the current line within the script or function being executed.                                                                                                                                                                                                                                                                   |
| OLDPWD    | The previous working directory set by the <b>cd</b> command.                                                                                                                                                                                                                                                                                        |
| OPTARG    | The value of the last option argument processed by the <b>getopts</b> special command.                                                                                                                                                                                                                                                              |
| OPTERR    | If set to 0, <b>OPTERR</b> will suppress error messages from the <b>getopts</b> special command. <b>OPTERR</b> is initially set to 1.                                                                                                                                                                                                               |
| OPTIND    | The index of the last option argument processed by the <b>getopts</b> special command.                                                                                                                                                                                                                                                              |
| PPID      | The process number of the parent of the shell.                                                                                                                                                                                                                                                                                                      |
| PWD       | The present working directory set by the <b>cd</b> command.                                                                                                                                                                                                                                                                                         |
| RANDOM    | Each time this parameter is evaluated, 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 <b>RANDOM</b> .                                                                                                                                 |
| REPLY     | Set by the <b>select</b> compound command, and by the <b>read</b> special command when no <i>name</i> is supplied.                                                                                                                                                                                                                                  |
| SECONDS   | Each time this parameter is referenced, the number of seconds since shell invocation is returned. If this parameter is assigned a value, the value returned upon reference is the value that was assigned plus the number of seconds since the assignment.                                                                                          |

- The following parameters are used by the shell:

|         |                                                                                                                                                                                                                                                                                   |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CDPATH  | The search path for the <b>cd</b> command, a list of directories separated by colons.                                                                                                                                                                                             |
| COLUMNS | If this variable is set, its value is used to define the width of the edit window for the shell edit modes and for printing <b>select</b> lists.                                                                                                                                  |
| EDITOR  | If the value of this variable ends in <b>emacs</b> , <b>gmacs</b> , or <b>vi</b> and the <b>VISUAL</b> variable is not set, the corresponding option is turned on (see the <b>set</b> special command).                                                                           |
| ENV     | If this parameter is set, parameter substitution is performed on the value to generate the path name of the script to be executed when the shell is invoked (see the <i>Invocation</i> subsection). This file is typically used for <b>alias</b> and <b>function</b> definitions. |
| FCEDIT  | The default editor name for the <b>fc</b> command.                                                                                                                                                                                                                                |
| FPATH   | The search path for function definitions, a list of directories separated by colons. This path is searched when a function with the <b>-u</b> attribute is referenced and when a command is not found. If an executable file is found, then it is read and executed in the        |

| S |

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | current environment.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>HISTFILE</b>    | If this parameter is set when the shell is invoked, its value is the path name of the file that is used to store the command history. The default value is <code>\$HOME/.sh_history</code> . If the user is a superuser and no <b>HISTFILE</b> is given, then no history file is used. See the <i>Command Reentry</i> subsection and the <i>WARNINGS</i> section.                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>HISTSIZE</b>    | If this parameter is set when the shell is invoked, the number of previously entered commands accessible to this shell will be greater than or equal to this number. The default is 128.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>HOME</b>        | The default argument (home directory) for the <b>cd</b> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>IFS</b>         | Internal field separators, normally space, tab, and newline, that are used to separate command words resulting from command or parameter substitution and for separating words with the special command <b>read</b> . The first character of the <b>IFS</b> parameter is used to separate arguments for the <code>\$*</code> substitution (see the <i>Quoting</i> subsection). If the value of <b>IFS</b> is space, tab, and newline, or if <b>IFS</b> is unset and it is being used to separate the results of command or parameter substitution, any sequence of <b>IFS</b> characters serves to delimit words; otherwise, each occurrence of a character in <b>IFS</b> serves to delimit a word. If the value of <b>IFS</b> is null, no word splitting is done. |
| <b>LANG</b>        | The locale of your system, which is made up of three parts: language, territory, and code set. The default is the <b>C</b> locale. See <i>environ</i> (5).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>LC_ALL</b>      | The overriding value for <b>LANG</b> and the <b>LC_*</b> variables. See <i>environ</i> (5).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>LC_COLLATE</b>  | The collating sequence to use when sorting names and when character ranges occur in patterns. See <i>environ</i> (5).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>LC_CTYPE</b>    | The character classification information to use. Changing the value of <b>LC_CTYPE</b> after the shell has started does not affect the lexical processing of shell commands in the current shell execution environment or its subshells. See <i>environ</i> (5).<br><br>The shell uses <b>LC_CTYPE</b> to detect nonprintable characters in the input and tries to handle them when the <b>emacs</b> , <b>gmacs</b> , or <b>vi</b> editing mode is selected. Not starting a new shell session after setting <b>LC_CTYPE</b> may affect the display of nonprintable input characters in the <b>emacs</b> , <b>gmacs</b> , or <b>vi</b> editing mode.                                                                                                                |
| <b>LC_MESSAGES</b> | The language in which system messages appear, and the language that the system expects for user input of <b>yes</b> and <b>no</b> strings. See <i>environ</i> (5).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>LC_MONETARY</b> | The currency symbol and monetary value format. See <i>environ</i> (5).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>LC_NUMERIC</b>  | The numeric format. See <i>environ</i> (5).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>LC_TIME</b>     | The date and time format. See <i>environ</i> (5).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>LINES</b>       | If this variable is set, the value is used to determine the column length for printing <b>select</b> lists. <b>select</b> lists print vertically until about two-thirds of <b>LINES</b> lines are filled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>MAIL</b>        | If this parameter is set to the name of a mail file and the <b>MAILPATH</b> parameter is not set, the shell informs the user of arrival of mail in the specified file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>MAILCHECK</b>   | How often (in seconds) the shell checks for changes in the modification time of any of the files specified by the <b>MAILPATH</b> or <b>MAIL</b> parameters. The default value is 600 seconds. When the time has elapsed, the shell checks before issuing the next prompt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>MAILPATH</b>    | A list of file names separated by colons. If this parameter is set, the shell informs the user of any modifications to the specified files that have occurred within the last <b>MAILCHECK</b> seconds. Each file name can be followed by a <b>?</b> and a message to be printed, in which case the message will undergo parameter and command substitution with the parameter <code>\$_</code> defined as the name of the changed file. The default message is <b>you have mail in \$_</b> .                                                                                                                                                                                                                                                                      |
| <b>NLSPATH</b>     | The search path for message catalogs, a list of directories separated by colons.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>PATH</b>        | The search path for commands, a list of directories separated by colons. See the <i>Execution</i> subsection.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

|               |                                                                                                                                                                                                                                                                            |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PS1</b>    | The value of this parameter is expanded for parameter substitution, to define the primary prompt string. The default value is "\$ ". The character ! in the primary prompt string is replaced by the command number. See the <i>Command Reentry</i> subsection.            |
| <b>PS2</b>    | Secondary prompt string for command completion. The default value is "> ".                                                                                                                                                                                                 |
| <b>PS3</b>    | Selection prompt string used within a <b>select</b> loop. If unset, it defaults to "#? ".                                                                                                                                                                                  |
| <b>PS4</b>    | Execution trace string that precedes each line of an execution trace. See the <b>set -x</b> special command. If unset, it defaults to "+ ".                                                                                                                                |
| <b>SHELL</b>  | The path name of the shell is kept in the environment. When invoked, the shell is restricted if the value of this variable contains an <b>r</b> in the base name.                                                                                                          |
| <b>TMOUT</b>  | If set to a value greater than zero, the shell will terminate if a command is not entered within the prescribed number of seconds after issuing the <b>PS1</b> prompt. (Note that the shell can be compiled with a maximum bound for this value which cannot be exceeded.) |
| <b>VISUAL</b> | Invokes the corresponding option when the value of this variable ends in <b>emacs</b> , <b>gmacs</b> , or <b>vi</b> . See the <b>set -o</b> special command.                                                                                                               |

The shell gives default values to **IFS**, **MAILCHECK**, **PATH**, **PS1**, **PS2**, and **TMOUT**. On the other hand, **MAIL**, **ENV**, **HOME**, and **SHELL** are never set automatically by the shell (although **HOME**, **MAIL**, and **SHELL** are set by **login**; see *login(1)*).

### Blank Interpretation

After parameter and command substitution, the results of substitution are scanned for field separator characters (defined in **IFS**), and split into distinct arguments when such characters are found. **sh** retains explicit null arguments (" " or ' ') but removes implicit null arguments (those resulting from parameters that have null values).

### File Name Generation

Following substitution, each command *word* is processed as a pattern for file name expansion unless expansion has been disabled with the **set -f** special command. The form of the patterns is the *Pattern Matching Notation* defined in *regex(5)*. The word is replaced with sorted file names matching the pattern. If no file name is found that matches the pattern, the word is left unchanged.

In addition to the notation described in *regex(5)*, **sh** recognizes composite patterns made up of one or more patterns separated from each other with a |. Composite patterns can be formed with one or more of the following:

|                           |                                                         |
|---------------------------|---------------------------------------------------------|
| ? ( <i>pattern-list</i> ) | Matches any one of the given patterns.                  |
| * ( <i>pattern-list</i> ) | Matches zero or more occurrences of the given patterns. |
| + ( <i>pattern-list</i> ) | Matches one or more occurrences of the given patterns.  |
| @ ( <i>pattern-list</i> ) | Matches exactly one of the given patterns.              |
| ! ( <i>pattern-list</i> ) | Matches anything, except one of the given patterns.     |

### Quoting

Each of the metacharacters (see the *Definitions* subsection) has a special meaning to the shell and terminates a word unless quoted. A character may be **quoted** (that is, made to stand for itself) by preceding it with a backslash (\). The pair *\newline* is ignored; the current and following lines are concatenated.

All characters enclosed between a pair of apostrophes ('...') are quoted. An apostrophe cannot appear within apostrophes.

Parameter and command substitution occurs inside quotation marks ("..."). \ quotes the characters \, ', and \$.

Inside grave accent marks (`...`), \ quotes the characters \, ', and \$. If the grave accents occur within quotation marks, \ also quotes the character ".

The meanings of \$\* and @\$ are identical when not quoted or when used as a parameter assignment value or as a file name. However, when used as a command argument, "\$\*" is equivalent to "\$1\$d\$2d...", whereas "\$@" is equivalent to "\$1"d"\$2"d..." (where *d* is the first character of **IFS**),

The special meaning of keywords or aliases can be removed by quoting any character of the name. The recognition of function names or special command names cannot be altered by quoting them.

### Arithmetic Evaluation

Integer arithmetic is provided with the special command **let**. Evaluations are performed using long integer arithmetic. Constants take the form *base*#*n* or *n*, where *base* is a decimal number between two and thirty-six representing the arithmetic base and *n* is a number in that base. If *base*# is omitted, base 10 is used.

An arithmetic expression uses the same syntax, precedence, and associativity of expression as the C language. All the integral operators, other than **++**, **--**, **?:**, and **,** are supported. Variables can be referenced by name within an arithmetic expression without using the parameter substitution syntax. When a variable is referenced, its value is evaluated as an arithmetic expression.

A variable can be typed as an integer with the **-i** option of the **typeset** special command, as in **typeset -i[base] name**. Arithmetic evaluation is performed on the value of each assignment to a variable with the **-i** attribute. If you do not specify an arithmetic base, the first assignment to the variable determines the arithmetic base. This base is used when parameter substitution occurs.

Since many of the arithmetic operators require quoting, an alternative form of the **let** command is provided. For any command beginning with **(**, all characters until the matching **)** are treated as a quoted expression. More precisely, **((...))** is equivalent to **let "..."**.

Arithmetic expressions given with **let** command, **((...))**, and **\$(...)** will be processed according to ISOC standard with the exception of **++**, **--**, **?:**, and **,** operators.

### Prompting

When used interactively, the shell prompts with the value of **PS1** before reading a command. Whenever a newline is received and further input is needed to complete a command, the secondary prompt (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. Word splitting and file name generation are not performed on the words between **[[** and **]]**. (See also the **test** special command.) Each expression can be constructed from one or more of the following unary or binary expressions:

|                                        |                                                                                                                                                                                     |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a file</b>                         | True, if <i>file</i> exists.                                                                                                                                                        |
| <b>-b file</b>                         | True, if <i>file</i> exists and is a block special file.                                                                                                                            |
| <b>-c file</b>                         | True, if <i>file</i> exists and is a character special file.                                                                                                                        |
| <b>-d file</b>                         | True, if <i>file</i> exists and is a directory.                                                                                                                                     |
| <b>-e file</b>                         | True, if <i>file</i> exists.                                                                                                                                                        |
| <b>-f file</b>                         | True, if <i>file</i> exists and is an ordinary file.                                                                                                                                |
| <b>-g file</b>                         | True, if <i>file</i> exists and has its setgid bit set.                                                                                                                             |
| <b>-h file</b>                         | True, if <i>file</i> exists and is a symbolic link.                                                                                                                                 |
| <b>-k file</b>                         | True, if <i>file</i> exists and has its sticky bit set.                                                                                                                             |
| <b>-n string</b>                       | True, if length of <i>string</i> is nonzero.                                                                                                                                        |
| <b>-o option</b>                       | True, if the set option named <i>option</i> is on.                                                                                                                                  |
| <b>-p file</b>                         | True, if <i>file</i> exists and is a fifo special file or a pipe.                                                                                                                   |
| <b>-r file</b>                         | True, if <i>file</i> exists and is readable by current process.                                                                                                                     |
| <b>-s file</b>                         | True, if <i>file</i> exists and has a size greater than zero.                                                                                                                       |
| <b>-t fildes</b>                       | True, if file descriptor number <i>fildes</i> is open and is associated with a terminal device.                                                                                     |
| <b>-u file</b>                         | True, if <i>file</i> exists and has its setuid bit set.                                                                                                                             |
| <b>-w file</b>                         | True, if <i>file</i> exists and is writable by the current process.                                                                                                                 |
| <b>-x file</b>                         | True, if <i>file</i> exists and is executable by the current process. If <i>file</i> exists and is a directory, then the current process has permission to search in the directory. |
| <b>-z string</b>                       | True, if length of <i>string</i> is zero.                                                                                                                                           |
| <b>-L file</b>                         | True, if <i>file</i> exists and is a symbolic link.                                                                                                                                 |
| <b>-O file</b>                         | True, if <i>file</i> exists and is owned by the effective user ID of this process.                                                                                                  |
| <b>-G file</b>                         | True, if <i>file</i> exists and its group matches the effective group ID of this process.                                                                                           |
| <b>-S file</b>                         | True, if <i>file</i> exists and is a socket.                                                                                                                                        |
| <i>file1</i> <b>-nt</b> <i>file2</i>   | True, if <i>file1</i> exists and is newer than <i>file2</i> .                                                                                                                       |
| <i>file1</i> <b>-ot</b> <i>file2</i>   | True, if <i>file1</i> exists and is older than <i>file2</i> .                                                                                                                       |
| <i>file1</i> <b>-ef</b> <i>file2</i>   | True, if <i>file1</i> and <i>file2</i> exist and refer to the same file.                                                                                                            |
| <i>string</i> <b>=</b> <i>pattern</i>  | True, if <i>string</i> matches <i>pattern</i> .                                                                                                                                     |
| <i>string</i> <b>!=</b> <i>pattern</i> | True, if <i>string</i> does not match <i>pattern</i> .                                                                                                                              |

| s |

|                                |                                                                                                   |
|--------------------------------|---------------------------------------------------------------------------------------------------|
| <i>string</i> < <i>string2</i> | True, if <i>string1</i> comes before <i>string2</i> based on the ASCII value of their characters. |
| <i>string</i> > <i>string2</i> | True, if <i>string1</i> comes after <i>string2</i> based on the ASCII value of their characters.  |
| <i>exp1</i> -eq <i>exp2</i>    | True, if <i>exp1</i> is equal to <i>exp2</i> .                                                    |
| <i>exp1</i> -ne <i>exp2</i>    | True, if <i>exp1</i> is not equal to <i>exp2</i> .                                                |
| <i>exp1</i> -lt <i>exp2</i>    | True, if <i>exp1</i> is less than <i>exp2</i> .                                                   |
| <i>exp1</i> -gt <i>exp2</i>    | True, if <i>exp1</i> is greater than <i>exp2</i> .                                                |
| <i>exp1</i> -le <i>exp2</i>    | True, if <i>exp1</i> is less than or equal to <i>exp2</i> .                                       |
| <i>exp1</i> -ge <i>exp2</i>    | True, if <i>exp1</i> is greater than or equal to <i>exp2</i> .                                    |

A compound expression can be constructed from these primitives by using any of the following, listed in decreasing order of precedence.

|                            |                                                         |
|----------------------------|---------------------------------------------------------|
| ( <i>exp</i> )             | True, if <i>exp</i> is true. Used to group expressions. |
| ! <i>exp</i>               | True, if <i>exp</i> is false.                           |
| <i>exp1</i> && <i>exp2</i> | True, if <i>exp1</i> and <i>exp2</i> are both true.     |
| <i>exp1</i>    <i>exp2</i> | True, if either <i>exp1</i> or <i>exp2</i> is true.     |

### Input/Output

Before a command is executed, its input and output can be redirected using a special notation interpreted by the shell. The following can appear anywhere in a simple-command or may precede or follow a command and are not passed on to the invoked command. Command and parameter substitution occurs before *word* or *digit* is used, except as noted below. File name generation occurs only if the pattern matches a single file and blank interpretation is not performed.

|                   |                                                                                                                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| < <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, it is created. If the file exists, and the <b>noclobber</b> option is on, an error occurs; otherwise, the file is truncated to zero length.                                                                                  |
| >  <i>word</i>    | Same as >, except that it overrides the <b>noclobber</b> option.                                                                                                                                                                                                                                                      |
| >> <i>word</i>    | Use file <i>word</i> as standard output. If the file exists, output is appended to it (by first searching for 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 matches <i>word</i> , or to an end-of-file. No parameter substitution, command substitution or file name generation is performed on <i>word</i> . The resulting document, called a <b>here-document</b> , becomes the standard input. See also the <b>WARNINGS</b> section. |
|                   | If any character of <i>word</i> is quoted, no interpretation is placed upon the characters of the document. Otherwise, parameter and command substitution occurs, <i>newline</i> is ignored, and \ must be used to quote the characters \, \$, `, and the first character of <i>word</i> .                            |
|                   | If - is appended to <<, all leading tabs are stripped from <i>word</i> and from the document.                                                                                                                                                                                                                         |
| <& <i>digit</i>   | The standard input is duplicated from file descriptor <i>digit</i> (see <i>dup(2)</i> ).                                                                                                                                                                                                                              |
| >& <i>digit</i>   | The standard output is duplicated to file descriptor <i>digit</i> (see <i>dup(2)</i> ).                                                                                                                                                                                                                               |
| <&-               | The standard input is closed.                                                                                                                                                                                                                                                                                         |
| >&-               | The standard output is closed.                                                                                                                                                                                                                                                                                        |
| <&p               | The input from the coprocess is moved to standard input.                                                                                                                                                                                                                                                              |
| >&p               | The output to the coprocess is moved to standard output.                                                                                                                                                                                                                                                              |

If any of the above redirections is preceded by a digit (0 to 9), the file descriptor used is the one specified by the digit, instead of the default 0 (standard input) or 1 (standard output). For example:

```
2>&1
```

means open file descriptor 2 for writing as a duplicate of file descriptor 1. Output directed to file descriptor 2 is written in the same location as output to file descriptor 1.



Order is significant in redirection. The shell evaluates each redirection in terms of the (*file descriptor*, *file*) assignment at the time of evaluation. For example:

```
1>fname 2>&1
```

first assigns file descriptor 1 to file *fname*. It then assigns file descriptor 2 to the file assigned to file descriptor 1 (that is, *fname*).

If the order of redirection is reversed, as in

```
2>&1 1>fname
```

file descriptor 2 is assigned to the file assigned to file descriptor 1 (probably the terminal) and then file descriptor 1 is assigned to file *fname*.

By using the redirection operators above, the input and output of a *coprocess* may be moved to a numbered file descriptor, allowing other commands to write to them and read from them. If the input of the current coprocess is moved to a numbered file descriptor, another coprocess may be started.

If a command is followed by **&** and job control is inactive, 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*(5)) is a list of name-value pairs passed to an executed program much like a normal argument list. The names must be identifiers and the values are character strings. The shell interacts with the environment in several ways. When invoked, the shell scans the environment and creates a parameter for each name found, gives it the corresponding value and marks it **export**. Executed commands inherit the environment. If the user modifies the values of these parameters or creates new ones by using the **export** or **typeset -x** special commands, the values 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 can be augmented by prefixing it with one or more parameter assignments. A parameter assignment argument takes the form *identifier=value*. For example, both the following

```
TERM=450 cmd args
(export TERM; TERM=450; cmd args)
```

are equivalent (as far as the execution of *cmd* is concerned, except for the special commands that are preceded by a percent sign (%)).

If the **-k** option is set, all parameter assignment arguments are placed in the environment, even if they occur after the command name. The following echo statement prints **a=b c**. After the **-k** option is set, the second echo statement prints only **c**:

```
echo a=b c → a=b c
set -k
echo a=b c → 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

The **function** command (described in the *Compound Commands* subsection) defines shell functions. Shell functions are read 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 the *Execution* subsection.)

Functions execute in the same process as the caller and share all files and current working directory with the caller. Traps defined by the caller remain in effect within the function until another **trap** command is executed. Traps set within a function remain in effect after the function returns. Ordinarily, variables are shared between the calling program and the function. However, the **typeset** special command can be used within a function to define local variables whose scope includes the current function and all functions it calls.

The **return** special command is used to return from function calls. Errors within functions return control to the caller.

Function identifiers can be listed with the **+f** option of the **typeset** special command. Function identifiers and the associated text of the functions can be listed with the **-f** option. Functions can be undefined with the **-f** option of the **unset** special command.

Ordinarily, functions are unset when the shell executes a shell script. The **-xf** option of the **typeset** command allows a function to be exported to scripts that are executed without reinvoking the shell. Functions that must be defined across separate invocations of the shell should be placed in the **ENV** file.

### 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 that looks like:

```
[1] 1234
```

indicating that job number 1 was started asynchronously and had one (top-level) process whose process ID was 1234.

If you are running a job and wish to do something else, you can type the suspend character (the **susp** character defined with **stty**; see *stty(1)*) to send a **SIGSTOP** signal to the current job. The shell then indicates that the job has been **Stopped**, and prints another prompt. Then you can manipulate the state of this job by putting it in the background with the **bg** command, running other commands, and eventually returning the job to the foreground with the **fg** command. A suspend takes effect immediately and resembles an interrupt, since pending output and unread input are discarded when the suspend is entered.

A job running in the background stops if it tries to read from the terminal. Background jobs normally are allowed to produce output, but can be disabled with the **stty tostop** command. If the user sets this terminal option, background jobs stop when trying to produce output.

There are several ways to refer to jobs in the shell. A job can be referred to by the process ID of any process in the job or by one of the following:

|                 |                                                        |
|-----------------|--------------------------------------------------------|
| <b>%number</b>  | The job with the given number.                         |
| <b>%string</b>  | Any job whose command line begins with <i>string</i> . |
| <b>??string</b> | Any job whose command line contains <i>string</i> .    |
| <b>%%</b>       | Current job.                                           |
| <b>%+</b>       | Equivalent to <b>%%</b> .                              |
| <b>%-</b>       | Previous job.                                          |

The shell learns immediately when a process changes state. It informs the user when a job is blocked and prevented from further progress, but only just before it prints a prompt.

When the monitor mode is on, each background job that completes triggers any trap set for **SIGCHLD**.

If you try to exit from shell while jobs are stopped, you are warned with the message **You have stopped jobs**. You can use the **jobs** command to identify them. If you immediately try to exit again, the shell will not warn you a second time, and the stopped jobs will be terminated.

If you try to leave the shell while jobs are running, you are not warned. The shell exits silently and sets the parent of the running jobs to the **init** process (number 1).

### Signals

The **SIGINT** and **SIGQUIT** signals for an invoked command are ignored if the command is followed by **&** and the **monitor** option is off. Otherwise, signals have the values inherited by the shell from its parent, with the exception of signal **SIGSEGV** (but see also the **trap** special command).

### Execution

Substitutions are made each time a command is executed. **sh** checks the command name to determine whether it matches a special command. If it does, it is executed within the current shell process.

Next, **sh** checks the command name to determine whether it matches one of the user-defined functions. If it does, **sh** saves the positional parameters, then sets them to the arguments of the function call. The positional parameter **0** is unchanged. When the function completes or issues a **return**, **sh** restores the positional parameter list. The value of a function is the value of the last command executed. A function is executed in the current shell process.

If a command name is not a user-defined function or a special command, **sh** creates a process and attempts to execute the command using an **exec\*( )** system call (see *exec(2)*).

The shell parameter **PATH** defines the search path for the directory containing the command. Alternative directory names are separated by a colon (:). The default path is **/usr/bin:** (specifying **/usr/bin**, and the current directory, in that order). Note that the current directory is specified by a null path name, which can appear immediately after the equal sign, between colon delimiters, or at the end of the path list. The search path is not used if the command name contains a /. Otherwise, each directory in the path is searched for an executable file. If the file has execute permissions but is not a directory or an executable object code file, it is assumed to be a script file, which is a file of data for an interpreter. If the first two characters of the script file are **#!**, **exec\*( )** expects an interpreter path name to follow. **exec\*( )** then attempts to execute the specified interpreter as a separate process to read the entire script file. If a call to **exec\*( )** fails, **sh** is spawned to interpret the script file. All nonexported aliases, functions, and named parameters are removed in this case. If the shell command file does not have read permission, or if the **setuid** and/or **setgid** bits are set on the file, the shell executes an agent to set up the permissions and execute the shell with the shell command file passed down as an open file. A parenthesized command is also executed in a subshell without removing nonexported quantities.

### Command Reentry

The text of the last **HISTSIZE** (default 128) 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 writable. A shell can access the commands of all interactive shells that use the same named **HISTFILE**. The special command **fc** 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 **fc**, the value of the parameter **FCEDIT** is used. If **FCEDIT** is not defined, **/usr/bin/ed** is used. The edited command is printed and reexecuted upon leaving the editor. The editor name **-** is used to skip the editing phase and to reexecute the command. In this case, a substitution parameter of the form *old=new* can be used to modify the command before execution. For example, if **r** is aliased to **fc -e -**, typing **r bad=good c** reexecutes the most recent command that starts with the letter **c** and replaces the first occurrence of the string **bad** with the string **good**.

The history file will be trimmed when all of the following conditions occurs:

- Its size is greater than four kilobytes.
- The number of commands in it is more than **HISTSIZE**.
- The file has not been modified in the last ten minutes.
- The user has write permission for the directory in which the history file resides.

If any one of the above conditions does not occur, the history file will not be trimmed. When the history file is trimmed, the latest **HISTSIZE** commands will be available in the history file.

### Command Line Editing

Normally, each command line typed at a terminal device is followed by a newline or return. If one of the **emacs**, **gmacs**, **vi**, or **viraw**, options is set, you can edit the command line. An editing option is automatically selected each time the **VISUAL** or **EDITOR** variable is assigned a value ending in one of these option names.

The editing features require that the user's terminal accept return without line feed and that a space (" ") must overwrite the current character on the screen. ADM terminal users should set the "space – advance" switch to "space". Hewlett-Packard terminal users should set the straps to "bcGHxZ etX".

The editing modes enable the user to look through a window at the current line. The default window width is 80, unless the value of **COLUMNS** is defined. If the line is longer than the window width minus two, a mark displayed at the end of the window notifies the user. The mark is one of:

- > The line extends to the right.
- < The line extends to the left.
- \* The line extends to both sides of the window.

As the cursor moves and reaches the window boundaries, the window is centered about the cursor.

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.

Changing the **LC\_TYPE** environment variable can affect the editors. See the *Parameter Substitution* subsection.

| S |

**emacs/gmacs Editing Mode**

This mode is invoked by either the **emacs** or **gmacs** option. The sole difference is how they handle Control-T.

To edit, the user moves the cursor to the point needing correction and inserts or deletes characters or words. All editing commands are control characters or escape sequences. The notation for control characters is caret (^) followed by a character. For example, ^F is the notation for Control-F. This is entered by holding down the Ctrl (control) key and pressing **f**. The shift key is *not* pressed. The notation ^? indicates the delete (DEL) key.

The notation for escape sequences is **M-** followed by a character. For example, **M-f** (pronounced *meta f*) is entered by pressing the escape key (**Esc**) followed by pressing **f**. **M-F** is the notation for escape followed by shift (capital) **F**.

All edit commands operate from any place on the line (not only at the beginning). Neither the return (^M) nor the newline (^J) key is entered after edit commands, except when noted.

|                 |                                                                                                                                                                                                                                                                                                                                                     |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>^F</b>       | Move cursor forward (right) one character.                                                                                                                                                                                                                                                                                                          |
| <b>M-f</b>      | Move cursor forward one word. (The editor's idea of a word is a string of characters consisting of only letters, digits and underscores.)                                                                                                                                                                                                           |
| <b>^B</b>       | Move cursor backward (left) one character.                                                                                                                                                                                                                                                                                                          |
| <b>M-b</b>      | Move cursor backward one word.                                                                                                                                                                                                                                                                                                                      |
| <b>^A</b>       | Move cursor to start of line.                                                                                                                                                                                                                                                                                                                       |
| <b>^E</b>       | Move cursor to end of line.                                                                                                                                                                                                                                                                                                                         |
| <b>^]char</b>   | Move cursor forward to character <i>char</i> on current line.                                                                                                                                                                                                                                                                                       |
| <b>M-^]char</b> | Move cursor backward to character <i>char</i> on current line.                                                                                                                                                                                                                                                                                      |
| <b>^X^X</b>     | Interchange the cursor and mark.                                                                                                                                                                                                                                                                                                                    |
| <i>erase</i>    | Delete previous character. The <i>erase</i> character is user-definable with the <b>stty</b> command; it is usually set to ^H. The system default is #.                                                                                                                                                                                             |
| <b>^D</b>       | Delete current character.                                                                                                                                                                                                                                                                                                                           |
| <i>eof</i>      | Terminate the shell if the current line is null. The <i>eof</i> character is user-definable with the <b>stty</b> command; it is usually set to ^D. The system default is ^D.                                                                                                                                                                        |
| <b>M-d</b>      | Delete current word.                                                                                                                                                                                                                                                                                                                                |
| <b>M-^H</b>     | Delete previous word (meta-backspace).                                                                                                                                                                                                                                                                                                              |
| <b>M-h</b>      | Delete previous word.                                                                                                                                                                                                                                                                                                                               |
| <b>M-^?</b>     | Delete previous word (meta-delete). If your interrupt character is ^? (DEL, the default), this command will not work.                                                                                                                                                                                                                               |
| <b>^T</b>       | In <b>emacs</b> mode, transpose current character with next character. In <b>gmacs</b> mode, transpose two previous characters.                                                                                                                                                                                                                     |
| <b>^C</b>       | Capitalize current character.                                                                                                                                                                                                                                                                                                                       |
| <b>M-c</b>      | Capitalize current word.                                                                                                                                                                                                                                                                                                                            |
| <b>M-l</b>      | Change the current word to lowercase.                                                                                                                                                                                                                                                                                                               |
| <b>^K</b>       | 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 the given position up to the cursor. If preceded by a numerical parameter whose value is greater than the current cursor position, then delete from the cursor up to the given position. |
| <b>^W</b>       | Kill from the cursor to the mark.                                                                                                                                                                                                                                                                                                                   |
| <b>M-p</b>      | Push the region from the cursor to the mark on the stack.                                                                                                                                                                                                                                                                                           |
| <i>kill</i>     | Kill the entire current line. If two kill characters are entered in succession, all subsequent consecutive kill characters cause a line feed (useful when using paper terminals). The <i>kill</i> character is user-definable with the <b>stty</b> command; it is usually set to ^X or ^U. The system default is @.                                 |
| <b>^Y</b>       | Restore last item removed from line (yank item back to the line).                                                                                                                                                                                                                                                                                   |
| <b>^L</b>       | Line feed and print current line.                                                                                                                                                                                                                                                                                                                   |
| <b>^@</b>       | Set mark (null character).                                                                                                                                                                                                                                                                                                                          |
| <b>M-</b>       | Set mark (meta-space).                                                                                                                                                                                                                                                                                                                              |
| <b>^J</b>       | Execute the current line (newline).                                                                                                                                                                                                                                                                                                                 |
| <b>^M</b>       | Execute the current line (return).                                                                                                                                                                                                                                                                                                                  |
| <b>^P</b>       | Fetch previous command. Each time ^P is entered, the previous command in the history list is accessed.                                                                                                                                                                                                                                              |
| <b>^N</b>       | Fetch next command. Each time ^N is entered the next command in the history list is accessed.                                                                                                                                                                                                                                                       |

| S |

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>M-<i>&lt;</i></b>   | Fetch the least recent (oldest) history line.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>M-<i>&gt;</i></b>   | Fetch the most recent (youngest) history line.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>^R<i>string</i></b> | 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 <b>^</b> , the matched line must begin with <i>string</i> . If <i>string</i> is omitted, 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. |
| <b>^O</b>              | Execute the current line and fetch the next line relative to current line from the history file.                                                                                                                                                                                                                                                                                                                                                                          |
| <b>M-<i>digits</i></b> | Define a numeric parameter. The digits are taken as a parameter to the next command. The commands that accept a parameter are <i>erase</i> , <b>^B</b> , <b>^C</b> , <b>^D</b> , <b>^F</b> , <b>^K</b> , <b>^N</b> , <b>^P</b> , <b>^R</b> , <b>^]</b> , <b>M-^H</b> , <b>M-.</b> , <b>M-<i>_</i></b> , <b>M-<i>b</i></b> , <b>M-<i>c</i></b> , <b>M-<i>d</i></b> , <b>M-<i>f</i></b> , <b>M-<i>h</i></b> , and <b>M-<i>l</i></b> .                                       |
| <b>M-<i>letter</i></b> | Your alias list is searched for an alias by the name <i>_letter</i> (underscore-letter). If an alias of this name is defined, its value is inserted on the input queue. This <i>letter</i> must not be one of the above metafunctions.                                                                                                                                                                                                                                    |
| <b>M-.</b>             | 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.                                                                                                                                                                                                                                                                                 |
| <b>M-<i>_</i></b>      | Same as <b>M-.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>M-*</b>             | Attempt file name generation on the current word.                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>M-<i>[</i></b>      | File name completion (meta-escape). Replaces the current word with the longest common prefix of all file names matching the current word with an asterisk appended. If the match is unique, a <b>/</b> is appended if the file is a directory and a space is appended if the file is not a directory.                                                                                                                                                                     |
| <b>M-<i>=</i></b>      | List files matching current word pattern as if an asterisk were appended.                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>^U</b>              | Multiply parameter of next command by 4.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>\</b>               | Escape next character. Editing characters and your erase, kill, and interrupt characters may be entered in a command line or in a search string, if preceded by a <b>\</b> . The <b>\</b> removes the next character's editing features (if any).                                                                                                                                                                                                                         |
| <b>^V</b>              | Display version of the shell.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>M-<i>#</i></b>      | Insert a <b>#</b> at the beginning of the line and execute it. This causes a comment to be inserted in the history file.                                                                                                                                                                                                                                                                                                                                                  |

## vi Editing Mode

The editor starts in insert mode until an escape (ESC) is received. This puts you in control mode in which you can move the cursor and perform editing commands. A return in either mode sends the line.

Most control commands accept an optional repeat *count* prior to the command.

In **vi** mode on most systems, canonical processing is initially enabled and the command is echoed again if the speed is 1200 baud or greater and contains any control characters, or if less than one second has elapsed since the prompt was printed. The escape (ESC) character terminates canonical processing for the remainder of the command and you can then modify the command line. This scheme has the advantages of canonical processing with the typeahead echoing of raw mode.

Setting the **viraw** option always disables canonical processing on the terminal. This mode is implicit for systems that do not support two alternate end-of-line delimiters, and may be helpful for certain terminals.

## Insert Edit Commands

By default, the editor is in insert mode.

|              |                                                                                                                                                                                                                          |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>erase</i> | Delete previous inserted character. The <i>erase</i> character is user-definable with the <b>stty</b> command; it is usually set to <b>^H</b> . The system default is <b>#</b> .                                         |
| <i>kill</i>  | Delete all current inserted characters. The <i>kill</i> character is user-definable with the <b>stty</b> command; it is usually set to <b>^X</b> or <b>^U</b> . The system default is <b>@</b> .                         |
| <b>\</b>     | Escape the next <i>erase</i> or <i>kill</i> character.                                                                                                                                                                   |
| <i>eof</i>   | Terminate the shell if the current line is null. The <i>eof</i> character is user-definable with the <b>stty</b> command; it is usually set to <b>^D</b> . The system default is <b>^D</b> .                             |
| <b>^V</b>    | Escape next character. Editing characters and erase or kill characters may be entered in a command line or in a search string if preceded by a <b>^V</b> , which removes the next character's editing features (if any). |
| <b>^W</b>    | Delete the previous blank-separated word.                                                                                                                                                                                |

**Motion Edit Commands**

These commands move the cursor. The use of *count* causes a repetition of the command the cited number of times.

|                          |                                                                                               |
|--------------------------|-----------------------------------------------------------------------------------------------|
| <i>[count]</i> <b>l</b>  | Cursor forward (right) one character.                                                         |
| <i>[count]</i> <b>w</b>  | Cursor forward one alphanumeric word.                                                         |
| <i>[count]</i> <b>W</b>  | Cursor forward to the beginning of the next word that follows a blank.                        |
| <i>[count]</i> <b>e</b>  | Cursor forward to the end of the word.                                                        |
| <i>[count]</i> <b>E</b>  | Cursor forward to end of the current blank-delimited word.                                    |
| <i>[count]</i> <b>h</b>  | Cursor backward (left) one character.                                                         |
| <i>[count]</i> <b>b</b>  | Cursor backward one word.                                                                     |
| <i>[count]</i> <b>B</b>  | Cursor backward to preceding blank-separated word.                                            |
| <i>[count]</i>           | Cursor to column <i>count</i> . Default is 1.                                                 |
| <i>[count]</i> <b>fc</b> | Find the next character <i>c</i> in the current line.                                         |
| <i>[count]</i> <b>Fc</b> | Find the previous character <i>c</i> in the current line.                                     |
| <i>[count]</i> <b>tc</b> | Equivalent to <b>fc</b> followed by <b>h</b> .                                                |
| <i>[count]</i> <b>Tc</b> | Equivalent to <b>Fc</b> followed by <b>l</b> .                                                |
| <i>[count]</i> ;         | Repeat the last single-character find command, <b>f</b> , <b>F</b> , <b>t</b> , or <b>T</b> . |
| <i>[count]</i> ,         | Reverses the last single character find command.                                              |
| <b>0</b>                 | Cursor to start of line.                                                                      |
| <b>^</b>                 | Cursor to first nonblank character in line.                                                   |
| <b>\$</b>                | Cursor to end of line.                                                                        |

**History Search Commands**

These commands access your command history file.

|                         |                                                                                                                                                                                                                                                                                            |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>[count]</i> <b>k</b> | Fetch previous command. Each time <b>k</b> is entered, the next earlier command in the history list is accessed.                                                                                                                                                                           |
| <i>[count]</i> -        | Equivalent to <b>k</b> .                                                                                                                                                                                                                                                                   |
| <i>[count]</i> <b>j</b> | Fetch next command. Each time <b>j</b> is entered, the next later command in the history list is accessed.                                                                                                                                                                                 |
| <i>[count]</i> +        | Equivalent to <b>j</b> .                                                                                                                                                                                                                                                                   |
| <i>[count]</i> <b>G</b> | The command number <i>count</i> is fetched. The default is the first command in the history list.                                                                                                                                                                                          |
| <i>/string</i>          | Search backward through history for a previous command containing <i>string</i> . <i>string</i> is terminated by a return or newline. If <i>string</i> is preceded by a <b>^</b> , the matched line must begin with <i>string</i> . If <i>string</i> is null, the previous string is used. |
| <b>?string</b>          | Same as <b>/</b> , but search in the forward direction.                                                                                                                                                                                                                                    |
| <b>n</b>                | Search for next match of the last pattern to the <b>/</b> or <b>?</b> commands.                                                                                                                                                                                                            |
| <b>N</b>                | Search for next match of the last pattern to <b>/</b> or <b>?</b> , but in reverse direction.                                                                                                                                                                                              |

| S |

**Text Modification Edit Commands**

These commands will modify the line.

|                                       |                                                                                                                                                                                                                                                 |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>a</b>                              | Enter insert mode after the current character.                                                                                                                                                                                                  |
| <b>A</b>                              | Append text to the end of the line. Equivalent to <b>\$a</b> .                                                                                                                                                                                  |
| <i>[count]</i> <b>c</b> <i>motion</i> | Move cursor forward to the character position specified by <i>motion</i> , deleting all characters between the original cursor position and the new position, and enter insert mode. If <i>motion</i> is <b>c</b> , the entire line is deleted. |
| <b>C</b>                              | Delete from the current character through the end of line and enter insert mode. Equivalent to <b>c\$</b> .                                                                                                                                     |
| <b>S</b>                              | Equivalent to <b>cc</b> .                                                                                                                                                                                                                       |
| <i>[count]</i> <b>d</b> <i>motion</i> | Move cursor to the character position specified by <i>motion</i> , deleting all characters between the original cursor position and the new position. If <i>motion</i> is <b>d</b> , the entire line will be deleted.                           |
| <b>D</b>                              | Delete from the current character through the end of line. Equivalent to <b>d\$</b> .                                                                                                                                                           |
| <b>i</b>                              | Enter insert mode before the current character.                                                                                                                                                                                                 |
| <b>I</b>                              | Enter insert mode before the beginning of the line. Equivalent to the two-character sequence <b>Oi</b> .                                                                                                                                        |
| <i>[count]</i> <b>P</b>               | Insert the previous text modification before the cursor.                                                                                                                                                                                        |
| <i>[count]</i> <b>p</b>               | Insert the previous text modification after the cursor.                                                                                                                                                                                         |

|                        |                                                                                                                                                                                                                                                                                                  |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>R</b>               | Enter insert mode and replace characters on the screen with characters you type, overlay fashion.                                                                                                                                                                                                |
| <code>[count]rc</code> | Replace the current character with <i>c</i> .                                                                                                                                                                                                                                                    |
| <code>[count]x</code>  | Delete the current character.                                                                                                                                                                                                                                                                    |
| <code>[count]X</code>  | Delete the preceding character.                                                                                                                                                                                                                                                                  |
| <code>[count].</code>  | Repeat the previous text modification command.                                                                                                                                                                                                                                                   |
| <code>~</code>         | Invert the case of the current character and advance the cursor.                                                                                                                                                                                                                                 |
| <code>[count]_</code>  | Append the <i>count</i> word of the previous command at the current cursor location and enter insert mode at the end of the appended text. The last word is used if <i>count</i> is omitted.                                                                                                     |
| <b>*</b>               | Append an <b>*</b> to the current word and attempt file name generation. If no match is found, ring the bell. If a match is found, replace the word with the matching string of file names and enter insert mode.                                                                                |
| escape                 |                                                                                                                                                                                                                                                                                                  |
| <code>\</code>         | Attempt file name completion on the current word. Replace the current word with the longest common prefix of all file names matching the current word with an asterisk appended. If the match is unique, append a / if the file is a directory or append a space if the file is not a directory. |

### Other Edit Commands

|                                     |                                                                                                                                                                                                                                                                               |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>[count]ymotion</code>         |                                                                                                                                                                                                                                                                               |
| <b>y</b> <code>[count]motion</code> | Yank current character through character that <i>motion</i> would move the cursor to and put them into the delete buffer. The text and cursor are unchanged.                                                                                                                  |
| <b>Y</b>                            | Yank from current position to end of line. Equivalent to <b>y\$</b> .                                                                                                                                                                                                         |
| <b>u</b>                            | Undo the last text-modifying command.                                                                                                                                                                                                                                         |
| <b>U</b>                            | Undo all the text-modifying commands performed on the line.                                                                                                                                                                                                                   |
| <code>[count]v</code>               | Execute the command <b>fc -e \${VISUAL:-\${EDITOR:-vi}}</b> <i>count</i> in the input buffer. If <i>count</i> is omitted, the current line is used. This executes an editor with the current line as the input "file". When you exit from the editor, the result is executed. |
| <b>^L</b>                           | Line feed and print current line.                                                                                                                                                                                                                                             |
| <b>^J</b>                           | Execute the current line, regardless of mode (newline).                                                                                                                                                                                                                       |
| <b>^M</b>                           | Execute the current line, regardless of mode (return).                                                                                                                                                                                                                        |
| <b>#</b>                            | Insert a <b>#</b> at the beginning of the current line and after each embedded newline, and execute the line. Useful for inserting the current command line in the history list without executing it.                                                                         |
| <b>=</b>                            | List the file names that match the current word if an asterisk were appended to it.                                                                                                                                                                                           |
| <code>@letter</code>                | Search your alias list for an alias with the name <i>_letter</i> (underscore letter). If an alias of this name is defined, its value is executed as a command sequence on the current line. This provides a simple macro capability.                                          |

| S |

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_COLLATE** determines the collating sequence used in evaluating pattern matching notation for file name generation. If it is not defined or is empty, it defaults to the value of **LANG**.

**LC\_CTYPE** determines the classification of characters as letters, and the characters matched by character class expressions in pattern matching notation. If it is not defined or is empty, it defaults to the value of **LANG**.

If **LANG** is not defined or is empty, it defaults to **C** (see *lang(5)*).

If any internationalization variable contains an invalid value, they all default to **C** (see *environ(5)*).

### International Code Set Support

Single- and multibyte character code sets are supported.

## RETURN VALUE

Errors detected by the shell, such as syntax errors, cause the shell to return a nonzero exit status. Otherwise, the shell returns the exit status of the last command executed. See also the **exit** special command.

If the shell is being used noninteractively, the execution of the shell file is abandoned. Runtime errors detected by the shell are reported by printing the command or function name and the error condition. If the line number on which the error occurred is greater than one, the line number is also printed in

brackets ([ ]) after the command or function name.

## WARNINGS

Some file descriptors are used internally by the POSIX shell. For HP-UX releases 10.10 and beyond, file descriptors 24 through 30 are reserved. HP-UX releases 10.00 and 10.01 reserve descriptors 54 through 60. Applications using these and forking a subshell should not depend upon them surviving in the subshell or its descendants.

If a command that is a tracked alias is executed, and 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 load and execute the original command. Use the **-t** option of the **alias** command to correct this situation.

If you move the current directory or one above it, **pwd** may not give the correct response. Use the **cd** command with a full path name to correct this situation.

Some very old shell scripts use a caret (^) as a synonym for the pipe character (|). **sh** does not recognize the caret as a pipe character.

If a command is piped into a shell command, all variables set in the shell command are lost when the command completes.

Using the **fc** built-in command within a compound command will cause the entire command to disappear from the history file.

The dot (.) special command, as in *. file*, reads the entire file before any commands are executed. Therefore, **alias** and **unalias** commands in the file will not apply to any functions defined in the file.

Traps are not processed while the shell is waiting for a foreground job. Thus, a trap on **SIGCHLD** is not executed until the foreground job terminates.

The **export** special command does not handle arrays properly. Only the first element of an array is exported to the environment.

Background processes started from a noninteractive shell cannot be accessed with job control commands.

The value of the **IFS** variable in the user's environment affects the behavior of scripts.

## Collating Order

In an international environment, character ordering is determined by the value of **LC\_COLLATE**, rather than by the binary ordering of character values in the machine collating sequence. This brings with it certain attendant dangers, particularly when using range expressions in file name generation patterns. For example, the command,

```
rm [a-z]*
```

might be expected to match all file names beginning with a lowercase alphabetic character. However, if dictionary ordering is specified by **LC\_COLLATE**, it would also match file names beginning with an uppercase character (as well as those beginning with accented letters). Conversely, it would fail to match letters collated after **z** in languages such as Danish or Norwegian.

The correct (and safe) way to match specific character classes in an international environment is to use a pattern (see *regex(5)*) of the form:

```
rm [[:lower:]]*
```

This uses **LC\_CTYPE** to determine character classes and works predictably for all supported languages and codesets. For shell scripts produced on noninternationalized systems (or without consideration for the above dangers), it is recommended that they be executed in a non-NLS environment. This requires that **LANG**, **LC\_COLLATE**, and so on, be set to **C** or not set at all.

## History File and Locales

The history file does not support mixing of locales in the same file. For users of multiple locales, you can assign a unique history file for each locale by setting **HISTFILE** as:

```
HISTFILE=$HOME/.sh_hist_${LANG}
```

On encountering a history file with invalid characters for the current locale setting, the shell will inform the user about it and continue processing user input. No history features will be available to the user in such a session. Restarting the shell after setting a new history file or after removing the current history file allows the user to access the history features.



**Here-Document Temp Files**

The contents of here-documents are stored in temporary files named `/var/tmp/shpid.number`. Usually, these temporary files are removed after they are used. However, due to design limitations, these temporary files may sometimes continue to exist after the shell exits. `pid` is the process ID of the shell. `number` is a sequence number for the here-document files.

**AUTHOR**

`sh` was developed by AT&T, OSF, and HP.

**FILES**

|                                    |                                                               |
|------------------------------------|---------------------------------------------------------------|
| <code>\$HOME/.profile</code>       | Read to set up user's custom environment                      |
| <code>/etc/passwd</code>           | To find home directories                                      |
| <code>/etc/profile</code>          | Read to set up system environment                             |
| <code>/etc/suid_profile</code>     | Security profile                                              |
| <code>/sbin/sh</code>              | Archived executable, especially for root access               |
| <code>/tmp/shpid.number</code>     | For here-documents if <code>/var/tmp</code> is not accessible |
| <code>/usr/bin/sh</code>           | Standard executable for the POSIX shell                       |
| <code>/var/tmp/shpid.number</code> | For here-documents if <code>/var/tmp</code> is accessible     |

**SEE ALSO**

`cat(1)`, `cd(1)`, `command(1)`, `echo(1)`, `ed(1)`, `env(1)`, `getopts(1)`, `kill(1)`, `ln(1)`, `login(1)`, `newgrp(1)`, `printf(1)`, `pwd(1)`, `read(1)`, `stty(1)`, `test(1)`, `time(1)`, `umask(1)`, `vi(1)`, `dup(2)`, `exec(2)`, `fork(2)`, `pipe(2)`, `stty(2)`, `ulimit(2)`, `umask(2)`, `wait(2)`, `rand(3C)`, `a.out(4)`, `profile(4)`, `environ(5)`, `lang(5)`, `regex(5)`, `signal(5)`.

**STANDARDS CONFORMANCE**

`sh`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`::`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`::`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`break`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`case`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`continue`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`eval`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`exec`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`exit`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`export`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`for`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`if`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`read`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`return`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`set`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`shift`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`time`: SVID2, SVID3, XPG2, XPG3, XPG4

`trap`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`unset`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`until`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`while`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| s |

**NAME**

sh - overview of various system shells

**SYNOPSIS****POSIX Shell:**

**sh** [**±aefhikmnoqrstuvx**] [**±o option**] ... [**-c string**] [**arg ...**]

**rsh** [**±aefhikmnoqrstuvx**] [**±o option**] ... [**-c string**] [**arg ...**]

**Korn Shell:**

**ksh** [**±aefhikmnoqrstuvx**] [**±o option**] ... [**-c string**] [**arg ...**]

**rksh** [**±aefhikmnoqrstuvx**] [**±o option**] ... [**-c string**] [**arg ...**]

**C Shell:**

**csh** [**-cefinstvxTVX**] [**command\_file**] [**argument\_list ...**]

**Key Shell:**

**keysh**

**DESCRIPTION****Remarks:**

The POSIX.2 standard requires that, on a POSIX-compliant system, executing the command **sh** activates the POSIX shell (located in file **/usr/bin/sh** on HP-UX systems), and executing the command **man sh** produces an on-line manual entry that displays the syntax of the POSIX shell command-line.

However, the **sh** command has historically been associated with the conventional Bourne shell, which could confuse some users. To meet standards requirements and also clarify the relationships of the various shells and where they reside on the system, this entry provides command-line syntax and a brief description of each shell, and lists the names of the manual entries where each shell is described in greater detail.

**NOTE :** The Bourne shell (**/usr/old/bin/sh**) is removed from the system starting with HP-UX 11i Version 1.5. Please use the POSIX shell (**/usr/bin/sh**) as an alternative.

**Shell Descriptions**

The HP-UX operating system supports the following shells:

- |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>sh</b>    | POSIX-conforming command programming language and command interpreter residing in file <b>/usr/bin/sh</b> . Can execute commands read from a terminal or a file. This shell conforms to current POSIX standards in effect at the time the HP-UX system release was introduced, and is similar to the Korn shell in many respects. Similar in many respects to the Korn shell, the POSIX shell contains a history mechanism, supports job control, and provides various other useful features. |
| <b>ksh</b>   | Korn-shell command programming language and commands interpreter residing in file <b>/usr/bin/ksh</b> . Can execute commands read from a terminal or a file. This shell, like the POSIX shell, contains a history mechanism, supports job control, and provides various other useful features.                                                                                                                                                                                                |
| <b>csh</b>   | A command language interpreter that incorporates a command history buffer, C-language-like syntax, and job control facilities.                                                                                                                                                                                                                                                                                                                                                                |
| <b>rsh</b>   | Restricted version of the POSIX shell command interpreter. Sets up a login name and execution environment whose capabilities are more controlled (restricted) than normal user shells.                                                                                                                                                                                                                                                                                                        |
| <b>rksh</b>  | restricted version of the Korn-shell command interpreter Sets up a login name and execution environment whose capabilities are more controlled (restricted) than normal user shells.                                                                                                                                                                                                                                                                                                          |
| <b>keysh</b> | An extension of the standard Korn Shell that uses hierarchical softkey menus and context-sensitive help.                                                                                                                                                                                                                                                                                                                                                                                      |

| S |

| To obtain:  | Use the command:              |
|-------------|-------------------------------|
| POSIX Shell | <code>/usr/bin/sh ...</code>  |
| Korn Shell  | <code>/usr/bin/ksh ...</code> |
| C Shell     | <code>/usr/bin/csh ...</code> |
| Key Shell   | <code>/usr/bin/keysh</code>   |

These shells can also be the default invocation, depending on the entry in the `/etc/passwd` file. See also *chsh*(1).

#### WARNINGS

Many manual entries contain descriptions of shell behavior or describe program or application behavior similar to “the shell” with a reference to “see *sh*(1)”.

#### SEE ALSO

For more information on the various individual shells, see:

*keysh*(1)      Key Shell (`/usr/bin/keysh`) description.  
*ksh*(1)        Korn Shell (`/usr/bin/ksh`) description.  
*sh-posix*(1)    POSIX Shell (`/usr/bin/sh`) description.  
*csh*(1)        C Shell (`/usr/bin/csh`) description.

**NAME**

shar - make a shell archive package

**SYNOPSIS**

**shar** [*options*] [*file* | *dir*] ... > *package*

**DESCRIPTION**

The **shar** command bundles the named files and directories into a single distribution package suitable for mailing or moving. The files can contain any data, including executables. The resulting package, written to standard output, is a shell script file that can be edited (to add messages at the beginning, etc.).

To unpack *package*, use the **sh** command with the package name as an argument as follows:

```
sh package
```

When unpacking, the files and directories in *package* are written to the path names recorded in the archive.

If a directory is specified and the **-d** option is not given, all files beneath that directory are archived.

If a special file is specified, the appropriate **mknod** commands are emitted to recreate the file (see *mknod*(1M)).

**shar** protects the contained files from mail processing, if necessary, by inserting an @ character at the beginning of each line. If the file contains unusual data, the data is transformed into **uencode** format, and a **udecode** script is included in *package* so that the package can still be unpacked correctly by **sh**. See WARNINGS for more information about mailers and file modifications.

Access modes are preserved for both directories and files.

**Options**

**shar** recognizes the following options:

- a** Assume that files can be shipped, regardless of their contents; do not protect them specially. **shar** is conservative, and might decide to **uencode** a file containing special characters (such as Ctrl-G) that the user knows do not need protection.
- A** Suppress warning messages regarding optional access control list entries. **shar** does not archive optional access control list entries in a file's access control list (see *acl*(5)). Normally, a warning message is printed for each file having optional access control list entries.
- b** Archive files under their base names, regardless of the original path names specified. The contents are thus unpacked into the current directory instead of to the originally specified path names. This allows you to archive files from many directories but unpack them into a single directory. It also allows you to unpack, for example, **/usr/share/lib/termcap** into **./termcap** instead of overwriting the original one in **/etc**.
- c** Append to the package a simple data-integrity check using **wc** to ensure that the contents were not damaged in transit (see *wc*(1)). This check is performed automatically after unpacking. Also see WARNINGS below.
- C** Insert a line of the form **--- cut here ---** before the archive.
- d** If a directory is specified, do not transmit its contents, but rather only create the empty directory.
- Ddir** Cause the archive to contain code that notifies the user if his or her current directory is not the same as *dir*, which must be an absolute path. If the user is not in *dir*, the unpacking can be continued by responding **yes** to the archive's question.
- e** Cause the archive to contain code that prevents **shar** from unpacking files that would overwrite existing files.
- f file** Read a list of file names from *file* and archive those files as if they were given as arguments.
- h** Follow symbolic links as if they were normal files or directories. If this option is not specified, **shar** archives the link.

| S |

- m** Retain modification and access times on files when they are unpacked.
- o** Preserve user and group ownership on files and directories.
- r** Cause the archive to contain code requiring that the user unpacking it be **root**. This is useful for processing system archives.
- s** Perform error checking using **sum** (see *sum(1)*). Both **-c** and **-s** can be specified for better error checking. Also see WARNINGS below.
- t** Write diagnostics and messages directly to your terminal instead of to the standard error. This is useful when invoking **shar** from programs (such as **vi** that normally combine standard error with standard output. Specifying **-t** also invokes the **-v** (verbose) option.
- u** Assume that the remote site has **uudecode** for unpacking. If this option is not specified, a version of **uudecode** is sent and compiled if any non-ASCII files are archived.
- v** Announce archived file names as they are packed. The **-t** option determines the destination for these announcements.
- Z** Compress files using **compress** (see *compress(1)*).

Most options are flagged in the header of the resulting package, thereby recording the format of the archive. The name of the archiver, system, and time/date of the archive are also recorded in the header.

### EXAMPLES

To archive all files under your home directory, type:

```
cd; shar -cmos .
```

or

```
shar -cmos $HOME
```

To preserve your **/dev** directory, type:

```
shar -mor /dev >save_dev_files
```

To send your newest programs in directory **newstuff** in your home directory to a friend, type:

```
cd; shar -cmos newstuff | mailx -s 'new source' friend
```

### RETURN VALUE

**shar** returns zero if successful; nonzero if problems with arguments occur.

### DIAGNOSTICS

If the **-b** option is specified, **shar** refuses to archive directories.

### WARNINGS

The modification and access time restoration does not take time zones into account.

Files with newline characters in their names scramble the table of contents.

Non-ASCII files with white space in their names do not unpack.

If a mailer such as *elm(1)* is used to transfer *package* to another system and the mailer is configured to expand tabs (by default or otherwise), any file in the archive will be modified if it contains tabs. If the **-c** or **-s** option is used to create the archive, the data-integrity check will fail during unpacking of any files in *package* that contain tab characters that were converted to spaces. (Some mailers that expand tabs when transferring files over a network may or may not expand tabs when transferring files to the sender or other users on the local system.) If an editor is used to modify any of the files in *package*, the data-integrity check will also fail for the files that were changed.

### AUTHOR

**shar** was invented in the public domain. This version of **shar** was developed by HP.

### FILES

**/dev/tty**

**\$TMPDIR/unpack\***

For unpacking non-ASCII files if **TMPDIR** environment variable is set and the directory specified in it is accessible.

|                                 |                                                                                                                                                                                                                              |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/var/tmp/unpack*</code>   | For unpacking non-ASCII files if <b>TMPDIR</b> environment variable is not set or the directory specified in it is not accessible and <code>/var/tmp</code> directory is accessible.                                         |
| <code>/tmp/unpack*</code>       | For unpacking non-ASCII files if <b>TMPDIR</b> environment variable is not set or the directory specified in it is not accessible and <code>/var/tmp</code> directory is not accessible.                                     |
| <code>\$TMPDIR/compress*</code> | For uncompressing files, which are packed using <b>-Z</b> option, if <b>TMPDIR</b> environment variable is set and the directory specified in it is accessible.                                                              |
| <code>/var/tmp/compress*</code> | For uncompressing files, which are packed using <b>-Z</b> option, if <b>TMPDIR</b> environment variable is not set or the directory specified in it is not accessible and <code>/var/tmp</code> directory is accessible.     |
| <code>/tmp/compress*</code>     | For uncompressing files, which are packed using <b>-Z</b> option, if <b>TMPDIR</b> environment variable is not set or the directory specified in it is not accessible and <code>/var/tmp</code> directory is not accessible. |

**SEE ALSO**

ar(1), compress(1), cpio(1), find(1), tar(1), acl(5).

| **s** |

**NAME**

shl - shell layer manager

**SYNOPSIS**

shl

**DESCRIPTION**

**shl** provides a means for interacting with more than one shell from a single terminal by using shell layers. A layer is a shell that is bound to a virtual device. The virtual device can be manipulated like an actual terminal by using **stty** and **ioctl()** (see *stty(1)* and *ioctl(2)*). Each layer has its own process group ID. The user controls these layers by using the commands described below.

The current layer is the layer that can receive input from the keyboard. Other layers attempting to read from the keyboard are blocked. Output from multiple layers is multiplexed onto the terminal. To block the output of a layer when it is not current, the **stty** option **loblk** can be set within the layer.

The **stty** character **swtch** (set to  $\hat{Z}$  if NUL) is used to switch control to **shl** from a layer. **shl** has its own prompt, **>>>**, to distinguish it from a layer.

**Definitions**

A **name** is a sequence of characters delimited by a space, tab, or new-line character. Only the first eight characters are significant. When provided as an argument to the **create** or **name** commands, *name* cannot be of the form *n* or (*n*), where *n* is a decimal number.

**Commands**

The following commands can be issued from the **shl** prompt level. Any unique prefix is accepted.

**create** [-*name*] | *name* [*command*]]

Create a layer called *name* and make it the current layer. If no argument is given, a layer is created with a name of the form (*n*), where *n* is the number of the next available slot in an internal table. Future references to this layer can be made with or without the parentheses. If *name* is followed by a command, that command is executed in the layer instead of a shell. If - is the first argument, a "login shell" is created in the layer. The shell prompt variable **PS1** is set to the name of the layer followed by a space.

**name** [*oldname*] *newname*

Rename the layer *oldname*, calling it *newname*. If *oldname* is not specified, the current layer name is changed.

**!** [*command*] Invoke a sub-shell and execute *command*. If no *command* is given, a shell is executed according to the **SHELL** environment variable.

**block** *name* [*name* ...]

For each *name*, block the output of the corresponding layer when it is not the current layer. This is equivalent to setting the **stty** **loblk** option within the layer.

**delete** *name* [*name* ...]

For each *name*, delete the corresponding layer. All processes in the process group of the layer are sent the **SIGHUP** signal (see *signal(5)*).

**help** (or **?**) Print the syntax of the **shl** commands.

**layers** [-**l**] [*name* ...]

For each *name*, list the layer name and its process group. The **-l** option produces a *ps(1)*-like listing. If no arguments are given, information is presented for all existing layers.

**resume** [*name*]

Change the status of the layer referred to by *name* to that of current layer. If no argument is given, the last existing current layer is changed.

**toggle** Change the status of the previous current layer to that of current layer.

**unblock** *name* [*name* ...]

For each *name*, do not block the output of the corresponding layer when it is not the current layer. This is equivalent to setting the **stty** **loblk** option within the layer.

**quit** Exit **shl**. All layers are sent the **SIGHUP** signal.

| S |

*name* Change the status of the layer referred to by *name* to that of current layer. Any unique prefix is accepted.

## WARNINGS

### Commands

The behavior of the **block** and **unblock shl** commands is not guaranteed when the SHELL environment variable is set to **/usr/bin/csh** (for *csh*(1)) or **/usr/bin/ksh** (for *ksh*(1)), or when the shell saves and restores the tty state (defined in *termio*(7)) before and after each command is invoked interactively from that shell. For both **/usr/bin/csh** and **/usr/bin/ksh**, the **loblk** or **-loblk** options of **stty** can be used from within the layer to block or unblock the output of that layer.

### Ptydaemon

For **shl** to function properly, the **ptydaemon** process must be running on the system. If your system has been installed with the Desktop HP-UX product, then **ptydaemon** will not be started by default. In order to start this daemon, change **PTYDAEMON\_START** from a "0" to a "1" in the **/etc/rc.config.d/ptydaemon** file. The system must either be rebooted for this change to take effect, or you can manually start this daemon by typing :

```
/usr/sbin/ptydaemon
```

Note that **ptydaemon** will also be disabled if the *DesktopConfig.LITECONFIG* fileset has been installed on the system, or if the system administrator has previously run the **SAM** utility and selected the **Apply Lite HP-UX Configuration Action** from within any of **SAM's Kernel Configuration** screens.

## FILES

**\$SHELL** Variable containing path name of the shell to use (default is **/usr/bin/sh**).

## SEE ALSO

sh(1), stty(1), ioctl(2), signal(5).

## STANDARDS CONFORMANCE

**sh1**: SVID2, SVID3, XPG2

| **s** |



**NAME**

size - print section sizes of object files

**SYNOPSIS**

**size** [-d] [-o] [-x] [-V] [-v] [-f] [-F] [-n] [-U] *files*

**DESCRIPTION**

**size** produces section size information for each section in the object files. The size of the text, data and bss (uninitialized data) sections are printed along with the total size of the object file. If an archive file is input to the **size** command, the information for all archive members is displayed.

**Options**

**size** recognizes the following options:

- d Print sizes in decimal. This is the default.
- o Print sizes in octal.
- x Print sizes in hexadecimal.
- V Print version information about the **size** command.
- v Print a verbose list of the subspaces in the object files. Each subspace is listed on a separate line with its size, physical address, and virtual address.
- f Print the size of each allocatable section (ELF only).
- F Print the size and permission bits of each loadable segment (ELF only).
- n Print the sizes of non loadable segments or non allocatable sections (ELF only).
- U Print the usage menu.

**EXTERNAL INFLUENCES****Environment Variables**

The following internationalization variables affect the execution of **size**:

**LANG**

Determines the locale category for native language, local customs and coded character set in the absence of **LC\_ALL** and other **LC\_\*** environment variables. If **LANG** is not specified or is set to the empty string, a default of **C** (see *lang(5)*) is used instead of **LANG**.

**LC\_ALL**

Determines the values for all locale categories and has precedence over **LANG** and other **LC\_\*** environment variables.

**LC\_MESSAGES**

Determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

**LC\_NUMERIC**

Determines the locale category for numeric formatting.

**LC\_CTYPE**

Determines the locale category for character handling functions.

**ST\_SIZECAT****NLSPATH**

Determines the location of message catalogues for the processing of **LC\_MESSAGES**.

If any internationalization variable contains an invalid setting, **size** behaves as if all internationalization variables are set to **C**. See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**DIAGNOSTICS**

- size: name: cannot open** *name* cannot be read.
- size: name: bad magic** *name* is not an appropriate object file.

**EXAMPLES**

Compare the sizes of the text, data, and bss sections for two versions of a program:

```
size ./version1 ./version2
```

**SEE ALSO****System Tools:**

*as*(1) translate assembly code to machine code  
*cc*(1) invoke the HP-UX C compiler  
*ld*(1) invoke the link editor

**Miscellaneous:**

*a.out*(4) assembler, compiler, and linker output  
*ar*(4) archive format

**STANDARDS CONFORMANCE**

**size:** SVID2, SVID3, XPG2, XPG4

**| s |**

**NAME**

sleep - suspend execution for an interval

**SYNOPSIS**

**sleep** *time*

**DESCRIPTION**

**sleep** suspends execution for *time* seconds. It is used to execute a command after a certain amount of time, as in:

```
(sleep 105; command)&
```

or to execute a command periodically, as in:

```
while true
do
 command
 sleep 37
done
```

**RETURN VALUE**

**sleep** exits with one of the following values:

- 0** The execution was successfully suspended for *time* seconds, or a SIGALRM signal was received.
- >0** If the *time* operand is missing, is not a decimal integer, is negative, or is greater than **UINT\_MAX**, **sleep** returns with exit status 2.

**SEE ALSO**

alarm(2), sleep(3C).

**STANDARDS CONFORMANCE**

**sleep**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| s |**

**NAME**

slp - set printing options for a non-serial printer

**SYNOPSIS**

**slp** [-a] [-b] [-c *cols*] [-d] [-i *indent*] [-k] [-l *lines*] [-n] [-o] [-r] [-C *pages*] [-O *pages*]

**DESCRIPTION**

**slp** sets printer formatting options such as the number of lines per page, number of characters per line, and indentation. These characteristics are controlled by the printer driver as described in *lp(7)*. **slp** acts on the current standard output.

**Options**

**slp** recognizes the following options and arguments:

- a** Report all option settings.
- b** Specify a character printer where backspace characters pass through the driver unchanged. The absence of this option indicates a line printer. The driver takes the necessary action to accommodate backspace characters.
- o** Resets the printer back to line-printer mode.
- c*cols*** Limit the number of columns to be printed to *cols*. Characters beyond the specified limit are truncated.
- d** Reset options to default for the device. This action is not taken until the next open occurs on the device.
- i*indent*** Indent *indent* columns before printing the first column.
- k** Select cooked mode. Cooked mode must be used with a cooked device special file which is identified by an *lp* mnemonic that is not preceded by the character **r**.
- l*lines*** Specify the number of *lines* per page. The last new-line character of each page is changed to a form-feed.
- n** Set the page size to infinity. Since the last new-line of the page is never encountered, no new-line characters are changed to form-feeds.
- r** Select a raw mode for graphics dumps. All other options are ignored except **-a**. If the **-r** option is not given, **-k** is assumed.
- C*pages*** Eject zero or more *pages* after the final close of the device.
- O*pages*** Eject zero or more *pages* when the device is opened.

**S****EXAMPLES**

In a typical case, the printer is set to 80 columns, no indentation, with no form-feeds between pages:

```
slp -c80 -i0 -n >/dev/lp
```

**WARNINGS**

Use of the **slp** command in conjunction with the **lp** spooler (see *lp(1)*) might cause undesirable side effects. The spooler model files make assumptions regarding the configuration and can get confused when the default values are altered. Although most options can be altered without difficulty, special problems sometimes result from adjusting the number of lines and the number of columns per page.

**AUTHOR**

**slp** was developed by HP.

**SEE ALSO**

*lp(1)*, *ioctl(2)*, *lp(7)*.

**NAME**

soelim - eliminate .so's from nroff input

**SYNOPSIS**

**soelim** [*file* ...]

**DESCRIPTION**

**soelim** reads the specified files or the standard input and performs the textual inclusion implied by **nroff** directives of the form

**.so** *some\_file*

when they appear at the beginning of input lines. This is useful when using programs such as *tbl*(1) that do not normally do this, allowing placement of individual tables or other text objects in separate files to be run as a part of a large document.

An argument consisting of a single minus (-) is taken to be a file name corresponding to the standard input.

Note that inclusion can be suppressed by using ' instead of . at the start of the line as in:

'so /usr/share/lib/tmac/tmac.s

**EXAMPLES**

**soelim** is often used in a context similar to the following:

**soelim** *exum?.n* | **tbl** | **nroff** -*mm* | **col** | **lp**

**WARNINGS**

The format of the source commands must involve no strangeness — exactly one blank must precede and no blanks follow the file name.

**SEE ALSO**

more(1), nroff(1), tbl(1).

| **s** |

**NAME**

sort - sort or merge files

**SYNOPSIS**

```
sort [-m] [-o output] [-bdfinrum] [-t char] [-k keydef] [-y [kmem]] [-z recsz] [-T dir]
[file ...]
```

```
sort [-c] [-Abdfinrum] [-t char] [-k keydef] [-y [kmem]] [-z recsz] [-T dir] [file ...]
```

**DESCRIPTION**

**sort** performs one of the following functions:

1. Sorts lines of all the named files together and writes the result to the specified output.
2. Merges lines of all the named (presorted) files together and writes the result to the specified output.
3. Checks that a single input file is correctly presorted.

The standard input is read if `-` is used as a file name or no input files are specified.

Comparisons are based on one or more sort keys extracted from each line of input. By default, there is one sort key, the entire input line. Ordering is lexicographic by characters using the collating sequence of the current locale. If the locale is not specified or is set to the **POSIX** locale, then ordering is lexicographic by bytes in machine-collating sequence. If the locale includes multi-byte characters, single-byte characters are machine-collated before multi-byte characters.

**Behavior Modification Options**

The following options alter the default behavior:

- A** Sorts on a byte-by-byte basis using each character's encoded value. On some systems, extended characters will be considered negative values, and so sort before ASCII characters. If you are sorting ASCII characters in a non-C/POSIX locale, this flag performs much faster.
- c** Check that the single input file is sorted according to the ordering rules. No output is produced; the exit code is set to indicate the result.
- m** Merge only; the input files are assumed to be already sorted.
- o *output*** The argument given is the name of an output file to use instead of the standard output. This file can be the same as one of the input files.
- u** Unique: suppress all but one in each set of lines having equal keys. If used with the **-c** option, check to see that there are no lines with duplicate keys, in addition to checking that the input file is sorted.
- y [*kmem*]** The amount of main memory used by the sort can have a large impact on its performance. If this option is omitted, **sort** begins using a system default memory size, and continues to use more space as needed. If this option is presented with a value, *kmem*, **sort** starts using that number of kilobytes of memory, unless the administrative minimum or maximum is violated, in which case the corresponding extremum will be used. Thus, **-y 0** is guaranteed to start with minimum memory. By convention, **-y** (with no argument) starts with maximum memory.
- z *recsz*** The size of the longest line read is recorded in the sort phase so that buffers can be allocated during the merge phase. If the sort phase is omitted via the **-c** or **-m** options, a popular system default size will be used. Lines longer than the buffer size will cause **sort** to terminate abnormally. Supplying the actual number of bytes in the longest line to be merged (or some larger value) will prevent abnormal termination.
- T *dir*** Use *dir* as the directory for temporary scratch files rather than the default directory, which is one of the following, tried in order: the directory as specified in the **TMPDIR** environment variable; **/var/tmp**, and finally, **/tmp**.

**Ordering Rule Options**

When ordering options appear before restricted sort key specifications, the ordering rules are applied globally to all sort keys. When attached to a specific sort key (described below), the ordering options override all global ordering options for that key.

The following options override the default ordering rules:

- d** Quasi-dictionary order: only alphanumeric characters and blanks (spaces and tabs), as defined by **LC\_CTYPE** are significant in comparisons (see *environ*(5)).  
(XPG4 only.) The behavior is undefined for a sort key to which **-i** or **-n** also applies.
- f** Fold letters. Prior to being compared, all lowercase letters are effectively converted into their uppercase equivalents, as defined by **LC\_CTYPE**.
- i** In non-numeric comparisons, ignore all characters which are non-printable, as defined by **LC\_CTYPE**. For the ASCII character set, octal character codes 001 through 037 and 0177 are ignored.
- n** The sort key is restricted to an initial numeric string consisting of optional blanks, an optional minus sign, zero or more digits with optional radix character, and optional thousands separators. The radix and thousands separator characters are defined by **LC\_NUMERIC**. The field is sorted by arithmetic value. An empty (missing) numeric field is treated as arithmetic zero. Leading zeros and plus or minus signs on zeros do not affect the ordering. The **-n** option implies the **-b** option (see below).
- r** Reverse the sense of comparisons.
- M** Compare as months. The first several non-blank characters of the field are folded to uppercase and compared with the *langinfo*(5) items **ABMON\_1** < **ABMON\_2** < ... < **ABMON\_12**. An invalid field is treated as being less than **ABMON\_1** string. For example, American month names are compared such that **JAN** < **FEB** < ... < **DEC**. An invalid field is treated as being less than all months. The **-M** option implies the **-b** option (see below).

### Field Separator Options

The treatment of field separators can be altered using the options:

- t char** Use *char* as the field separator character; *char* is not considered to be part of a field (although it can be included in a sort key). Each occurrence of *char* is significant (for example, <*char*><*char*> delimits an empty field). If **-t** is not specified, <blank> characters will be used as default field separators; each maximal sequence of <blank> characters that follows a non-<blank> character is a field separator.
- b** Ignore leading blanks when determining the starting and ending positions of a restricted sort key. If the **-b** option is specified before the first **-k** option (*+pos1* argument), it is applied to all **-k** options (*+pos1* arguments). Otherwise, the **-b** option can be attached independently to each **-k** *field\_start* or *field\_end* option (*+pos1* or *-pos2* argument; see below). Note that the **-b** option is only effective when restricted sort key specifications are given.

### Restricted Sort Key

- k keydef** The *keydef* argument defines a restricted sort key. The format of this definition is

*field\_start* [*type*] [, *field\_end* [*type*]]

which defines a key field beginning at *field\_start* and ending at *field\_end*. The characters at positions *field\_start* and *field\_end* are included in the key field, providing that *field\_end* does not precede *field\_start*. A missing *field\_end* means the end of the line. Fields and characters within fields are numbered starting with 1. Note that this is different than the obsolete form of restricted sort keys, where numbering starts at 0. See *WARNINGS* below.

Specifying *field\_start* and *field\_end* involves the notion of a field, a minimal sequence of characters followed by a field separator or a new-line. By default, the first blank of a sequence of blanks acts as the field separator. All blanks in a sequence of blanks are considered to be part of the next field; for example, all blanks at the beginning of a line are considered to be part of the first field.

The arguments *field\_start* and *field\_end* each have the form *m.n* which are optionally followed by one or more of the *type* options **b**, **d**, **f**, **i**, **n**, **r**, or **M**. These modifiers have the functionality for this key only, that their command-line counterparts have for the entire record.

A *field\_start* position specified by *m.n* is interpreted to mean the *n*th character in the *m*th field. A missing *n* means *.1*, indicating the first character of the *m*th field. If the **-b** option is in effect, *n* is counted from the first non-blank character in the *m*th field.

A *field\_end* position specified by *m.n* is interpreted to mean the *n*th character in the *m*th field. If *n* is missing, the *m*th field ends at the last character of the field. If the **-b** option is in effect, *n* is counted from the first non-<blank> character in the *m*th field.

Multiple **-k** options are permitted and are significant in command line order. A maximum of 9 **-k** options can be given. If no **-k** option is specified, a default sort key of the entire line is used. When there are multiple sort keys, later keys are compared only after all earlier keys compare equal. Lines that otherwise compare equal are ordered with all bytes significant. If all the specified keys compare equal, the entire record is used as the final key.

The **-k** option is intended to replace the obsolete [*+pos1* [*+pos2*]] notation, using *field\_start* and *field\_end* respectively. The fully specified [*+pos1* [*+pos2*]] form:

```
+w.x-y.z
```

is equivalent to:

```
-k w+1.x+1,y.0 (if z == 0)
```

```
-k w+1.x+1,y+1.z (if z > 0)
```

### Obsolete Restricted Sort Key

The notation *+pos1 -pos2* restricts a sort key to one beginning at *pos1* and ending at *pos2*. The characters at positions *pos1* and *pos2* are included in the sort key (provided that *pos2* does not precede *pos1*). A missing *-pos2* means the end of the line.

Specifying *pos1* and *pos2* involves the notion of a field, a minimal sequence of characters followed by a field separator or a new-line. By default, the first blank (space or tab) of a sequence of blanks acts as the field separator. All blanks in a sequence of blanks are considered to be part of the next field; for example, all blanks at the beginning of a line are considered to be part of the first field.

*pos1* and *pos2* each have the form *m.n* optionally followed by one or more of the flags **bdfinrM**. A starting position specified by *+m.n* is interpreted to mean character *n+1* in field *m+1*. A missing *.n* means *.0*, indicating the first character of field *m+1*. If the **b** flag is in effect, *n* is counted from the first non-blank in field *m+1*; *+m.0b* refers to the first non-blank character in field *m+1*.

A last position specified by *-m.n* is interpreted to mean the *n*th character (including separators) after the last character of the *m*th field. A missing *.n* means *.0*, indicating the last character of the *m*th field. If the **b** flag is in effect, *n* is counted from the last leading blank in field *m+1*; *-m.1b* refers to the first non-blank in field *m+1*.

| S |

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_COLLATE** determines the default ordering rules applied to the sort.

**LC\_CTYPE** determines the locale for interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files) and the behavior of character classification for the **-b**, **-d**, **-f**, **-i**, and **-n** options.

**LC\_NUMERIC** determines the definition of the radix and thousands separator characters for the **-n** option.

**LC\_TIME** determines the month names for the **-M** option.

**LC\_MESSAGES** determines the language in which messages are displayed.

**LC\_ALL** determines the locale to use to override the values of all the other internationalization variables.

**NLS\_PATH** determines the location of message catalogs for the processing of **LC\_MESSAGES**.

**LANG** provides a default value for the internationalization variables that are unset or null. If **LANG** is unset or null, the default value of "C" (see *lang(5)*) is used.

If any of the internationalization variables contains an invalid setting, **sort** behaves as if all internationalization variables are set to "C". See *environ(5)*.



**International Code Set Support**

Single- and multi-byte character code sets are supported.

**EXAMPLES**

Sort the contents of **infile** with the second field as the sort key:

```
sort -k 2,2 infile
```

Sort, in reverse order, the contents of **infile1** and **infile2**, placing the output in **outfile** and using the first two characters of the second field as the sort key:

```
sort -r -o outfile -k 2.1,2.2 infile1 infile2
```

Sort, in reverse order, the contents of **infile1** and **infile2**, using the first non-blank character of the fourth field as the sort key:

```
sort -r -k 4.1b,4.1b infile1 infile2
```

Print the password file (**/etc/passwd**) sorted by numeric user ID (the third colon-separated field):

```
sort -t: -k 3n,3 /etc/passwd
```

Print the lines of the presorted file **infile**, suppressing all but the first occurrence of lines having the same third field:

```
sort -mu -k 3,3 infile
```

**DIAGNOSTICS**

**sort** exits with one of the following values:

- 0 All input files were output successfully, or **-c** was specified and the input file was correctly presorted.
- 1 Under the **-c** option, the file was not ordered as specified, or if the **-c** and **-u** options were both specified, two input lines were found with equal keys. This exit status is not returned if the **-c** option is not used.
- >1 An error occurred such as when one or more input lines are too long.

When the last line of an input file is missing a new-line character, **sort** appends one, prints a warning message, and continues.

If an error occurs when accessing the tables that contain the collation rules for the specified language, **sort** prints a warning message and defaults to the **POSIX** locale.

If a **-d**, **-f**, or **-i** option is specified for a language with multi-byte characters, **sort** prints a warning message and ignores the option.

**WARNINGS**

Numbering of fields and characters within fields (**-k** option) has changed to conform to the POSIX standard. Beginning at HP-UX Release 9.0, the **-k** option numbers fields and characters within fields, starting with **1**. Prior to HP-UX Release 9.0, numbering started at **0**.

A field separator specified by the **-t** option is recognized only if it is a single-byte character.

The character type classification categories **alpha**, **digit**, **space**, and **print** are not defined for multi-byte characters. For languages with multi-byte characters, all characters are significant in comparisons.

For **non-text** input files, the behaviour is undefined.

**AUTHOR**

**sort** was developed by OSF and HP.

**FILES**

```
/var/tmp/stm???
/tmp/stm???
```

**SEE ALSO**

comm(1), join(1), uniq(1), environ(5), lang(5).

**sort(1)**

**sort(1)**

**STANDARDS CONFORMANCE**

**sort:** SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| **s** |

**NAME**

spell, hashmake, spellin, hashcheck - find spelling errors

**SYNOPSIS**

```
spell [-v] [-b] [-x] [-l] [-i] [+local_file] [files]
/usr/sbin/spell/hashmake
/usr/sbin/spell/spellin n
/usr/sbin/spell/hashcheck spelling_list
```

**DESCRIPTION**

The **spell** command collects words from the named *files* and looks them up in a spelling list. Words that neither occur among nor are derivable (by applying certain inflections, prefixes, and/or suffixes) from words in the spelling list are printed on the standard output. If no *files* are named, words are collected from the standard input.

The **spell** command ignores most **troff**, **tbl**, and **eqn** constructions.

**Options**

The **spell** command recognizes the following options:

- v** All words not literally in the spelling list are printed, and plausible derivations from the words in the spelling list are indicated.
- b** British spelling is checked. Besides preferring **centre**, **colour**, **programme**, **speciality**, **travelled**, etc., this option insists upon **-ise** in certain words, such as in **standardise**.
- x** Every plausible stem is printed with = for each word.

By default, **spell** follows chains of included files much like **deroff** (see *deroff(1)*) which recognizes the **troff/nroff** intrinsics **.so** and **.nx**, *unless* the names of such included files begin with **/usr/share/lib**. If the **-l** option is used, **spell** follows the chains of *all* included files. With the **-i** option, **spell** ignores all chains of included files.

If the **+local\_file** option is used, words found in *local\_file* are removed from **spell**'s output. *local\_file* is the name of a user-provided file containing a sorted list of words, one per line. With this option, the user can specify a set of words that are correct spellings (in addition to *spell*'s own spelling list) for each job.

The spelling list is based on many sources, and while more haphazard than an ordinary dictionary, is also more effective with respect to proper names and popular technical words. Coverage of the specialized vocabularies of biology, medicine, and chemistry is light.

Pertinent auxiliary files can be specified by name arguments, indicated below with their default settings (see **FILES** and **VARIABLES**). Copies of all output are accumulated in the history file. The stop list filters out misspellings (such as **thier=thy-y+ier**) that would otherwise pass.

Three routines help maintain and check the hash lists used by **spell**:

- hashmake** Reads a list of words from the standard input and writes the corresponding nine-digit hash code on the standard output.
- spellin n** Reads *n* hash codes from the standard input and writes a compressed spelling list on the standard output. Information about the hash coding is printed on standard error.
- hashcheck** Reads a compressed *spelling\_list* and recreates the nine-digit hash codes for all the words in it; it writes these codes on the standard output.

**EXTERNAL INFLUENCES****Environment Variables**

- D\_SPELL** Your hashed spelling list (default is **D\_SPELL=/usr/share/dict/hlist(ab)**)
- H\_SPELL** Spelling history (default is **H\_SPELL=/var/adm/spellhist**).
- S\_SPELL** Your hashed stop list (default is **S\_SPELL=/usr/share/dict/hstop**).
- TMPDIR** Directory for temporary files; overrides the default **/tmp**.

**EXAMPLES**

To check spelling of a single *word*:

```
echo word | spell
```

If *word* is spelled correctly, a prompt is returned. If it is spelled incorrectly, *word* is printed before the prompt is returned. To check spelling of multiple words, they can also be typed as a group on the same command line:

```
echo worda wordb wordc ... | spell
```

To create a personal spelling list that incorporates the words already present in the default American spelling list file `/usr/share/dict/hlista`:

```
cat /usr/share/dict/hlista | /usr/sbin/spell/hashcheck >tmp1
/usr/sbin/spell/hashmake <addwds >>tmp1
sort -u -o tmp1 tmp1
/usr/sbin/spell/spellin `wc -l <tmp1` <tmp1 >hlista
```

To modify the default British spelling list file `/usr/share/dict/hlistb`, replace all occurrences of `hlista` with `hlistb` in the above example.

To add words to the default spelling list, change login to `root`, change the current working directory to `/usr/share/dict` and execute the commands listed in the above example.

**WARNINGS**

The spelling list's coverage is uneven. When undertaking the use of `spell` as a new tool, it may be advisable to monitor the output for several months to gather local additions. Typically, these are kept in a separate local file that is added to the hashed *spelling\_list* via `spellin`, as shown above.

The British spelling feature was developed by an American.

Start-up versions of files `hlista`, `hlistb`, and `hstop` are available in directory `/usr/newconfig/usr/share/dict`. If these files or a suitable equivalent are not present in directory `/usr/share/dict`, `spell` complains:

```
spell: cannot initialize hash table
spell: cannot initialize hash table
```

The `spell` command is likely to be withdrawn from X/Open standards. Applications using this command might not be portable to other vendors' systems.

**FILES**

|                                        |                                              |
|----------------------------------------|----------------------------------------------|
| <code>/usr/share/dict/hlist[ab]</code> | Hashed spelling lists, American and British. |
| <code>/usr/share/dict/hstop</code>     | Hashed stop list.                            |
| <code>/var/adm/spellhist</code>        | History file.                                |
| <code>/usr/sbin/spell/spellprog</code> | Executable program file.                     |

**SEE ALSO**

`deroff(1)`, `sed(1)`, `sort(1)`, `tbl(1)`, `tee(1)`.

**STANDARDS CONFORMANCE**

`spell`: SVID2, SVID3, XPG2, XPG3

| S |

**NAME**

split - split a file into pieces

**SYNOPSIS**

**split** [-l *line\_count*] [-a *suffix\_length*] [*file* [*name*]]

**split** [-b *n*[**k**|**m**]] [-a *suffix\_length*] [*file* [*name*]]

**Obsolescent**

**split** [-n] [*file* [*name*]]

**DESCRIPTION**

**split** reads *file* and writes it in pieces (default 1000 lines) onto a set of output files. The name of the first output file is *name* with **aa** appended, and so on lexicographically, up to **zz** (only ASCII letters are used, a maximum of 676 files). If no output *name* is given, **x** is the default.

If no input *file* is given, or if **-** is given instead, the standard input file is used.

**Options**

**split** recognizes the following command-line options and arguments:

**-l** *line\_count* The input file is split into pieces *line\_count* lines in size.

**-a** *suffix\_length*

*suffix\_length* letters are used to form the suffix of the output filenames. This option allows creation of more than 676 output files. The output file names created cannot exceed the maximum file name length allowed in the directory containing the files.

**-b** *n* The input file is split into pieces *n* bytes in size.

**-b** *n***k** The input file is split into pieces  $n \times 1024$  bytes in size. No space separates the *n* from the **k**.

**-b** *n***m** The input file is split into pieces  $n \times 1048576$  bytes in size. No space separates the *n* from the **m**.

**-n** The input file is split into pieces *n* lines in size. This option is obsolescent and is equivalent to using the **-l** *line\_count* option.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the locale for the interpretation of text as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **split** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**SEE ALSO**

csplit(1).

**STANDARDS CONFORMANCE**

**split**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| S |

**NAME**

ssp - remove multiple line-feeds from output

**SYNOPSIS**

**ssp**

**DESCRIPTION**

**ssp** (single-space) removes redundant blank lines from the standard input and sends the result to the standard output. All blank lines at the beginning of a file are removed, and all multiple blank lines elsewhere in the file (including end-of-file) are reduced to a single blank line.

**ssp** is typically used in pipelines such as

```
nroff -ms file1 | ssp
```

**ssp** is equivalent to the 4.2BSD **cat -s** command.

To remove all blank lines from a file except at beginning of file, use **rmnl** (see *rmnl(1)*). To remove all blank lines from a file including beginning of file, use **rmnl** piped to **ssp**, or **ssp** piped to **rmnl**.

**SEE ALSO**

cat(1), rmnl(1).

| **s** |

**NAME**

strings - find the printable strings in an object or other binary file

**SYNOPSIS**

**strings** [-a] [-t *format*] [-n *number*] [*file*] ...

**Obsolescent**

**strings** [-a] [-o] [-*number*] [*file*] ...

**DESCRIPTION**

**strings** looks for ASCII strings in a file. If no *file* is specified, standard input is used. A string is any sequence of four or more printing characters ending with a newline or null character.

**strings** is useful for identifying random object files and many other things.

**Options**

**strings** recognizes the following options:

- a** By default, *strings* looks only in the initialized data space of object files (as recognized by their magic numbers). If this flag is used, the entire file is inspected. This flag is always set if standard input is being read or the file is not recognized as an object file. For backward compatibility, **-** is understood as a synonym for **-a**.
- t *format*** Write each string preceded by its byte offset from the start of the file. The format is dependent on the single character used as the *format* option-argument:
  - d* The offset is written in decimal.
  - o* The offset is written in octal.
  - x* The offset is written in hexadecimal.
- n *number*** Specify *number* as the minimum string length, rather than the default 4.
- o** Each string is preceded by its offset in the file (in octal). This option is obsolescent and is equivalent to specifying the **-t o** option.
- number*** Specify *number* as the minimum string length, rather than the default 4. This option is obsolescent and is equivalent to using the **-n *number*** option.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the locale for the interpretation of text as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

**NLSPATH** Determine the location of message catalogues for the processing of **LC\_MESSAGES**.

If any internationalization variable contains an invalid setting, **strings** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**WARNINGS**

The algorithm for identifying strings is extremely primitive.

**AUTHOR**

**strings** was developed by the University of California, Berkeley.

**SEE ALSO**

od(1).

**STANDARDS CONFORMANCE**

**strings**: XPG4, POSIX.2

**NAME**

strip - strip symbol and line number information from an object file

**SYNOPSIS**

**strip** [-l] [-x] [-r] [-v] [-U] *filename* ...

**DESCRIPTION**

**strip** removes the symbol table and line number information from object files, including archives. Thereafter, no symbolic debugging access is available for that file; thus, this command is normally run only on production modules that have been debugged and tested. The effect is nearly identical to using the **-s** option of *ld*.

**Options**

The amount of information stripped from the symbol table can be controlled by using any of the following options:

- l** Strip line number information only; do not strip any symbol table information.
- x** Do not strip static or external symbol information.  
Note that the **-l** and **-x** options are synonymous because the symbol table contains only static and external symbols. Either option strips only symbolic debugging information and unloadable data.
- r** Reset the relocation indexes into the symbol table (SOM only). Obsolete for ELF files. This option allows **strip** to be run on relocatable files, in which case the effect is also to strip only symbolic debugging information and unloadable data.
- v** Print the version of the strip command on the standard error output.
- U** Print the usage menu.

If there are any relocation entries in the object file and any symbol table information is to be stripped, **strip** complains and terminates without stripping *filename* unless the **-r** option is used.

If **strip** is executed on an archive file (see *ar(4)*), the archive symbol table is removed. The archive symbol table must be restored by executing **ar** with its **s** operator (see *ar(1)*) before the archive can be used by the **ld** command (see *ld(1)*). **strip** instructs the user with appropriate warning messages when this situation arises.

The purpose of this command is to reduce file storage overhead consumed by the object file.

**EXTERNAL INFLUENCES****Environment Variables**

The following internationalization variables affect the execution of **strip**:

**LANG**

Determines the locale category for native language, local customs and coded character set in the absence of **LC\_ALL** and other **LC\_\*** environment variables. If **LANG** is not specified or is set to the empty string, a default of **C** (see *lang(5)*) is used instead of **LANG**.

**LC\_ALL**

Determines the values for all locale categories and has precedence over **LANG** and other **LC\_\*** environment variables.

**LC\_MESSAGES**

Determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

**LC\_NUMERIC**

Determines the locale category for numeric formatting.

**LC\_CTYPE**

Determines the locale category for character handling functions.

**ST\_STRIPCAT****NLSPATH**

Determines the location of message catalogues for the processing of **LC\_MESSAGES**.

If any internationalization variable contains an invalid setting, **strip** behaves as if all internationalization variables are set to **C**. See *environ(5)*.



In addition, the following environment variable affects **strip**:

**TMPDIR**

Specifies a directory for temporary files (see *tmpnam(3S)*).

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**DIAGNOSTICS**

**strip: name: cannot open**

*name* cannot be read.

**strip: name: bad magic**

*name* is not an appropriate object file.

**strip: name: relocation entries present; cannot strip**

*name* contains relocation entries and the **-r** option was not specified. Symbol table information cannot be stripped.

**EXAMPLES**

Strip symbol table and debug information from the shared library **libfoo.so** in the current directory to reduce its size. Symbol information required to use the library is preserved:

```
strip ./libfoo.so
```

**FILES**

**/var/tmp/SGSstrp\*** temporary files

**SEE ALSO****System Tools:**

*ar*(1) create archived libraries  
*as*(1) translate assembly code to machine code  
*cc*(1) invoke the HP-UX C compiler  
*ld*(1) invoke the link editor

**Miscellaneous:**

*a.out*(4) assembler, compiler, and linker output  
*ar*(4) archive format

**STANDARDS CONFORMANCE**

**strip**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| **s** |

**NAME**

stty - set the options for a terminal port

**SYNOPSIS**

stty [-a | -g | *options* ]

**DESCRIPTION**

**stty** sets or reports current settings of certain terminal I/O options for the device that is the current standard input. The command takes four forms:

|                     |                                                                                                   |
|---------------------|---------------------------------------------------------------------------------------------------|
| <b>stty</b>         | Report the settings of a system-defined set of options;                                           |
| <b>stty -a</b>      | Report all of current option settings;                                                            |
| <b>stty -g</b>      | Report current settings in a form that can be used as an argument to another <b>stty</b> command. |
| <b>stty options</b> | Set terminal I/O options as defined by <i>options</i> .                                           |

For detailed information about the modes listed below from Control Modes through Local Modes as they relate to asynchronous lines, see *termio*(7). For detailed information about the modes listed under Hardware Flow Control Modes below, see *termiox*(7).

Options in the Combination Modes group are implemented using options in the previous groups. Note that many combinations of options make no sense, but no sanity checking is performed.

*options* are selected from the following:

**Control Modes**

|                                                                                                                             |                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>rows</b> <i>number</i>                                                                                                   | Set the terminal window row size equal to <i>number</i> .                                                                                                                             |
| <b>columns</b> <i>number</i>                                                                                                | Set the terminal window column size (width) equal to <i>number</i> . <b>cols</b> can be used as an abbreviation for <b>columns</b> .                                                  |
| <b>parenb</b> ( <b>-parenb</b> )                                                                                            | Enable (disable) parity generation and detection.                                                                                                                                     |
| <b>parodd</b> ( <b>-parodd</b> )                                                                                            | Select odd (even) parity.                                                                                                                                                             |
| <b>cs5 cs6 cs7 cs8</b>                                                                                                      | Select character size (see <i>termio</i> (7)).                                                                                                                                        |
| <b>0</b>                                                                                                                    | Hang up phone line immediately.                                                                                                                                                       |
| <b>50 75 110 134.5 150 200 300 600 900 1200 1800 2400<br/>3600 4800 7200 9600 19200 38400 57600 115200 230400 exta extb</b> | Set terminal baud rate to the number given, if possible (some hardware interfaces do not support all of the speeds listed here). Speeds above 38400 are supported on Series 700 only. |
| <b>ispeed</b> <i>number</i>                                                                                                 | Set terminal input baud rate to <i>number</i> . If <i>number</i> is zero, the input baud rate is set to the value of the output baud rate.                                            |
| <b>ospeed</b> <i>number</i>                                                                                                 | Set terminal output baud rate to <i>number</i> . If <i>number</i> is zero, the modem control lines are released, which in turn disconnects the line.                                  |
| <b>hupcl</b> ( <b>-hupcl</b> )                                                                                              | Hang up (do not hang up) modem connection on last close.                                                                                                                              |
| <b>hup</b> ( <b>-hup</b> )                                                                                                  | Same as <b>hupcl</b> ( <b>-hupcl</b> ).                                                                                                                                               |
| <b>cstopb</b> ( <b>-cstopb</b> )                                                                                            | Use two (one) stop bits per character.                                                                                                                                                |
| <b>cread</b> ( <b>-cread</b> )                                                                                              | Enable (disable) the receiver.                                                                                                                                                        |
| <b>crts</b> ( <b>-crts</b> )                                                                                                | Enable (disable) request-to-send.                                                                                                                                                     |
| <b>clocal</b> ( <b>-clocal</b> )                                                                                            | Assume a line without (with) modem control.                                                                                                                                           |
| <b>loblk</b> ( <b>-loblk</b> )                                                                                              | Block (do not block) output from a noncurrent layer.                                                                                                                                  |
| <b>+resetGSP</b>                                                                                                            | Reset the Guardian Service Processor (GSP) of the console. This mode can be used only by the superuser. This is supported only on specific hardware.                                  |

**Input Modes**

|                                  |                                        |
|----------------------------------|----------------------------------------|
| <b>ignbrk</b> ( <b>-ignbrk</b> ) | Ignore (do not ignore) break on input. |
|----------------------------------|----------------------------------------|

|                           |                                                                                                                            |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>ienqak (-ienqak)</b>   | Enable (disable) ENQ-ACK Handshaking.                                                                                      |
| <b>brkint (-brkint)</b>   | Signal (do not signal) INTR on break.                                                                                      |
| <b>ignpar (-ignpar)</b>   | Ignore (do not ignore) parity errors.                                                                                      |
| <b>parmrk (-parmrk)</b>   | Mark (do not mark) parity errors (see <i>termio(7)</i> ).                                                                  |
| <b>inpck (-inpck)</b>     | Enable (disable) input parity checking.                                                                                    |
| <b>istrip (-istrip)</b>   | Strip (do not strip) input characters to seven bits.                                                                       |
| <b>inlcr (-inlcr)</b>     | Map (do not map) newline character to carriage return (CR) on input.                                                       |
| <b>igncr (-igncr)</b>     | Ignore (do not ignore) CR on input.                                                                                        |
| <b>icrnl (-icrnl)</b>     | Map (do not map) CR to a newline character on input.                                                                       |
| <b>iuclc (-iuclc)</b>     | Map (do not map) uppercase alphabetic characters to lowercase on input.                                                    |
| <b>ixon (-ixon)</b>       | Enable (disable) START/STOP output control. Output is stopped by sending an ASCII DC3 and started by sending an ASCII DC1. |
| <b>ixany (-ixany)</b>     | Allow any character (only DC1) to restart output.                                                                          |
| <b>ixoff (-ixoff)</b>     | Request that the system send (not send) START/STOP characters when the input queue is nearly empty/full.                   |
| <b>imaxbel (-imaxbel)</b> | Echo (do not echo) BEL when the input line is too long.                                                                    |

### Output Modes

|                            |                                                                                               |
|----------------------------|-----------------------------------------------------------------------------------------------|
| <b>opost (-opost)</b>      | Post-process output (do not post-process output; ignore all other output modes).              |
| <b>olcuc (-olcuc)</b>      | Map (do not map) lowercase alphabetic characters to uppercase on output.                      |
| <b>onlcr (-onlcr)</b>      | Map (do not map) newline character to a carriage-return/newline character sequence on output. |
| <b>ocrnl (-ocrnl)</b>      | Map (do not map) CR to newline character on output.                                           |
| <b>onocr (-onocr)</b>      | Do not (do) output CRs at column zero.                                                        |
| <b>onlret (-onlret)</b>    | On the terminal, a newline character performs (does not perform) the CR function.             |
| <b>ofill (-ofill)</b>      | Use fill characters (use timing) for delays.                                                  |
| <b>ofdel (-ofdel)</b>      | Fill characters are DELs ( NULs).                                                             |
| <b>cr0 cr1 cr2 cr3</b>     | Select style of delay for carriage returns (see <i>termio(7)</i> ).                           |
| <b>nl0 nl1</b>             | Select style of delay for newline characters (see <i>termio(7)</i> ).                         |
| <b>tab0 tab1 tab2 tab3</b> | Select style of delay for horizontal tabs (see <i>termio(7)</i> ).                            |
| <b>bs0 bs1</b>             | Select style of delay for backspaces (see <i>termio(7)</i> ).                                 |
| <b>ff0 ff1</b>             | Select style of delay for form-feeds (see <i>termio(7)</i> ).                                 |
| <b>vt0 vt1</b>             | Select style of delay for vertical tabs (see <i>termio(7)</i> ).                              |

### Local Modes

|                         |                                                                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>isig (-isig)</b>     | Enable (disable) the checking of characters against the special control characters INTR and QUIT.                                                 |
| <b>icanon (-icanon)</b> | Enable (disable) canonical input (ERASE and KILL processing).                                                                                     |
| <b>iexten (-iexten)</b> | Enable (disable) any implementation-defined special control characters not currently controlled by <i>icanon</i> , <i>isig</i> , or <i>ixon</i> . |
| <b>xcase (-xcase)</b>   | Canonical (unprocessed) uppercase and lowercase presentation.                                                                                     |
| <b>echo (-echo)</b>     | Echo back (do not echo back) every character typed.                                                                                               |
| <b>echoe (-echoe)</b>   | Echo (do not echo) ERASE character as a backspace-space-backspace string. Note: this mode erases the ERASEed character on many CRT terminals.     |

However, it does *not* keep track of column position and, as a result, may not correctly erase escaped characters, tabs, and backspaces.

|                                    |                                                                                     |
|------------------------------------|-------------------------------------------------------------------------------------|
| <b>echok</b> ( <b>-echok</b> )     | Echo (do not echo) a newline character after a KILL character.                      |
| <b>lfkc</b> ( <b>-lfkc</b> )       | (obsolete) Same as <b>echok</b> ( <b>-echok</b> ).                                  |
| <b>echonl</b> ( <b>-echonl</b> )   | Echo (do not echo) newline character.                                               |
| <b>noflsh</b> ( <b>-noflsh</b> )   | Disable (enable) flush after INTR or QUIT.                                          |
| <b>echoctl</b> ( <b>-echoctl</b> ) | Echo (do not echo) control characters as ^char, delete as ^?                        |
| <b>echopr</b> ( <b>-echopr</b> )   | Echo (do not echo) erase character as character is erased.                          |
| <b>echoke</b> ( <b>-echoke</b> )   | BS-SP-BS erase (do not BS-SP-BS erase) entire line on line kill.                    |
| <b>flusho</b> ( <b>-flusho</b> )   | Output is (is not) being flushed.                                                   |
| <b>pendin</b> ( <b>-pendin</b> )   | Retype (do not retype) pending output at next read or input character.              |
| <b>tostop</b> ( <b>-tostop</b> )   | Enable (disable) generation of SIGTTOU signals when background jobs attempt output. |

### Hardware Flow Control Modes

The following options are reserved for use with those devices that support hardware flow control through the termiox interface. If the functionality is supported, this interface must be used.

|                                    |                                                                               |
|------------------------------------|-------------------------------------------------------------------------------|
| <b>rtsxoff</b> ( <b>-rtsxoff</b> ) | enable (disable) RTS hardware flow control on input (see <i>termiox</i> (7))  |
| <b>ctsxon</b> ( <b>-ctsxon</b> )   | enable (disable) CTS hardware flow control on output (see <i>termiox</i> (7)) |

### Control Assignments

*control-character c*

Set *control-character* to *c*, where *control-character* is **erase**, **kill**, **intr**, **quit**, **eof**, **eol**, **eol2**, **werase**, **lnext**, **min**, or **time** (**min** and **time** are used with **-icanon**; see *termio*(7)). For systems that support job control, **susp** and **dsusp** characters can also be set. For systems that support shell layers (see *shl*(1)) **swtch** can also be set. If *c* is preceded by an (escaped from the shell) circumflex (^), the value used is the corresponding control character (for example, ^d represents **Ctrl-d**); ^? is interpreted as DEL and ^-is interpreted as undefined.

|                      |                                                                                                                           |
|----------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>line</b> <i>i</i> | Set line discipline to <i>i</i> where the value of <i>i</i> ranges from zero through 127 decimal (See <i>termio</i> (7)). |
|----------------------|---------------------------------------------------------------------------------------------------------------------------|

### Combination Modes

|                                                  |                                                                                                                                                                                              |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>evenp</b> or <b>parity</b>                    | Enable <b>parenb</b> and <b>cs7</b> .                                                                                                                                                        |
| <b>oddp</b>                                      | Enable <b>parenb</b> , <b>cs7</b> , and <b>parodd</b> .                                                                                                                                      |
| <b>-parity</b> , <b>-evenp</b> , or <b>-oddp</b> | Disable <b>parenb</b> and set <b>cs8</b> .                                                                                                                                                   |
| <b>raw</b> ( <b>-raw</b> or <b>cooked</b> )      | Enable (disable) raw input and output (no ERASE, KILL, INTR, QUIT, EOT, or output post processing). See WARNINGS.                                                                            |
| <b>nl</b> ( <b>-nl</b> )                         | Unset (set) <b>icrnl</b> and <b>onlcr</b> . In addition <b>-nl</b> unsets <b>inlcr</b> , <b>igncr</b> , <b>ocrnl</b> , and <b>onlret</b> .                                                   |
| <b>lcase</b> ( <b>-lcase</b> )                   | Set (unset) <b>xcase</b> , <b>iuclc</b> , and <b>olcuc</b> .                                                                                                                                 |
| <b>LCASE</b> ( <b>-LCASE</b> )                   | Same as <b>lcase</b> ( <b>-lcase</b> ).                                                                                                                                                      |
| <b>tabs</b> ( <b>-tabs</b> or <b>tab3</b> )      | Preserve (expand to spaces) tabs when printing.                                                                                                                                              |
| <b>ek</b>                                        | Reset ERASE and KILL characters back to default # and @.                                                                                                                                     |
| <b>sane</b>                                      | Reset all modes to some reasonable values.                                                                                                                                                   |
| <b>term</b>                                      | Set all modes suitable for the terminal type <i>term</i> , where <i>term</i> is one of <b>tty33</b> , <b>tty37</b> , <b>vt05</b> , <b>tn300</b> , <b>ti700</b> , <b>hp</b> , or <b>tek</b> . |

| S |

**Reporting Functions**

|                  |                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>size</b>      | Print terminal window size to standard output in a rows-and-columns format.                                                                                                    |
| <b>+queryGSP</b> | Print the status of the Guardian Service Processor (GSP) of the console. This function can be used only by the superuser. This feature is available only on specific hardware. |

**Control Character Default Assignments**

The control characters are assigned default values when the terminal port is opened, see *termio(7)*. The default values used are those specified by the System V Interface Definition, Third Edition (SVID3), except for the **werase** and **lnext** control characters, which are set to **\_POSIX\_VDISABLE** to maintain binary compatibility with previous releases of HP-UX.

The default values for the control characters may be changed by a user with root capability by using **stty** and redirecting stdin to the device **/dev/ttyconf**. Any of the four command forms specified in the Description section above may be used. However, only the control character defaults will be reported or altered. It will have no effect on the defaults for any of the other modes.

Note that these defaults will be used for all terminal ports in the system, except the system console, and the changes will not become effective for a particular port until it is (re)opened. The default control character assignment will not work with the system console because the system console is never closed while the system is running, and therefore cannot be reopened.

Care should be exercised when re-assigning the control character defaults. Control character values should be tested with applications before assigning them as a default value.

**EXTERNAL INFLUENCES****Environment Variables**

LC\_CTYPE determines the valid control characters for printing.

If LC\_CTYPE is not specified in the environment or is set to the empty string, the value of LANG is used as a default for each unspecified or empty variable. If LANG is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of LANG. If any internationalization variable contains an invalid setting, **stty** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single-byte character code sets are supported.

**EXAMPLES**

The command:

```
stty kill '^X' intr '^C'
```

sets the delete-line character to **^X** (Ctrl-X) and the interrupt character to **^C**. This command is usually found in the **.login** or **.profile** file so that **^X** and **^C** need not be set by the user at each login session.

The command:

```
stty kill '^X' intr '^C' werase '^W' </dev/ttyconf
```

sets the default values for the delete-line character to **^X** (Ctrl-X), the interrupt character to **^C**, and the word erase character to **^W**. Any terminal port opened after this command is issued will see these new default values for the **kill**, **intr**, and **werase** control characters.

**WARNINGS**

Use of **raw** mode produces certain side effects which have varied from release to release in the past and may vary in the future. Relying on these side effects in applications can lead to unreliable results in the future and is therefore discouraged.

**DEPENDENCIES**

Refer to the DEPENDENCIES section of *termio(7)* for a further description of capabilities that are not supported.

**SEE ALSO**

shl(1), tabs(1), ioctl(2), termio(7), termiox(7).

**STANDARDS CONFORMANCE**

**stty:** SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| **s** |

**NAME**

su - switch user

**SYNOPSIS**

**su** [-] [*username* [*arguments*]]

**su** [-] -d [*username*]

**DESCRIPTION**

The **su** (set user or superuser) command allows one user to become another user without logging out.

*username* is the name of a user defined in the `/etc/passwd` file (see *passwd*(4)). The default name is **root** (that is, superuser).

To use **su**, the appropriate password must be supplied unless the current user is superuser and is not using the **-d** option. If a valid password is entered, **su** executes a new shell with the real and effective user ID, real and effective group ID, and group access list set to that of the specified user. The new shell is the one specified in the shell field of the new user's entry in the password file, `/etc/passwd`.

The *arguments* are passed along to the new shell for execution, permitting the user to run shell procedures with the new user's privileges.

When exiting from the new shell, the previous *username* and environment are restored.

All attempts to become another user are logged in `/var/adm/sulog`, including failures. Successful attempts are flagged with +; failures, with -. They are also logged with **syslog**( ) (see *syslog*(3C)).

**Options**

**su** recognizes the following options:

- If the **-** option is specified, the new shell starts up as if the new user had initiated a new login session. If the **-** option is omitted, the new shell starts as if a subshell was invoked. See more details below.
- d** If DCE (Distributed Computing Environment) is being used as the authentication mechanism, the **-d** option must be specified. With this option, even superuser will be prompted for the user's password. The reason for this is because DCE credentials for a user cannot be obtained without that user's password.

This option cannot be used with shell *arguments*.

If the **-** option is specified, the new shell starts up as if the new user had initiated a new login session. Exceptions are as follows:

- The **HOME** variable is reset to the new user's home directory.
- If the new user name is **root**, the path and prompt variables are reset:

```
PATH=/usr/bin:/usr/sbin:/sbin
PS1=#
```

For other user names:

```
PATH=/usr/bin
PS1=$
```

- The **TERM** variable is retained.
- The rest of the environment is deleted and reset to the login state. However, the login files are normally executed anyway, usually restoring the expected value of **PATH** and other variables.

If the **-** option is omitted, the new shell starts as if a subshell was invoked. Exceptions are as follows:

- If the new user name is **root**, the path and prompt variables are reset:

```
PATH=/usr/bin:/usr/sbin:/sbin
PS1=#
```

- The previously defined **HOME** and **ENV** environment variables are removed.
- The rest of the environment is retained.

If the shell specified in `/etc/passwd` is `/usr/bin/sh`, **su** sets the value of parameter 0 in the new shell (referenced as **\$0**) to **su**. If the **-** option of the **su** command is specified, **su** sets parameter 0 to

| S |

**-su.**

If the shell specified in `/etc/passwd` is not `/usr/bin/sh`, **su** sets the value of parameter `0` in the new shell to *shellname*. If the `-` option of the **su** command is specified, **su** sets parameter `0` to `-shellname`. For example, if the Korn shell is invoked, the value of *shellname* will be either **ksh** or **-ksh**.

By comparison, the **login** command always sets parameter `0` to `-shellname`.

### HP-UX Smart Card Login

If the user account is configured to use a Smart Card, the user password is stored in the card. This password has characteristics identical to a normal password stored on the system.

In order to **su** using a Smart Card account, the Smart Card from the destination user account must be inserted into the Smart Card reader. The user is prompted for a PIN instead of a password during authentication.

#### Enter PIN:

The password is retrieved automatically from the Smart Card when a valid PIN is entered. Therefore, it is not necessary to know the password, only the PIN.

The card is locked if an incorrect PIN is entered three consecutive times. It may be unlocked only by the card issuer.

### SECURITY FEATURES

Except for user **root**, users cannot use **su** to change to an account that has been locked because of expired passwords or other access restrictions.

Refer to the `/etc/default/security` file in the *security*(4) manual page for detailed information on configurable parameters that affect the behavior of this command. Currently, the supported parameters for the **su** command are:

```
SU_ROOT_GROUP
SU_DEFAULT_PATH
SU_KEEP_ENV_VARS
```

### EXTERNAL INFLUENCES

#### Environment Variables

|                |                                                                                                                                                                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>HOME</b>    | User's home directory                                                                                                                                                                                                                                                                       |
| <b>LANG</b>    | The language in which messages are displayed. If <b>LANG</b> is not specified or is null, it defaults to <b>C</b> (see <i>lang</i> (5)). If any internationalization variable contains an invalid setting, all internationalization variables default to <b>C</b> (see <i>environ</i> (5)). |
| <b>LOGNAME</b> | User's login name                                                                                                                                                                                                                                                                           |
| <b>PATH</b>    | Command name search path                                                                                                                                                                                                                                                                    |
| <b>PS1</b>     | Default prompt                                                                                                                                                                                                                                                                              |
| <b>SHELL</b>   | Name of the user's shell                                                                                                                                                                                                                                                                    |

#### International Code Set Support

Characters in the 7-bit US-ASCII code sets are supported in login names (see *ascii*(5)).

### EXAMPLES

Become user **bin** while retaining the previously exported environment:

```
su bin
```

Become user **bin** but change the environment to what would be expected if **bin** had originally logged in:

```
su - bin
```

Execute the command, `'echo hello'`, using the temporary environment and permissions of user **bin**. In this example, user **bin**'s shell is invoked with the arguments `-c 'echo hello'`.

```
su bin -c 'echo hello'
```

Become user **DCEPrincipal** in the DCE environment:



```
su -d DCEPrincipal
```

**WARNINGS**

After a valid password is supplied, **su** uses information from **/etc/passwd** and **/etc/logingroup** to determine the user's group ID and group access list. If **/etc/group** is linked to **/etc/logingroup**, and group membership for the user trying to log in is managed by the Network Information Service (NIS), and no NIS server is able to respond, **su** waits until a server does respond.

**DEPENDENCIES****Pluggable Authentication Modules (PAM)**

PAM is an Open Group standard for user authentication, password modification, and account validation. In particular, **pam\_authenticate()** is invoked to perform all functions related to **su**. This includes password retrieval, account validation, and error message displays.

**FILES**

|                              |                                         |
|------------------------------|-----------------------------------------|
| <b>\$HOME/.profile</b>       | User's profile                          |
| <b>/etc/logingroup</b>       | System's default group access list file |
| <b>/etc/passwd</b>           | System's password file                  |
| <b>/etc/profile</b>          | System's profile                        |
| <b>/var/adm/sulog</b>        | Log of all attempts                     |
| <b>/etc/default/security</b> | Security defaults configuration file    |

**SEE ALSO**

env(1), login(1), sh(1), initgroups(3C), syslog(3C), group(4), passwd(4), profile(4), security(4), environ(5).

**Pluggable Authentication Modules (PAM)**

pam\_acct\_mgmt(3), pam\_authenticate(3).

**HP-UX Smart Card Login**

scpin(1).

**STANDARDS CONFORMANCE**

**su**: SVID2, SVID3, XPG2

**NAME**

sum - print checksum and block or byte count of file(s)

**SYNOPSIS**

sum [-r] [-p] [file ...]

**Remarks**

**sum** is obsolescent and should not be used in new applications that are intended to be portable between systems. Use **cksum** instead (see *cksum(1)*).

**DESCRIPTION**

**sum** calculates and prints to standard output a checksum for each named file, and also prints the size of the file in 512 byte blocks, rounded up.

The default algorithm is a 16-bit sum of the bytes in which overflow is ignored. Alternate algorithms can be selected with the **-r** and **-p** options.

Standard input is used if no file names are given.

**sum** is typically used to verify data integrity when copying files between systems.

**Options**

**sum** recognizes the following options:

- r** Use an alternate algorithm in which the 16-bit sum is right rotated with each byte in computing the checksum.
- p** Use the 32-bit cyclical redundancy check (CRC) algorithm used by **cksum**.

**RETURN VALUE**

**sum** returns the following values upon completion:

- 0** All files were processed successfully.
- >0** One or more files could not be read or some other error occurred.

If an inaccessible file is encountered, **sum** continues processing any remaining files, but the final exit status is affected.

**DIAGNOSTICS**

Read error conditions are indistinguishable from end of file on most devices; check the block or byte count.

**| S |****WARNINGS**

This command is likely to be withdrawn from X/Open standards. Applications using this command might not be portable to other vendors' platforms. The usage of *cksum(1)* is recommended.

**SEE ALSO**

cksum(1), wc(1).

**STANDARDS CONFORMANCE**

**sum**: SVID2, SVID3, XPG2, XPG3

**NAME**

tabs - set tabs on a terminal

**SYNOPSIS**

**tabs** [*tabspec*] [+**m** *n*] [-**T** *type*]

**DESCRIPTION**

**tabs** sets the tab stops on the user's terminal according to the tab specification *tabspec*, after clearing any previous settings. The user's terminal must have remotely-settable hardware tabs.

If you are using a non-HP terminal, you should keep in mind that behavior will vary for some tab settings.

Four types of tab specification are accepted for *tabspec*: "canned", repetitive, arbitrary, and file. If no **tabspec** is given, the default value is **-8**; i.e., UNIX "standard" tabs. The lowest column number is 1. Note that for *tabs*, column 1 always refers to the left-most column on a terminal, even one whose column markers begin at 0.

**-code** Gives the name of one of a set of "canned" tabs. Recognized *codes* and their meanings are as follows:

- a** 1,10,16,36,72  
Assembler, IBM S/370, first format
- a2** 1,10,16,40,72  
Assembler, IBM S/370, second format
- c** 1,8,12,16,20,55  
COBOL, normal format
- c2** 1,6,10,14,49  
COBOL compact format (columns 1-6 omitted). Using this code, the first typed character corresponds to card column 7, one space gets you to column 8, and a tab reaches column 12. Files using this tab setup should have **tabs** specify a format specification file as defined by **-file** below. The *file* should have the following format specification:  

```
<:t-c2 m6 s66 d:>
```
- c3** 1,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62,67  
COBOL compact format (columns 1-6 omitted), with more tabs than **-c2**. This is the recommended format for COBOL. The appropriate format specification is:  

```
<:t-c3 m6 s66 d:>
```
- f** 1,7,11,15,19,23  
FORTRAN
- p** 1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61  
PL/I
- s** 1,10,55  
SNOBOL
- u** 1,12,20,44  
UNIVAC 1100 Assembler

In addition to these "canned" formats, three other types exist:

- n** A repetitive specification requests tabs at columns 1+n, 1+2×n, etc. Of particular importance is the value **-8**: this represents the UNIX "standard" tab setting, and is the most likely tab setting to be found at a terminal. Another special case is the value **-0**, implying no tabs at all.
- n1 ,n2 ,...** The arbitrary format permits the user to type any chosen set of numbers, separated by commas, in ascending order. Up to 40 numbers are allowed. If any number (except the first one) is preceded by a plus sign, it is taken as an increment to be added to the previous value. Thus, the tab lists 1,10,20,30 and 1,10,+10,+10 are considered identical.
- file** If the name of a file is given, **tabs** reads the first line of the file, searching for a format specification. If it finds one there, it sets the tab stops according to it, otherwise it sets them as **-8**. This type of specification can be used to ensure that a tabbed file is printed with

| t |

correct tab settings, and is suitable for use with the **pr** command (see *pr(1)*):

**tabs -- file; pr file**

Any of the following can be used also; if a given option occurs more than once, the last value given takes effect:

- Ttype** **tabs** usually needs to know the type of terminal in order to set tabs and always needs to know the type to set margins. *type* is a name listed in *term(5)*. If no **-T** option is supplied, **tabs** searches for the **\$TERM** value in the *environment* (see *environ(5)*). If **TERM** is not defined in the environment, **tabs** tries a sequence that will work for many terminals.
- +mn** The margin argument can be used for some terminals. It causes all tabs to be moved over *n* columns by making column *n+1* the left margin. If **+m** is given without a value of *n*, the value assumed is 10. The normal (left-most) margin on most terminals is obtained by **+m0**. The margin for most terminals is reset only when the **+m** option is given explicitly.

Tab and margin setting is performed via the standard output.

**EXTERNAL INFLUENCES**

**Environment Variables**

**LC\_CTYPE** determines the interpretation of text within file as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **tabs** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**DIAGNOSTICS**

**illegal tabs**

Arbitrary tabs are ordered incorrectly.

**illegal increment**

A zero or missing increment found in an arbitrary specification.

**unknown tab code**

A "canned" code cannot be found.

**can't open**

**--file** option was used and file cannot be opened.

**file indirection**

**--file** option was used and the specification in that file points to yet another file. Indirection of this form is not permitted.

| t |

**WARNINGS**

There is no consistency among different terminals regarding ways of clearing tabs and setting the left margin.

It is generally impossible to usefully change the left margin without also setting tabs.

**tabs** clears only 20 tabs (on terminals requiring a long sequence), but is willing to set 64.

**SEE ALSO**

*nroff(1)*, *pr(1)*, *tset(1)*, *environ(5)*, *term(5)*.

**STANDARDS CONFORMANCE**

**tabs**: SVID2, SVID3, XPG2, XPG3, XPG4

**NAME**

tail - deliver the last part of a file

**SYNOPSIS**

**tail** [-f] [-b *number*] [*file*]

**tail** [-f] [-c *number*] [*file*]

**tail** [-f] [-n *number*] [*file*]

**Obsolescent:**

**tail** [ $\pm$ *number*][**l|b|c**] [-f] [*file*]

**DESCRIPTION**

**tail** copies the named *file* to the standard output beginning at a designated place. If no *file* is named, standard input is used.

**Command Forms**

**tail** can be used in three forms as indicated above:

**tail -b** *number*... Copy file starting at *number* blocks from end or beginning of file.

**tail -c** *number*... Copy file starting at *number* bytes from end or beginning of file.

**tail -n** *number*...

**tail** *number*... Copy file starting at *number* lines from end or beginning of file.

**tail** with no options specified is equivalent to **tail -n 10**...

**Options and Command-Line Arguments**

**tail** recognizes the following options and command-line arguments:

- |                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-f</b>               | Follow option. If the input file is a regular file or if <i>file</i> specifies a FIFO, do not terminate after the last line of the input file has been copied, but read and copy further bytes from the input file when they become available ( <b>tail</b> enters an endless loop wherein it sleeps for one second then attempts to read and copy further records from the input file). This is useful when monitoring text being written to a file by another process. If no <i>file</i> argument is specified and the input is a pipe (FIFO), the <b>-f</b> option is ignored. |
| <i>number</i>           | Decimal integer indicating quantity of output to be copied, measured in units specified by accompanying option. If <i>number</i> is preceded by a <b>+</b> character, copy operation starts <i>number</i> units from beginning of file. If <i>number</i> is preceded by a <b>-</b> character or the option name, copy operation starts <i>number</i> units from end of file. If <i>number</i> is not preceded by a <b>b</b> , <b>c</b> , or <b>n</b> option, <b>-n</b> is assumed. If both the option and <i>number</i> are not specified, <b>-n 10</b> is assumed.               |
| <b>-b</b> <i>number</i> | Copy file beginning <i>number</i> 512-byte blocks from end or beginning of file. If <i>number</i> is not specified, <b>-b 10</b> is assumed. See <i>number</i> description above.                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>-c</b> <i>number</i> | Copy file beginning <i>number</i> bytes from end or beginning of file. If <i>number</i> is not specified, <b>-c 10</b> is assumed. See <i>number</i> description above.                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>-n</b> <i>number</i> | Copy file beginning <i>number</i> lines from end or beginning of file. If <i>number</i> is not specified, <b>-n 10</b> is assumed. See <i>number</i> description above.                                                                                                                                                                                                                                                                                                                                                                                                           |
| <i>file</i>             | Name of file to be copied. If not specified, the standard input is used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

If the **-c** option is specified, the input file can contain arbitrary data. Otherwise, the input file should be a text file.

**Obsolescent Form**

In the obsolescent form, option letters can be concatenated after the *number* argument to select blocks, bytes, or lines. If this syntax is used,  $\pm$ *number* must be the first argument given. If *number* is not specified, **-10** is assumed. This version is provided for backward compatibility only. The forms discussed previously are recommended for portability.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files).

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **tail** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported. However, the **b** and **c** options can break multi-byte characters and should be used with caution in a multi-byte locale environment.

**EXAMPLES**

Print the last three lines in file **file1** to the standard output, and leave **tail** in "follow" mode:

```
tail -fn 3 file1
tail -3 -f file1
```

Print the last 15 bytes of file **logfile** followed by any lines that are appended to **logfile** after **tail** is initiated until it is killed:

```
tail -fc15 logfile
tail -f -c 15 logfile
```

Three ways to print an entire file:

```
tail -b +1 file
tail -c +1 file
tail -n +1 file
```

**WARNINGS**

Various kinds of anomalous behavior may occur with character special files. For piped output, **tail** is limited in its output and depends on process limits.

**SEE ALSO**

dd(1), head(1).

**STANDARDS CONFORMANCE**

**tail**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| t |

**NAME**

talk - talk to another user

**SYNOPSIS**

**talk** *talk\_party* [*ttyname* ]

**DESCRIPTION**

The **talk** utility is a two-way, screen-oriented communication program.

The command argument *talk\_party* can take one the following forms:

```
user
user@host
host !user
host :user
host .user
```

where *user* is a login name and *host* is a host name.

The optional command argument, *ttyname*, can be used to specify the terminal to be used when contacting a user who is logged in more than once. In absence of this argument, **talk** will try to contact the user on the user's most recently used terminal.

When first invoked, **talk** sends the following message to the party it tries to connect to (*callee*):

```
Message from Talk_Daemon@callee_host
...
talk: connection requested by caller@caller_host
talk: respond with: talk caller@caller_host
```

At this point, the recipient of the message can reply by typing:

```
talk caller@caller_host
```

Once communication is established, the two parties can type simultaneously, with their output displayed in separate regions of the screen. Characters are processed as follows:

- Typing characters from LC\_CTYPE classifications **print** or **space** will cause those characters to be sent to the recipient's terminal.
- Typing <control>-L will cause the sender's screen to be refreshed.
- Typing the erase, kill or kill word character will delete the last character, line or word on the sender's terminal, with the action propagated to the recipient's terminal.
- Typing the interrupt character will terminate the local *talk* utility. Once the *talk* session has been terminated on one side, the other side of the *talk* session will be notified that the session has been terminated and will be able to do nothing except exit.
- Other non-printable characters typed on the sender's terminal are converted to printable characters before they are sent to the recipient's terminal.

Permission to be a recipient of a **talk** message can be denied or granted by using the **mesg** utility. However, a user may need other privileges to be able to access other users' terminals. The **talk** utility will fail when the user lacks the appropriate privileges.

**SEE ALSO**

mesg(1), who(1), write(1).

**STANDARDS CONFORMANCE**

**talk**: XPG4

| t |

**NAME**

tar - tape file archiver

**SYNOPSIS**

**tar** [-*key* [*arg* ...] [*file* | -**C** *directory*] ...

**DESCRIPTION**

The **tar** command saves and restores archives of files on a magnetic tape, a flexible disk, or a regular file. The default archive file is **/dev/rmt/0m**. See the **-f** option below. Its actions are controlled by the *key* argument.

**Arguments**

- key* is a string of characters containing exactly one function letter and zero or more function modifiers, specified in any order. Whitespace is not permitted in *key*. The *key* string can be preceded by a hyphen (-), as when specifying options in other HP-UX commands, but it is not necessary.
- arg* ... The **b** and **f** function modifiers each require an *arg* argument (see below). If both **b** and **f** are specified, the order of the *arg* arguments must match the order of the modifiers. If specified, the *arg* arguments must be separated from the key and each other by whitespace.
- file* specifies a file being saved or restored. If *file* is a directory name, it refers to the files and (recursively) the subdirectories contained in that directory.
- C** *directory* causes **tar** to perform a **chdir()** to *directory* (see *chdir(2)*). Subsequent *file* and **-C** *directory* arguments are relative to *directory*. This allows multiple directories not related by a close or common parent to be archived using short relative path names.

The value of *file* is stored in the archive. The value of *directory* is not stored.

**Function Keys**

The function portion of the *key* is specified by exactly one of the following letters:

- c** Create a new archive. Write from the beginning of the archive instead of appending after the last file. Any previous information in the archive is overwritten.
- r** Add the named *file* to the end of the archive. The same blocking factor used to create the archive must be used to append to it. This option cannot be used if the archive is a tape.
- t** List the names of all the files in the archive. Adding the **v** function modifier expands this listing to include the file modes and owner numbers. The names of all files are listed each time they occur on the tape.
- u** Add any named *file* to the archive if it is not already present or has been modified since it was last written in the archive. The same blocking factor used to create the archive must be used to update it.
- x** Extract the named *file* from the archive and restore it to the system. If a named *file* matches a directory whose contents were written to the archive, this directory is (recursively) extracted. If a named *file* on tape does not exist on the system, the *file* is created as follows:
  - The user, group, and other protections are restored from the tape.
  - The modification time is restored from the tape unless the **m** function modifier is specified.
  - The file user ID and group ID are normally those of the restoring process.
  - The set-user-ID, set-group-ID, and sticky bits are not set automatically. The **o** and **p** function modifiers control the restoration of protection; see below for more details.

If the files exist, their modes are not changed, but the set-user-id, set-group-id and sticky bits are cleared. If no *file* argument is given, the entire content of the archive is extracted. Note that if several files with the same name are on the archive, the last one overwrites all earlier ones.

**Function Modifier Keys**

The following function modifiers can be used in addition to the function letters listed above (note that some modifiers are incompatible with some functions):



- A** Suppress warning messages that **tar** did not archive a file's access control list. By default, **tar** writes a warning message for each file with optional ACL entries.
- b** Use the next *arg* argument as the blocking factor for archive records. The default is 20; the maximum is at least 20. However, if the **f** - modifier is used to specify standard input, the default blocking factor is 1.

The blocking factor is determined automatically when reading nine-track tapes (key letters **x** and **t**). On nine-track tapes, the physical tape record length is the same as the block size. The block size is defined as the logical record size times the blocking factor (number of logical records per block).

The blocking factor must be specified when reading flexible disks and cartridge tapes if they were written with a blocking factor other than the default.

If a **tar** file is read using a blocking factor not equal to the one used when the file was written, an error may occur at the end of the file but there may or may not be an actual error in the read. To prevent this problem, a blocking factor of **1** can be used, although performance may be reduced somewhat.

**tar** writes logical records of 512 bytes, independent of how logical records may be defined elsewhere by other programs (such as variable-length records (lines) within an ASCII text file).

- e** Fail if the extent attributes are present in the files to be archived. If **tar** fails for this reason, the partially created destination file is not be removed.
- f** Use the next *arg* argument as the name of the archive instead of the default, **/dev/rmt/0m**. If the name of the file is **-**, **tar** writes to standard output or reads from standard input, whichever is appropriate, and the default blocking factor becomes 1. Thus, **tar** can be used as the head or tail of a pipeline (see EXAMPLES).
- h** Force **tar** to follow symbolic links as if they were normal files or directories. Normally, **tar** does not follow symbolic links.
- l** Tell **tar** to complain if it cannot resolve all of the links to the files being saved. If **l** is not specified, no error messages are printed.
- m** Tell **tar** not to restore the modification time written on the archive. The modification time of the file will be the time of extraction.
- N** Write a POSIX format archive. This format allows file names of up to 256 characters in length, and correctly archives and restores the following file types: regular files, character and block special devices, links, symbolic links, directories, and FIFO special files. It also stores the user and group name of each file and attempts to use these names to determine the user-ID and group-ID of a file when restoring it with the **p** function modifier. This is the default format.
- o** Suppress writing certain directory information that older versions of **tar** cannot handle on input. **tar** normally writes information specifying owners and modes of directories in the archive. Earlier versions of **tar**, when encountering this information, give error messages of the form:
 

```
name - cannot create
```

When **o** is used for reading, it causes the extracted *file* to take on the user and group IDs of the user running the program rather than those on the tape. This is the default for the ordinary user and can be overridden, to the extent that system protections allow, by using the **p** function modifier.
- O** Write a pre-POSIX format archive.
- p** Cause *file* to be restored to the original modes and ownerships written on the archive, if possible. This is the default for the superuser, and can be overridden by the **o** function modifier. If system protections prevent the ordinary user from executing **chown**( ), the error is ignored, and the ownership is set to that of the restoring process (see *chown*(2)). The set-user-id, set-group-id, and sticky bit information are restored as allowed by the protections defined by **chmod**( ) if the **chown**( ) operation above succeeds.
- n d* Specify a particular nine-track tape drive and density, where *n* is a tape drive number: **0**–**7**, and *d* is the density: **l** = low (800 bpi); **m** = medium (1600 bpi); **h** = high (6250 bpi). This modifier selects the drive on which the nine-track tape is mounted. The default is **0m**.

| t |

- v** Normally, **tar** does its work silently. The **v** (verbose) function modifier causes **tar** to type the name of each file it treats, preceded by the function letter. With the **t** function, **v** gives more information about the archive entries than just the name.
- V** Same as the **v** function modifier except that, when using the **t** option, **tar** also prints out a letter indicating the type of the archived file.
- w** Cause **tar** to print the action being taken, followed by the name of the file, then wait for the user's confirmation. If the user answers **y**, the action is performed. Any other input means "no".

When end-of-tape is reached, **tar** prompts the user for a new special file and continues.

If a nine-track tape drive is used as the output device, it must be configured in Berkeley-compatibility mode (see *mt(7)*).

The **O** and **N** function modifiers specify the format in which **tar** writes the archive. Upon extraction, **tar** can read either format, regardless of the function modifiers used.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_TIME** determines the format and contents of date and time strings output when listing the contents of an archive with the **-v** option.

**LANG** determines the language equivalent of **y** (for yes/no queries).

If **LC\_TIME** is not specified in the environment or is set to the empty string, the value of **LANG** is used as the default.

If **LANG** is not specified or is set to the empty string, it defaults to "C" (see *lang(5)*).

If any internationalization variable contains an invalid setting, **tar** behaves as if all internationalization variables are set to "C". See *environ(5)*.

### International Code Set Support

Single- and multibyte character code sets are supported.

## DIAGNOSTICS

**tar** issues self-explanatory messages about bad key characters, tape read/write errors, and if not enough memory is available to hold the link tables.

## EXAMPLES

Create a new archive on **/dev/rfd.0** and copy **file1** and **file2** onto it, using the default blocking factor of 20. The *key* is made up of one function letter (**c**) and two function modifiers (**v** and **f**):

```
tar cvf /dev/rfd.0 file1 file2
```

Archive files from **/usr/include** and **/etc**:

```
tar cv -C /usr/include . -C /etc .
```

Use **tar** in a pipeline to copy the entire file system hierarchy under *fromdir* to *todir*:

```
cd fromdir ; tar cf - . | (cd todir ; tar xf -)
```

Archive all files and directories in directory **my\_project** in the current directory to a file called **my\_project.TAR**, also in the current directory:

```
tar -cvf my_project.TAR my_project
```

## WARNINGS

Because of industry standards and interoperability goals, **tar** does not support the archival of files of size 8GB or larger or files that have user/group IDs 2MB or greater. Files with user/group IDs of 2MB or greater are archived and restored under the user/group ID of the current process, unless the *uname/gname* exists, (see *tar(4)*).

The default format has changed from **O** to **N**, beginning with HP-UX Release 8.0.

Due to internal limitations in the header structure, not all file names of fewer than 256 characters fit when using the **N** function modifier. If a file name does not fit, **tar** prints a message and does not archive the file.

Link names are still limited to 100 characters when using the **N** function modifier.

There is no way to ask for the *n*-th occurrence of a file.

Tape errors are handled ungracefully.

The **u** function key can be slow.

If the archive is a file on disk, flexible disk, or cartridge tape, and if the blocking factor specified on output is not the default, the same blocking factor must be specified on input, because the blocking factor is not explicitly stored in the archive. Updating or appending to the archive without following this rule can destroy it.

Some previous versions of **tar** have claimed to support the selective listing of file names using the **t** function key with a list. This appears to be an error in the documentation because the capability does not appear in the original source code.

There is no way to restore an absolute path name to a relative position.

**tar** always pads information written to an archive up to the next multiple of the block size. Therefore, if you are creating a small archive and write out one block of information, **tar** reports that one block was written, but the actual size of the archive might be larger if the **b** function modifier is used.

Note that **tar c0m** is not the same as **tar cm0**.

Do not create archives on block special devices. Attempting to do so can cause excessive wear, leading to premature drive hardware failure.

## DEPENDENCIES

The **r** and **u** function keys are not supported on QIC or 8mm devices. If these options are used with QIC or 8mm devices, **tar** fails and displays the message:

```
tar: option not supported for this device
```

## AUTHOR

**tar** was developed by AT&T, the University of California, Berkeley, HP, and POSIX.

## FILES

```
/dev/rmt/*
/dev/rfd.*
/tmp/tar*
```

## SEE ALSO

ar(1), cpio(1), acl(5), mt(7).

## STANDARDS CONFORMANCE

**tar**: SVID2, SVID3, XPG2, XPG3

| t |

**NAME**

tbl - format tables for nroff

**SYNOPSIS**

**tbl** [-**TX**] [*file ...*]

**DESCRIPTION**

**tbl** is a preprocessor that formats tables for *nroff*(1). The input files are copied to the standard output, except for lines between **.TS** and **.TE** command lines, which are assumed to describe tables and are reformatted by **tbl**. (The **.TS** and **.TE** command lines are not altered by **tbl**).

**.TS** is followed by global options. The available global options are:

**center** center the table (default is left-adjust);  
**expand** make the table as wide as the current line length;  
**box** enclose the table in a box;  
**doublebox**  
 enclose the table in a double box;  
**allbox** enclose each item of the table in a box;  
**tab** (*x*) use the character *x* instead of a tab to separate items in a line of input data.

The global options, if any, are terminated with a semi-colon (;).

Next come lines describing the format of each line of the table. Each such format line describes one line of the actual table, except that the last format line (which must end with a period) describes *all* remaining lines of the table. Each column of each line of the table is described by a single key-letter, optionally followed by specifiers that determine the font and point size of the corresponding item, indicate where vertical bars are to appear between columns, or determine column width, inter-column spacing, etc. The available key-letters are:

**c** center item within the column;  
**r** right-adjust item within the column;  
**l** left-adjust item within the column;  
**n** numerically adjust item in the column: units positions of numbers are aligned vertically;  
**s** span previous item on the left into this column;  
**a** center longest line in this column, then left-adjust all other lines in this column with respect to that centered line;  
<sup>^</sup> span down previous entry in this column;  
 \_ replace this entry with a horizontal line;  
 = replace this entry with a double horizontal line.

The characters **B** and **I** stand for the bold (font position 3) and italic (font position 2) fonts, respectively; the character | indicates a vertical line between columns.

The format lines are followed by lines containing the actual data for the table, followed finally by **.TE**. Within such data lines, data items are normally separated by tab characters.

If a data line consists of only \_ or =, a single or double line, respectively, is drawn across the table at that point; if a *single item* in a data line consists of only \_ or =, then that item is replaced by a single or double line.

The **-TX** option forces **tbl** to use only full vertical line motions, making the output more suitable for devices that cannot generate partial vertical line motions (such as line printers).

If no file names are given as arguments (or if - is specified as the last argument), **tbl** reads the standard input, and thus can be used as a filter. When used with **neqn**, **tbl** should be used first to minimize the volume of data passed through pipes (see *neqn*(1)).

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the interpretation of text as single- and/or multi-byte characters.

**LC\_NUMERIC** determines the radix character used in numerical data.

**LANG** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_NUMERIC** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG**

is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **tbl** behaves as if all internationalization variables are set to "C". See *environ(5)*.

### International Code Set Support

Single- and multi-byte character code sets are supported.

### EXAMPLES

If we redefine the tab character to a semicolon, then the input:

```
.TS
center box tab(;) ;
CB s s
cI | cI s
^ | c c
l | n n.
Household Population
-
Town;Households
;Number;Size
=
Bedminster;789;3.26
Bernards Twp.;3087;3.74
Bernardsville;2018;3.30
Bound Brook;3425;3.04
Bridgewater;7897;3.81
Far Hills;240;3.19
.TE
```

yields:

| Household Population |            |      |
|----------------------|------------|------|
| Town                 | Households |      |
|                      | Number     | Size |
| Bedminster           | 789        | 3.26 |
| Bernards Twp.        | 3087       | 3.74 |
| Bernardsville        | 2018       | 3.30 |
| Bound Brook          | 3425       | 3.04 |
| Bridgewater          | 7897       | 3.81 |
| Far Hills            | 240        | 3.19 |

The **tbl** command is used most often with **nroff** and **col** (see *col(1)*). A common usage is:

```
tbl filename | nroff -macro_package_name | col
```

| t |

### WARNINGS

See WARNINGS under *nroff(1)*.

### SEE ALSO

*col(1)*, *mm(1)*, *neqn(1)*, *nroff(1)*, *soelim(1)*, *mm(5)*.

**NAME**

tcpdchk - check tcp wrapper configuration

**SYNOPSIS**

`/usr/bin/tcpdchk [-a] [-d] [-i inet_conf] [-v]`

**DESCRIPTION**

**tcpdchk** examines the tcp wrapper configuration and reports all potential and real problems it can encounter. The command examines the **tcpd** access control files (by default, these are `/etc/hosts.allow` and `/etc/hosts.deny`), and compares the entries in these files against entries in the **inetd** configuration file.

**tcpdchk** reports the following types of problems:

- non-existent pathnames,
- services that appear in **tcpd** access control rules but are not controlled by **tcpd**,
- services that should not be wrapped,
- non-existent host names or non-internet address forms,
- occurrences of host aliases instead of official host names,
- hosts with a name/address conflict,
- inappropriate use of wildcard patterns,
- inappropriate use of NIS netgroups or references to non-existent NIS netgroups,
- references to non-existent options,
- invalid arguments to options.

Wherever possible, **tcpdchk** provides a helpful suggestion to fix the problem.

**Options**

The following options are supported by **tcpdchk**. If no options are specified, then it uses the default location of the files.

- a** Report access control rules that permit access without an explicit **ALLOW** keyword.
- d** Examine the **hosts.allow** and **hosts.deny** files in the current directory instead of the default ones.
- i** *inet\_conf*  
Specify this option when **tcpdchk** is unable to find your **inetd.conf** configuration file, or when you suspect that **tcpdchk** is using the wrong file. *inet\_conf* is the path name of the **inetd.conf** configuration file whose entries you want to examine.
- v** Display the contents of each access control rule. Daemon lists, client lists, shell commands and options are shown in a printable format. The display helps you find any discrepancies between what you want and what **tcpdchk** understands for the access control rules.

**AUTHOR**

Wietse Venema (wietse@wzv.win.tue.nl),  
Department of Mathematics and Computing Science,  
Eindhoven University of Technology  
Den Dolech 2, P.O. Box 513,  
5600 MB Eindhoven, The Netherlands

**FILES**

The default locations of the **tcpd** access control tables are:

- `/etc/hosts.allow` (daemon, client) pairs that are granted access.
- `/etc/hosts.deny` (daemon, client) pairs that are denied access.

**SEE ALSO**

*tcpdmatch*(1), explains what **tcpd** would do in specific cases.

*inetd.conf*(4), format of the **inetd** control file.

*hosts\_access*(5), format of the **tcpd** access control tables.

*hosts\_options*(5), format of the language extensions.

**NAME**

tcpdmatch - evaluate tcp wrapper service requests

**SYNOPSIS**

```
/usr/bin/tcpdmatch [-d] [-i inet_conf] daemon client
```

```
/usr/bin/tcpdmatch [-d] [-i inet_conf] daemon [@server] [user@]client
```

**DESCRIPTION**

**tcpdmatch** predicts how the tcp wrapper would handle a specific request for service. Examples are given below.

The program examines the **tcpd** access control tables (default **/etc/hosts.allow** and **/etc/hosts.deny**) and prints its conclusion. For maximum accuracy, it extracts additional information from the **inetd** configuration file.

When **tcpdmatch** finds a match in the access control tables, it identifies the matched rule. In addition, it displays the optional shell commands or options in a printable format. The display helps you find any discrepancies between what you want and what **tcpdmatch** understands for the access control rules.

**Arguments**

The *daemon* and *client* arguments are always required.

*daemon* A daemon process name. Typically, the last component of a daemon executable pathname.

*client* A host name or network address, or one of the 'unknown' or 'paranoid' wildcard patterns.

When a client host name is specified, **tcpdmatch** gives a prediction for each address listed for that client.

When a client address is specified, **tcpdmatch** predicts what **tcpd** would do when the client name lookup fails.

Optional information specified with the *daemon@server* form:

*server* A host name or network address, or one of the 'unknown' or 'paranoid' wildcard patterns. The default server name is 'unknown'.

Optional information specified with the *user@client* form:

*user* A client user identifier. Typically, a login name or a numeric userid. The default user name is 'unknown'.

**Options**

**-d** Examine **hosts.allow** and **hosts.deny** files in the current directory instead of the default ones.

**-i** *inet\_conf*

Specify this option when **tcpdmatch** is unable to find your **inetd.conf** configuration file, or when you suspect that **tcpdmatch** is using the wrong file. *inet\_conf* is the path name of the **inetd.conf** configuration file whose entries you want to examine.

**EXAMPLES**

To predict how **tcpd** would handle a telnet request from the local system:

```
tcpdmatch telnetd localhost
```

The same request, pretending that hostname lookup failed:

```
tcpdmatch telnetd 127.0.0.1
```

To predict what **tcpd** would do when the client name does not match the client address:

```
tcpdmatch telnetd paranoid
```

**AUTHOR**

Wietse Venema (wietse@wzv.win.tue.nl),  
Department of Mathematics and Computing Science,  
Eindhoven University of Technology  
Den Dolech 2, P.O. Box 513,  
5600 MB Eindhoven, The Netherlands



**FILES**

The default locations of the **tcpd** access control tables are:

**/etc/hosts.allow** (daemon, client) pairs that are granted access.

**/etc/hosts.deny** (daemon, client) pairs that are denied access.

**SEE ALSO**

*tcpdchk*(1), **tcpd** configuration checker.

*inetd.conf*(4), format of the inetd control file.

*hosts\_access*(5), format of the **tcpd** access control tables.

*hosts\_options*(5), format of the language extensions.

**NAME**

tee - pipe fitting

**SYNOPSIS**

tee [-i] [-a] [*file*] ...

**DESCRIPTION**

The **tee** command transcribes the standard input to the standard output and makes copies in the *files*.

**Options**

- i This option ignores interrupts.
- a This option appends the output to the *files* rather than overwriting the *files*.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **tee** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**RETURN VALUE**

The **tee** command returns zero upon successful completion, or nonzero if the command fails.

**EXAMPLES**

Write a list of users to the screen and also append the list to the file **hunt**:

```
who | tee -a hunt
```

**STANDARDS CONFORMANCE**

**tee**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| t |

**NAME**

telnet - user interface to the TELNET protocol

**SYNOPSIS**

```
telnet [[options] host [port]]
```

**DESCRIPTION**

**telnet** is used to communicate with another host using the TELNET protocol. If **telnet** is invoked without arguments, it enters command mode, indicated by its prompt (**telnet>**). In this mode, it accepts and executes the commands listed below. If **telnet** is invoked with arguments, it performs an **open** command (see below) with those arguments.

Once a connection has been opened, **telnet** enters an input mode. The input mode will be either "character at a time" or "line by line", depending on what the remote system supports.

In "character at a time" mode, most text typed is immediately sent to the remote host for processing.

In "line by line" mode, all text is echoed locally, and (normally) only completed lines are sent to the remote host. The "local echo character" (initially **^E**) can be used to turn-off and turn-on the local echo (this would mostly be used to enter passwords without the password being echoed).

In either mode, if the **localchars** toggle is TRUE (the default in line mode; see below), the user's **quit** and **intr** characters are trapped locally, and sent as TELNET protocol sequences to the remote side. There are options (see **toggle autoflush** and **toggle autosynch** below) that cause this action to flush subsequent output to the terminal (until the remote host acknowledges the TELNET sequence) and flush previous terminal input (in the case of **quit** and **intr**).

While connected to a remote host, **telnet** command mode can be entered by typing the **telnet** "escape character" (initially **^]**). When in command mode, the normal terminal editing conventions are available.

**telnet** supports 8-bit characters when communicating with the server on the remote host. To use eight-bit characters you may need to reconfigure your terminal or the remote host appropriately (see *stty(1)*). Furthermore, you may have to use the **binary** toggle to enable an 8-bit data stream between **telnet** and the remote host.

**Note:** Some remote hosts may not provide the necessary support for eight-bit characters.

If, at any time, **telnet** is unable to read from or write to the server over the connection, the message **Connection closed by foreign host.** is printed on standard error. **telnet** then exits with a value of 1.

**telnet** supports the TAC User ID (also known as the TAC Access Control System, or TACACS User ID) option. Enabling the option on a host server allows the user to **telnet** into that host without being prompted for a second login sequence. The TAC User ID option uses the same security mechanism as **rlogin** for authorizing access by remote hosts and users. The system administrator must enable the (telnetd) option only on systems, which are designated as participating hosts. The system administrator must also assign to each user of TAC User ID the very same UID on every system for which he is allowed to use the feature. See the *telnetd(1M)* manpage and the *Managing Systems and Workgroups* manual.

**Options**

The following **telnet** options are available:

- 8** Enable cs8 (8 bit transfer) on local tty.
- e c** Set the **telnet** command mode escape character to be **^c** instead of its default value of **^]**.
- 1** Disable the TAC User ID option if enabled on the client, to cause the user to be prompted for login username and password. Omitting the **-1** option executes the default setting.

**Kerberos-Specific Options**

By default, or by use of the **-a** or the **-1** option, the Kerberos version of **telnet** behaves as a client which supports authentication based on Kerberos V5. As a Kerberos client, **telnet** will authenticate and authorize the user to access the remote system. See the *sis(5)* manpage for details on Kerberos authentication and authorization. However, it will not support integrity-checked or encrypted sessions.

The default Kerberos options for the applications are set in the **krb5.conf** configuration file. Refer to the *appdefaults* Section in the *krb5.conf(4)* manpage for more information. The options **-a**, **-f**, and **-F** described in the subsequent paragraphs, can be set in the **krb5.conf** file with the tag names

**autologin**, **forward**, and **forwardable** respectively. Refer to the *appdefaults* Section of the *krb5.conf(4)* manpage for more information.

The **fallback** option can be set in the **krb5.conf** file within *appdefaults* Section. If **fallback** is set to true and the kerberos authentication fails, **telnet** will use the non-secure mode of authentication.

**Note:** Command line options override configuration file options.

The following options are Kerberos-specific:

**-a** This option is applicable only in a secure environment based on Kerberos V5. Attempt automatic login into the Kerberos realm and disable the TAC User ID option.

**Note:** This is the default login mode.

Sends the user name via the NAME subnegotiation of the Authentication option. The name used is that of the current user as returned by the USER environment variable. If this variable is not defined, the name used is that returned by **getpwnam()** (see *getpwent(3C)*) if it agrees with the current user ID. Otherwise, it is the name associated with the user ID.

**-P** This option is applicable only in a secure environment based on Kerberos V5. Disable use of Kerberos authentication and authorization. When this option is specified, a password is required that is sent across the network in a readable form. See the *sis(5)* manpage.

**-f** This option is applicable only in a secure environment based on Kerberos V5. Allows local credentials to be forwarded to the remote system. Only one **-f** or **-F** option is allowed. They cannot be used together.

**-F** This option is applicable only in a secure environment based on Kerberos V5. Allows local credentials to be forwarded to the remote system including any credentials that have already been forwarded into the local environment. Only one **-f** or **-F** option is allowed. They cannot be used together.

**-l user** This option is applicable only in a secure environment based on Kerberos V5. Attempt automatic login into the Kerberos realm as the specified user and disable the TAC User ID option. The user name specified is sent via the NAME subnegotiation of the Authentication option. Omitting the **-l** option executes the default setting. Only one **-l** option is allowed.

## Commands

The following commands are available in command mode. You need to type only sufficient prefix of each command to uniquely identify it (this is also true for arguments to the **mode**, **set**, **toggle**, and **display** commands).

**open host [port]**

Open a connection to the named host at the indicated port. If no port is specified, **telnet** attempts to contact a TELNET server at the standard TELNET port. In the IPv4 environment, hostname can be either the official name or an alias as understood by **gethostbyname()** (see *gethostent(3N)*) or an Internet address specified in the dot notation as described in *hosts(4)*. In the IPv6 environment, hostname can be either the official name or an alias as understood by **getnameinfo()** (see *getnameinfo(3N)*) or an Internet address specified in the colon notation as described in *hosts(4)*. If no hostname is given, **telnet** prompts for one.

**close** Close a TELNET session. If the session was started from command mode, **telnet** returns to command mode; otherwise **telnet** exits.

**quit** Close any open TELNET session and exit **telnet**. An end of file (in command mode) will also close a session and exit.

**z** Suspend **telnet**. If **telnet** is run from a shell that supports job control, (such as *csh(1)* or *ksh(1)*), the **z** command suspends the TELNET session and returns the user to the shell that invoked **telnet**. Then the job can be resumed by using the **fg** command (see *csh(1)* or *ksh(1)*).

**mode mode** Change **telnet**'s user input mode to *mode*, which can be **character** (for "character at a time" mode) or **line** (for "line by line" mode). The remote host is asked for permission to go into the requested mode. If the remote host is capable of entering that mode, the requested mode is entered. In **character** mode, **telnet** sends each character to the remote host as it is typed. In **line** mode, **telnet** gathers user input into lines and transmits each line to the remote host when the user types carriage return, linefeed, or

EOF (normally  $\text{^D}$ ; see *stty*(1)).

**Note:** Setting line-mode also sets local echo. Applications that expect to interpret user input character by character (such as **more**, **csch**, **ksh**, and **vi**) do not work properly in line mode.

**status** Show current status of **telnet**. **telnet** reports the current escape character. If **telnet** is connected, it reports the host to which it is connected and the current **mode**. If **telnet** is not connected to a remote host, it reports **No connection**. Once **telnet** has been connected, it reports the local flow control toggle value.

**display** [*argument ...*]  
Displays all or some of the **set** and **toggle** values (see below).

? [*command*] Get help. With no arguments, **telnet** prints a help summary. If a command is specified, **telnet** prints the help information available about that command only. Help information is limited to a one-line description of the command.

! [*shell\_command*]  
Shell escape. The **SHELL** environment variable is checked for the name of a shell to use to execute the command. If no *shell\_command* is specified, a shell is started and connected to the user's terminal. If **SHELL** is undefined, **/usr/bin/sh** is used.

**send arguments**  
Sends one or more special character sequences to the remote host. Each *argument* can have any of the following values (multiple *arguments* can be specified with each **send** command):

**escape** Sends the current **telnet** escape character (initially  $\text{^]}$ ).

**synch** Sends the TELNET SYNCH sequence. This sequence causes the remote system to discard all previously typed (but not yet read) input. This sequence is sent as TCP urgent data (and may not work to some systems -- if it doesn't work, a lower case "r" may be echoed on the terminal).

**brk** Sends the TELNET BRK (Break) sequence, which may have significance to the remote system.

**ip** Sends the TELNET IP (Interrupt Process) sequence, which should cause the remote system to abort the currently running process.

**ao** Sends the TELNET AO (Abort Output) sequence, which should cause the remote system to flush all output *from* the remote system *to* the user's terminal.

**ayt** Sends the TELNET AYT (Are You There) sequence, to which the remote system may or may not choose to respond.

**ec** Sends the TELNET EC (Erase Character) sequence, which should cause the remote system to erase the last character entered.

**el** Sends the TELNET EL (Erase Line) sequence, which should cause the remote system to erase the line currently being entered.

**ga** Sends the TELNET GA (Go Ahead) sequence, which is likely to have no significance to the remote system.

**nop** Sends the TELNET NOP (No Operation) sequence.

? Prints out help information for the **send** command.

**set** *variable\_name value*

Set any of the **telnet** variables to a specific value. The special value **off** turns off the function associated with the variable. The values of variables can be shown by using the **display** command. The following *variable\_names* can be specified:

**echo** This is the value (initially  $\text{^E}$ ) which, toggles between doing local echoing of entered characters (for normal processing), and suppressing echoing of entered characters (for entering, for example, a password) in line-by-line mode.

| t |

**escape** This is the **telnet** escape character (initially `^]`) that causes entry into **telnet** command mode (when connected to a remote system).

**interrupt**

If **telnet** is in **localchars** mode (see **toggle localchars** below) and the *interrupt* character is typed, a TELNET IP sequence (see **send ip** above) is sent to the remote host. The initial value for the interrupt character is taken to be the terminal's **intr** character.

**quit** If **telnet** is in **localchars** mode (see **toggle localchars** below) and the **quit** character is typed, a TELNET BRK sequence (see **send brk** above) is sent to the remote host. The initial value for the quit character is taken to be the terminal's **quit** character.

**flushoutput**

If **telnet** is in **localchars** mode (see **toggle localchars** below) and the **flushoutput** character is typed, a TELNET AO sequence (see **send ao** above) is sent to the remote host. The initial value for the flush character is `^O`.

**erase** If **telnet** is in **localchars** mode (see **toggle localchars** below), and if **telnet** is operating in character-at-a-time mode, then when this character is typed, a TELNET EC sequence (see **send ec** above) is sent to the remote system. The initial value for the erase character is taken to be the terminal's **erase** character.

**kill** If **telnet** is in **localchars** mode (see **toggle localchars** below), and if **telnet** is operating in character-at-a-time mode, then when this character is typed, a TELNET EL sequence (see **send el** above) is sent to the remote system. The initial value for the kill character is taken to be the terminal's **kill** character.

**eof** If **telnet** is operating in line-by-line mode, entering this character as the first character on a line causes this character to be sent to the remote system. The initial value of the **eof** character is taken to be the terminal's **eof** character.

**toggle arguments ...**

Toggle (between TRUE and FALSE ) various flags that control how **telnet** responds to events. More than one argument can be specified. The state of these flags can be shown by using the **display** command. Valid arguments are:

**localchars**

If TRUE, the **flush**, **interrupt**, **quit**, **erase**, and **kill** characters (see **set** above) are recognized locally, and transformed into appropriate TELNET control sequences (respectively **ao**, **ip**, **brk**, **ec**, and **el**; see **send** above). The initial value for this toggle is **TRUE** in line-by-line mode, and **FALSE** in character-at-a-time mode.

**autoflush**

If **autoflush** and **localchars** are both TRUE, whenever the **ao**, **intr**, or **quit** characters are recognized (and transformed into TELNET sequences – see **set** above for details), **telnet** refuses to display any data on the user's terminal until the remote system acknowledges (via a TELNET *Timing Mark* option) that it has processed those TELNET sequences. The initial value for this toggle is TRUE.

**autologin**

This flag is available only in a secure environment based on Kerberos V5. Enable or disable automatic login into the Kerberos realm. Using this option yields the same results as using the **-a** option. The initial value for this toggle is TRUE.

**autosynch**

If **autosynch** and **localchars** are both TRUE, when either the **intr** or **quit** character is typed (see **set** above for descriptions of the **intr** and **quit** characters), the resulting TELNET sequence sent is followed by the TELNET SYNCH sequence. This procedure causes the remote system to begin

| t |

- discarding all previously typed input until both of the TELNET sequences have been read and acted upon. The initial value of this toggle is FALSE.
- binary** Enable or disable the TELNET BINARY option on both input and output. This option should be enabled in order to send and receive 8-bit characters to and from the TELNET server.
- crlf** If TRUE, end-of-line sequences are sent as an ASCII carriage-return and line-feed pair. If FALSE, end-of-line sequences are sent as an ASCII carriage-return and NUL character pair. The initial value for this toggle is FALSE.
- crmod** Toggle carriage return mode. When this mode is enabled, any carriage return characters received from the remote host are mapped into a carriage return and a line feed. This mode does not affect those characters typed by the user; only those received. This mode is only required for some hosts that require the client to do local echoing, but output “naked” carriage returns. The initial value for this toggle is FALSE.
- echo** Toggle local echo mode or remote echo mode. In local echo mode, user input is echoed to the terminal by the local **telnet** before being transmitted to the remote host. In remote echo, any echoing of user input is done by the remote host. Applications that handle echoing of user input themselves, such as C shell, Korn shell, and **vi** (see *cs(1)*, *ksh(1)*, and *vi(1)*), do not work correctly with local echo.
- options** Toggle viewing of TELNET options processing. When options viewing is enabled, all TELNET option negotiations are displayed. Options sent by **telnet** are displayed as ``SENT``, while options received from the TELNET server are displayed as ``RCVD``. The initial value for this toggle is FALSE.
- netdata** Toggles the display of all network data (in hexadecimal format). The initial value for this toggle is FALSE.
- ?** Displays the legal **toggle** commands.

## RETURN VALUE

In the event of an error, or if the TELNET connection is closed by the remote host, **telnet** returns a value of 1. Otherwise, it returns 0.

## DIAGNOSTICS

The following diagnostic messages are displayed by **telnet**:

### Error! Could not retrieve authentication type.

There are two authentication mechanisms used by TELNET. One authentication mechanism is based on Kerberos and the other is not. The type of authentication mechanism is obtained from a system file, which is updated by **inetsvcs\_sec**. If the system file on either the local host or the remote host does not contain known authentication types, the above error is displayed.

### telnet/tcp: Unknown service

**telnet** was unable to find the TELNET service entry in the *services(4)* database.

### hostname: Unknown host

**telnet** was unable to map the host name to an Internet address. Your next step should be to contact the system administrator to check whether there is an entry for the remote host in the **hosts** database (see *hosts(4)*).

### ?Invalid command

An invalid command was typed in **telnet** command mode.

### system call>: ...

An error occurred in the specified system call. See the appropriate manual entry for a description of the error.

## AUTHOR

**telnet** was developed by the University of California, Berkeley.

**SEE ALSO**

cs(1), ksh(1), login(1), rlogin(1), stty(1), telnetd(1M), inetd(1M), hosts(4), krb5.conf(4), services(4), sis(5), termio(7).

**| t |**



**NAME**

test - condition evaluation command

**SYNOPSIS**

```
test expr
[expr]
```

**DESCRIPTION**

The **test** command evaluates the expression *expr* and, if its value is True, returns a zero (true) exit status; otherwise, a nonzero (false) exit status is returned. **test** also returns a nonzero exit status if there are no arguments. The following primitives are used to construct *expr*:

|                                     |                                                                                                                                                                                                      |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r</b> <i>file</i>               | True if <i>file</i> exists and is readable.                                                                                                                                                          |
| <b>-w</b> <i>file</i>               | True if <i>file</i> exists and is writable.                                                                                                                                                          |
| <b>-x</b> <i>file</i>               | True if <i>file</i> exists and is executable.                                                                                                                                                        |
| <b>-f</b> <i>file</i>               | True if <i>file</i> exists and is a regular file.                                                                                                                                                    |
| <b>-d</b> <i>file</i>               | True if <i>file</i> exists and is a directory.                                                                                                                                                       |
| <b>-c</b> <i>file</i>               | True if <i>file</i> exists and is a character special file.                                                                                                                                          |
| <b>-b</b> <i>file</i>               | True if <i>file</i> exists and is a block special file.                                                                                                                                              |
| <b>-p</b> <i>file</i>               | True if <i>file</i> exists and is a named pipe (fifo).                                                                                                                                               |
| <b>-u</b> <i>file</i>               | True if <i>file</i> exists and its set-user-ID bit is set.                                                                                                                                           |
| <b>-g</b> <i>file</i>               | True if <i>file</i> exists and its set-group-ID bit is set.                                                                                                                                          |
| <b>-k</b> <i>file</i>               | True if <i>file</i> exists and its sticky bit is set.                                                                                                                                                |
| <b>-s</b> <i>file</i>               | True if <i>file</i> exists and has a size greater than zero.                                                                                                                                         |
| <b>-h</b> <i>file</i>               | True if <i>file</i> exists and is a symbolic link.                                                                                                                                                   |
| <b>-t</b> [ <i>fil</i> <i>des</i> ] | True if the open file whose file descriptor number is <i>fil</i> <i>des</i> (1 by default) is associated with a terminal device.                                                                     |
| <b>-z</b> <i>s1</i>                 | True if the length of string <i>s1</i> is zero.                                                                                                                                                      |
| <b>-n</b> <i>s1</i>                 | True if the length of the string <i>s1</i> is non-zero.                                                                                                                                              |
| <i>s1</i> = <i>s2</i>               | True if strings <i>s1</i> and <i>s2</i> are identical.                                                                                                                                               |
| <i>s1</i> != <i>s2</i>              | True if strings <i>s1</i> and <i>s2</i> are <i>not</i> identical.                                                                                                                                    |
| <i>s1</i>                           | True if <i>s1</i> is <i>not</i> the null string.                                                                                                                                                     |
| <i>n1</i> <b>-eq</b> <i>n2</i>      | True if the integers <i>n1</i> and <i>n2</i> are algebraically equal. Any of the comparisons <b>-ne</b> , <b>-gt</b> , <b>-ge</b> , <b>-lt</b> , and <b>-le</b> can be used in place of <b>-eq</b> . |

These primaries can be combined with the following operators:

|                 |                                                                        |
|-----------------|------------------------------------------------------------------------|
| <b>!</b>        | Unary negation operator.                                               |
| <b>-a</b>       | Binary AND operator.                                                   |
| <b>-o</b>       | Binary OR operator ( <b>-a</b> has higher precedence than <b>-o</b> ). |
| ( <i>expr</i> ) | Parentheses for grouping.                                              |

Note that all the operators and flags are separate arguments to **test**. Note also that parentheses are significant to the shell and therefore must be escaped. All file test operators return success if the argument is a symbolic link that points to a file of the file type being tested.

**test** is interpreted directly by the shell, and therefore does not exist as a separate executable program.

**EXTERNAL INFLUENCES****International Code Set Support**

Single byte and multibyte character code sets are supported.

**EXAMPLES**

Exit if there are not two or three arguments:

```
if [$# -lt 2 -o $# -gt 3]; then exit 1; fi
```

Create a new file containing the text string **default** if the file does not already exist:

```
[! -f thisfile] && echo default > thisfile
```

Wait for myfile to become non-readable:

```
while test -r myfile
do
 sleep 30
done
echo '"myfile" is no longer readable'
```

**WARNINGS**

When the [ form of this command is used, the matching ] must be the final argument, and both must be separate arguments from the arguments they enclose (white space delimiters required).

Parentheses and other special shell metacharacters intended to be handled by test must be escaped or quoted when invoking **test** from a shell.

Avoid such problems when comparing strings by inserting a non-operator character at the beginning of both operands:

```
test "X$response" = "Xexpected string"
```

This approach does not work with numeric comparisons or the unary operators because it would affect the operand being checked.

**AUTHOR**

**test** was developed by the University of California, Berkeley and HP.

**SEE ALSO**

find(1), sh-posix(1), sh(1).

**STANDARDS CONFORMANCE**

**test**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

[ : SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| t |

**NAME**

tftp - trivial file transfer program

**SYNOPSIS**

**tftp** [-**B** *val*] [-**s**] [-**t** *val*] [*host* [*port*]]

**DESCRIPTION**

**tftp** is the user interface to the Internet TFTP (Trivial File Transfer Protocol), that allows users to transfer files to and from a remote machine. The remote *host* can be specified on the command line, in which case **tftp** uses *host* as the default host for future transfers (see the **connect** command below).

**Note**

**tftp** now conforms to the RFCs: 2347, 2348, and 2349.

**Options**

**tftp** supports the following new options:

- B** *val*      Set the block size option for data transfer, in octets. See **blksize** command for more information.
- s**            Set the use of transfer size option. See **tsize** command for more information.
- t** *val*      Set the retransmission timeout option, in seconds. See **newrexmt** command for more information.

**Commands**

Once **tftp** is running, it issues the prompt **tftp>** and recognizes the following commands:

**connect** *host* [*port*]

Set the *host* (and optionally *port*) for transfers. Note that the TFTP protocol, unlike the FTP protocol, does not maintain connections between transfers; thus, the **connect** command does not actually create a connection, but merely remembers what host is to be used for transfers. You do not have to use the **connect** command; the remote host can be specified as part of the **get** or **put** commands.

**mode** *transfer-mode*

Set the mode for transfers; *transfer-mode* can be one of **ascii** or **binary** (default is **ascii**).

**put** *file*

**put** *localfile remotefile*

**put** *file1 file2 ... fileN remotedirectory*

Put a file or set of files to the specified remote file or directory. The destination can be in one of two forms: a filename on the remote host if the host has already been specified, or a string of the form *host:filename* to specify both a host and filename at the same time. If the latter form is used, the hostname specified becomes the default for future transfers. If the remote-directory form is used, the remote host is assumed to be a UNIX-like machine.

**get** *filename*

**get** *remotename localname*

**get** *file1 file2 ... fileN*

Get a file or set of files from the specified *source*s. *source* can be in one of two forms: a filename on the remote host if the host has already been specified, or a string of the form *host:filename* to specify both a host and filename at the same time. If the latter form is used, the last hostname specified becomes the default for future transfers.

**quit**            Exit **tftp**. Typing the end-of-file character also causes an exit.

**verbose**        Toggle verbose mode.

**trace**          Toggle packet tracing.

**status**         Show current status.

**rexmt** *retransmission-timeout*

Set the per-packet retransmission timeout, in seconds.

**timeout** *total-transmission-timeout*

Set the total transmission timeout, in seconds.

- ascii** Shorthand for "mode ascii"
- binary** Shorthand for "mode binary"
- blksize** *val*  
Set the block size for data transfer. The client and the server communicate to arrive upon a block size that is suitable for use on the network medium. The valid range is 8 to 65464 octets. The default value is 512 octets.
- newrext** *val*  
Set the retransmission timeout, in seconds. The client and the server communicate to arrive upon a retransmission timeout value. The valid range is 1 to 255 seconds. The default value is 5 seconds.
- tsize** Toggles the use of transfer size. This option is implemented for binary mode transfers only. By default, this option is disabled. If this option is enabled, it allows the receiving side to determine the size of the file being transferred. When the **get** command is used in binary mode and the size of the file is greater than the free disk space, transfer will be aborted immediately. When the **put** command is used in binary mode, the size of the file will be sent to the server.
- ? [*command-name ...*]  
Print help information.

**WARNINGS**

Since there is no user-login or validation within the TFTP protocol, the remote site probably has some sort of file-access restrictions in place. The exact methods are specific to each site and are therefore difficult to document here.

**AUTHOR**

**tftp** was developed by the University of California, Berkeley.

**SEE ALSO**

tftpd(1M).

| t |

**NAME**

time - time a command

**SYNOPSIS**

**time** *command*

**XPG4 only**

**time** [-p] *utility* [*argument* ...]

**DESCRIPTION**

*command* is executed. Upon completion, **time** prints the elapsed time during the command, the time spent in the system, and the time spent executing the command. Times are reported in seconds.

Execution time can depend on the performance of the memory in which the program is running.

The times are printed on standard error.

Note that the shell also has a keyword **time** that times an entire pipeline if used anywhere in the pipeline, unlike *time(1)* command which times a particular command if used in a pipeline.

**Options**

**time** recognizes the following option:

**-p** (XPG4 only.) Writes the timing statistics to standard error.

**SEE ALSO**

sh(1), timex(1), times(2).

**STANDARDS CONFORMANCE**

**time**: SVID2, XPG2, XPG3, XPG4, POSIX.2

**| t |**

**NAME**

timex - time a command; report process data and system activity

**SYNOPSIS**

**timex** [-o] [-p[**fhkmrt**]] [-s] *command*

**DESCRIPTION**

**timex** reports in seconds the elapsed time, user time, and system time spent in execution of the given *command*. Optionally, process accounting data for *command* and all its children can be listed or summarized, and total system activity during the execution interval can be reported.

The output of **timex** is written on the standard error. Timex returns an exit status of 1 if it is used incorrectly, if it is unable to fork, or if it could not exec *command*. Otherwise, timex returns the exit status of *command*.

**Options**

- o** Report the total number of blocks read or written and total characters transferred by *command* and all its children. This option works only if the process accounting software is installed.
- p[**fhkmrt**]** List process accounting records for *command* and all its children. The suboptions **f**, **h**, **k**, **m**, **r**, and **t** modify the data items reported. They behave as defined in *acctcom*(1M). The number of blocks read or written and the number of characters transferred are always reported.  
  
This option works only if the process accounting software is installed and */usr/lib/acct/turnacct* has been invoked to create */var/adm/pacct*.
- s** Report total system activity (not just that due to *command*) that occurred during the execution interval of *command*. All the data items listed in *sar*(1M) are reported.

**EXAMPLES**

A simple example:

```
timex -ops sleep 60
```

A terminal session of arbitrary complexity can be measured by timing a sub-shell:

```
timex -opskmt sh
 session commands
EOT
```

**WARNINGS**

Process records associated with *command* are selected from the accounting file */var/adm/pacct* by inference, since process genealogy is not available. Background processes having the same user-ID, terminal-ID, and execution time window are spuriously included.

**SEE ALSO**

*sar*(1M), *acctcom*(1M).

**STANDARDS CONFORMANCE**

**timex**: SVID2, SVID3

| t |

**NAME**

top - display and update information about the top processes on the system

**SYNOPSIS**

**top** [-s *time*] [-d *count*] [-q] [-u] [-w] [-h] [-P] [-n *number*] [-f *filename*] [-p *pset\_id*]

**DESCRIPTION**

**top** displays the top processes on the system and periodically updates the information. Raw CPU percentage is used to rank the processes. On a PA-RISC, hard-partitionable platform, **top** issues a warning that it reports only for the current partition and it has to be run on other partitions if their status is required.

**Options**

**top** recognizes the following command-line options:

- s** *time* Set the delay between screen updates to *time* seconds. The default delay between updates is 5 seconds.
- d** *count* Show only *count* displays, then exit. A display is considered to be one update of the screen. This option is used to select the number of displays to be shown before the program exits.
- q** This option runs the **top** program at the same priority as if it is executed via a **nice -20** command so that it will execute faster (see *nice(1)*). This can be very useful in discovering any system problem when the system is very sluggish. This option is accessible only to users who have appropriate privileges.
- u** User ID (uid) numbers are displayed instead of usernames. This improves execution speed by eliminating the additional time required to map uid numbers to user names.
- h** Hides the individual CPU state information for systems having multiple processors. Only the average CPU status will be displayed.
- n** *number* Show only *number* processes per screen. Note that, if *number* is greater than the maximum number of processes that can be displayed per screen, this option is ignored. But when used with **-f** option, there is no limit on the maximum number of processes that can be displayed.
- f** *filename* Output is redirected to *filename*. By default this option will redirect 16 processes.
- p** *pset\_id* Show load averages and process state break down for system and processor set *pset\_id*. Show only the processes running on the processor set *pset\_id*. This option is supported only if the kernel supports processor sets functionality.
- P** Adds column *PSET* before column *CPU* for individual CPU information. Adds columns *PSET* before column *CPU* for each process information. This option is supported only if the kernel supports processor sets functionality.
- w** Show individual CPU information. By default, **top** does not display any individual CPU information. The user can toggle between individual process information and individual CPU information by using the **w** screen-control key. When used with the **-h** option, the **-w** option overrides the **-h** option.

**Screen-Control Commands**

When displaying multiple-screen data, **top** recognizes the following keyboard screen-control commands:

- j** Display next screen if the current screen is not the last screen.
- k** Display previous screen if the current screen is not the first screen.
- t** Display the first (top) screen.
- w** Display individual CPU information in place of individual process information and vice versa.

**Program Termination**

To exit the program and resume normal user activities, type **q** at any time.

**Display Description**

Three general classes of information are displayed by **top**:

**System Data**

The first few lines at the top of the display show general information about the state of the system, including:

- System name and current time.
- Load averages in the last one, five, and fifteen minutes of all the active processors in the system.
- Number of existing processes and the number of processes in each state (sleeping, waiting, running, starting, zombie, and stopped).
- Percentage of time spent in each of the processor states (user, nice, system, idle, interrupt and swapper) per active processor on the system.
- Average value for each of the active processor states (only on multi-processor systems).

**Memory Data**

Includes virtual and real memory in use (with the amount of memory considered "active" in parentheses) and the amount of free memory.

**Process Data**

Information about individual processes on the system. When process data cannot fit on a single screen, **top** divides the data into two or more screens. To view multiple-screen data, use the **j**, **k**, and **t** commands described previously. Note that the system- and memory-data displays are present in each screen of multiple-screen process data.

Process data is displayed in a format similar to that used by *ps(1)*:

|                 |                                                                                                                                                                                                                                        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CPU</b>      | Processor number on which the process is executing (only on multi-processor systems).                                                                                                                                                  |
| <b>TTY</b>      | Terminal interface used by the process.                                                                                                                                                                                                |
| <b>PID</b>      | Process ID number.                                                                                                                                                                                                                     |
| <b>PSET</b>     | ID of the processor set to which the processor belongs. This is shown only when <b>-P</b> option is used.                                                                                                                              |
| <b>USERNAME</b> | Name of the owner of the process. When the <b>-u</b> option is specified, the user ID (uid) is displayed instead of <b>USERNAME</b> .                                                                                                  |
| <b>PRI</b>      | Current priority of the process.                                                                                                                                                                                                       |
| <b>NI</b>       | Nice value ranging from -20 to +20.                                                                                                                                                                                                    |
| <b>SIZE</b>     | Total virtual size of the process in kilobytes. This includes virtual sizes of text, data, stack, mmap regions, shared memory regions and IO mapped regions. This may also include virtual memory regions shared with other processes. |
| <b>RES</b>      | Resident size of the process in kilobytes. It includes the sizes of all private regions in the process. The resident size information is, at best, an approximate value.                                                               |
| <b>STATE</b>    | Current state of the process. The various states are <b>sleep</b> , <b>wait</b> , <b>run</b> , <b>idl</b> , <b>zomb</b> , or <b>stop</b> .                                                                                             |
| <b>TIME</b>     | Number of system and CPU seconds the process has consumed.                                                                                                                                                                             |
| <b>%WCPU</b>    | Weighted CPU (central processing unit) percentage.                                                                                                                                                                                     |
| <b>%CPU</b>     | Raw CPU percentage. This field is used to sort the top processes.                                                                                                                                                                      |
| <b>COMMAND</b>  | Name of the command the process is currently running.                                                                                                                                                                                  |

**EXAMPLES**

**top** can be executed with or without command-line options.

To display five screens of data at two-second intervals then automatically exit, use:



```
top -s2 -d5
```

```
top -P -s2 -d5
```

To display information about pset 2, use:

```
top -p 2
```

```
top -P -p 2
```

To display individual CPU information in place of individual process information, use:

```
top -w
```

and press the **w** key.

**AUTHOR**

**top** was developed by HP and William LeFebvre of Rice University.

**| t |**

**NAME**

touch - update access, modification, and/or change times of file

**SYNOPSIS**

```
touch [-amc] [-r ref_file | -t time] file_name ...
```

**Obsolescent:**

```
touch time_str file_name ...
```

**DESCRIPTION**

**touch** updates the access, modification, and last-change times of each argument. The file name is created if it does not exist. If no time is specified (see *date*(1)) the current time is used.

The **-r** and **-t** options are mutually exclusive.

**Options**

The following options are available:

- a** Change the access time of *file\_name* to *time*, or to the current time if *time* is not specified. Do not change the modification time unless **-m** is also specified.
- m** Change the modification time of *file\_name* to *time*, or to the current time if *time* is not specified. Do not change the access time unless **-a** is also specified.
- c** Silently prevent **touch** from creating the file if it did not previously exist. Do not write any diagnostic messages concerning this condition.
- r** *ref\_file* Use the corresponding time of *ref\_file* instead of the current time.
- t** *time* Use the specified *time* instead of the current time. The option argument is a decimal number of the form:

```
[[CC]YY]MMDDhhmm[.SS]
```

where each two digits represents the following:

|           |                                    |
|-----------|------------------------------------|
| <i>CC</i> | The first two digits of the year.  |
| <i>YY</i> | The second two digits of the year. |
| <i>MM</i> | The month of the year (01-12).     |
| <i>DD</i> | The day of the month (01-31).      |
| <i>hh</i> | The hour of the day (00-23).       |
| <i>mm</i> | The minute of the hour (00-59).    |
| <i>SS</i> | The second of the minute (00-61).  |

If neither *CC* nor *YY* is given, the current year is assumed. If *YY* is specified, but *CC* is not, *CC* is derived as follows: (taken into account the local time factor)

|                  |                    |
|------------------|--------------------|
| If <i>YY</i> is: | <i>CC</i> becomes: |
| 69-99            | 19                 |
| 00-68            | 20                 |

If the resulting time value precedes the Epoch (00:00:00 January 1, 1970 Greenwich Mean Time), **touch** exits immediately with an error status.

The range for *SS* is 00 through 61 rather than 00 through 59 to accommodate leap seconds. If *SS* is 60 or 61, and the resulting time, as affected by the **TZ** environment variable, does not refer to a leap second, the resulting time is one second after a time where *SS* is 59. If *SS* is not given a value, it is assumed to be 0.

The syntax shown by the second SYNOPSIS line is recognized when neither the **-r** option, the **-t** option, nor the **--** option delimiter is specified, and the first operand consists of all decimal digits. This operand is interpreted as the *time* argument instead of as a file name. However, in this case, *time\_str* is assumed to be of the form:

```
MMDDhhmm[YY]
```

This is for backward compatibility. The `-t` form given above is recommended for future portability. The `--` option delimiter can be used before the first *file\_name* if there is a possibility that *file\_name* consists of all digits, in order to ensure that the first syntax is used.

**touch** succeeds *only* when invoked by the *owner* of the file if any of the following are true:

- A time is specified,
- Only the access time of the file is being updated, or
- Only the modification time of the file is being updated.

In addition, **touch** succeeds when invoked by a user with write permission on the file if both of the following are true:

- No time is specified, *and*
- *Both* the access time and modification time of the file are being updated.

## EXTERNAL INFLUENCES

### Environment Variables

**TZ** If the time is specified via the `-t` option, **TZ** is used to interpret the time for the specified time zone.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **touch** behaves as if all internationalization variables are set to "C". See *environ(5)*.

### International Code Set Support

Single- and multi-byte character code sets are supported.

## RETURN VALUE

**touch** returns zero if all *file\_name* arguments were successfully changed.

**touch** returns non-zero and prints out a diagnostic message if an invalid time or a time earlier than the Epoch was specified with the `-t` option, or if the `-r` and `-t` options were both specified, or if one or more of the *file\_name* arguments could not be accessed.

## EXAMPLES

The following command sets the modification and access times of the file named "bastille" to midnight, July 14, 1989, creating the file if it does not already exist.

```
touch -t 8907140000 bastille
```

The following command does the same thing using the backward-compatible syntax:

```
touch 0714000089 bastille
```

The following command sets the time of the two files named "0714000089" and "bastille" to the current time, creating them if they do not exist:

```
touch -- 0714000089 bastille
```

To create a zero-length file, use any of the following:

```
touch file
cat /dev/null >file
cp /dev/null file
```

## DEPENDENCIES

### NFS:

An attempt to touch a file owned by the super-user on a remote server might fail, even if the invoking user has write permission on the file.

## SEE ALSO

date(1), utime(2).

**touch(1)**

**touch(1)**

**STANDARDS CONFORMANCE**

**touch:** SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| **t** |

**NAME**

tput - query terminfo database

**SYNOPSIS**

```
tput [-T type] capname ...
tput [-T type] capname [parms ...]
tput -s
```

**DESCRIPTION**

The **tput** command uses the **terminfo** database to make terminal-dependent capabilities and information available to the shell (see *terminfo*(4)). The **tput** command outputs a string if the attribute (*capname*) is of type string, or an integer if the attribute is of type integer. If the attribute is of type boolean, **tput** simply sets the exit code (0 for TRUE, 1 for FALSE), and produces no output.

**Command-line Arguments**

The **tput** command recognizes the following command-line arguments:

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-T</b> <i>type</i> | Indicates the type of terminal. Normally this flag is unnecessary because the default is taken from the environment variable <b>TERM</b> .                                                                                                                                                                                                                                                                                      |
| <i>capname</i>        | Indicates the attribute from the <b>terminfo</b> database. See <i>terminfo</i> (4). In addition, the following <i>capnames</i> are supported: <ul style="list-style-type: none"> <li><b>clear</b> Echo the clear-screen sequence for the current terminal.</li> <li><b>init</b> Echo the initialize sequence for the current terminal.</li> <li><b>reset</b> Echo the sequence that will reset the current terminal.</li> </ul> |
| <i>parms</i>          | If the <i>capname</i> takes optional numeric parameters, the <i>parms</i> will be placed in the string output by <b>tput</b> .                                                                                                                                                                                                                                                                                                  |
| <b>-s</b>             | The <i>capnames</i> are read from stdin and multiple <i>capnames</i> are allowed. Only one <i>capname</i> is allowed per line when reading from stdin.                                                                                                                                                                                                                                                                          |

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_ALL** determines the locale to use. This overrides settings of other environment variables.

**LC\_MESSAGES** determines the language to use for messages.

**TERM** determines the terminal type if the **-T** option is not specified.

**EXAMPLES**

Echo clear-screen sequence for the current terminal.

```
tput clear
```

Print the number of columns for the current terminal.

```
tput cols
```

Print the number of columns for the 70092 terminal.

```
tput -T70092 cols
```

Set shell variable **bold** to stand-out-mode sequence for current terminal.

```
bold=`tput smso`
```

This might be followed by a prompt:

```
echo "${bold}Please type in your name: \c"
```

Set exit code to indicate if current terminal is a hard copy terminal.

```
tput hc
```

| t |

Clear the screen, move the cursor to line 10, column 20 and turn on bold.

```
tput -S <<EOF
clear
cup 10 20
bold
EOF
```

#### RETURN VALUE

If *capname* is of type boolean, then the exit code is set to 0 for true and 1 for false.

If *capname* is not of type boolean and **tput** fails, an error message is printed, and exit code is set to one of the following depending on the failure:

- 0 The capability name is of type integer and does not exist.
- 2 Usage error.
- 3 Unknown terminal type.
- 4 Unknown capability name.
- >4 An error occurred.

If the exit code is 0, a -1 is printed if a capability name of type integer is requested for a terminal that has no entry for that capability name in the **terminfo** database (such as **tput -Thp70092 vt**).

#### FILES

|                                    |                    |
|------------------------------------|--------------------|
| <b>/usr/share/lib/terminfo/?/*</b> | Terminfo data base |
| <b>/usr/include/curses.h</b>       | Definition files   |
| <b>/usr/include/term.h</b>         |                    |

#### SEE ALSO

stty(1), untic(1M), terminfo(4).

#### STANDARDS CONFORMANCE

**tput**: SVID2, SVID3, XPG4

| t |

**NAME**

tr - translate characters

**SYNOPSIS**

tr [-Acs] *string1 string2*

tr -s [-Ac] *string1*

tr -d [-Ac] *string1*

tr -ds [-Ac] *string1 string1*

**DESCRIPTION**

**tr** copies the standard input to the standard output with substitution or deletion of selected characters. Input characters from *string1* are replaced with the corresponding characters in *string2*. If necessary, *string1* and *string2* can be quoted to avoid pattern matching by the shell.

**tr** recognizes the following command line options:

- A** Translates on a byte-by-byte basis. When this flag is specified **tr** does not support extended characters.
- c** Complements the set of characters in *string1*, which is the set of all characters in the current character set, as defined by the current setting of **LC\_CTYPE**, except for those actually specified in the *string1* argument. These characters are placed in the array in ascending collation sequence, as defined by the current setting of **LC\_COLLATE**.
- d** Deletes all occurrences of input characters or collating elements found in the array specified in *string1*.

If **-c** and **-d** are both specified, all characters except those specified by *string1* are deleted. The contents of *string2* are ignored, unless **-s** is also specified. Note, however, that the same string cannot be used for both the **-d** and the **-s** flags; when both flags are specified, both *string1* (used for deletion) and *string2* (used for squeezing) are required.

If **-d** is not specified, each input character or collating element found in the array specified by *string1* is replaced by the character or collating element in the same relative position specified by *string2*.

- s** Replaces any character specified in *string1* that occurs as a string of two or more repeating characters as a single instance of the character in *string2*.

If the *string2* contains a character class, the argument's array contains all of the characters in that character class. For example:

```
tr -s '[:space:]'
```

In a case conversion, however, the *string2* array contains only those characters defined as the second characters in each of the **toupper** or **tolower** character pairs, as appropriate. For example:

```
tr -s '[:upper:]' '[:lower:]'
```

The following abbreviation conventions can be used to introduce ranges of characters, repeated characters or single-character collating elements into the strings:

*c1-c2* or [*c1-c2*] Stands for the range of collating elements *c1* through *c2*, inclusive, as defined by the current setting of the **LC\_COLLATE** locale category.

[*:class*:] or [[*:class*:]] Stands for all the characters belonging to the defined character class, as defined by the current setting of **LC\_CTYPE** locale category. The following character class names will be accepted when specified in *string1*: **alnum**, **alpha**, **blank**, **cntrl**, **digit**, **graph**, **lower**, **print**, **punct**, **space**, **upper**, or **xdigit**. Character classes are expanded in collation order.

When the **-d** and **-s** flags are specified together, any of the character class names are accepted in *string2*; otherwise, only character class names **lower** or **upper** are accepted in *string2* and then only if the corresponding character class (**upper** and **lower**, respectively) is specified in the same relative position in *string1*. Such a specification is interpreted as a request for case conversion.

When `[:lower:]` appears in *string1* and `[:upper:]` appears in *string2*, the arrays contain the characters from the `toupper` mapping in the `LC_CTYPE` category of the current locale. When `[:upper:]` appears in *string1* and `[:lower:]` appears in *string2*, the arrays contain the characters from the `tolower` mapping in the `LC_CTYPE` category of the current locale.

`[=c=]` or `[[=c=]]` Stands for all the characters or collating elements belonging to the same equivalence class as *c*, as defined by the current setting of `LC_COLLATE` locale category. An equivalence class expression is allowed only in *string1*, or in *string2* when it is being used by the combined `-d` and `-s` options.

`[a*n]` Stands for *n* repetitions of *a*. If the first digit of *n* is 0, *n* is considered octal; otherwise, *n* is treated as a decimal value. A zero or missing *n* is interpreted as large enough to extend *string2*-based sequence to the length of the *string1*-based sequence.

The escape character `\` can be used as in the shell to remove special meaning from any character in a string. In addition, `\` followed by 1, 2, or 3 octal digits represents the character whose ASCII code is given by those digits.

An ASCII NUL character in *string1* or *string2* can be represented only as an escaped character; i.e. as `\000`, but is treated like other characters and translated correctly if so specified. NUL characters in the input are not stripped out unless the option `-d "\000"` is given.

## EXTERNAL INFLUENCES

### Environment Variables

`LANG` provides a default value for the internationalization variables that are unset or null. If `LANG` is unset or null, the default value of "C" (see *lang(5)*) is used. If any of the internationalization variables contains an invalid setting, `tr` will behave as if all internationalization variables are set to "C". See *environ(5)*.

`LC_ALL` If set to a non-empty string value, overrides the values of all the other internationalization variables.

`LC_CTYPE` determines the interpretation of text as single and/or multi-byte characters, the classification of characters as printable, and the characters matched by character class expressions in regular expressions.

`LC_MESSAGES` determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

`NLSPATH` determines the location of message catalogues for the processing of `LC_MESSAGES`.

## RETURN VALUE

`tr` exits with one of the following values:

- 0 All input was processed successfully.
- >0 An error occurred.

## EXAMPLES

For the ASCII character set and default collation sequence, create a list of all the words in *file1*, one per line in *file2*, where a word is taken to be a maximal string of alphabetic. Quote the strings to protect the special characters from interpretation by the shell (012 is the ASCII code for a new-line (line feed) character):

```
tr -cs "[A-Z][a-z]" "[\012*]" <file1 >file2
```

Same as above, but for all character sets and collation sequences:

```
tr -cs "[:alpha:]" "[\012*]" <file1 >file2
```

Translate all lower case characters in *file1* to upper case and write the result to standard output.

```
tr "[:lower:]" "[:upper:]" <file1
```

Use an equivalence class to identify accented variants of the base character `e` in *file1*, strip them of diacritical marks and write the result to *file2*:

```
tr "[=e=]" "[e*]" <file1 >file2
```



Translate each digit in *file1* to a # (number sign), and write the result to *file2*.

```
tr "0-9" "[#*]" <file1 >file2
```

The \* (asterisk) tells **tr** to repeat the # (number sign) enough times to make the second string as long as the first one.

**AUTHOR**

**tr** was developed by OSF and HP.

**SEE ALSO**

ed(1), sh(1), ascii(5), environ(5), lang(5), regexp(5).

**STANDARDS CONFORMANCE**

**tr**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| t |**

**NAME**

`true`, `false` - return exit status zero or one respectively

**SYNOPSIS**

`true`  
`false`

**DESCRIPTION**

The command `true` does nothing, and returns exit code zero. The command `false` does nothing, and returns exit code one. They are typically used to construct command procedures.

**RETURN VALUE**

Exit values are:

|   |                            |
|---|----------------------------|
| 0 | always from <i>true</i> .  |
| 1 | always from <i>false</i> . |

**EXAMPLES**

This command loop repeats without end:

```
while true
do
 command
done
```

**WARNINGS**

`true` is typically used in shell scripts. Some shells provide a built-in version of `true` (and `false`) that is more efficient than the standalone versions.

**SEE ALSO**

`csh(1)`, `ksh(1)`, `sh(1)`, `sh-posix(1)`.

**STANDARDS CONFORMANCE**

`true`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

`false`: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| **t** |

**NAME**

tryfrom, sffinger - utility programs for TCP Wrappers

**SYNOPSIS**

`/usr/bin/tryfrom`

`/usr/bin/sffinger`

**DESCRIPTION****tryfrom**

This program identifies the end-point details related to a connection. This program must be called via a remote shell command as shown below. It is used to find out if the hostname and the address are properly recognized, and also if the username lookup works properly.

```
remsh host /usr/bin/tryfrom
```

**tryfrom** prints the following output when it is invoked:

```
client address (%a):
client hostname (%n):
client username (%u):
client info (%c):
server address (%A):
server hostname (%N):
server process (%d):
server info (%s):
```

The information related to the client describes how the remote host recognizes the client in terms of address, name and user name. The information related to the server gives the remote host's details.

See *hosts\_access*(5) manual page for more information on *%letter* expressions.

**sffinger**

This is a wrapper program to the *finger*(1) client which offers better protection against the possible damage from data sent by the remote **finger** server. This command accepts all the options supported by *finger*(1).

It is recommended to use this program in the implementation of traps in the access control language of `/etc/hosts.allow` and `/etc/hosts.deny`. See *hosts\_access*(5) man page for more information on setting traps.

This program restricts each line length to 128 bytes and total response to a maximum of 100000 bytes. **sffinger** uses 60 seconds time-out value in getting the response from the remote host's **finger** server.

**AUTHOR**

Wietse Venema (wietse@wzv.win.tue.nl),  
 Department of Mathematics and Computing Science,  
 Eindhoven University of Technology  
 Den Dolech 2, P.O. Box 513,  
 5600 MB Eindhoven, The Netherlands

**SEE ALSO**

*finger*(1), *tcpd*(1M), *hosts\_access*(5).

| t |

**NAME**

tset, reset - terminal-dependent initialization

**SYNOPSIS**

```
tset [options] [-m [ident] [test baudrate] :type]... [type]
reset
```

**DESCRIPTION**

**tset** sets up the terminal when logging in on an HP-UX system. It does terminal-dependent processing, such as setting erase and kill characters, setting or resetting delays, and sending any sequences needed to properly initialize the terminal. It first determines the *type* of terminal involved, then does the necessary initializations and mode settings. The type of terminal attached to each HP-UX port is specified in the `/etc/ttytype` data base. Type names for terminals can be found in the files under the `/usr/share/lib/terminfo` directory (see *terminfo*(4)). If a port is not wired permanently to a specific terminal (not hardwired), it is given an appropriate generic identifier, such as **dialup**.

**reset** performs a similar function, setting the terminal to a sensible default state.

In the case where no arguments are specified, **tset** simply reads the terminal type out of the environment variable **TERM** and re-initializes the terminal. The rest of this manual entry concerns itself with mode and environment initialization, typically done once at login, and options used at initialization time to determine the terminal type and set up terminal modes.

When used in a startup script (`.profile` for *sh*(1), or `.login` for *cs*h(1) users), it is desirable to give information about the type of terminal that will normally be used on ports that are not hardwired. These ports are identified in `/etc/ttytype` as **dialup** or **plugboard**, etc. To specify what terminal type you usually use on these ports, the `-m` (map) option flag is followed by the appropriate port type identifier, an optional baud rate specification, and the terminal type. (The effect is to "map" from some conditions to a terminal type; that is, to tell **tset** that "If I am on this kind of port, I will probably be on this kind of terminal.") If more than one mapping is specified, the first applicable mapping prevails. A missing port type identifier matches all identifiers. A *baudrate* is specified as with **stty** (see *stty*(1)), and is compared with the speed of the diagnostic output (which should be the control terminal). The baud rate *test* can be any combination of `>`, `=`, `<`, `@`, and `!`. `@` is a synonym for `=`, and `!` inverts the sense of the test. To avoid problems with metacharacters, it is best to place the entire argument to `-m` within single quotes; users of *cs*h(1) must also put a `\` before any `!` used.

Thus,

```
tset -m 'dialup>300:2622' -m 'dialup:2624' -m 'plugboard:?2623'
```

causes the terminal type to be set to an HP 2622 if the port in use is a dialup at a speed greater than 300 baud, or to an HP 2624 if the port is otherwise a dialup (i.e., at 300 baud or less). If the *type* finally determined by **tset** begins with a question mark, the user is asked for verification that the type indicated is really the one desired. A null response means to use that type; otherwise, another type can be entered. Thus, in the above case, if the user is on a plugboard port, he or she will be asked whether or not he or she is actually using an HP 2623.

If no mapping applies and a final *type* option, not preceded by a `-m`, is given on the command line, that type is used. Otherwise, the identifier found in the `/etc/ttytype` data base is taken to be the terminal type. The latter should always be the case for hardwired ports.

It is usually desirable to return the terminal type, as finally determined by *tset*, and information about the terminal's capabilities to a shell's environment. This can be done using the `-s` option. From *sh*(1), the command:

```
eval `tset -s options...`
```

or using the C shell, (*cs*h(1)):

```
set noglob; eval `tset -s options...`
```

These commands cause **tset** to generate as output a sequence of shell commands which place the variable **TERM** in the environment; see *environ*(5).

Once the terminal type is known, **tset** engages in terminal mode setting. This normally involves sending an initialization sequence to the terminal, setting the single character erase (and optionally the full line erase or line-kill) characters, and setting special character delays. Tab and new-line expansion are turned off during transmission of the terminal initialization sequence.

On terminals that can backspace but not overstrike (such as a CRT), and when the erase character is the default erase character (**#** on standard systems), the erase character is changed to Backspace.

### Options

**tset** recognizes the following options:

- ec** Set the erase character to be the named character *c*; *c* defaults to what the terminfo database (see *terminfo*(4)) entry reports to be the character sent by the Backspace key (usually **^H**). The character *c* can either be typed directly, or entered using circumflex notation used here (e.g., the circumflex notation for control-H is **^H**).
- kc** Set the kill character to *c*. The default *c* is **^X**. If *c* is not specified, the kill character remains unchanged unless the original value of the kill character is null, in which case the kill character is set to **@**.
- Report terminal type. Whatever type is decided on is reported. If no other flags are given, the only effect is to write the terminal type on the standard output. Has no effect if used with **-s**.
- s** Generate appropriate commands (depending on current **SHELL** environment variable) to set **TERM**.
- I** Suppress transmitting terminal initialization strings.
- Q** Suppress printing the **Erase set to** and **Kill set to** messages.
- A** Ask the user for the **TERM** type.
- S** Output the strings that would be assigned to **TERM** in the environment rather than generating commands for a shell. In *sh*(1), the following is an alternate way of setting **TERM**:
 

```
set -- `tset -S ...`
TERM=$1
```
- h** Force a read of **/etc/ttytype**. When **-h** is not specified, the terminal type is determined by reading the environment unless some mapping is specified.

For compatibility with earlier versions of **tset**, the following flags are accepted, but their use is discouraged:

- r** Report to the user in addition to other flags.
- Ec** Set the erase character to *c* only if the terminal can backspace. *c* defaults to what the terminfo database (see *terminfo*(4)) entry reports to be the character sent by the Backspace key (usually **^H**).

In addition to capabilities described in **terminfo** (see *termio*(7) and *terminfo*(4)), the following boolean terminfo capabilities are understood by **tset** and **reset**, and can be included in the terminfo database for the purpose of terminal setup:

- UC** "Uppercase" mode sets character mapping for terminals that support only uppercase characters. Equivalent to **stty lcase**.
- LC** "Lowercase" mode permits input and output of lowercase characters. Equivalent to **stty -lcase**.
- EP** Set "even parity". Equivalent to **stty parenb -parodd**
- OP** Set "odd parity". Equivalent to **stty parenb parodd**.
- NL** Set "new line" mode. Equivalent to **stty onlret**.
- HD** Set "half-duplex" mode. Equivalent to **stty -echo**.
- pt** Set "print tabs" mode. Equivalent to **stty tabs**.

### EXTERNAL INFLUENCES

#### Environment Variables

- SHELL** if **cs**h, generate **cs**h commands; otherwise generate *sh*(1) commands.
- TERM** the (canonical) terminal name.

### EXAMPLES

These examples all assume the *sh*(1). Note that a typical use of **tset** in a **.profile** also uses the **-e** and **-k** options, and often the **-m** or **-Q** options as well. These options have been omitted here to keep

the examples small.

Assume, for the moment, that you are on an HP 2622. This is suitable for typing by hand but not for a `.profile` unless you are *always* on a 2622.

```
export TERM; TERM=`tset - 2622`
```

Assume you have an HP 2623 at home that you dial up on, but your office terminal is hardwired and known in `/etc/ttytype`.

```
export TERM; TERM=`tset - -m dialup:2623`
```

Suppose you are accessing the system through a switching network that can connect any system to any incoming modem line in an arbitrary combination, making it nearly impossible to key on what port you are coming in on. Your office terminal is an HP 2622, and your home terminal is an HP 2623 running at 1200 baud on dial-up switch ports. Sometimes you use someone else's terminal at work, so you want it to verify what terminal type you have at high speeds, but at 1200 baud you are always on a 2623. Note the placement of the question mark and the quotes to protect the `>` and `?` from interpretation by the shell.

```
export TERM; TERM=`tset - -m 'switch>1200:?2622'
-m 'switch<=1200:2623'`
```

All of the above entries fall back on the terminal type specified in `/etc/ttytype` if none of the conditions hold. The following entry is appropriate if you always dial up, always at the same baud rate, on many different kinds of terminals. Your most common terminal is an HP 2622. It always asks you what kind of terminal you are on, defaulting to 2622.

```
export TERM; TERM=`tset - ?2622`
```

If the file `/etc/ttytype` is not properly installed and you want to key entirely on the baud rate, the following can be used:

```
export TERM; TERM=`tset - -m '>1200:2624' 2622`
```

#### AUTHOR

`tset` was developed by the University of California, Berkeley.

#### FILES

```
/etc/ttytype
 port-name to terminal-type mapping data base

/usr/share/lib/terminfo/??/*
 terminal information data base
```

#### SEE ALSO

`csh(1)`, `sh(1)`, `stty(1)`, `ttytype(4)`, `environ(5)`.

| t |

**NAME**

tsm - Terminal Session Manager

**SYNOPSIS**

tsm

**DESCRIPTION**

**tsm** allows a user to interact with more than one shell or application (session) from a single terminal. Each session is bound to a virtual device emulating the physical terminal. The emulation includes maintaining display state, softkeys, and terminal modes for each session. The virtual device can be manipulated like the actual terminal by using **stty** and **ioctl** (see *stty*(1) and *ioctl*(2)). Additionally **tsm** supports cut and paste between sessions, and provides an interface for a local lp device. Each session has its own process group ID.

**Definitions**

A session is **current** if it is being displayed and is the recipient of keyboard input.

The **standard search path** is:

```
./
$HOME
$TSMSPATH
/usr/tsm/
```

Configuration files and such are searched for in the order indicated and defined by these paths.

**Commands**

There are two methods of interacting with **tsm**: a pull-down menu, and a command line interface. The pull-down menu (when configured) can be activated from a session by pressing the **tsm menu hot key** (default is **^T**) and should be self explanatory. The command line interface can be activated by pressing the **tsm hot key** (default is **^W**) in a session. Pressing a "hot key" twice passes the "hot key" character to the session instead of activating **tsm** command or menu mode.

Commands to **tsm** generally have single character invocation, in some cases the user is prompted for more input. The following commands can be issued from the **tsm** prompt level:

- 0-9** Pressing a number at the command prompt selects the session of the same number to become the current session.
- +** Select the next higher numbered session.
- Select the next lower numbered session.
- l** Select the last session.
- ?** Display a help screen describing **tsm** commands.
- c** Copy (cut): Three types:
  - Text (Lines including new-lines). This is the default. Select with **T** when cut prompt is displayed.
  - String (Lines strung together with white space in place of new-lines). Select with **T** when cut prompt is displayed.
  - Block (A rectangle). Select with **T** when cut prompt is displayed.

The user is prompted for the "cut extents". The extents are defined by using arrow keys or the keys **u**, **d**, **l**, and **r** to move the cursor as desired. Pressing the space bar aborts the cut operation. The selected text is placed in the **cut buffer**. Trailing whitespace and character attribute information are ignored.

- p** Paste: the contents of the **cut buffer** is echoed to the current session as if it were typed from the keyboard.
- r** Run a program as a new session. The user is prompted for the program name.
- s** Start a new session containing a shell.
- o** Output the current display to a printer (screen dump). The print mechanism is specified in a file named **.tsmprint** searched for in the standard way. Character attribute information is ignored.

| t |

- k** Load the softkeys of the current session from a file. To load **tsm** defaults, specify “file” +. To load terminal defaults, specify “file” -.
- g** Same as **k** above but softkeys are loaded “globally” into all sessions.
- x** Access extended **tsm** commands as described in the **tsm** reference manual or on the **tsm** help screen.
- q** Quit **tsm**: **SIGHUP** is sent to all processes started under **tsm**, and **tsm** exits.

#### EXTERNAL INFLUENCES

In general **tsm** environment variables must be set prior to **tsm** invocation. **TSMLP** is the *lp(1)* name of a printer that gets its output redirected to the printer port of the terminal.

**TSMPATH** specifies an alternate search path for **tsm** files.

**TSMTERM** specifies an alternate terminal information file to be used by **tsm** instead of that specified by **TERM**. **TSMHOTKEY** specifies an alternate **tsm** hotkey for invocation of the **tsm** command line.

#### WARNINGS

Some operations are not supported on certain terminals.

#### AUTHOR

**tsm** was developed by Structured Software Solutions, Inc.

#### FILES

|                          |                                                                                                |
|--------------------------|------------------------------------------------------------------------------------------------|
| <b>/usr/tsm/.tsm</b>     | <b>tsm</b> main configuration file (default). Copy to <b>\$HOME</b> for user customization.    |
| <b>/usr/tsm/.tsmkeys</b> | <b>tsm</b> softkey configuration file (default). Copy to <b>\$HOME</b> for user customization. |
| <b>/usr/tsm/term/*</b>   | terminal description files                                                                     |

#### SEE ALSO

tsm.info(1), tsm.command(1), tsm.lpadmin(1M), shl(1).



**NAME**

`tsm.command` - send commands to the Terminal Session Manager (TSM)

**SYNOPSIS**

`/usr/tsm/bin/tsm.command command`

**DESCRIPTION**

`tsm.command` is used to send a command string programmatically to the Terminal Session Manager (TSM), as if the string were typed on the TSM command line. `tsm.command` fails unless it is run from inside a TSM session. Actions caused by `tsm.command` affect only the instance of TSM that `tsm.command` is run under. *command* can have any value that is a valid key sequence for the TSM command line. The sequence should not include the "hotkey" character that normally initiates the command line mode of TSM. The sequence should end at the point where TSM exits command mode. If it ends prematurely TSM behaves as though escape was pressed, which exits command mode, usually canceling the command. `\r` should be used to indicate a return key. If no arguments are given on the command line, the program prompts for input from the user. If a `^C` terminates the sequence, the remainder of the sequence is accepted from the user.

**AUTHOR**

`tsm.command` was developed by Structured Software Solutions, Inc.

**SEE ALSO**

`tsm(1)`.

**NAME**

tsm.info - get Terminal Session Manager state information

**SYNOPSIS**

*/usr/tsm/bin/tsm.info request*

**DESCRIPTION**

**tsm.info** is used to obtain information about TSM. When run from inside a TSM session it returns valid information; otherwise it fails with a nonzero error code. Information returned is written to standard output. *request* can have any of the following values:

|                               |                                                                                                                                             |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>is_a_window</b>            | Successful (returns zero) if executed from a TSM session, nonzero error code otherwise.                                                     |
| <b>session_number</b>         | Writes the session number of the session the <b>tsm.info</b> command is executed in.                                                        |
| <b>current_session_number</b> | Writes the <i>session_number</i> of the TSM session the user currently has active.                                                          |
| <b>active_session_numbers</b> | Writes the session numbers (separated by whitespace) of all active sessions (sessions not idle).                                            |
| <b>idle_session_numbers</b>   | Writes the session numbers (separated by whitespace) of all idle sessions.                                                                  |
| <b>program_name</b>           | Writes the program name (as assigned in <i>.tsm</i> or with <b>tsm.command</b> ) of the session the <b>tsm.info</b> command is executed in. |
| <b>program_name_n</b>         | Writes the program name of session <i>n</i> .                                                                                               |

**AUTHOR**

**tsm.info** was developed by Structured Software Solutions, Inc.

**SEE ALSO**

tsm(1)

| t |

**NAME**

tsort - topological sort

**SYNOPSIS**

**tsort** [*file*]

**DESCRIPTION**

**tsort** produces on the standard output a totally ordered list of items consistent with a partial ordering of items mentioned in the input text *file*. If no *file* is specified, the standard input is understood. **tsort** is generally used in conjunction with the **lorder** command to sort the objects to be installed in a library by **ar** (see *lorder*(1) and *ar*(1)).

The input consists of pairs of text items (nonempty strings) separated by blanks. Pairs of different items indicate ordering. Pairs of identical items indicate presence, but not ordering.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the locale for the interpretation of text as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **tsort** behaves as if all internationalization variables are set to "C". See *environ*(5).

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**DIAGNOSTICS****Odd data**

There is an odd number of fields in the input file.

**WARNINGS**

Libraries and object files cannot be **tsorted** directly.

**tsort** uses a quadratic algorithm that is not considered worth fixing given its typical use of ordering a library archive file.

**SEE ALSO**

*lorder*(1).

**STANDARDS CONFORMANCE**

**tsort**: SVID2, SVID3, XPG2, XPG3, XPG4

| t |

**NAME**

tty, pty - get the name of the terminal

**SYNOPSIS**

tty [-s]

pty [-s]

**DESCRIPTION**

tty and pty print the path name of the user's terminal. The -s option inhibits printing of the terminal path name and any diagnostics, providing a means to test only the exit code.

**RETURN VALUE**

Exit status codes for tty are:

- 2 Invalid options were specified,
- 1 The standard input is not a terminal or pseudo-terminal,
- 0 The standard input is a terminal or pseudo-terminal.

Exit status codes for pty are:

- 2 Invalid options were specified,
- 1 The standard input is not a pseudo-terminal,
- 0 The standard input is a pseudo-terminal.

**DIAGNOSTICS**

not a tty

standard input is not a terminal or pseudo-terminal for tty.

not a pty

standard input is not a pseudo-terminal for pty.

**AUTHOR**

tty was developed by AT&T. pty was developed by HP.

**STANDARDS CONFORMANCE**

tty: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| t |

**NAME**

ttytype - terminal identification program

**SYNOPSIS**

ttytype [-apsv] [-t *type* ]

**DESCRIPTION**

**ttytype** automatically identifies the current terminal type by sending an identification request sequence to the terminal. This method works for local, modem, and remote terminal connections, as well as for the **hpterm** and **xterm** terminal emulators.

Once the terminal has been identified, **ttytype** prints the terminal's type to the standard output (see *terminfo*(4)). This string is usually used as the value for the **TERM** environment variable.

If **ttytype** is unable to determine the correct terminal type, it prompts the user for the correct terminal identification string.

**Options**

**ttytype** recognizes the following options:

**-a** Causes **ttytype** to return an ID of "unknown" instead of prompting for the terminal type if auto-identification fails. If this option is not present, **ttytype** interactively prompts the user for the terminal type if it is unable to determine the correct type automatically.

**-p** Causes **ttytype** to prompt for the terminal type before it sends the terminal identification request sequence. If the user responds with only a carriage return, **ttytype** proceeds with the automatic terminal identification process. Any other response is taken as the correct terminal type. Note that the **LINES** and **COLUMNS** variables are not set if the user manually enters a terminal type.

The **-p** option is normally used only for terminals that do not behave well when presented with **ttytype**'s terminal identification request sequence. It gives the user a chance to respond with the correct terminal type before any escape sequences are sent that could have an adverse effect on the terminal.

The **-a** option can be used in conjunction with the **-p** option. The **-a** option only inhibits interactive prompting after **ttytype** has failed to identify the terminal by other means.

**-s** Tells **ttytype** to print a series of shell commands to set the **TERM**, **LINES**, and **COLUMNS** environment variables to appropriate values. In addition, the variable **ERASE** is set to the two-character sequence representing the appropriate erase character for the terminal (DEL for ANSI terminals, backspace for all others). This two-character sequence can then be used as an argument to **stty** or **tset** (see *stty*(1) and *tset*(1)).

The **SHELL** environment variable is consulted to see which shell syntax to use for setting the environment variables. This output is normally used with a command of the form:

```
eval `ttytype -s`
```

**-t *type*** **ttytype** normally attempts identification of Wyse, ANSI and HP terminals. The **-t *type*** argument can be used to restrict the inquiry to that required for terminals of the specified type. The accepted types are **ansi**, **hp**, and **wyse**. Multiple **-t** options can be specified.

**-v** Enable verbose messages to standard error.

**Notes**

Use of the **-s** option is highly recommended because many terminals support variable-size displays. This option provides the only means for automatically configuring the user environment in such a manner that applications can handle these terminals correctly. Note that **LINES** and **COLUMNS** are not set if the **-p** option is used and the user manually enters a terminal type.

The following steps are performed in the order indicated when identifying a terminal:

1. **ttytype** tries the Wyse 30/50/60 id request sequence.
2. **ttytype** tries the standard ANSI id request sequence. If a response is received, it is converted to a string according to an internal table.
3. **ttytype** tries the HP id request sequence.
4. If none of the above steps succeed, **ttytype** prompts interactively for the correct terminal type unless the **-a** option has been given.

**ttytype** may skip one or more of the first three steps, depending on the presence of **-t** options.

The HP ID-request sequence can switch some ANSI terminals into an unexpected operating mode. Recovery from such a condition sometimes requires cycling power on the terminal. To avoid this problem, **ttytype** always sends the HP identification sequence last.

## EXAMPLES

**ttytype** is most commonly used as part of the login sequence. The following shell script fragment can be used during login shell initialization:

```
#
If TERM is not set, see if our port is listed in /etc/ttytype.
If /etc/ttytype doesn't have information for our port, run
ttytype(1) to try to determine the type of terminal we have.
#
To have ttytype(1) prompt for the terminal type before trying
to automatically identify the terminal, add the "-p" option
to the "ttytype -s" command below.
#
if [-z "$TERM" -o "$TERM" = network]; then
 unset TERM
 eval `tset -s -Q`
 if [-z "$TERM" -o "$TERM" = unknown]; then
 eval `ttytype -s`
 tset -Q -e ${ERASE:-\^h} $TERM
 fi
fi
```

## WARNINGS

The terminal identification sequences sent by **ttytype** can cause unexpected behavior on terminals other than the Wyse 30/50/60, standard ANSI or HP terminals. If you have such terminals in your configuration, use the **-t** or **-p** options to prevent **ttytype** from sending sequences that cause unexpected behavior.

| t |

## AUTHOR

**ttytype** was developed by HP.

## SEE ALSO

csh(1), ksh(1), sh(1), stty(1), ttytype(4), environ(5).

**NAME**

**ul** - do underlining

**SYNOPSIS**

**ul** [-**t** *terminal*] [-**i**] [*name ...*]

**DESCRIPTION**

**ul** reads the named files (or standard input if none are given) and translates occurrences of underscores to the sequence which indicates underlining for the terminal in use, as specified by the environment variable **TERM**. The **-t** option overrides the terminal type specified in the environment. The *terminfo(4)* file corresponding to **TERM** is read to determine the appropriate sequences for underlining. If the terminal is incapable of underlining, but is capable of a standout mode, the standout mode is used instead. If the terminal can overstrike, or handles underlining automatically, **ul** degenerates to **cat**. If the terminal cannot underline, underlining is ignored.

The **-i** option causes **ul** to indicate underlining onto by a separate line containing appropriate dashes -; this is useful when you want to look at the underlining present in an **nroff** output stream on a CRT terminal.

**EXTERNAL INFLUENCES****International Code Set Support**

Single- and multi-byte character code sets are supported with the exception that multi-byte-character file names are not supported.

**WARNINGS**

**nroff** usually outputs a series of backspaces and underlines intermixed with the text to indicate underlining. No attempt is made to optimize the backward motion.

**AUTHOR**

**ul** was developed by the University of California, Berkeley.

**FILES**

**/usr/share/lib/terminfo/?/\*** terminal capability files

**SEE ALSO**

**col(1)**, **man(1)**, **nroff(1)**.

**| u |**

**NAME**

umask - set or display the file mode creation mask

**SYNOPSIS**

**Set Mask**

**umask** *mask*

**Display Mask**

**umask** [-S]

**DESCRIPTION**

The **umask** command sets the value of the file mode creation mask or displays the current one. The mask affects the initial value of the file mode (permission) bits for subsequently created files.

**Setting the File Mode Creation Mask**

The **umask** *mask* command sets a new file mode creation mask for the current shell execution environment. *mask* can be a symbolic or numeric (obsolescent) value.

A symbolic mask provides a flexible way of modifying the mask permission bits individually or as a group. A numeric mask specifies all the permission bits at one time.

When a mask is specified, no output is written to standard output.

**Symbolic Mask Value**

A symbolic *mask* replaces or modifies the current file mode creation mask. It is specified as a comma-separated list of operations in the following format. Whitespace is not permitted.

[*who*][*operator*][*permissions*][, ...]

The fields can have the following values:

*who* One or more of the following letters:

- u** Modify permissions for user (owner).
- g** Modify permissions for group.
- o** Modify permissions for others.

Or:

**a** Modify permissions for all (**a = ugo**).

*operator* One of the following symbols:

- +** Add *permissions* to the existing mask for *who*.
- Delete *permissions* from the existing mask for *who*.
- =** Replace the existing mask for *who* with *permissions*.

*permissions* One or more of the following letters:

- r** The read permission.
- w** The write permission.
- x** The execute/search permission.

If one or two of the fields are omitted, the following table applies:

| <b>Format Entered</b>      | <b>Effect</b>                             | <b>Input</b> | <b>Equals</b> |
|----------------------------|-------------------------------------------|--------------|---------------|
| <i>who</i>                 | Delete current permissions for <i>who</i> | <b>g</b>     | <b>g=</b>     |
| <i>operator</i>            | No action                                 | <b>-</b>     | (none)        |
| <i>permissions</i>         | Equal to: <b>a+permissions</b>            | <b>rw</b>    | <b>a+rw</b>   |
| <i>who=</i>                | Delete current permissions for <i>who</i> | <b>u=</b>    | <b>u=</b>     |
| <i>who+</i>                | No action                                 | <b>u+</b>    | (none)        |
| <i>who-</i>                | No action                                 | <b>u-</b>    | (none)        |
| <i>whopermissions</i>      | Equal to: <i>who=permissions</i>          | <b>ux</b>    | <b>u=x</b>    |
| <i>operatorpermissions</i> | Equal to: <b>aoperatorpermissions</b>     | <b>-rw</b>   | <b>a-rw</b>   |

| **u** |



**Numeric Mask Value (Obsolescent)**

A numeric *mask* replaces the current file mode creation mask. It is specified as an unsigned octal integer, constructed from the logical OR (sum) of the following mode bits (leading zeros can be omitted):

|      |                      |                                        |
|------|----------------------|----------------------------------------|
| 0400 | ( <b>a=rwx,u-r</b> ) | Read by owner                          |
| 0200 | ( <b>a=rwx,u-w</b> ) | Write by owner                         |
| 0100 | ( <b>a=rwx,u-x</b> ) | Execute (search in directory) by owner |
| 0040 | ( <b>a=rwx,g-r</b> ) | Read by group                          |
| 0020 | ( <b>a=rwx,g-w</b> ) | Write by group                         |
| 0010 | ( <b>a=rwx,g-x</b> ) | Execute/search by group                |
| 0004 | ( <b>a=rwx,o-r</b> ) | Read by others                         |
| 0002 | ( <b>a=rwx,o-w</b> ) | Write by others                        |
| 0001 | ( <b>a=rwx,o-x</b> ) | Execute/search by others               |

**Displaying the Current Mask Value**

To display the current file mode creation mask value, use one of the commands:

**umask -s** Print the current file mode creation mask in a symbolic format:

**u=[r][w][x],g=[r][w][x],o=[r][w][x]**

The characters **r** (read), **w** (write), and **x** (execute/search) represent the bits that are clear in the mask for **u** (user/owner), **g** (group), and **o** (other). All other bits are set.

**umask** Print the current file mode creation mask as an octal value.

**0[o][o][o]**

The zero bits in the numeric value correspond to the displayed **r**, **w**, and **x** permission characters in the symbolic value. The one bits in the numeric value correspond to the missing permission characters in the symbolic value.

Depending on implementation, the display consists of one to four octal digits; the first digit is always zero (see **DEPENDENCIES**). The rightmost three digits (leading zeros implied as needed) represent the bits that are set or clear in the mask.

Both forms produce output that can be used as the *mask* argument to set the mask in a subsequent **umask** command.

**General Operation**

When a new file is created (see *creat(2)*), each bit that is set in the file mode creation mask causes the corresponding permission bit in the file mode to be cleared (disabled). Conversely, bits that are clear in the mask allow the corresponding file mode bits to be enabled in newly created files.

For example, the mask **u=rwx,g=rx,o=rx** (octal **022**) disables group and other write permissions. As a result, files normally created with a file mode shown by the **ls -l** command as **-rwxrwxrwx** (octal **777**) become mode **-rwxr-xr-x** (octal **755**); while files created with file mode **-rw-rw-rw-** (octal **666**) become mode **-rw-r--r--** (octal **644**).

Note that the file creation mode mask does not affect the set-user-id, set-group-id, or "sticky" bits.

The file creation mode mask is also used by the **chmod** command (see *chmod(1)*).

Since **umask** affects the current shell execution environment, it is generally provided as a shell regular built-in (see **DEPENDENCIES**).

If **umask** is called in a subshell or separate utility execution environment, such as one of the following:

```
(umask 002)
nohup umask ...
find . -exec umask ...
```

it does not affect the file mode creation mask of the calling environment.

The default mask is **u=rwx,g=rwx,o=rwx** (octal **000**).

| u |

**RETURN VALUE**

**umask** exits with one of the following values:

- 0 The file mode creation mask was successfully changed or no *mask* operand was supplied.
- >0 An error occurred.

**EXAMPLES**

In these examples, each line show an alternate way of accomplishing the same task.

Set the **umask** value to produce read and write permissions for the file's owner and read permissions for all others (**ls -l** displays **-rw-r--r--** on newly created files):

```
umask u=rwx,g=rx,o=rx symbolic mode
umask a=rx,u+w symbolic mode
umask 022 numeric mode
```

Set the **umask** value to produce read, and write permissions for the file's owner, read-only for others users in the same group, and no access to others (**-rw-r-----**):

```
umask a-rwx,u+rw,g+r symbolic mode
umask u=rw,g=r,o= symbolic mode
umask ,u+rw,g+r symbolic mode
umask 137 numeric mode
```

Set the **umask** value to deny read, write, and execute permissions to everyone (**-----**):

```
umask a= symbolic mode
umask 777 numeric mode
```

Add the write permission to the current mask for everyone (there is no equivalent numeric mode):

```
umask a+w symbolic mode
```

**WARNINGS**

If you set a mask that prevents read or write access for the user (owner), many programs, such as editors, that create temporary files will fail because they cannot access the file data.

**DEPENDENCIES**

The **umask** command is implemented both as a separate executable file (**/usr/bin/umask**) and as built-in shell commands.

**POSIX Shell and Separate File**

All features are supported (see *sh-posix(1)*). The numeric mask display uses a minimum of two digits.

**Korn Shell**

The **-s** option is not supported in the Korn shell built-in command (see *ksh(1)*). The numeric mask display uses a minimum of two digits.

**u**

**C Shell**

The **-s** option and symbolic mask values are not supported in the C shell built-in command (see *csh(1)*). The numeric mask display uses a minimum of one digit.

**SEE ALSO**

*chmod(1)*, *csh(1)*, *ksh(1)*, *sh-posix(1)*, *sh(1)*, *chmod(2)*, *creat(2)*, *umask(2)*.

**STANDARDS CONFORMANCE**

**umask**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**NAME**

umodem - XMODEM-protocol file transfer program

**SYNOPSIS**

**umodem** [*options*] *files* ...

**umodem -c**

**DESCRIPTION**

**umodem** is a file transfer program that incorporates the well-known XMODEM protocol used on CP/M systems and on the HP 110 portable computer.

**Options**

**umodem** recognizes the following options and command-line arguments:

- 1** (one) Employ TERM II FTP 1.
- 3** Enable TERM FTP 3 (CP/M UG).
- 7** Enable 7-bit transfer mask.
- a** Turn on ARPA Net flag.
- c** Enter command mode.
- d** Do not delete **umodem.log** before starting.
- l** (ell) Turn on entry logging.
- m** Allow overwriting of files.
- p** Print all messages.
- r[t|b]** Receive file. Specify **t** for text, or **b** for binary.
- s[t|b]** Send file. Specify **t** for text, or **b** for binary.
- y** Display file status only.
- files* Name of file or files to be transferred.

**WARNINGS**

When a binary file is transferred, the target file may have some extra bytes added at the end. This is due to the limitation of the underlying **XMODEM** protocol.

**EXAMPLES**

Receive a text file:

**umodem -rt7 file**

Receive a binary file:

**umodem -rb file**

Send a text file:

**umodem -st7 file**

Send a binary file:

**umodem -sb file**

**AUTHOR**

**umodem** is in the public domain.

**SEE ALSO**

cu(1), kermi(1), uucp(1).

| **u** |

**NAME**

uname - display information about computer system; set node name (system name)

**SYNOPSIS**

uname [-ailmnrsv]

uname [-S *nodename*]

**DESCRIPTION**

In the first form above, the **uname** command displays selected information about the current computer system, derived from the **utsname** structure (see *uname(2)*).

In the second form, **uname** sets the node name (system name) that is used in the **utsname** structure.

**Options**

**uname** recognizes the options listed below. If you enter several options, the output is always in the order shown for the **-a** option.

|                           |                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| none                      | Equivalent to <b>-s</b> .                                                                                                                                                                                                                                                                                                                                     |
| <b>-a</b>                 | Display the options below in the following order, separated by blanks.<br><b>-s -n -r -v -m -i -l</b>                                                                                                                                                                                                                                                         |
| <b>-i</b>                 | Display the machine identification number (or the node name, if the machine identification number cannot be determined). This option cannot display the unique machine identification number. For getting the unique machine identification number refer to the <b>getconf</b> command or <b>confstr</b> call. See <i>getconf(1)</i> and <i>confstr(3C)</i> . |
| <b>-l</b>                 | Display the license level of the operating system. 128-, 256-, and unlimited-user licenses are shown as <b>unlimited-user license</b> .                                                                                                                                                                                                                       |
| <b>-m</b>                 | Display the machine hardware and model names. On Itanium(R)-based systems, this option always displays <b>ia64</b> . See <i>WARNINGS</i> .                                                                                                                                                                                                                    |
| <b>-n</b>                 | Display the node name (system name) by which the system is usually known in a UUCP network. See <i>WARNINGS</i> .                                                                                                                                                                                                                                             |
| <b>-r</b>                 | Display the current release level of the operating system.                                                                                                                                                                                                                                                                                                    |
| <b>-s</b>                 | Display the name of the operating system. On standard HP-UX systems, this option always displays <b>HP-UX</b> .                                                                                                                                                                                                                                               |
| <b>-v</b>                 | Display the current version level of the operating system.                                                                                                                                                                                                                                                                                                    |
| <b>-S <i>nodename</i></b> | Change the node name (system name) to <i>nodename</i> . <i>nodename</i> is restricted to <b>UTSLEN-1</b> characters (see <i>uname(2)</i> ). See <i>WARNINGS</i> . Only users with appropriate privileges can use the <b>-S</b> option.                                                                                                                        |

**u****EXAMPLES**

When you execute the command **uname -a**, it produces output like the following:

```
HP-UX myhost A.09.01 C 9000/750 2015986034 32-user license
```

The displayed fields are interpreted as follows:

|                        |                                                                          |
|------------------------|--------------------------------------------------------------------------|
| <b>HP-UX</b>           | The operating system name (option <b>-s</b> ).                           |
| <b>myhost</b>          | The UUCP network system name by which the system is known ( <b>-n</b> ). |
| <b>B.11.00</b>         | The operating system release identifier ( <b>-r</b> ).                   |
| <b>A</b>               | The operating system version identifier ( <b>-v</b> ).                   |
| <b>9000/889</b>        | The machine and model numbers ( <b>-m</b> ).                             |
| <b>2015986034</b>      | The machine identification number ( <b>-i</b> ).                         |
| <b>32-user license</b> | The operating system license level ( <b>-l</b> ).                        |

**WARNINGS**

It is recommended that the **model** command or the **getconf** command be used to obtain the model name, since future model names may not be compatible with **uname**. See *model(1)* and *getconf(1)*.

Many types of networking services are supported on HP-UX, each of which uses a separately assigned system name and naming convention. To ensure predictable system behavior, it is essential that system names (also called host names or node names) be assigned in such a manner that they do not create conflicts when the various networking facilities interact with each other.

The system does not rely on a single system name in a specific location, partly because different services use dissimilar name formats as explained below. The **hostname** and **uname** commands assign system names as follows:

| <b>Node Name</b> | <b>Command</b>              | <i>name</i> <b>Format</b>       | <b>Used By</b>                   |
|------------------|-----------------------------|---------------------------------|----------------------------------|
| Internet name    | <b>hostname</b> <i>name</i> | <i>sys</i> [ <i>.x.y.z...</i> ] | ARPA and NFS Services            |
| UUCP name        | <b>uname -s</b> <i>name</i> | <i>sys</i>                      | <b>uucp</b> and related programs |

where *sys* represents the assigned system name. It is *strongly* recommended that *sys* be identical for all commands and locations and that the optional *.x.y.z...* follow the specified notation for the particular ARPA/NFS environment.

Internet names are also frequently called host names or domain names (which are different from NFS domain names). Refer to *hostname(5)* for more information about Internet naming conventions.

Whenever the system name is changed in any file or by the use of any of the above commands, it should also be changed in all other locations as well. Other files or commands in addition to those above (such as **/etc/uucp/Permissions** if used to circumvent **uname**, for example) may contain or alter system names. To ensure correct operation, they should also use the same system name.

System names are normally assigned by the **/sbin/init.d/hostname** script at start-up, and should not be altered elsewhere.

**SEE ALSO**

getconf(1), hostname(1), model(1), setuname(1M), gethostname(2), sethostname(2), uname(2), hostname(5).

**STANDARDS CONFORMANCE**

**uname**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**NAME**

unget - undo a previous get of an SCCS file

**SYNOPSIS**

**unget** [-r *SID*] [-s] [-n] *file* ...

**DESCRIPTION**

The **unget** command undoes the effect of a **get -e** done prior to creating the intended new delta. If *file* is a directory name, **unget** treats each file in the directory as a file to be processed, except that non-SCCS files and unreadable files are silently ignored. If **-** is specified for *file*, the standard input is read with each line being taken as the name of an SCCS file to be processed. Refer to *sact*(1), which describes how to determine what deltas are currently binding for an s-file.

**Options**

**unget** recognizes the following options and command-line arguments. Options and arguments apply independently to each named *file*.

- r** *SID* Uniquely identifies which delta is no longer wanted (this would have been specified by **get** as the "new delta"). This option is necessary only if two or more outstanding **gets** for editing on the same SCCS file were done by the same person (login name). **unget** prints a diagnostic message if the specified *SID* is ambiguous, or if it is required but not present on the command line (see *sact*(1)).
- s** Silent option. Suppress printing the intended delta's *SID* on the standard output.
- n** Retain the file deposited in the current directory by the previous **get**. Normally this file is removed by **unget**.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **unget** behaves as if all internationalization variables are set to "C". See *environ*(5).

**International Code Set Support**

Single-byte and multi-byte character code sets are supported.

**DIAGNOSTICS**

Use *sccshelp*(1) for explanations.

**u****WARNINGS**

Only the user who did the corresponding **get -e** can execute **unget**. Any other user must either use **su** to change user ID to that user (see *su*(1)), or edit the *p-file* directly (which can be done either by the *s-file* owner or a user who has appropriate privileges).

**FILES**

*p-file* See *delta*(1).  
*g-file* See *delta*(1).

**SEE ALSO**

*delta*(1), *get*(1), *sccshelp*(1), *sact*(1).

**STANDARDS CONFORMANCE**

**unget**: SVID2, SVID3, XPG2, XPG3, XPG4

**NAME**

unifdef - remove preprocessor lines

**SYNOPSIS**

```
unifdef [-clt] [[-D sym] [-U sym] [-iD sym] [-iU sym]] ... [file]
```

**DESCRIPTION**

**unifdef** simulates some of the actions of **cpp** in interpreting C language preprocessor command lines (see *cpp*(1)). For **unifdef**, a valid preprocessor command line contains as its first character a **#** and one of the following keywords: **ifdef**, **ifndef**, **if**, **else**, or **endif**. The **#** character and its associated keyword must appear on the same line, but they can be separated by spaces, tabs, and commented text. When appropriate, the portions of code surrounded by and including the targeted preprocessor directives are removed, and the resultant text is written to the standard output.

Unlike **cpp**, **unifdef** does not insert included files, interpret macros, or strip comment lines. This means, among other things, that **#define** and **#undef** macros occurring within the input text are not interpreted.

Since **unifdef** is language-independent, it can be used for processing source files for languages other than the C language. For example, **unifdef** can be used on FORTRAN language source files, provided the C language preprocessor commands are used.

**Options**

**unifdef** recognizes the following command-line options:

- c** Complement the normal behavior by printing only the rejected lines.
- iD*sym*** Ignore text delimited by **#ifdef *sym***. In other words, text that would otherwise be affected by some action is not touched when found within the context of a preprocessor command using *sym*.
- iU*sym*** Ignore text delimited by **#ifndef *sym***.
- l** Replace rejected lines with blank lines in the text written to the standard output.
- t** Treat the input source as plain text. C-language comment and quoting constructs are not recognized.
- D*sym*** Define symbol *sym*.
- U*sym*** Cause symbol *sym* to be undefined.

**RETURN VALUE**

The **unifdef** command returns the following exit values:

- 0 Output is an exact copy of the input.
- 1 Output is not an exact copy of the input.
- 2 The **unifdef** command fails. The failure might be due to a premature EOF or to an inappropriate **else**, **elif**, or **endif**.

**EXAMPLES**

Assume file **foo.f** contains the following:

```
PROGRAM TEST1
 INTEGER I, J
#ifdef ANSI77
 DO I=1,10
#else
 DO 100 I=1,10
#endif
 J=J+1
#ifdef (DEBUG) || defined (TEST)
 PRINT *,J
#endif
#ifdef ANSI77
 ENDDO
#else
```

```
 100 CONTINUE
#endif
 END
```

The command sequence:

```
unifdef -DANSI77 -UDEBUG -DTEST foo.f > /tmp/foo.f
```

produces the following result in file `/tmp/foo.f`:

```
PROGRAM TEST1
INTEGER I, J
DO I=1,10
J=J+1
PRINT *,J
ENDDO
END
```

#### WARNINGS

Any symbol name defined in the file must be specified in the `unifdef` command line; otherwise, `unifdef` will ignore the line.

#### AUTHOR

`unifdef` was developed in the public domain.

#### SEE ALSO

`cpp(1)`.

| **u** |



**NAME**

uniq - report repeated lines in a file

**SYNOPSIS**

**uniq** [-udc [-f *fields*] [-s *chars*] [*input\_file* [*output\_file*]]

**DESCRIPTION**

**uniq** reads the input text file *input\_file*, comparing adjacent lines, and copies the result to *output\_file*. If *input\_file* is not specified, the standard input and standard output are used. If *input\_file* is specified, but *output\_file* is not, results are printed to standard output. *input\_file* and *output\_file* must not be the same file.

**Line-Comparison Options**

**uniq** recognizes the following options when comparing adjacent lines:

- u** Print *only* those lines that are *not* repeated in the original file.
- d** Print *one* copy only of each repeated line in the input file.
- c** Generate an output report in default style except that each line is preceded by a count of the number of times it occurred. If this option is specified, the **-u** and **-d** options are ignored if either or both are also present.

If none of the options **u**, **d**, or **c** are present, **uniq** prints the results of the union of the **-u** and **-d** options, producing a copy of the original input file with the second and succeeding copies of any repeated lines removed. (Note that repeated lines must be adjacent in order to be found — see *sort*(1)).

**Field-Skip Options**

Two options are provided for skipping an initial portion of each line when making comparisons:

- f *fields*** Ignore the first *fields* fields, together with any blanks before each. *fields* is a positive decimal integer. A field is defined as a string of non-space, non-tab characters separated by tabs and/or spaces from its neighbors.
- s *chars*** Ignore the first *chars* characters. *chars* is a positive decimal integer. Each line in the input is assumed to be terminated with a new line character for purposes of comparison. Fields are skipped before characters.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_COLLATE** must be equal to the value it had when the input files were sorted.

**LC\_CTYPE** determines the interpretation of text within files as single- and/or multi-byte characters, and defines a space character when the **-f** or **-s** option is used.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_COLLATE**, **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **uniq** behaves as if all internationalization variables are set to "C". See *environ*(5).

| u |

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**RETURN VALUE**

Exit values are:

- 0** Successful completion.
- >0** Error condition occurred.

**AUTHOR**

**uniq** was developed by OSF and HP.

**SEE ALSO**

*comm*(1), *sort*(1).

**STANDARDS CONFORMANCE**

**uniq:** SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| u |**

**NAME**

units - conversion program

**SYNOPSIS**

**units** [- *file*]

**DESCRIPTION**

**units** converts quantities expressed in various standard scales to their equivalents in other scales. It works interactively as follows:

| System Prompt | User Response |
|---------------|---------------|
| You have:     | inch          |
| You want:     | cm            |

The system responds with two factors; one used if multiplying (preceded by \*), the other if dividing (preceded by /):

```
* 2.540000e+00
/ 3.937008e-01
```

After providing the conversion factors, **units** prompts for the next set of values. To terminate **units**, press the interrupt character as defined for your login.

A quantity is specified as a multiplicative combination of units optionally preceded by a numeric multiplier. Powers are indicated by suffixed positive integers, and division by the usual sign:

| System Prompt | User Response    |
|---------------|------------------|
| You have:     | 15 lbs force/in2 |
| You want:     | atm              |

The system responds with:

```
* 1.020689e+00
/ 9.797299e-01
```

**units** only does multiplicative scale changes; thus it can convert Kelvin to Rankine, but not Celsius to Fahrenheit. Most familiar units, abbreviations, and metric prefixes are recognized, together with a generous leavening of exotica and a few constants of nature including:

|              |                                         |
|--------------|-----------------------------------------|
| <b>pi</b>    | ratio of circumference to diameter      |
| <b>c</b>     | speed of light                          |
| <b>e</b>     | charge on an electron                   |
| <b>g</b>     | acceleration of gravity                 |
| <b>force</b> | same as <b>g</b> ,                      |
| <b>mole</b>  | Avogadro's number,                      |
| <b>water</b> | pressure head per unit height of water, |
| <b>au</b>    | astronomical unit.                      |

Units must be provided in lowercase only. **pound** is not recognized as a unit of mass; **lb** is. Compound names are run together, (e.g., **lightyear**). British units that differ from their U.S. counterparts are prefixed thus: **brgallon**. For a complete list of units, examine the file:

**/usr/share/lib/unittab**

An alternate unit database file can be specified for use with the - *file* option. **units** looks in this file rather than the default **/usr/share/lib/unittab** for the table of conversions. This must be in the same format as **/usr/share/lib/unittab**. This is useful in defining your own units and conversions.

**WARNINGS**

The monetary exchange rates are out of date.

**FILES**

**/usr/share/lib/unittab**

**NAME**

uptime, w - show how long system has been up, and/or who is logged in and what they are doing

**SYNOPSIS**

**uptime** [-hlsuw] [*user*]

**uptime** [-p[*pset\_list* ]]

**w** [-hlsuw] [*user*]

**w** [-p[*pset\_list* ]]

**DESCRIPTION**

**uptime** prints the current time, the length of time the system has been up, the number of users logged on to the system, and the average number of jobs in the run queue over the last 1, 5, and 15 minutes for the active processors.

**w** is linked to **uptime** and prints the same output as **uptime -w**, displaying a summary of the current activity on the system.

**Options**

**uptime** and **w** recognize the following options:

- h** Suppress the first line and the heading line. This option should not be used with the **-u** option. This option assumes the use of the **-w** option to **uptime**.
- l** Use long output. This option assumes the use of the **-w** option to **uptime**.
- p** Print the current time, the length of time the system has been up, and the number of users logged on to the system, in the first line of the output. The load averages over the last 1, 5, and 15 minutes for the **processor sets** (psets) given in the command line, *pset\_list*, are displayed in the subsequent lines. If no arguments are given, the load averages are displayed for all the psets in the system.  
  
If pset id of an empty pset is given in the command line, corresponding message will be displayed. If the kernel does not have the pset capability, uptime gives an error. The **-p** option can not be used along with other options.
- s** Use the short form of output for displaying terminal information. The terminal name is abbreviated; the login time and CPU times are suppressed.
- u** Print only the first line describing the overall state of the system. This is the default for the **uptime** command.
- w** Print a summary of the current activity on the system for each user. This is the default for the **w** command.

**EXAMPLES**

The command:

```
uptime
```

produces text resembling the following:

```
2:30pm up 14days, 2:39, 33 users, load average: 1.71, 1.88, 1.80
```

depending upon the current status of the system.

The command:

```
uptime -p
```

gives the load average of all the psets in the system. If 0, 94,95,and 97 are the psets existing in the system, the output will look like the following:

```
7:59pm up 11 days, 6:25, 12 users
load average for pset 0 : 1.06, 1.15, 1.15
load average for pset 94 : 0.00, 0.00, 0.00
load average for pset 95 : 0.00, 0.00, 0.00
load average for pset 97 : 1.00, 1.00, 1.01
```

The command:

```
uptime -p94,95 or uptime -p"94 95"
```

gives the output which looks like the following, if 94 and 95 are valid pset ids:

```
7:59pm up 11 days, 6:25, 12 users
load average for pset 94 : 0.00, 0.00, 0.00
load average for pset 95 : 0.00, 0.00, 0.00
```

**AUTHOR**

**uptime** was developed by the University of California, Berkeley and HP.

**| u |**

**NAME**

users - compact list of users who are on the system

**SYNOPSIS**

**users**

**DESCRIPTION**

**users** lists the login names of the users currently on the system in a compact, one-line format.

The login names are sorted in ascending collation order (see Environment Variables below).

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_COLLATE** determines the order in which the output is sorted.

If **LC\_COLLATE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default. If **LANG** is not specified or is set to the empty string, a default of “C” (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, *users* behaves as if all internationalization variables are set to “C” (see *environ(5)*).

**AUTHOR**

**users** was developed by the University of California, Berkeley and HP.

**FILES**

**/etc/utmp**

**utmps** database      **who** database

**SEE ALSO**

who(1), utmpd(1M), gettutsent(3C).

| **u** |

**NAME**

uucp, uulog, uuname, uutry - UNIX system to UNIX system copy

**SYNOPSIS**

**uucp** [*options*] *source\_files destination\_file*

**uulog** **-f** *system* [**-x**] [**-number**]

**uulog** [**-s** *system*] ... [**-x**] [**-number**]

**uuname** [**-l**]

**uutry** **-r1 -s** *system* [**-x** *debug\_level*]

**DESCRIPTION**

**uucp** copies files named by the *source\_files* argument to the destination identified by the *destination\_file* argument. When copying files to or from a remote system, *source\_files* and *destination\_file* can be a path name on the local system, or have the form:

*system\_name !path\_name*

where *system\_name* is the name of a remote system in a list of system names known to **uucp**. When copying files to (but not from) a remote system, *system\_name* can also be a chained list of remote system names such as:

*system\_name !system\_name !... !system\_name !path\_name*

in which case an attempt is made to send the file, via the specified route, to the destination. Care should be taken to ensure that intermediate nodes in the route are configured to forward information (see WARNINGS for restrictions).

The shell metacharacters `?`, `*` and `[ . . . ]` appearing in *path\_name* are expanded on the appropriate system.

*path\_name* can be one of:

- A full path name.
- A path name preceded by `~user` where *user* is a login name on the specified system and `~user` is replaced by that user's login directory. (If an invalid login is specified, the default public directory (`/var/spool/uucppublic`) is used instead.
- A path name preceded by `~/destination` where *destination* is appended to `/var/spool/uucppublic`.

NOTE: This destination is treated as a file name unless more than one file is being transferred by this request or the destination is already a directory. To ensure that *destination* is a directory, append a `/` to the destination argument. For example, `~/dan/` as the destination argument causes directory `/var/spool/uucppublic/dan` to be created if it does not already exist, and places the requested file or files in that directory.

- Anything else is prefixed by the current directory.

If an erroneous path name is specified for the remote system, the copy fails. If *destination\_file* is a directory, the file-name part of the *source\_file* argument is used.

**uucp** preserves execute permissions across the transmission and sets read and write permissions to 0666 (see `chmod(2)` and Access Control Lists below).

**Options**

**uucp** recognizes the following options:

- |                |                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c</b>      | Do not copy local file to the spool directory for transfer to the remote machine (default).                                                                                |
| <b>-C</b>      | Force the copy of local files to the spool directory for transfer.                                                                                                         |
| <b>-d</b>      | Make all necessary directories for the file copy (default).                                                                                                                |
| <b>-f</b>      | Do not make intermediate directories for the file copy.                                                                                                                    |
| <b>-ggrade</b> | <i>grade</i> is a single letter or number. A lower ASCII sequence value for <i>grade</i> causes the job to be transmitted earlier in a given conversation between systems. |

- j** Output the ASCII job identification string on standard output. This job identification can be used by **uustat** to obtain the status or terminate a job (see **uustat(1)**).
- mfile** Send mail to the requester when the copy is completed.
- nuser** Notify *user* on the remote system that a file was sent.
- r** Do not start the file transfer; just queue the job.
- sfile** Report status of the transfer to *file*. Note that *file* must be a full path name.
- xdebug\_level** Produce debugging on standard output. *debug\_level* is a number between 0 and 9; higher numbers give more information.

**uulog**

**uulog** queries a log file of **uucp** transactions in a file `/var/uucp/.Log/uucico/system`.

The following options cause **uulog** to print logging information:

- s system** Print information about work involving *system*.
- f system** Do a **tail -f** (see **tail(1)**) of the file transfer log for *system*.

Other options used in conjunction with the **-s** and **-f** options above are:

- x** Search for the given system in the `/var/uucp/.Log/uuxqt/system` file instead of in the *uucico* log file.
- number** Do a **tail(1)** command of *number* lines.

**uuname**

**uuname** lists the **uucp** names of known systems. **uuname -l** returns the local system's default name.

**uutry**

**uutry** tests for the successful login to the remote system. This is executed for checking communication channel.

- r1** Starts **uutry** in the MASTER mode. The default is SLAVE mode.
- s system** Do work only for the system specified by *system*. If there is no work for *system* on the local spool directory, initiate a connection to *system* to determine if *system* has work for the local system. This option must be used if **-r1** is specified.
- x debug\_level** Produce debugging on standard output. *debug\_level* is a number between 0 and 9; higher numbers give more information.

When started by a local program, **uutry** is considered the MASTER and attempts a connection to a remote system. If **uutry** is started by a remote system, it is considered to be in SLAVE mode.

**u****Access Control Lists (ACLs)**

A file's optional ACL entries are not preserved across **uucp** transmission. Instead, new files have a summary of the access modes (as returned in **st\_mode** by **stat()**; see **stat(2)**).

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_TIME** determines the format and contents of date and time strings displayed by **uucp** and **uulog** commands.

**LANG** determines the language in which messages are displayed by **uucp** and **uuname** commands.

If **LC\_TIME** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see **lang(5)**) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **uucp**, **uulog**, and **uuname** behave as if all internationalization variables are set to "C". See **environ(5)**.

**International Code Set Support**

Single- and multi-byte character code sets are supported with the exception that multi-byte-character file names are not supported.



**WARNINGS**

The domain of remotely accessible files can (and for obvious security reasons, usually should) be severely restricted. In most cases, you cannot fetch files by path name from a remote system. Ask a responsible person on the remote system to send them to you. For the same reasons, you probably cannot send files to arbitrary path names. As distributed, remotely accessible files are those whose names begin `/var/spool/uucppublic` (equivalent to `~/`).

All files received by **uucp** are owned by **uucp**.

The **-m** option only works when sending files or when receiving a single file. Receiving multiple files specified by special shell characters `? * [ ... ]` does not activate the **-m** option.

Protected files and files in protected directories owned by the requester can be sent by **uucp**. However, if the requester is root and the directory is not searchable by **other** or the file is not readable by **other**, the request fails.

**uutry** should be executed only with request file that exists in the directory `/var/spool/uucp/system_name/`.

**FILES**

|                                    |                                            |
|------------------------------------|--------------------------------------------|
| <code>/etc/uucp</code>             | configuration files                        |
| <code>/var/uucp</code>             | log and error files                        |
| <code>/var/spool/uucp</code>       | spool directories                          |
| <code>/var/spool/locks</code>      | lock files                                 |
| <code>/var/spool/uucppublic</code> | public directory for receiving and sending |

**SEE ALSO**

mail(1), uux(1), uucico(1M), chmod(2), stat(2), acl(5).

Tim O'Reilly and Grace Todino,  
*Managing UUCP and Usenet*, O'Reilly & Associates, Inc. USA.

Grace Todino and Dale Dougherty,  
*Using UUCP and Usenet*, O'Reilly & Associates, Inc. USA.

**STANDARDS CONFORMANCE**

**uucp**: SVID2, SVID3, XPG2, XPG3

**uulog**: SVID2, SVID3, XPG2, XPG3

**uuname**: SVID2, SVID3, XPG2, XPG3

| u |

**NAME**

uuencode, uudecode - encode/decode a binary file for transmission by mailer

**SYNOPSIS**

**uuencode** [*source*] *remotedest*

**uudecode** [*file*]

**DESCRIPTION**

**uuencode** and **uudecode** can be used to send a binary file to another machine by means of such services as *elm*(1), *mailx*(1), or *uucp*(1) (see *elm*(1), *mailx*(1), and *uucp*(1)).

**uuencode** takes the named source file (default standard input) and produces an encoded version on the standard output. The encoding uses only printing ASCII characters, includes the original mode of the input file, and preserves the value of the *remotedest* argument which is the intended name for the file when it is restored later on the remote system.

**uudecode** reads an encoded file, ignores any leading and trailing lines added by mailers, and recreates the original file with the specified mode and name.

The encoded file is an ordinary ASCII text file and can be edited with any text editor to change the mode or remote name.

**EXAMPLES**

To encode and send a compiled program **foo** to user **friend**:

```
uuencode foo foo | mailx -s 'new program' friend
```

After receiving the mail message, user **friend** can decode the program by first saving the message in a file **foo.mail** and executing the command:

```
uudecode foo.mail
```

**WARNINGS**

The file is expanded by 35% (three bytes become four plus control information) causing it to take longer to transmit.

The user on the remote system who is invoking **uudecode** (often **uucp**) must have write permission for the specified file.

If an encoded file has the same name as the destination name specified in it, **uudecode** starts overwriting the encoded file before decoding is completed.

**SEE ALSO**

*elm*(1), *mail*(1), *mailx*(1), *shar*(1), *uucp*(1), *uux*(1), *uuencode*(4).

**STANDARDS CONFORMANCE**

**uuencode**, **uudecode**: XPG4, POSIX.2

| **u** |

## (TO BE OBSOLETE)

**NAME**

uupath, mkuupath - access and manage the pathalias database

**SYNOPSIS**

**uupath** [-f *pathsfile*] *mailaddress*

**mkuupath** [-v] *pathsfile*

**DESCRIPTION**

The **uucp** commands, including **uupath** and **mkuupath**, are targeted for removal from HP-UX; see the **WARNINGS** below.

**uupath** provides electronic message routing by expanding a simple UUCP address into a full UUCP path (see *uucp(1)*). For example, *host!user* could be expanded into *hostA!hostB!host!user*.

**uupath** expands an address by parsing *mailaddress* for the dominant host (see below) and looking up the host in the appropriate **pathalias** database (see *pathalias(1)*). If the host is found in the database, the expanded address is written to the standard output. If the host is not found, **uupath** writes the original address to the standard output and returns an exit status of 1. **uupath** expects *mailaddress* to be in UUCP format (*host!...!hostZ!user*) or ARPANET format (*user@host*).

The **-f** option opens the **pathalias** database based on *pathsfile* rather than the default database based on `/usr/lib/mail/paths`. This database must be a database created by **mkuupath**, consisting of the two files *pathsfile.dir* and *pathsfile.pag*.

The dominant host is the left-most UUCP host in *mailaddress*. If no UUCP host is found (no **!** is in the address), **uupath** assumes that the address is in the simple ARPANET format *user@host*. If the address does not match either format, **uupath** writes the original address to the standard output and returns an exit status of 1.

**mkuupath** constructs a mail routing database by using the *pathsfile* data file obtained from **pathalias** (see *pathalias(1)*) as input. The recommended *pathsfile* location is `/usr/lib/mail/paths`, because this is the default database used by **uupath**. The database files *pathsfile.dir* and *pathsfile.pag* are created by **mkuupath**. If these files already exist, they must be removed prior to running **mkuupath**.

The **-v** option specifies verbose mode, which writes a line to the standard output for each entry written to the database.

**DIAGNOSTICS**

**uupath** returns an exit status of 1 and writes the original *mailaddress* to the standard output if the address is not found or is incorrectly formatted. **uupath** returns an exit status of 2 and prints a diagnostic message if the database files are not accessible, or if improper parameters are given. Otherwise, **uupath** returns an exit status of 0.

If the database files *pathsfile.dir* and *pathsfile.pag* already exist prior to running **mkuupath**, the message **mkuupath: pathsfile.dir: File exists** is displayed. These files must be removed before running **mkuupath**.

**WARNINGS**

Use of **uucp** commands, including **uupath** and **mkuupath**, is discouraged because they are targeted for removal from HP-UX. Use *ftp(1)* or *rcp(1)* instead.

**AUTHOR**

**uupath** was developed by University of California, Berkeley.

**FILES**

`/usr/lib/mail/paths`  
`/usr/lib/mail/paths.dir`  
`/usr/lib/mail/paths.pag`

**SEE ALSO**

*pathalias(1)*, *uucp(1)*.

**NAME**

uustat - uucp status inquiry and job control

**SYNOPSIS**

```
uustat -a
uustat -m
uustat -p
uustat -q
uustat -k jobid]
uustat -r jobid]
uustat [-sys] [-user]
```

**DESCRIPTION**

**uustat** displays the status of, or cancels, previously specified **uucp** commands, or provide general status on **uucp** connections to other systems (see *uucp(1)*). Only one of the following options can be specified with **uustat** per command execution:

- a** Output all jobs in queue.
- m** Report the status of accessibility of all machines.
- p** Execute a **ps -flp** for all the process IDs that are in the lock files.
- q** List the jobs queued for each machine. If a status file exists for the machine, its date, time and status information are reported. In addition, if a number appears in ( ) next to the number of **C** or **X** files it is the age in days of the oldest **C.** or **X.** file for that system. The Retry field is the number of hours until the next possible call. The Count field is the number of failure attempts. Note that for systems with a moderate number of outstanding jobs, this could take 30 seconds or more of real time to execute. As an example of the output produced by **uustat -q** :

```
eagle 3C 04/07-11:07 NO DEVICES AVAILABLE
mh3bs3 2C 07/07-10:42 SUCCESSFUL
```

The above output tells how many command files are waiting for each system. Each command file can have zero or more files to be sent (a command file with no files to be sent causes the **uucp** system to call the remote system and see if work is waiting). The date and time refer to the previous interaction with the system followed by the status of interaction.

- k *jobid*** Kill the **uucp** request whose job identification is *jobid*. The killed **uucp** request must belong to the person issuing the **uustat** command unless the command is executed by the super-user.
- r *jobid*** Rejuvenate *jobid*. The files associated with *jobid* are touched so that their modification time is set to the current time. This prevents the cleanup daemon from deleting the job until the jobs modification time reaches the limit imposed by the cleanup daemon.

The following options can be used singly or together but cannot be used with the options listed above:

- s *sys*** Report the status of all **uucp** requests for remote system *sys*.
- u *user*** Report the status of all **uucp** requests issued by *user*.

Output for both the **-s** and **-u** options has the following format:

```
eaglen0000 4/07-11:01:03 (POLL)
eagleN1bd7 4/07-11:07 S eagle dan 522 /usr/dan/A
eagleC1bd8 4/07-11:07 S eagle dan 59 D.3b2a12cd4924
 4/07-11:07 S eagle dan rmail mike
```

With the **-s** and **-u** options, the first field is the *jobid* of the job. This is followed by the date and time. The next field is either an **S** or **R**, depending on whether the job is to send or request a file. The next field is the destination system name. This is followed by the user ID of the user who queued the job. The next field contains the size of the file, or in the case of a remote execution the name of the command (such as **rmail** which is the command used for remote mail). When the size appears in this field, the file name is also given. This can either be the name given by the user or an internal name (such as **D.3b2alce4924**) that is created for data files associated with remote execution (**rmail** in this example).

When no options are given, **uustat** outputs the status of all **uucp** requests issued by the current user. The format used is the same as with the **-s** or **-u** options.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_TIME** determines the format and contents of date and time strings.

**LANG** determines the language in which messages are displayed.

If **LC\_TIME** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **uustat** behaves as if all internationalization variables are set to "C". See *environ(5)*.

## FILES

`/var/spool/uucp/*` spool directories

## SEE ALSO

`uucp(1)`.

Tim O'Reilly and Grace Todino,

*Managing UUCP and Usenet*, O'Reilly & Associates, Inc. USA.

Grace Todino and Dale Dougherty,

*Using UUCP and Usenet*, O'Reilly & Associates, Inc. USA.

## STANDARDS CONFORMANCE

**uustat**: SVID2, SVID3, XPG2, XPG3, XPG4

| u |

**NAME**

uuto, uupick - public UNIX system to UNIX system file copy

**SYNOPSIS**

**uuto** [*options*] *source-files destination*

**uupick** [**-s** *system*]

**DESCRIPTION**

**uuto** sends *source-files* to *destination*. **uuto** uses the **uucp** facility to send files (see *uucp(1)*), while allowing the local system to control the file access. A source-file name is a path name on your machine. Destination has the form:

*system !user*

where *system* is taken from a list of system names that **uucp** knows about (see **uuname** in *uucp(1)* manual entry). *user* is the login name of someone on the specified system.

**uuto** recognizes the following options:

- p** Copy the source file into the spool directory immediately, and send the copy.
- m** Send mail to the requester when the copy is complete.

The files (or sub-trees if directories are specified) are sent to *PUBDIR* on *system*, where *PUBDIR* is the UUCP public directory (*/var/spool/uucppublic*). Specifically the files are sent to

*PUBDIR/receive/user/mysystem/files*.

The recipient is notified by electronic mail when the files arrive.

**uupick** accepts or rejects the files transmitted to the recipient. Specifically, **uupick** searches *PUBDIR* for files destined for the user. For each entry (file or directory) found, the following message is printed on the standard output:

**from system :** [**file** *file-name*] [**dir** *dirname*] ?

**uupick** then reads a line from the standard input to determine the disposition of the file:

- <new-line> Go on to next entry.
- d** Delete the entry.
- m** [*dir*] Move the entry to named directory *dir* (current directory is default). Note that, if the current working directory is desired for *dir*, do *not* specify any parameter with **m**. A construction such as **m.** is erroneous, and results in loss of data.
- a** [*dir*] Same as **m** except move all the files sent from *system*.
- p** Print the contents of the file.
- q** Stop.
- EOT (control-D) Same as **q**.
- !command** Escape to the shell to execute *command*.
- \*** Print a command summary.

**uupick** invoked with the **-ssystem** option searches only the *PUBDIR* for files sent from *system*.

**WARNINGS**

To send files that begin with a dot (such as **.profile**) the filename must contain a corresponding dot. For example: **.profile**, **.prof\***, and **.profil?** are correct, whereas **\*prof\*** and **?profile** are incorrect.

**FILES**

*PUBDIR* */var/spool/uucppublic* public directory

**SEE ALSO**

mail(1), uuclean(1M), uucp(1), uustat(1), uux(1).

Tim O'Reilly and Grace Todino,

*Managing UUCP and Usenet*, O'Reilly & Associates, Inc. USA.

**uuto(1)**

**uuto(1)**

Grace Todino and Dale Dougherty,  
*Using UUCP and Usenet*, O'Reilly & Associates, Inc. USA.

**STANDARDS CONFORMANCE**

**uuto**: SVID2, SVID3, XPG2, XPG3, XPG4

**uupick**: SVID2, SVID3, XPG2, XPG3, XPG4

| **u** |

**NAME**

uux - UNIX system to UNIX system command execution

**SYNOPSIS**

**uux** [*options*] *command-string*

**DESCRIPTION**

**uux** gathers zero or more files from various systems, executes a command on a specified system, then sends standard output to a file on a specified system. Note that, for security reasons, many installations limit the list of commands executable on behalf of an incoming request from **uux**. Many sites will permit little more than the receipt of mail (see *mail(1)*, *mailx(1)*, and *elm(1)*) via **uux**.

The *command-string* is made up of one or more arguments that look like a shell command line, except that the command and file names may be prefixed by *system-name*!. A null *system-name* is interpreted as the local system.

File names can be one of the following:

- A full path name;
- A path name preceded by *~xxx* where *xxx* is a login name on the specified system and is replaced by that user's login directory. Note that if an invalid login is specified, the default will be to the public directory (*/var/spool/uucppublic*);
- A path name preceded by *~/destination* where *destination* is appended to */var/spool/uucppublic*.
- A simple file name (which is prefixed by the current directory). See *uucp(1)* for details.

For example, the command

```
uux " !diff usg!/usr/dan/file1 pwba!/a4/dan/file2 > !~/dan/file.diff"
```

gets files **file1** and **file2** from machines **usg** and **pwba**, and executes a *diff(1)* command, placing the results in **file.diff** in the local directory */var/spool/uucppublic*.

Any special shell characters such as *<*, *>*, *;*, or *|* should be quoted, either by quoting the entire *command-string*, or quoting the special characters as individual arguments.

**uux** attempts to get all files to the execution system. For files that are output files, the file name must be escaped using parentheses. For example, the command

```
uux a!cut -f1 b!/usr/file \(c!/usr/file\)
```

gets */usr/file* from system **b** and sends it to system **a**, performs a **cut** command on the file, and sends the result of the cut command to system **c**.

**uux** notifies you if the requested command on the remote system was disallowed. The list of commands allowed is specified in the **Permissions** file in */etc/uucp*. The response comes by remote mail from the remote machine.

**u**

**uux** recognizes the following *options*:

- The standard input to **uux** is made the standard input to the *command-string*.
- aname* Use *name* as the user identification replacing the initiator user-ID (notification is returned to the user).
- b** Return whatever standard input was provided to the **uux** command if the exit status is non-zero.
- c** Do not copy the local file to the spool directory for transfer to the remote machine (default).
- C** Force the copy of local files to the spool directory for transfer.
- ggrade* *grade* is a single letter/number; lower ASCII sequence characters cause the job to be transmitted earlier during a particular conversation.
- j** Output the *jobid* (the job identification ASCII string) on the standard output. This job identification can be used by **uustat** to obtain the status or terminate a job (see *uustat(1)*).



- n** Do not notify the user if the command fails.
- r** Do not start the file transfer, just queue the job.
- sfile** Report status of the transfer in *file*.
- xdebug\_level** Produce debugging output on standard output. The *debug\_level* is a number between 0 and 9. The higher the number, the more detailed the information returned.
- z** Send success notification to user.

## WARNINGS

Only the first command of a shell pipeline can have a *system-name* !. All other commands are executed on the system of the first command.

The use of the shell metacharacter **\*** will probably not do what you want it to do. The shell tokens **<<** and **>>** are not implemented.

The execution of commands on remote systems takes place in an execution directory known to the UUCP subsystem. All files required for the execution are put into this directory unless they already reside on that machine. Therefore, the simple file name (without path or machine reference) must be unique within the **uux** request. The following command does *not* work:

```
uux "a!diff b!/usr/dan/xyz c!/usr/dan/xyz > !xyz.diff"
```

but the command:

```
uux "a!diff a!/usr/dan/xyz c!/usr/dan/xyz > !xyz.diff"
```

works (if **diff** is a permitted command).

Protected files and files that are in protected directories that are owned by the requester can be sent in commands using **uux**. However, if the requester is **root**, and the directory is not searchable by **other**, the request fails.

## FILES

|                              |                     |
|------------------------------|---------------------|
| <b>/etc/uucp</b>             | configuration files |
| <b>/var/uucp</b>             | log and error files |
| <b>/var/spool/uucp</b>       | spool directories   |
| <b>/var/spool/locks</b>      | lock files          |
| <b>/var/spool/uucppublic</b> | public directory    |

## SEE ALSO

mail(1), uuclean(1M), uucp(1).

Tim O'Reilly and Grace Todino,  
*Managing UUCP and Usenet*, O'Reilly & Associates, Inc. USA.

Grace Todino and Dale Dougherty,  
*Using UUCP and Usenet*, O'Reilly & Associates, Inc. USA.

## STANDARDS CONFORMANCE

**uux**: SVID2, SVID3, XPG2, XPG3, XPG4

| u |

**NAME**

vacation - return "I am not here" indication

**SYNOPSIS**

```
vacation -i
```

```
vacation [[-a alias] ...] login
```

**DESCRIPTION**

The **vacation** program returns a message to the sender of a message telling them that you are currently not reading your mail. The intended use is in a **.forward** file in **\$HOME**. For example, your **.forward** file might contain:

```
\eric, "|/usr/bin/vacation -a allman eric"
```

which would send messages to you (assuming your login name was **eric**) and reply to any messages for **eric** or **allman**. The **\** preceding **eric** is required to force direct delivery to **eric**'s mailbox and prevent an infinite loop through the **.forward** file. The double quotes are needed to tell **sendmail(1M)** to treat the enclosed as a unit, rather than separate recipients. It is also important to specify the full path for the vacation program, and there must be no white space between the **|** character and the start of the path name.

No message is sent unless *login* or an *alias* supplied using the **-a** option is a substring of either the **To:** or **Cc:** headers of the mail. No messages from **???-REQUEST**, **Postmaster**, **UUCP**, **MAILER**, or **MAILER-DAEMON** are replied to, nor is a notification sent if a **Precedence: bulk** or **Precedence: junk** line is included in the mail headers. Only one message per week is sent to each unique sender (at each unique host system). The people who have sent you messages are recorded in a database in the files **.vacation.pag** and **.vacation.dir** in your home directory.

The **vacation** program expects a file **.vacation.msg**, in your home directory, containing a message to be sent back to each sender. It should be an entire message (including headers). For example, it might say:

```
>From: eric@ucbmonet.Berkeley.EDU (Eric Allman)
Subject: I am on vacation
X-Delivered-By-The-Graces-Of: The vacation program
Precedence: bulk

I am on vacation until July 22. If you have something urgent,
please contact Joe Kalash <kalash@ucbingres.Berkeley.EDU>.
--eric
```

Header lines in this file must be left justified and must not be preceded by any other lines, including blank lines (see **sendmail(1M)**). If there is no **.vacation.msg** file, **vacation** uses the following file (if it exists):

```
/usr/share/lib/vacation.def
```

Otherwise, it logs an error.

**vacation** reads the first line from the standard input (the incoming mail message in the example **.forward** file above) for a UNIX style **From** line to determine the sender. **sendmail(1M)** includes this **From** line automatically, and its absence indicates non-mail input.

**Options**

The **vacation** program supports the following options:

- i**           Initializes the vacation database files. This option should be used before modifying a **.forward** file.
- a alias**    Identifies another name that can legitimately appear in the **To:** line of the mail header instead of your login name. More than one **-a** option can be specified.

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the language in which error messages are printed.

**DIAGNOSTICS**

On error, **vacation** exits with a value from `<sysexits.h>` and causes **sendmail** to report an error back to the sender of the original message. Errors such as the absence of **.vacation.msg** or calling **vacation** with incorrect arguments, are logged using **syslogd** on the system where **vacation** actually runs (see *syslogd(1M)*). The **syslog** file (`/var/adm/syslog/mail.log` by default – see `/etc/syslog.conf` and *syslogd(1M)* for customizations) should be inspected when **vacation** generates mailer error messages.

Remember that if the machine is configured for shared mail, inbound mail is handled at the mail server rather than on mail client nodes. This means that **syslog** diagnostics appear in the mail server's **syslog**; not the client's **syslog**.

**WARNINGS**

Errors in the **.forward** file can lead to loss of mail and infinite mail loops.

Always send test mail to yourself after configuring **vacation** to be sure that it is working properly. This is akin to checking telephone forwarding before leaving for an extended period, and can prevent loss of messages.

Some mail recipients look for RFC-822 compliant **To:** header in their incoming mail messages. Such recipients have to set the option **NoRecipientAction** to **add\_to** in the **sendmail** configuration file, **sendmail.cf**.

**AUTHOR**

**vacation** was developed by Eric Allman and the University of California, Berkeley.

**FILES**

|                                          |                                             |
|------------------------------------------|---------------------------------------------|
| <code>\$HOME/.vacation.dir</code>        | Database file.                              |
| <code>\$HOME/.vacation.msg</code>        | Message to send.                            |
| <code>\$HOME/.vacation.pag</code>        | Database file.                              |
| <code>/usr/share/lib/vacation.def</code> | System-wide default header and message.     |
| <code>/etc/syslog.conf</code>            | Dictates where error messages are recorded. |

**SEE ALSO**

*sendmail(1M)*, *syslogd(1M)*, *ndbm(3X)*.

**NAME**

val - validate SCCS file

**SYNOPSIS**

val -

val [-s] [-r *SID*] [-m *name*] [-y *type*] [-v] *files*

**DESCRIPTION**

The **val** command reads one or more *files* to determine whether each file read is an SCCS file meeting the characteristics specified by the optional argument list. Command-line options may appear in any order, and are described below.

**Options**

The **val** command recognizes the following options and command-line arguments. The effects of each option apply independently to each specified *file*.

- s** Silent option. Suppress diagnostic messages normally generated on the standard output when an error is encountered while processing any specified *file*.
- r *SID*** Check existence of revision *SID* in *file* where *SID* (SCCS IDentification string) is an SCCS delta number. *SID* is first checked to ensure that it is unambiguous and valid before checking *file*. For example, **\*-r1** is ambiguous because it physically does not exist but implies 1.1, 1.2, etc., which may exist; **\*-r1.0** and **\*-r1.1.0** are invalid because they have a zero suffix which never appears in a valid delta number.
- m *name*** *name* is compared with the SCCS **%M%** keyword in *file*.
- y *type*** *type* is compared with the SCCS **%Y%** keyword in *file*.
- v** Verbose option. Prints additional detailed diagnostic messages on the standard output for any corruption detected while processing each named *file*. The messages are intended for use with the information contained in *sccsfile*(4) when fixing the file.
- file* One or more SCCS files to be processed. If - is used as a *file* argument, **val** reads the standard input until an end-of-file condition is encountered. Each line read is independently processed as if it were a command-line argument list.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the interpretation of text within file as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **val** behaves as if all internationalization variables are set to "C". See *environ*(5).

| **V** |

**International Code Set Support**

Single-byte and multi-byte character code sets are supported.

**RETURN VALUE**

The 8-bit code returned by **val** is a disjunction of the possible errors; i. e., can be interpreted as a bit string where (moving from left to right) set bits are interpreted as follows:

**Bit Interpretation**

- 0 Missing *file* argument;
- 1 Unknown or duplicate option argument;
- 2 Corrupt SCCS file;
- 3 Cannot open *file* or *file* is not an SCCS file;
- 4 *SID* is invalid or ambiguous;
- 5 *SID* does not exist;
- 6 **%Y%** does not match **-y *type*** argument;

7 %M% does not match **-m name** argument;

Note that **val** can process two or more files on a given command line, and in turn can process multiple command lines (when reading the standard input). In these cases an aggregate code is returned; a logical **OR** of the codes generated for each command line and file processed.

**DIAGNOSTICS**

**val** generates diagnostic messages on the standard output for each command line and file processed, and also returns a single 8-bit code upon exit as described earlier under RETURN VALUE. Use the *sccshelp*(1) command for explanations.

**SEE ALSO**

admin(1), delta(1), get(1), sccshelp(1), prs(1), sccsfile(4).

**STANDARDS CONFORMANCE**

**val**: SVID2, SVID3, XPG2, XPG3, XPG4

**NAME**

vc - substitutes assigned values in place of identification keywords.

**SYNOPSIS**

vc [-a] [-t] [-c *char*] [-s] [*keyword=value* ... *keyword=value*]

**DESCRIPTION**

The **vc**, or *version control* command copies lines from the standard input to the standard output under control of command line *arguments* and *control statements* encountered in the standard input. In the process of performing the copy operation, user declared *keywords* can be replaced by their string *value* when they appear in plain text and/or control statements. The copying of lines from the standard input to the standard output is conditional, based on tests of keyword values specified in control statements or on **vc** command arguments.

Replacement of keywords by values is done whenever a keyword surrounded by control characters is encountered on a version control statement. The **-a** option forces replacement of keywords in *all* lines of text. An uninterpreted control character can be included in a value by preceding it with \. If a literal \ is desired, it too must be preceded by \.

The **vc** command is part of the SCCS (Source Code Control System) command suite.

**Options**

**vc** recognizes the following options and arguments:

- a** Replace keywords surrounded by control characters with their assigned value in *all* text lines and not just in **vc** statements.
- t** Ignore all characters from the beginning of a line up to and including the first *tab* character for the purpose of detecting a control statement. If one is found, all characters up to and including the *tab* are discarded.
- cchar** Specify a control character to be used in place of **:**.
- s** Silence warning messages (not errors) that are normally printed on the diagnostic output.

**Control Statements**

A control statement is a single line beginning with a control character, and the default control character is colon (**:**) (Unless the **-t** and **-c** options are used [See above]). Input lines beginning with a backslash (\) followed by the control character are not control lines, and are copied to the standard output with the backslash removed. Lines beginning with a backslash followed by a non-control character are copied in their entirety.

A keyword is composed of 9 or fewer alphanumeric characters of which the first character is alphabetic. A value is any ASCII string that can be created using **ed** (see *ed(1)*); a numeric value is an unsigned string of digits. Keyword values must not contain spaces or tabs.

Version control statements occur in the following forms:

**:dcl** keyword[, ..., keyword]

Used to declare keywords. All keywords must be declared.

**:asg** keyword=value

Used to assign values to keywords. An **asg** statement overrides the assignment for the corresponding keyword on the **vc** command line and all previous **asgs** for that keyword. Keywords declared, but not assigned values have null values.

**:if** condition

...

**:end**

Used to skip lines of the standard input. If the condition is true, all lines between the *if* statement and the matching *end* statement are copied to the standard output. If the condition is false, all intervening lines are discarded, including control statements. Note that intervening *if* statements and matching *end* statements are recognized solely for the purpose of maintaining the proper *if-end* matching.

The syntax of a condition may include the following:

```

<cond> ::= ["not"] <or>
<or> ::= <and> | <and> "|" <or>
<and> ::= <exp> | <exp> "&" <and>
<exp> ::= "(" <or> ")" | <value> <op> <value>
<op> ::= "=" | "!=" | "<" | ">"
<value> ::= <arbitrary ASCII string> | <numeric string>

```

The following are available operators and their meanings:

```

= equal
!= not equal
& and
| or
> greater than
< less than
() used for logical groupings
not allowed only immediately after the if, and when present, inverts
 the value of the entire condition

```

The > and < operate only on unsigned integer values (such as : 012 > 12 is false). All other operators take strings as arguments (for example, : 012 != 12 is true). The precedence of the operators (from highest to lowest) is as follows:

```

= != > < all of equal precedence
&
|

```

Parentheses can be used to alter the order of precedence. Values must be separated from operators or parentheses by at least one space or tab.

```

: :text Used for keyword replacement on lines that are copied to the standard output.
 The two leading control characters are removed, and keywords surrounded by
 control characters in text are replaced by their value before the line is copied to
 the output file. This action is independent of the -a option.

: on
: off Turn on or off keyword replacement on all lines.

: ctlchar Change the control character to char.

: msgmessage Prints the given message on the diagnostic output.

: errmessage Prints the given message followed by:
 ERROR: err statement on line ... (915)
 on the diagnostic output. vc halts execution and returns an exit code of 1.

```

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_CTYPE** determines the interpretation of keywords, values, the control character assigned through **ctl** and within text as single- and/or multi-byte characters.

**LANG** determines the language in which messages are displayed.

If **LC\_CTYPE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **vc** behaves as if all internationalization variables are set to "C". See *environ(5)*.

## RETURN VALUE

**vc** returns 0 on normal completion; 1 if an error occurs.

## DIAGNOSTICS

Use *scshelp(1)* for explanations.

**SEE ALSO**  
ed(1), sccshelp(1).

**| v |**



**NAME**

vi, view, vedit - screen-oriented (visual) text editor

**SYNOPSIS**

**vi** [-] [-l] [-r] [-R] [-t *tag*] [-v] [-V] [-wsize] [-x] [-C] [+*command*] [*file* ...]

**XPG4 Synopsis**

**vi** [-rR] [-c *command*] [-t *tag*] [-w *size*] [*file* ...]

**Obsolescent Options**

**vi** [-rR] [+*command*] [-t *tag*] [-w *size*] [*file* ...]

**view** [-] [-l] [-r] [-R] [-t *tag*] [-v] [-V] [-wsize] [-x] [-C] [+*command*] [*file* ...]

**vedit** [-] [-r] [-R] [-l] [-t *tag*] [-v] [-V] [-wsize] [-x] [-C] [+*command*] [*file* ...]

**Remarks**

The program names **ex**, **edit**, **vi**, **view**, and **vedit** are separate personalities of the same program. This manual entry describes the behavior of the **vi/view/vedit** personality.

**DESCRIPTION**

The **vi** (visual) program is a display-oriented text editor that is based on the underlying **ex** line editor (see *ex(1)*). It is possible to switch back and forth between the two and to execute **ex** commands from within **vi**. The line-editor commands and the editor options are described in *ex(1)*. Only the visual mode commands are described here.

The **view** program is identical to **vi** except that the **readonly** editor option is set (see *ex(1)*).

The **vedit** program is somewhat friendlier for beginners and casual users. The **report** editor option is set to **1**, and the **nomagic**, **novice**, and **showmode** editor options are set.

In **vi**, the terminal screen acts as a window into a memory copy of the file being edited. Changes made to the file copy are reflected in the screen display. The position of the cursor on the screen indicates the position within the file copy.

The environment variable **TERM** must specify a terminal type that is defined in the **terminfo** database (see *terminfo(4)*). Otherwise, a message is displayed and the line-editor is invoked.

As with **ex**, editor initialization scripts can be placed in the environment variable **EXINIT**, or in the file **.exrc** in the current or home directory.

**Options and Arguments**

**vi** recognizes the following command-line options and arguments:

- |               |                                                                                                                                                                                                                                                                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -             | Suppress all interactive-user feedback. This is useful when editor commands are taken from scripts.                                                                                                                                                                                                                                                        |
| -l            | Set the <b>lisp</b> editor option (see <i>ex(1)</i> ). Provides indents appropriate for <b>lisp</b> code. The <b>(</b> , <b>)</b> , <b>{</b> , <b>}</b> , <b>[</b> , and <b>]</b> commands in <b>vi</b> are modified to function with <b>lisp</b> source code.                                                                                             |
| -r            | Recover the specified <i>files</i> after an editor or system crash. If no <i>file</i> is specified, a list of all saved files is printed. You must be the owner of the saved file in order to recover it (superuser cannot recover files owned by other users).                                                                                            |
| -R            | Set the <b>readonly</b> editor option to prevent overwriting a file inadvertently (see <i>ex(1)</i> ).                                                                                                                                                                                                                                                     |
| -t <i>tag</i> | Execute the <b>tag</b> <i>tag</i> command to load and position a predefined file. See the <b>tag</b> command and the <b>tags</b> editor option in <i>ex(1)</i> .                                                                                                                                                                                           |
| -v            | Invoke visual mode ( <b>vi</b> ). Useful with <b>ex</b> , it has no effect on <b>vi</b> .                                                                                                                                                                                                                                                                  |
| -V            | Set verbose mode. Editor commands are displayed as they are executed when input from a <b>.exrc</b> file or a source file (see the <b>source</b> command in <i>ex(1)</i> ).                                                                                                                                                                                |
| -wsize        | Set the value of the <b>window</b> editor option to <i>size</i> . If <i>size</i> is omitted, it defaults to <b>3</b> .                                                                                                                                                                                                                                     |
| -x            | Set encryption mode. You are prompted for a key to allow for the creation or editing of an encrypted file. This command makes an educated guess to determine whether text read in is encrypted or not. The temporary buffer file is encrypted also, using a transformed version of the key typed in for the <b>-x</b> option (see the <b>crypt</b> command |

| **V** |

- in *ex*(1)).
- C** Encryption option. Same as the **-x** option, except that all text read in is assumed to have been encrypted.
  - c *command*** (XPG4 only.)
  - +*command*** (Obsolescent) Begin editing by executing the specified **ex** command-mode commands. As with the normal **ex** command-line entries, the **command** option-argument can consist of multiple **ex** commands separated by vertical-line commands (|). The use of commands that enter input mode in this manner produces undefined results.
  - file* Specify the file or files to be edited. If more than one *file* is specified, they are processed in the order given. If the **-r** option is also specified, the files are read from the recovery area.

(XPG4 only.) If both the **-t** tag and **-c** command (or the obsolescent **+command**) options are given, the **-t** tag will be processed first, that is, the file containing the tag is selected by **-t** and then the command is executed.

When invoked, **vi** is in *command mode*. *input mode* is initiated by several commands used to insert or change text.

In input mode, ESC (escape) is used to leave input mode; however, two consecutive ESC characters are required to leave input mode if the **doubleescape** editor option is set (see *ex*(1)).

In command mode, ESC is used to cancel a partial command; the terminal bell sounds if the editor is not in input mode and there is no partially entered command.

**WARNING:** ESC *completes* a "bottom line" command (see below).

The last (bottom) line of the screen is used to echo the input for search commands (/ and ?), **ex** commands (:), and system commands (!). It is also used to report errors or print other messages.

The receipt of **SIGINT** during text input or during the input of a command on the bottom line terminates the input (or cancels the command) and returns the editor to command mode. During command mode, **SIGINT** causes the bell to be sounded. In general the bell indicates an error (such as an unrecognized key).

Lines displayed on the screen containing only a ~ indicate that the last line above them is the last line of the file (the ~ lines are past the end of the file). Terminals with limited local intelligence might display lines on the screen marked with an @. These indicate space on the screen not corresponding to lines in the file. (These lines can be removed by entering a ^R, forcing the editor to retype the screen without these holes.)

If the system crashes or **vi** aborts due to an internal error or unexpected signal, **vi** attempts to preserve the buffer if any unwritten changes were made. Use the **-r** command line option to retrieve the saved changes.

The **vi** text editor supports the **SIGWINCH** signal, and redraws the screen in response to window-size changes.

| **V** |

## Command Summary

Most commands accept a preceding number as an argument, either to give a size or position (for display or movement commands), or as a repeat count (for commands that change text). For simplicity, this optional argument is referred to as *count* when its effect is described.

The following operators can be followed by a movement command to specify an extent of text to be affected: **c**, **d**, **y**, **<**, **>**, **!**, and **=**. The region specified begins at the current cursor position and ends just prior to the cursor position indicated by the move. If the command operates on lines only, all the lines that fall partly or wholly within this region are affected. Otherwise the exact marked region is affected.

In the following description, control characters are indicated in the form ^X, which represents Ctrl-X. Whitespace is defined to be the characters space, tab, and alternative space. Alternative space is the first character of the **ALT\_PUNCT** item described in *langinfo*(5) for the language specified by the **LANG** environment variable (see *environ*(5)).

Unless otherwise specified, the commands are interpreted in command mode and have no special effect in input mode.

- ^B** Scroll backward to display the previous window of text. A preceding *count* specifies the number of windows to go back. Two lines of overlap are kept if possible.
- ^D** Scroll forward a half-window of text. A preceding *count* gives the number of (logical) lines to scroll, and is remembered for future **^D** and **^U** commands.
- ^D** (input mode) Backs up over the indentation provided by **autoindent** or **^T** to the next multiple of **shiftwidth** spaces. Whitespace inserted by **^T** at other than the beginning of a line cannot be backed over using **^D**. A preceding **^** removes all indentation for the current and subsequent input lines of the current input mode until new indentation is established by inserting leading whitespace, either by direct input or by using **^T**.
- ^E** Scroll forward one line, leaving the cursor where it is if possible.
- ^F** Scroll forward to display the window of text following the current one. A preceding *count* specifies the number of windows to advance. Two lines of overlap are kept if possible.
- (XPG4 only.) The current line is displayed and the cursor is moved to the first non-blank character of the current line or the first character if the line is a blank line.
- ^G** Print the current file name and other information, including the number of lines and the current position (equivalent to the **ex** command **f**).
- ^H** Move one space to the left (stops at the left margin). A preceding *count* specifies the number of spaces to back up. (Same as **h**).
- ^H** (input mode) Move the cursor left to the previous input character without erasing it from the screen. The character is deleted from the saved text.
- ^J** Move the cursor down one line in the same column, if possible. A preceding *count* specifies the number of lines to move down. (Same as **^N** and **j**).
- ^L** Clear and redraw the screen. Use when the screen is scrambled for any reason.
- ^M** Move to the first nonwhitespace character in the next line. A preceding *count* specifies the number of lines to advance.
- ^N** Same as **^J** and **j**.
- ^P** Move the cursor up one line in the same column. A preceding *count* specifies the number of lines to move up (same as **k**).
- ^R** Redraw the current screen, eliminating the false lines marked with **@** (which do not correspond to actual lines in the file).
- ^T** Pop the tag stack. See the **pop** command in *ex(1)*.
- ^T** (input mode) Insert **shiftwidth** whitespace. If at the beginning of the line, this inserted space can only be backed over using **^D**.
- ^U** Scroll up a half-window of text. A preceding *count* gives the number of (logical) lines to scroll, and is remembered for future **^D** and **^U** commands.
- ^V** In input mode, **^V** quotes the next character to permit the insertion of special characters (including ESC) into the file.
- ^W** In input mode, **^W** backs up one word; the deleted characters remain on the display.
- ^Y** Scroll backward one line, leaving the cursor where it is, if possible.
- ^[** Cancel a partially formed command; **^[** sounds the bell if there is no partially formed command.
- In input mode, **^[** terminates input mode. However, two consecutive ESC characters are required to terminate input mode if the **doubleescape** editor option is set (see *ex(1)*).
- When entering a command on the bottom line of the screen (**ex** command line or search pattern with **\** or **?**), terminate input and execute command.
- On many terminals, **^[** can be entered by pressing the ESC or ESCAPE key.

| v |

- ^\  
^]  
^^  
space  
erase  
kill  
susp  
!  
"  
\$  
%  
&  
'  
'  
'  
[ [  
]]  
^  
(**
- Exit **vi** and enter *ex* command mode. If in input mode, terminate the input first.
- Take the word at or after the cursor as a tag and execute the **tagMbobC** editor command (see *ex(1)*).
- Return to the previous file (equivalent to **:ex #**).
- Move one space to the right (stops at the end of the line). A preceding *count* specifies the number of spaces to go forward (same as **l**).
- Erase, where *erase* is the user-designated erase character (see *stty(1)*). Same as **^H**.
- Kill, where *kill* is the user-designated kill character (see *stty(1)*). In input mode, *kill* backs up to the beginning of the current input line without erasing the line from the screen display.
- Suspend the editor session and return to the calling shell, where *susp* is the user-designated process-control suspend character (see *stty(1)*). See *ex(1)* for more information on the **suspend** editor command.
- An operator that passes specified lines from the buffer as standard input to the specified system command, and replaces those lines with the standard output from the command. The **!** is followed by a movement command specifying the lines to be passed (lines from the current position to the end of the movement) and then the command (terminated as usual by a return). A preceding *count* is passed on to the movement command after **!**.
- Doubling **!** and preceding it by *count* causes that many lines, starting with the current line, to be passed.
- Use to precede a named buffer specification. There are named buffers **1** through **9** in which the editor places deleted text. The named buffers **a** through **z** are available to the user for saving deleted or yanked text; see also **y**, below.
- Move to the end of the current line. A preceding *count* specifies the number of lines to advance (for example, **2\$** causes the cursor to advance to the end of the next line).
- Move to the parenthesis or brace that matches the parenthesis or brace at the current cursor position.
- Same as the *ex* command **&** (that is, **&** repeats the previous **substitute** command).
- When followed by a **'**, **vi** returns to the previous context, placing the cursor at the beginning of the line. (The previous context is set whenever a nonrelative move is made.) When followed by a letter **a-z**, returns to the line marked with that letter (see the **m** command), at the first nonwhitespace character in the line.
- When used with an operator such as **d** to specify an extent of text, the operation takes place over complete lines (see also **`**).
- When followed by a **`**, **vi** returns to the previous context, placing the cursor at the character position marked (the previous context is set whenever a nonrelative move is made). When followed by a letter **a-z**, returns to the line marked with that letter (see the **m** command), at the character position marked.
- When used with an operator such as **d** to specify an extent of text, the operation takes place from the exact marked place to the current position within the line (see also **'**).
- Back up to the previous section boundary. A section is defined by the value of the **sections** option. Lines that start with a form feed (**^L**) or **{** also stop **[ [**.
- If the option **lisp** is set, the cursor stops at each **(** at the beginning of a line.
- Move forward to a section boundary (see **[ [**).
- Move to the first nonwhitespace position on the current line.
- Move backward to the beginning of a sentence. A sentence ends at a **.**, **!**, or **?** followed by either the end of a line or by two spaces. Any number of closing **)**, **]**, **"**, and **'** characters can appear between the **.**, **!**, or **?** and the spaces or end of line. If a *count* is specified, the cursor moves back the specified number of sentences.
- If the **lisp** option is set, the cursor moves to the beginning of a **lisp** s-expression. Sentences also begin at paragraph and section boundaries (see **{** and **[ [**).

- ) Move forward to the beginning of a sentence. If a *count* is specified, the cursor advances the specified number of sentences (see `()`).
- { Move back to the beginning of the preceding paragraph. A paragraph is defined by the value of the **paragraphs** option. A completely empty line and a section boundary (see `[]` above) are also interpreted as the beginning of a paragraph. If a *count* is specified, the cursor moves backward the specified number of paragraphs.
- } Move forward to the beginning of the next paragraph. If a *count* is specified, the cursor advances the specified number of paragraphs (see `{}`).
- | Requires a preceding *count*; the cursor moves to the specified column of the current line (if possible).
- + Move to the first nonwhitespace character in the next line. If a *count* is specified, the cursor advances the specified number of lines (same as `^M`).
- ,
- The comma (`,`) performs the reverse action of the last **f**, **F**, **t**, or **T** command issued, by searching in the opposite direction on the current line. If a *count* is specified, the cursor repeats the search the specified number of times.
- The hyphen character (`-`) moves the cursor to the first nonwhitespace character in the previous line. If a *count* is specified, the cursor moves back the specified number of times.
- \_ The underscore character (`_`) moves the cursor to the first nonwhitespace character in the current line. If a *count* is specified, the cursor advances the specified number of lines, with the current line being counted as the first line; no *count* or a *count* of 1 specifies the current line.
- .
- Repeat the last command that changed the buffer. If a *count* is specified, the command is repeated the specified number of times.
- /
- Read a string from the last line on the screen, interpret it as a regular expression, and scan forward for the next occurrence of a matching string. The search begins when the user types a carriage return to terminate the pattern; the search can be terminated by sending **SIGINT** (or the user-designated interrupt character).
- When used with an operator to specify an extent of text, the defined region begins with the current cursor position and ends at the beginning of the matched string. Entire lines can be specified by giving an offset from the matched line (by using a closing `/` followed by a `+n` or `-n`).
- 0 Move to the first character on the current line (the `0` is not interpreted as a command when preceded by a nonzero digit).
- :
- The colon character (`:`) begins an **ex** command. The `:` and the entered command are echoed on the bottom line; the **ex** command is executed when the user types a carriage return.
- ;
- Repeat the last single character find using **f**, **F**, **t**, or **T**. If a *count* is specified, the search is repeated the specified number of times.
- <
- An operator that shifts lines to the left by one **shiftwidth**. The `<` can be followed by a move to specify lines. A preceding *count* is passed through to the move command.
- When repeated (`<<`), shifts the current line (or *count* lines starting at the current one).
- >
- An operator that shifts lines right one **shiftwidth** (see `<`).
- =
- If the **lisp** option is set, `=` reindents the specified lines, as if they were typed in with **lisp** and **autoindent** set. `=` can be preceded by a *count* to indicate how many lines to process, or followed by a move command for the same purpose.
- ?
- Scan backwards, the reverse of `/` (see `/`).
- @buffer Execute the commands stored in the named *buffer*. Be careful not to include a `<return>` character at the end of the buffer contents unless the `<return>` is part of the command stream. Commands to be executed in *ex* mode should be preceded by a colon (`:`).

- ~ The tilde (~) switches the case of the character under the cursor (if it is a letter), then moves one character to the right, stopping at the end of the line). A preceding *count* specifies how many characters in the current line are switched.
- A** Append at the end of line (same as **\$a**).
- B** Back up one word, where a word is any nonblank sequence, placing the cursor at the beginning of the word. If a *count* is specified, the cursor moves back the specified number of words.
- C** Change the rest of the text on the current line (same as **c\$**).
- D** Delete the rest of the text on the current line (same as **d\$**).
- E** Move forward to the end of a word, where a word is any nonblank sequence. If a *count* is specified, the cursor advances the specified number of words.
- F** Must be followed by a single character; scans backwards in the current line, searching for that character and moving the cursor to it, if found. If a *count* is specified, the search is repeated the specified number of times.
- G** Go to the line number given as preceding argument, or the end of the file if no preceding *count* is given.
- H** Move the cursor to the top line on the screen. If a *count* is given, the cursor moves to *count* number of lines from the top of the screen. The cursor is placed on the first nonwhitespace character on the line. If used as the target of an operator, entire lines are affected.
- I** Insert at the beginning of a line (same as ^ followed by **i**).
- J** Join the current line with the next one, supplying appropriate whitespace: one space between words, two spaces after a period, and no spaces at all if the first character of the next line is a closing parenthesis ()). A preceding *count* causes the specified number of lines to be joined, instead of just two.
- L** Move the cursor to the first nonwhitespace character of the last line on the screen. If a *count* is given, the cursor moves to *count* number of lines from the bottom of the screen. When used with an operator, entire lines are affected.
- M** Move the cursor to the middle line on the screen, at the first nonwhitespace position on the line.
- N** Scan for the next match of the last pattern given to / or ?, but in the opposite direction; this is the reverse of **n**.
- O** Open a new line above the current line and enter input mode.
- P** Put back (replace) the last deleted or yanked text before/above the cursor. Entire lines of text are returned above the cursor if entire lines were deleted or yanked. Otherwise, the text is inserted just before the cursor.  
(XPG4 only.) In this case, the cursor is moved to last column position of the inserted characters.  
If **P** is preceded by a named buffer specification (*x*), the contents of that buffer are retrieved instead.
- Q** Exit **vi** and enter **ex** command mode.
- R** Replace characters on the screen with characters entered, until the input is terminated with ESC.
- S** Change entire lines (same as **cc**). A preceding *count* changes the specified number of lines.
- T** Must be followed by a single character; scan backwards in the current line for that character, and, if found, place the cursor just after that character. A *count* is equivalent to repeating the search the specified number of times.
- U** Restore the current line to its state before the cursor was last moved to it.  
(XPG4 only.) The cursor position is set to the column position 1 or to the position indicated by the previous line if the **autoindent** is set.

- W** Move forward to the beginning of a word in the current line, where a word is a sequence of nonblank characters. If the current position is at the beginning of a word, the current position is within a bigword or the character at that position cannot be a part of a bigword, the current position shall move to the first character of the next bigword. If no subsequent bigword exists on the current line, the current position shall move to the first character of the first bigword on the first following line that contains the bigword. For this command, an empty or blank line is considered to contain exactly one bigword. The current line is set to the line containing the bigword selected and the current position is set to the first character of the bigword selected. A preceding *count* specifies the number of words to advance.
- X** Delete the character before the cursor. A preceding *count* repeats the effect, but only characters on the current line are deleted.
- Y** Place (yank) a copy of the current line into the unnamed buffer (same as **yy**). If a *count* is specified, *count* lines are copied to the buffer. If the **Y** is preceded by a buffer name, the lines are copied to the named buffer.
- ZZ** Exit the editor, writing out the buffer if it was changed since the last write (same as the **ex** command **x**). Note that if the last write was to a different file and no changes have occurred since, the editor exits without writing out the buffer.
- a** Enter input mode, appending the entered text after the current cursor position. A preceding *count* causes the inserted text to be replicated the specified number of times, but only if the inserted text is all on one line.
- b** Back up to the previous beginning of a word in the current line. A word is a sequence of alphanumeric characters or a sequence of special characters. A preceding *count* repeats the effect.
- c** Must be followed by a movement command. Delete the specified region of text, and enter input mode to replace deleted text with new text. If more than part of a single line is affected, the deleted text is saved in the numeric buffers. If only part of the current line is affected, the last character deleted is marked with a **\$**. A preceding *count* passes that value through to the move command. If the command is **cc**, the entire current line is changed.
- d** Must be followed by a movement command. Delete the specified region of text. If more than part of a line is affected, the text is saved in the numeric buffers. A preceding *count* passes that value through to the move command. If the command is **dd**, the entire current line is deleted.
- e** Move forward to the end of the next word, defined as for **b**. A preceding *count* repeats the effect.
- f** Must be followed by a single character; scan the rest of the current line for that character, and moves the cursor to it if found. A preceding *count* repeats the action that many times.
- h** Move the cursor one character to the left (same as **^H**). A preceding *count* repeats the effect.
- i** Enter input mode, inserting the entered text before the cursor (see **a**).
- j** Move the cursor one line down in the same column (same as **^J** and **^N**).
- k** Move the cursor one line up (same as **^P**).
- l** Move the cursor one character to the right (same as **<space>**).
- mx** Mark the current position of the cursor. **x** is a lowercase letter, **a-z**, that is used with the **`** and **'** commands to refer to the marked line or line position.
- n** Repeat the last / or ? scanning commands.
- o** Open a line below the current line and enter input mode; otherwise like **O**.
- p** Put text after/below the cursor; otherwise like **P**.
- r** Must be followed by a single character; the character under the cursor is replaced by the specified one. (The new character can be a new-line.) If **r** is preceded by a *count*, *count* characters are replaced by the specified character.

- s** Delete the single character under the cursor and enter input mode; the entered text replaces the deleted character. A preceding *count* specifies how many characters on the current line are changed. The last character being changed is marked with a \$, as for **c**.
- t** Must be followed by a single character; scan the remainder of the line for that character. The cursor moves to the column prior to the character if the character is found. A preceding *count* is equivalent to repeating the search *count* times.
- u** Reverse the last change made to the current buffer. If repeated, **u** alternates between these two states; thus is its own inverse. When used after an insertion of text on more than one line, the lines are saved in the numerically named buffers.
- w** Move forward to the beginning of the next word (where word is defined as in **b**). A preceding *count* specifies how many words the cursor advances.
- x** Delete the single character under the cursor. When **x** is preceded by a *count*, **x** deletes the specified number of characters forward from the cursor position, but only on the current line.
- y** Must be followed by a movement command; the specified text is copied (yanked) into the unnamed temporary buffer. If preceded by a named buffer specification, "x, the text is placed in that buffer also. If the command is **yy**, the entire current line is yanked.
- z** Redraw the screen with the current line placed as specified by the following options: **z**<return> specifies the top of the screen, **z.** the center of the screen, and **z-** the bottom of the screen. The commands **z^** and **z+** are similar to **^B** and **^F**, respectively. However, **z^** and **z+** do not attempt to maintain two lines of overlap. A *count* after the **z** and before the following character to specifies the number of lines displayed in the redrawn screen. A *count* before the **z** gives the number of the line to use as the reference line instead of the default current line.

### Keyboard Editing Keys

At initialization, the editor automatically maps some terminal keyboard editing keys to equivalent visual mode commands. These mappings are only established for keys that are listed in the following table and defined in the *terminfo*(4) database as valid for the current terminal (as specified by the **TERM** environment variable).

Both command and input mode mappings are created (see the **map** command in *ex*(1)). With the exception of the **insert char** keys, which simply toggle input mode on and off, the input mode mappings exit input mode, perform the same action as the command mode mapping, and then reenter input mode.

On certain terminals, the character sequence sent by a keyboard editing key, which is then mapped to a visual mode command, can be the same character sequence a user might enter to perform another command or set of commands. This is most likely to happen with the input mode mappings; therefore, on these terminals, the input mode mappings are disabled by default. Users can override the disabling and enabling of both the command and input mode keyboard editing key mappings by setting the **keyboardedit** and **keyboardedit!** editor options as appropriate (see *ex*(1)). The **timeout**, **timeoutlen**, and **doubleescape** editor options are alternative methods of addressing this problem.

| **V** |



| terminfo entry | command mode map | input mode map | map name | description            |
|----------------|------------------|----------------|----------|------------------------|
| key_ic         | i                | ^[             | inschar  | insert char            |
| key_eic        | i                | ^[             | inschar  | end insert char        |
| key_up         | k                | ^[ka           | up       | arrow up               |
| key_down       | j                | ^[ja           | down     | arrow down             |
| key_left       | h                | ^[ha           | left     | arrow left             |
| key_right      | l                | ^[la           | right    | arrow right            |
| key_home       | H                | ^[Ha           | home     | arrow home             |
| key_il         | o^[              | ^[o^[a         | insline  | insert line            |
| key_dl         | dd               | ^[dda          | delline  | delete line            |
| key_clear      | ^L               | ^[^La          | clear    | clear screen           |
| key_eol        | d\$              | ^[d\$a         | clreol   | clear line             |
| key_sf         | ^E               | ^[^Ea          | scrollf  | scroll down            |
| key_dc         | x                | ^[xa           | delchar  | delete char            |
| key_npage      | ^F               | ^[^Fa          | npage    | next page              |
| key_ppage      | ^B               | ^[^Ba          | ppage    | previous page          |
| key_sr         | ^Y               | ^[^Ya          | sr       | scroll up              |
| key_eos        | dG               | ^[dGa          | clreos   | clear to end of screen |

## EXTERNAL INFLUENCES

Support for international codes and environment variables are as follows:

### Environment Variables

**UNIX95** specifies using the XPG4 behaviour for this command.

**COLUMNS** overrides the system-selected horizontal screen size.

**LINES** overrides the system-selected vertical screen size, used as the number of lines in a screenful and the vertical screen size in visual mode.

**SHELL** is a variable that shall be interpreted as the preferred command-line interpreter for use in **!**, **shell**, **read**, and other commands with an operand of the form **!string**. For the **shell** command the program shall be invoked with the two arguments **-c** and **string**. If this variable is null or not set, the **sh** utility shall be used.

**TERM** is a variable that shall be interpreted as the name of the terminal type. If this variable is unset or null, an unspecified default terminal type shall be used.

**PATH** determines the search path for the shell command specified in the editor commands, **shell**, **read**, and **write**. **EXINIT** determines a list of *ex* commands that will be executed on editor startup, before reading the first file. The list can contain multiple commands by separating them using a vertical line (|) character.

**HOME** determines a pathname of a directory that will be searched for an editor startup file named **.exerc**.

**LC\_ALL** This variable shall determine the locale to be used to override any values for locale categories specified by the setting of **LANG** or any environment variables beginning with **LC\_**.

**LC\_MESSAGES** determines the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

**LC\_COLLATE** determines the collating sequence used in evaluating regular expressions and in processing the *tags* file. **LC\_CTYPE** determines the interpretation of text as single and/or multi-byte characters, the classification of characters as uppercase or lowercase letters, the shifting of letters between uppercase and lowercase, and the characters matched by character class expressions in regular expressions.

**LANG** determines the language in which messages are displayed.

**LANGOPTS** specifies options determining how text for right-to-left languages is stored in input and output files. See *environ*(5).

If **LC\_COLLATE** or **LC\_CTYPE** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**. If any internationalization variable contains an invalid setting, the editor behaves as if all internationalization variables are set to "C". See *environ*(5).

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**WARNINGS**

See also the WARNINGS section in *ex*(1).

**Program Limits**

**vi** places the following limits on files being edited.

**Maximum Line Length**

**4096** characters including 2-3 bytes for overhead. Thus, a line length up to 4092 characters should cause no problem.

If you load a file that contain lines longer than the specified limit, the lines are truncated to the stated maximum length. Saving the file will write the truncated version over the original file, thus overwriting the original lines completely.

Attempting to create lines longer than the allowable maximum for the editor produces a **line too long** error message.

**Maximum File Size**

The maximum file length of 234,239 lines is silently enforced.

**Other limits**

- 256 characters per global command list.
- 128 characters in a file name in **vi** or **ex open** mode. On short-file-name HP-UX systems, the maximum file name length is 14 characters.
- 128 characters in a previous insert/delete buffer.
- 100 characters in a shell-escape command.
- 63 characters in a string-valued option (**:set** command).
- 30 characters in a program tag name.
- 32 or fewer macros defined by **map** command.
- 512 or fewer characters total in combined **map** macros.

Do not use the **-C** option to edit unencrypted files. The **-C** option is meant to be used only on files that are already encrypted. If the **-C** option is used on files which are not yet encrypted, a write in the edit session is likely to corrupt the file.

**AUTHOR**

**vi** was developed by the University of California, Berkeley. The 16-bit extensions to **vi** are based in part on software of the Toshiba Corporation.

**V****SEE ALSO**

*ctags*(1), *ed*(1), *ex*(1), *stty*(1), *write*(1), *terminfo*(4), *environ*(5), *lang*(5), *regexp*(5).

*The Ultimate Guide to the vi and ex Text Editors*, Benjamin/Cummings Publishing Company, Inc., ISBN 0-8053-4460-8, HP part number 97005-90015.

**STANDARDS CONFORMANCE**

**vi**: SVID2, SVID3, XPG2, XPG3, XPG4

**NAME**

vis, inv - make unprintable and non-ASCII characters in a file visible or invisible

**SYNOPSIS**

**vis** [-n] [-s] [-t] [-u] [-x] *file* ...

**inv** [-n] [-s] [-t] [-u] [-x] *file* ...

**DESCRIPTION**

**vis** reads characters from each *file* in sequence and writes them to the standard output, converting those that are not printable or not ASCII into a visible form. *inv* performs the inverse function, reading printable characters from each *file*, returning them to non-printable or non-ASCII form, if appropriate, then writing them to standard output;

Non-printable ASCII characters are represented using C-like escape conventions:

|     |                                                                             |
|-----|-----------------------------------------------------------------------------|
| \\  | backslash                                                                   |
| \b  | backspace                                                                   |
| \e  | escape                                                                      |
| \f  | form-feed                                                                   |
| \n  | new-line                                                                    |
| \r  | carriage return                                                             |
| \s  | space                                                                       |
| \t  | horizontal tab                                                              |
| \v  | vertical tab                                                                |
| \n  | the character whose ASCII code is the 3-digit octal number <i>n</i> .       |
| \xn | the character whose ASCII code is the 2-digit hexadecimal number <i>n</i> . |

Non-ASCII single- or multi-byte characters are examined one byte at a time. For each byte, if it can be displayed as an ASCII character, it is treated as if it is an ASCII character; Otherwise, it is represented in the following conventions:

|     |                                                                                   |
|-----|-----------------------------------------------------------------------------------|
| \n  | the 8-bit character whose code value is the 3-digit octal number <i>n</i> .       |
| \xn | the 8-bit character whose code value is the 2-digit hexadecimal number <i>n</i> . |

Space, horizontal-tab, and new-line characters can be treated as printable (and therefore passed unaltered to the output) or non-printable depending on the options selected. Backslash, although printable, is expanded by *vis*, to a pair of backslashes so that when they are passed back through *inv*, they convert back to a single backslash.

If no input file is given, or if the argument - is encountered, **vis** and *inv* read from the standard input.

**Options**

**vis** and **inv** recognize the following options:

- n Treat new-line, space, and horizontal tab as non-printable characters. **vis** expands them visibly as \n, \s, and \t, rather than passing them directly to the output. **inv** discards these characters, expecting only the printable expansions. New-line characters are inserted by **vis** every 16 bytes so that the output will be in a form that is usable by most editors.
- s Make **vis** and **inv** silent about non-existent files, identical input and output, and write errors. Normally, no input file can be the same as the output file unless it is a special file.
- t Treat horizontal-tab and space characters as non-printable in the same manner that -n treats them.
- u Cause output to be unbuffered (byte-by-byte); normally, output is buffered.
- x Cause **vis** output to be in hexadecimal form rather than the default octal form. Either form is accepted to **inv** as input.

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the language in which messages are displayed.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**WARNINGS**

Redirecting output to an input file destroys the original data. Therefore, command forms such as

```
vis file1 file2 >file1
```

should be avoided unless the source file can be safely discarded.

**AUTHOR**

**vis** was developed by HP.

**SEE ALSO**

cat(1), echo(1), od(1).

| **v** |

**NAME**

vmstat - report virtual memory statistics

**SYNOPSIS**

**vmstat** [-**dns**] [*interval* [*count*]]

**vmstat** -**f** | -**s** | -**z**

**DESCRIPTION**

The **vmstat** command reports certain statistics kept about process, virtual memory, trap, and CPU activity. It also can clear the accumulators in the kernel **sum** structure.

**Options**

**vmstat** recognizes the following options:

- d** Report disk transfer information as a separate section, in the form of transfers per second.
- n** Provide an output format that is more easily viewed on an 80-column display device. This format separates the default output into two groups: virtual memory information and CPU data. Each group is displayed as a separate line of output. On multiprocessor systems, this display format also provides CPU utilization on a per CPU basis for the active processors.
- S** Report the number of processes swapped in and out (**si** and **so**) instead of page reclaims and address translation faults (**re** and **at**).
- interval* Display successive lines which are summaries over the last *interval* seconds. The first line reported is for the time since a reboot and each subsequent line is for the last interval only. If *interval* is zero, the output is displayed once only. If the **-d** option is specified, the column headers are repeated. If **-d** is omitted, the column headers are not repeated.  
  
The command **vmstat 5** prints what the system is doing every five seconds. This is a good choice of printing interval since this is how often some of the statistics are sampled in the system; others vary every second.
- count* Repeat the summary statistics *count* times. If *count* is omitted or zero, the output is repeated until an interrupt or quit signal is received. From the terminal, these are commonly **^C** and **^\**, respectively (see *stty*(1)).
- f** Report on the number of forks and the number of pages of virtual memory involved since boot-up.
- s** Print the total number of several kinds of paging-related events from the kernel **sum** structure that have occurred since boot-up or since **vmstat** was last executed with the **-z** option.
- z** Clear all accumulators in the kernel **sum** structure. This option is restricted to the super user.

If none of these options is given, **vmstat** displays a one-line summary of the virtual memory activity since boot-up or since the **-z** option was last executed.

**Column Descriptions**

The column headings and the meaning of each column are:

|               |                                                                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>procs</b>  | Information about numbers of processes in various states.                                                                                                                    |
|               | <b>r</b> In run queue                                                                                                                                                        |
|               | <b>b</b> Blocked for resources (I/O, paging, etc.)                                                                                                                           |
|               | <b>w</b> Runnable or short sleeper (< 20 secs) but swapped                                                                                                                   |
| <b>memory</b> | Information about the usage of virtual and real memory. Virtual pages are considered active if they belong to processes that are running or have run in the last 20 seconds. |
|               | <b>avm</b> Active virtual pages                                                                                                                                              |
|               | <b>free</b> Size of the free list                                                                                                                                            |

**page** Information about page faults and paging activity. These are averaged each five seconds, and given in units per second.

- re** Page reclaims (without **-S**)
- at** Address translation faults (without **-S**)
- si** Processes swapped in (with **-S**)
- so** Processes swapped out (with **-S**)
- pi** Pages paged in
- po** Pages paged out
- fr** Pages freed per second
- de** Anticipated short term memory shortfall
- sr** Pages scanned by clock algorithm, per second

**faults** Trap/interrupt rate averages per second over last 5 seconds.

- in** Device interrupts per second (nonclock)
- sy** System calls per second
- cs** CPU context switch rate (switches/sec)

**cpu** Breakdown of percentage usage of CPU time for the active processors

- us** User time for normal and low priority processes
- sy** System time
- id** CPU idle

**EXAMPLES**

The following examples show the output for various command options. For formatting purposes, some leading blanks have been deleted.

1. Display the default output.

vmstat

| procs |     | memory |      |     |    | page |    |    |    |    |    |    |
|-------|-----|--------|------|-----|----|------|----|----|----|----|----|----|
| r     | b   | w      | cpu  |     |    | re   | at | pi | po | fr | de | sr |
|       | in  | sy     | cs   | us  | sy | id   |    |    |    |    |    |    |
| 0     | 0   | 0      | 1158 | 511 | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  |
|       | 111 | 18     | 7    | 0   | 0  | 100  |    |    |    |    |    |    |

2. Add the disk transfer information to the default output.

vmstat -d

| procs |     | memory |      |     |    | page |    |    |    |    |    |    |
|-------|-----|--------|------|-----|----|------|----|----|----|----|----|----|
| r     | b   | w      | cpu  |     |    | re   | at | pi | po | fr | de | sr |
|       | in  | sy     | cs   | us  | sy | id   |    |    |    |    |    |    |
| 0     | 0   | 0      | 1158 | 511 | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  |
|       | 111 | 18     | 7    | 0   | 0  | 100  |    |    |    |    |    |    |

**Disk Transfers**

| device | xfer/sec |
|--------|----------|
| c0t6d0 | 0        |
| c0t1d0 | 0        |
| c0t3d0 | 0        |
| c0t5d0 | 0        |

3. Display the default output in 80-column format.

vmstat -n

| VM | memory | page | faults |
|----|--------|------|--------|
|----|--------|------|--------|

```

 avm free re at pi po fr de sr in sy cs
 1158 430 0 0 0 0 0 0 0 111 18 7
CPU
 cpu procs
 us sy id r b w
 0 0 100 0 0 0

```

- Replace the page reclaims and address translation faults with process swapping in the default output.

vmstat -S

```

 procs memory page
 r b faults cpu si so pi po fr de sr
 in sy cs us sy id
 0 0 0 1158 430 0 0 0 0 0 0 0
 111 18 7 0 0 100

```

- Display the default output twice at five-second intervals. Note that the headers are *not* repeated.

vmstat 5 2

```

 procs memory page
 r b faults cpu re at pi po fr de sr
 in sy cs us sy id
 0 0 0 1158 456 0 0 0 0 0 0 0
 111 18 7 0 0 100
 0 0 0 1221 436 5 0 5 0 0 0 0
 108 65 18 0 1 99

```

- Display the default output twice in 80-column format at five-second intervals. Note that the headers are *not* repeated.

vmstat -n 5 2

```

VM
 memory page faults
 avm free re at pi po fr de sr in sy cs
 1221 436 0 0 0 0 0 0 0 111 18 7
CPU
 cpu procs
 us sy id r b w
 1221 435 2 0 2 0 0 0 0 109 35 17
 0 1 99 0 0 0

```

- Display the default output and disk transfers twice in 80-column format at five-second intervals. Note that the headers *are* repeated.

vmstat -dn 5 2

```

VM
 memory page faults
 avm free re at pi po fr de sr in sy cs
 1221 435 0 0 0 0 0 0 0 111 18 7
CPU
 cpu procs
 us sy id r b w
 0 0 100 0 0 0

Disk Transfers
 device xfer/sec
 c0t6d0 0
 c0t1d0 0
 c0t3d0 0
 c0t5d0 0

```



```

VM
 memory
 avm free re at pi po fr de sr in sy cs
1219 425 0 0 0 0 0 0 0 111 54 15
CPU
 cpu procs
 us sy id r b w
 1 8 92 0 0 0

Disk Transfers
 device xfer/sec
 c0t6d0 0
 c0t1d0 0
 c0t3d0 0
 c0t5d0 0

```

8. Display the number of forks and pages of virtual memory since boot-up.

```
vmstat -f
```

```
24558 forks, 1471595 pages, average= 59.92
```

9. Display the counts of paging-related events.

```
vmstat -s
```

```

0 swap ins
0 swap outs
0 pages swapped in
0 pages swapped out
1344563 total address trans. faults taken
542093 page ins
2185 page outs
602573 pages paged in
4346 pages paged out
482343 reclaims from free list
504621 total page reclaims
124 intransit blocking page faults
1460755 zero fill pages created
404137 zero fill page faults
366022 executable fill pages created
71578 executable fill page faults
0 swap text pages found in free list
162043 inode text pages found in free list
196 revolutions of the clock hand
45732 pages scanned for page out
4859 pages freed by the clock daemon
36680636 cpu context switches
1497746186 device interrupts
1835626 traps
87434493 system calls

```

| V |

## WARNINGS

Users of `vmstat` must not rely on the exact field widths and spacing of its output, as these will vary depending on the system, the release of HP-UX, and the data to be displayed.

## AUTHOR

`vmstat` was developed by the University of California, Berkeley and HP.

## SEE ALSO

`iostat(1)`.



**NAME**

vt - log into another system over lan

**SYNOPSIS**

`/usr/bin/vt nodename [lan_device]`

`/usr/bin/vt -p [lan_device]`

**DESCRIPTION**

**vt** enables a user to log into another HP 9000 system (*nodename*) over an HP local area network. The **-p** option causes **vt** to send a poll request over the local area network to find out what systems currently have **vtdaemon** running (see *vtdaemon*(1M)). An asterisk (\*) following a *nodename* in the response indicates that the system is a **vt** gateway. Plus signs (+) following the *nodename* indicate how many **vt** gateways must be traversed to reach that system.

The optional argument *lan\_device* specifies a character special device name to use instead of the default device name to send and receive data to and from the local area network. The major number for this device must correspond to a CIO IEEE 802.3 local area network device.

Once a connection has been established, **vt** enters input mode. In this mode, text typed is sent to the remote host. To issue **vt** commands when in input mode, precede them with the **vt** escape character (see Commands below). When in command mode, normal terminal editing conventions are available.

The connection should terminate automatically upon termination of the login shell on the remote machine. If the connection is not terminated upon exit, it is likely that the **ptydaemon** on the remote system has either been terminated or restarted. To terminate a **vt** connection, enter command mode and use the **quit** command to terminate the connection.

**Commands**

**vt** recognizes the following commands. Commands can be abbreviated by typing enough of the command to uniquely identify it.

|                                                    |                                                                                                                                                                                                                                              |
|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>cd</b> <i>remote-directory</i>                  | Change the working directory on the remote machine to <i>remote-directory</i> . This command is applicable for file transfer only.                                                                                                           |
| <b>escape</b> [ <i>escape-char</i> ]               | Set the <b>vt</b> escape character. If a character is not specified <b>vt</b> prompts for one. To print or display the current <b>vt</b> escape character simply press Return in response to this prompt.                                    |
| <b>help</b> or <b>?</b>                            | Print a <b>vt</b> command summary.                                                                                                                                                                                                           |
| <b>lcd</b> [ <i>directory</i> ]                    | Change the local working directory. If no <i>directory</i> is specified, use the user's home directory. This command is applicable for file transfer and shell escapes only.                                                                 |
| <b>get</b> <i>remote-file local-file</i>           |                                                                                                                                                                                                                                              |
| <b>receive</b> <i>remote-file local-file</i>       | Copy <i>remote-file</i> to local machine and store as <i>local-file</i> . <b>vt</b> prompts for the file names if they are not specified. The file transfer can be aborted by typing the interrupt character or pressing the break key.      |
| <b>put</b> <i>local-file remote-file</i>           |                                                                                                                                                                                                                                              |
| <b>send</b> <i>local-file remote-file</i>          | Copy <i>local-file</i> to the remote machine and store as <i>remote-file</i> . <b>vt</b> prompts for the file names if they are not specified. The file transfer can be aborted by typing the interrupt character or pressing the break key. |
| <b>quit</b>                                        | Terminate the connection and exit <i>vt</i> .                                                                                                                                                                                                |
| <b>user</b> <i>user-name</i> [: <i>password</i> ]] | Identify yourself to the remote <b>vt</b> server. <b>vt</b> prompts for a password (after disabling local echo) if a colon (:) is appended to <i>user-name</i> . This command must be executed before any file transfer command can be used. |
| <b>!</b> [ <i>shellcommand</i> ]                   | Shell escape. The given command is passed to a sub-shell for execution. If no command is given, a shell is started and connected to the user's terminal.                                                                                     |

**Access Control Lists (ACLs)**

When sending or receiving files using **vt**, optional ACL entries are removed. New files have a summary of the access modes (as returned in **st\_mode** by **stat()** of the file being transferred (see *stat*(2)).

**DIAGNOSTICS**

The diagnostics produced by **vt** are intended to be self-explanatory.

**WARNINGS**

**vt** uses the Hewlett-Packard LLA (Link Level Access) direct interface to the HP network drivers. **vt** uses the multicast address **0x01AABBCCBBAA**. It should not be used or deleted by other applications accessing the network. **vt** uses the following IEEE 802.3 *sap* (service access point) values: **0x90**, **0x94**, **0x98**, **0x9C**, **0xA0**, **0xA4**, **0xA8**, **0xAC**, **0xB0**, **0xB4**, **0xB8**, **0xBC**, **0xC0**, **0xC4**, **0xC8**, **0xCC**, **0xD0**, and **0xD4**. They should not be used by other applications accessing the network.

When using **vt** on a system with multiple LAN cards installed, the optional command-line argument *lan\_device* may be required if the remote system is not accessible through the default LAN device. The appropriate *lan\_device* is the one connected (either directly or by way of other gateways) to the remote system.

**Desktop HP-UX**

If your system has been installed with the Desktop HP-UX product, neither **ptydaemon** nor **vtdaemon** will be started by default. To start these daemons, change *PTYDAEMON\_START* and *VTDAEMON\_START* from a **0** to a **1** in the */etc/rc.config.d/ptydaemon* and */etc/rc.config.d/vt* files, respectively. The system must be either rebooted for these changes to take effect, or you can start both daemons manually by typing the following commands:

```
/usr/sbin/ptydaemon
/usr/sbin/vtdaemon /dev/lan0
```

where */dev/lan0* is the character special device file corresponding to the IEEE 802.3 local area network device.

**FILES**

```
/dev/lan0 Default lan device name.
/etc/rc.config.d/ptydaemon
/etc/rc.config.d/vt
```

**SEE ALSO**

vtdaemon(1M), stat(2), lan(7), acl(5).

**NAME**

wait - await process completion

**SYNOPSIS**

**wait** [*pid*]

**DESCRIPTION**

If no argument is specified, **wait** waits until all processes (started with **&**) of the current shell have completed, and reports on abnormal terminations. If a numeric argument *pid* is given and is the process ID of a background process, **wait** waits until that process has completed. Otherwise, if *pid* is not a background process, **wait** exits without waiting for any processes to complete.

Because the **wait()** system call must be executed in the parent process, the shell itself executes **wait** without creating a new process (see *wait(2)*).

**Command-Line Arguments**

**wait** supports the following command line arguments:

**pid**           The unsigned decimal integer process ID of a command, whose termination **wait** is to wait for.

**WARNINGS**

Some processes in a 2-or-more-stage pipeline may not be children of the shell, and thus cannot be waited for.

**SEE ALSO**

*csh(1)*, *ksh(1)*, *sh-posix(1)*, *sh(1)*, *wait(2)*.

**STANDARDS CONFORMANCE**

**wait**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| w |**

**NAME**

wc - count words, lines, and bytes or characters in a file

**SYNOPSIS**

```
wc [-c | -m] [-l w] [file]...
```

**DESCRIPTION**

The **wc** command counts lines, words, and bytes or characters in the named files, or in the standard input if no file names are specified. It also keeps a total count for all named files.

A word is a string of characters delimited by spaces, tabs, or newlines.

**Options**

**wc** recognizes the following options:

- c Report the number of bytes in each input file.
- l Report the number of newline characters in each input file.
- m Report the number of characters in each input file.
- w Report the number of words in each input file.

The **c** and **m** options are mutually exclusive. Otherwise, the **l**, **w**, and **c** or **m** options can be used in any combination to specify that a subset of lines, words, and bytes or characters are to be reported.

When any option is specified, **wc** reports only the information requested. If no option is specified, the default output is **-lwc**.

When a *file* is specified on the command line, its name is printed along with the counts.

**Standard Output (XPG4 only)**

By default, the standard output contains an entry for each input file in the form:

```
newlines words bytes file
```

If the **-m** option is specified, the number of characters replaces the *bytes* field in this format.

If any option is specified, the fields for the unspecified options are omitted.

If no *file* operand is specified, neither the file name nor the preceding blank character is written.

If more than one *file* operand is specified, an additional line is written at the end of the output, of the same format as the other lines, except that the word **total** (in the POSIX locale) is written instead of a file name and the total of each column is written as appropriate.

**EXIT STATUS**

**wc** exits with one of the following values:

- 0 Successful completion.
- >0 An error occurred.

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the range of graphics and space characters, and the interpretation of text as single- and/or multibyte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is null, they default to the value of **LANG**.

If **LANG** is not specified or is null, it defaults to **C** (see *lang(5)*).

If any internationalization variable contains an invalid setting, they all default to **C**. See *environ(5)*.

**International Code Set Support**

Single- and multibyte character code sets are supported. with a newline character, the count will be off by one.

**WARNINGS**

The **wc** command counts the number of newlines to determine the line count. If a text file has a final line that is not terminated

**EXAMPLES**

Print the number of words and characters in **file1**:

```
wc -wm file1
```

The following is printed when the above command is executed:

```
words chars file1
```

where *words* is the number of words and *chars* is the number of characters in **file1**.

**STANDARDS CONFORMANCE**

**wc**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| w |**

**NAME**

what - get SCCS identification information

**SYNOPSIS**

**what** [-s] *file* ...

**DESCRIPTION**

The **what** command searches the given files for all occurrences of the pattern that *get(1)* substitutes for **%Z%** (currently **@(#)** at this printing) and prints out what follows until the first **", >**, new-line, **\**, or null character. For example, if the C program in file **f.c** contains

```
char ident[] = "@(#)identification information";
```

and **f.c** is compiled to yield **f.o** and **a.out**, the command

```
what f.c f.o a.out
```

prints

```
f.c: identification information
```

```
f.o: identification information
```

```
a.out: identification information
```

**what** is intended to be used in conjunction with the SCCS **get** command (see *get(1)*) which automatically inserts identifying information, but it can also be used where the information is inserted manually.

**Options**

**what** recognizes the following option:

```
-s Quit after finding the first occurrence of pattern in each file.
```

**EXTERNAL INFLUENCES****Environment Variables**

**LC\_CTYPE** determines the interpretation of the pattern substituted for **%Z%** as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang(5)*) is used instead of **LANG**. If any internationalization variable contains an invalid setting, **what** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single-byte and multi-byte character code sets are supported.

**DIAGNOSTICS**

Exit status is 0 if any matches are found, otherwise 1. Use **help** for explanations (see *sccshelp(1)*).

**W****WARNINGS**

The pattern **@(#)** may occasionally appear unintentionally in random files, but this causes no harm in nearly all cases.

**SEE ALSO**

*get(1)*, *sccshelp(1)*.

**STANDARDS CONFORMANCE**

**what**: SVID2, SVID3, XPG2, XPG3, XPG4

**NAME**

whereis - locate source, binary, and/or manual for program

**SYNOPSIS**

```
whereis [-bsm] [-u] [-BMS dir ... -f] name ...
```

**DESCRIPTION**

**whereis** locates source, binary, and manuals sections for specified files. The supplied names are first stripped of leading path name components and any (single) trailing extension of the form *.ext* (such as *.c*). Prefixes of *s.* resulting from use of SCCS are also dealt with. **whereis** then attempts to locate the desired program in a list of standard places.

**Options**

**whereis** recognizes the following command-line options:

- b** Limit search to binary files. Can be used in conjunction with **-s** or **-m**.
- s** Limit search to source-code files. Can be used in conjunction with **-b** or **-m**.
- m** Limit search to manual entry files. Can be used in conjunction with **-b** or **-s**.
- u** Search for unusual entries. A file is said to be unusual if it does not have one entry of each requested type. Thus, **whereis -m -u \*** searches for those files in the current directory that have no corresponding manual entry.
- B dir [dir ...]** Limit search to binaries located in one or more specified directories. Can be used in conjunction with **-S** or **-M**.
- S dir [dir ...]** Limit search to source files located in one or more specified directories. Can be used in conjunction with **-B** or **-M**.
- M dir [dir ...]** Limit search to manual entry files located in one or more specified directories. Can be used in conjunction with **-B** or **-S**.
- f** Terminates the last directory list (**-B**, **-S**, or **-M** options) and identifies the start of file names.

**EXTERNAL INFLUENCES****International Code Set Support**

Single- and multi-byte character code sets are supported.

**EXAMPLES**

Find all the files in `/usr/bin` that are not documented in `/usr/share/man/man1` with source files in `/usr/src/cmd`:

```
cd /usr/bin
whereis -u -M /usr/share/man/man1 -S /usr/src/cmd -f *
```

**WARNINGS**

**whereis** uses the `chdir()` system call (see `chdir(2)`) to run faster. Therefore, path names given with the **-B**, **-M**, and **-S** options must be absolute path names.

**AUTHOR**

**whereis** was developed by the University of California, Berkeley.

**FILES**

```
/usr/src/*
/usr/sbin, /sbin, /usr/bin, /usr/sbin, /usr/ccs/bin
/usr/share/man/*
/usr/local/{man/*, bin, games, include, lib}
/usr/contrib/{man/*, bin, games, include, lib}
/usr/share/man/$LANG/*
/usr/local/man/$LANG/*
/usr/contrib/man/$LANG/*
```

| **W** |

**NAME**

which - locate a program file including aliases and paths

**SYNOPSIS**

**which** [*name* ...]

**DESCRIPTION**

For each *name* given, **which** searches for the file that would be executed if *name* were given as a command, and displays the absolute path of that file. Each argument is expanded if it is aliased, and searched for along the user's path. Both aliases and path are determined by sourcing (executing) the user's **.cshrc** file.

**DIAGNOSTICS**

A diagnostic is given for names that are aliased to more than a single word, or if an executable file with the argument name was not found in the path.

**EXAMPLES**

The command:

```
which sh
```

specifies where the executable program of the *sh*(1) command is found. For example, the response might be:

```
/usr/bin/sh
```

if the *sh*(1) being used is located in **/usr/bin**.

**WARNINGS**

**which** reports **.cshrc** aliases even when not invoked from *csh*.

**which** cannot find **csh** built-in commands (e.g. jobs).

**which**'s information may be incorrect because it is unaware of any path or alias changes that have occurred in the current shell session.

**AUTHOR**

**which** was developed by the University of California, Berkeley.

**FILES**

**~/ .cshrc** source of aliases and path values

| **W** |



**NAME**

who - who is on the system

**SYNOPSIS**

who [-muTlHqpdbrtasARW] [*file*]

who am i

who am I

**DESCRIPTION**

The **who** command can list the user's name, terminal line, login time, elapsed time since input activity occurred on the line, the user's host name, and the process-ID of the command interpreter (shell) for each current system user. It examines the **utmps** database to obtain the information. If *file* is given, that file is examined, *file* should be a **utmp** like file.

The **who** command with the **am i** or **am I** option identifies the invoking user.

Except for the default **-s** option, the general format for output entries is:

```
name [state] line time activity pid [comment] [exit]
```

With options, **who** can list logins, logoffs, reboots, and changes to the system clock, as well as other processes spawned by the **init** process.

**Options**

- m** Output only information about the current terminal. This option is equivalent to the **am i** and **am I** options described above.
- u** Lists only those users who are currently logged in. *name* is the user's login name. *line* is the name of the line as found in the directory **/dev**. The *time* field indicates when the user logged in.  
  
*activity* is the number of hours and minutes since input activity last occurred on that particular line. A dot (.) indicates that the terminal has seen activity in the last minute and is therefore "current". If more than twenty-four hours have elapsed or the line has not been used since boot time, the entry is marked **old**. This field is useful when trying to determine whether a person is working at the terminal or not. The *pid* is the process-ID of the user's login process. The *comment* is the comment field associated with this line as found in **/etc/inittab** (see *inittab*(4)). This can contain information about where the terminal is located, the telephone number of the dataset, type of terminal if hard-wired, etc. If no such information is found, then **who** prints, as the *comment*, the user's host name as it was stored in the **utmps** database or named *file*. Note that the user's host name is printed instead of comments from the **/etc/inittab** file if the **-u** option is used in conjunction with the **-R** option.
- T** Same as the **-u** option, except that the *state* of the terminal line is printed. *state* describes whether someone else can write to that terminal. A **+** appears if the terminal is writable by anyone; a **-** appears if it is not. **root** can write to all lines having a **+** or a **-** in the *state* field. If a bad line is encountered, a **?** is printed.  
  
(XPG4 only.) Only the following fields are displayed: *name state line time*
- l** Lists only those lines on which the system is waiting for someone to login. The *name* field is **LOGIN** in such cases. Other fields are the same as for user entries except that the *state* field does not exist.
- H** Prints column headings above the regular output.
- q** A quick **who**, displaying only the names and the number of users currently logged in. When this option is used, all other options are ignored.
- p** Lists any other process which is currently active and has been previously spawned by *init*. The *name* field is the name of the program executed by **init** as found in **/etc/inittab**. The *state*, *line*, and *activity* fields have no meaning. The *comment* field shows the *id* field of the line from **/etc/inittab** that spawned this process. See *inittab*(4).

| **W** |

- d** This option displays all processes that have expired and have not been respawned by **init**. The *exit* field appears for dead processes and contains the termination and exit values of the dead process (as returned by **wait()** — see *wait(2)*). This can be useful in determining why a process terminated.
- b** Indicates the time and date of the last reboot.
- r** Indicates the current *run-level* of the **init** process. The last three fields contain the current state of **init**, the number of times that state has been previously entered, and the previous state. These fields are updated each time **init** changes to a different run state.
- t** Indicates the last change to the system clock (via the **date** command) by **root**. See *su(1)*.
- a** Processes **utmps** database or the named *file* with all options turned on.
- s** Default. Lists only the *name*, *line*, and *time* fields.
- A** When the */var/adm/wtmp* file is specified, (the **-W** option can be used to examine the */var/adm/wtmp* file) this option indicates when the accounting system was turned on or off using the **startup** or **shutacct** commands (see *acctsh(1M)*). The *name* field is a dot (.). The *line* field is **acctg on**, **acctg off**, or a reason that was given as an option to the **shutacct** command. The *time* is the time that the on/off activity occurred.
- R** Displays the user's host name. If the user is logged in on a tty, **who** displays the string returned from **gethostname()** (see *gethostname(2)*). If the user is not logged in on a tty and the host name stored in the **utmps** database or named **utmp** like file has not been truncated when stored (meaning that the entire host name was stored with no loss of information), it is displayed as it was stored. Otherwise, the **gethostbyaddr()** (IPv4) or **getipnodebyaddr()** (IPv6) function is called with the internet address of the host (see *gethostent(3N)*). The host name returned by **gethostbyaddr()** (IPv4) or **getipnodebyaddr()** (IPv6) is displayed unless it returns an error, in which case the truncated host name is displayed.
- W** Gets the information from */var/adm/wtmp* file.

(XPG4 only. The **-s** option can not be used with **-d**, **-a** or **-T** options. If **-u** option is used with **-T**, the idle time is added to the end of the **-T** format.)

## EXTERNAL INFLUENCES

### Environment Variables

**LANG** determines the locale to use for the locale categories when both **LC\_ALL** and the corresponding environment variable (beginning with **LC\_**) do not specify a locale. If **LANG** is not set or is set to the empty string, a default of "C" (see *lang(5)*) is used.

**LC\_CTYPE** determines the locale for interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files).

**LC\_TIME** determines the format and contents of date and time strings.

**LC\_MESSAGES** determines the language in which messages are displayed.

If any internationalization variable contains an invalid setting, **who** behaves as if all internationalization variables are set to "C". See *environ(5)*.

### International Code Set Support

Single- and multi-byte character code sets are supported.

## EXAMPLES

Check who is logged in on the system:

```
who
```

Check whether or not you can write to the terminal that another user is using:

```
who -T
```

and look for a plus (+) after the user ID.

**AUTHOR**

**who** was developed by AT&T and HP.

**FILES**

**/etc/inittab**  
**/etc/utmp**  
**/var/adm/wtmp**  
**/var/adm/wtmps**

**SEE ALSO**

**date(1)**, **login(1)**, **mesg(1)**, **su(1)**, **init(1M)**, **utmpd(1M)**, **gethostname(2)**, **wait(2)**, **gethostent(3N)**, **getutsent(3C)**, **getbwent(3C)**, **inittab(4)**, **utmp(4)**.

**STANDARDS CONFORMANCE**

**who**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| w |**

**NAME**

whoami - print effective current user id

**SYNOPSIS**

whoami

**DESCRIPTION**

**whoami** prints your *current* user name, even if you have used **su** to change it since your initial login (see *su(1)*). The command **who am i** reports your initial login name because it uses **/etc/utmp**.

**FILES**

**/etc/passwd** name data base

**AUTHOR**

**whoami** was developed by the University of California, Berkeley.

**SEE ALSO**

who(1).

| **w** |

**NAME**

whois - Internet user name directory service

**SYNOPSIS**

**whois** [-h *hostname*] *name*

**DESCRIPTION**

**whois** looks up records in the Network Information Center database.

The operands specified to **whois** are concatenated together (separated by white-space) and presented to the **whois** server.

The default action, unless directed otherwise with a special name, is to do a very broad search, looking for matches to name in all types of records and most fields (name, nicknames, hostname, net address, etc.) in the database. For more information as to what name operands have special meaning, and how to guide the search, use the special name "help".

**Options**

**whois** supports the following option:

**-h** use the specified host instead of the default NIC.

**EXAMPLES**

Look up user John Doe

**whois** *Doe,John*

**DEPENDENCIES**

The machine making the **whois** enquiry must be able to route the request over the network to the specified host, or to the default NIC.

**AUTHOR**

**whois** was developed by the University of California, Berkeley.

**SEE ALSO**

RFC 854: Nicname/Whois

| **w** |

**NAME**

write - interactively write (talk) to another user

**SYNOPSIS**

**write** *user* [*terminal*]

**DESCRIPTION**

The **write** command copies lines from your terminal to that of another user. When first called, it sends the message:

**Message from** *yourname* (*yourterminal*) [ *date* ] ...

to the receiving *user*'s terminal. When it has successfully completed the connection, it also sends two bells to your own terminal to indicate that what you are typing is being sent.

To set up two-way communication, the recipient of the message (*user*) must execute the command:

**write** *yourname* [*yourterminal*]

(*yourterminal* is only required if the originator is logged in more than once.)

Communication continues until an end of file is read from the terminal, an interrupt is sent, or the recipient executes **mesg n**. At that point, **write** writes <EOT> on the other terminal and exits.

To write to a user who is logged in more than once, use the *terminal* argument to indicate which line or terminal to send to (e.g., **tty00**). Otherwise, the first writable instance of the user found in **utmps** database is assumed and the following message is displayed:

**user is logged on more than one place.**  
**You are connected to "terminal".**  
**Other locations are:**  
*terminal*  
 ...

Permission to write may be denied or granted with the **mesg** command (see *mesg(1)*). Writing to others is normally allowed by default. Certain commands, in particular **nroff** and **pr** disallow messages in order to prevent interference with their output. However, if the user has the appropriate privileges, messages can be forced onto a write-inhibited terminal.

If the character **!** is found at the beginning of a line, **write** calls the POSIX shell (see *sh-posix(1)*) to execute the rest of the line as a command.

The following protocol is suggested for using **write**: When you first **write** to another user, wait for the user to **write** back before starting to send. Each person should end a message with a distinctive signal (such as "(o)" for "over") so that the other person knows when to reply. Similarly, the signal "(oo)" (for "over and out") can be used to indicate the end of the conversation.

**EXTERNAL INFLUENCES****Environment Variables**

**LANG** determines the locale to use for the locale categories when both **LC\_ALL** and the corresponding environment variable (beginning with **LC\_**) do not specify a locale. If **LANG** is not set or is set to the empty string, a default of "C" (see *lang(5)*) is used.

**LC\_TIME** determines the format and contents of date and time strings.

**LC\_MESSAGES** determines the language in which messages are displayed.

If any internationalization variable contains an invalid setting, **write** behaves as if all internationalization variables are set to "C". See *environ(5)*.

**International Code Set Support**

Single- and multi-byte character code sets are supported.

**DIAGNOSTICS**

**user is not logged on.**

The user you are trying to write to is not logged on.

**Can no longer write to terminal**

Your correspondent has denied write permission (**mesg n**) after your **write** session started. Your **write** session is ended.

**<EOT>**

Your correspondent sent end-of-file, or you set your terminal to **mesg n** and your correspondent tried to write to you. If you have a **write** session established, you can continue to write to your correspondent.

**Permission denied.**

The user you are trying to write to has denied write permission (with **mesg n**).

**Warning: You have your terminal set to "mesg -n". No reply possible.**

Your terminal is set to **mesg n** and the recipient cannot respond to you.

**EXAMPLES**

By issuing the command:

```
write matthew
```

user **linda** sends a message to user **matthew**'s screen. If **matthew** responds:

```
write linda
```

two-way communication between **matthew** and **linda** is established.

**FILES**

|                       |                             |
|-----------------------|-----------------------------|
| <b>utmps</b> database | To find user                |
| <b>/usr/bin/sh</b>    | To execute ! shell commands |

**SEE ALSO**

elm(1), mail(1), mailx(1), mesg(1), nroff(1), pr(1), sh-posix(1), sh(1), who(1), utmpd(1M), gettysent(3C).

**STANDARDS CONFORMANCE**

**write**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

**| w |**

**NAME**

xargs - construct argument list(s) and execute command

**SYNOPSIS**

**xargs** [*options*] [*command* [*initial-arguments*]]

**DESCRIPTION**

**xargs** combines the fixed *initial-arguments* with arguments read from standard input to execute the specified *command* one or more times. The number of arguments read for each *command* invocation and the manner in which they are combined are determined by the options specified.

*command*, which can be a shell file, is searched for, using the **\$PATH** environment variable. If *command* is omitted, **/usr/bin/echo** is used.

Arguments read in from standard input are defined to be contiguous strings of characters delimited by one or more blanks, tabs, or new-lines; empty lines are always discarded. Spaces and tabs can be embedded as part of an argument if escaped or quoted. Characters enclosed in quotes (single or double) are taken literally, and the delimiting quotes are removed. Outside of quoted strings, a backslash (\) escapes the next character.

The amount of memory available for the execution of *command* is limited by the system parameter **ARG\_MAX**. By default, the size of the argument list is limited to **LINE\_MAX** bytes. See *limits(5)* and *sysconf(2)* for a description of these system parameters and how their values can be determined. To increase the available argument list space, use the **-s** option.

Each argument list is constructed starting with the *initial-arguments*, followed by some number of arguments read from standard input (exception: see **-i** or **-I** option). The **-i**, **-I**, **-l**, **-L**, and **-n** options determine how arguments are selected for each command invocation. When none of these options is specified, the *initial-arguments* are followed by arguments read continuously from standard input until an internal buffer is full, then *command* is executed with the accumulated args. This process is repeated until there are no more args. When there are option conflicts (such as **-l** or **-L** versus **-n**), the last option has precedence. *option* values are:

- L** *number*     *command* is executed for each non-empty *number* lines of arguments from standard input. The last invocation of *command* will be with fewer lines of arguments if fewer than *number* remain. A line is considered to end with the first new-line unless the last character of the line is a blank or a tab; a trailing blank/tab signals continuation through the next non-empty line. The **-L**, **-l**, and **-n** options are mutually exclusive. The last one specified takes effect.
- l**[ *number* ] This option is equivalent to the **-L** option. **1** is assumed if *number* is omitted or is given as the empty string ("" ). Option **-x** is forced.
- I** *replstr*     Insert mode: *command* is executed for each line from standard input, taking the entire line as a single arg, inserting it in *initial-arguments* for each occurrence of *replstr*. A maximum of 5 arguments in *initial-arguments* can each contain one or more instances of *replstr*. Blanks and tabs at the beginning of each line are discarded. Constructed arguments must not grow larger than 255 bytes, and option **-x** is also forced. The **-I** and **-i** options are mutually exclusive. The last one specified takes effect.
- i**[ *repstr* ]   This option is equivalent to the **-I** option. { } is assumed if *repstr* is omitted or is given as the empty string ("" ).
- n** *number*     Execute *command* using as many standard input arguments as possible, up to *number* arguments maximum. Fewer arguments are used if their total size is greater than *size* bytes, and for the last invocation if there are fewer than *number* arguments remaining. If option **-x** is also coded, each *number* arguments must fit in the *size* limitation or **xargs** terminates execution.
- s** *size*        The maximum total size of each argument list is set to *size* bytes; *size* must be a positive integer less than **LINE\_MAX** (see *limits(5)*, *sysconf(2)*). If **-s** is not coded, **LINE\_MAX** is taken as the default. Note that the bytes count for *size* includes one extra bytes for each argument and the count of bytes in the command name.
- t**              Trace mode: The *command* and each constructed argument list are echoed to standard error just prior to their execution.

| X |



- p** Prompt mode: The user is asked whether to execute *command* prior to each invocation. Trace mode (**-t**) is turned on to print the command instance to be executed, followed by a **?...** prompt. An affirmative reply (by default, an affirmative reply is **y** optionally followed by anything) executes the command; anything else, including pressing Return, skips that particular invocation of *command*.
- x** Causes *xargs* to terminate if any argument list would be greater than *size* bytes. **-x** is forced by the options **-i**, **-I**, **-l**, and **-L**. When none of the options **-i**, **-I**, **-l**, **-L**, or **-n** is coded, the total length of all arguments must be within the *size* limit.
- e[ eofstr ]** *eofstr* is taken as the logical end-of-file string. Underscore (**\_**) is assumed for the logical **EOF** string if neither **-e** nor **-E** is used. The value **-e** with *eofstr* given as the empty string (**""**) turns off the logical **EOF** string capability (underscore is taken literally). *xargs* reads standard input until either end-of-file or the logical **EOF** string is encountered.
- E eofstr** Specify a logical end-of-file string to replace the default underscore (**\_**) character. Equivalent to the **-e** option above.

*xargs* terminates if it receives a return code of **-1** from *command* or if it cannot execute *command*. When *command* is a shell program, it should explicitly **exit** (see *sh*(1)) with an appropriate value to avoid accidentally returning with **-1**.

## RETURN VALUE

**xargs** exits with one of the following values:

- 0** All invocations of *command* completed successfully.
- 1-125** One or more invocations of *command* did not complete successfully.
- 126** The *command* specified was found but could not be invoked.
- 127** The *command* specified could not be found.

## EXTERNAL INFLUENCES

### Environment Variables

**LC\_CTYPE** determines the space characters and the interpretation of text as single- and/or multi-byte characters.

**LC\_MESSAGES** determines the language in which messages are displayed, and the local language equivalent of an affirmative reply when the **-p** prompt option is specified.

If **LC\_CTYPE** or **LC\_MESSAGES** is not specified in the environment or is set to the empty string, the value of **LANG** is used as a default for each unspecified or empty variable. If **LANG** is not specified or is set to the empty string, a default of "C" (see *lang*(5)) is used instead of **LANG**.

If any internationalization variable contains an invalid setting, **xargs** behaves as if all internationalization variables are set to "C". See *environ*(5).

### International Code Set Support

Single- and multi-byte character code sets are supported.

## EXAMPLES

Move all files from directory **\$1** to directory **\$2**, and echo each move command just before doing it:

```
ls $1 | xargs -i -t mv $1/{ } $2/{ }
```

Combine the output of the parenthesized commands onto one line, then echo to the end of file **log**:

```
(logname; date; echo $0 $*) | xargs >>log
```

Ask the user which files in the current directory are to be archived then archive them into **arch** one at a time:

```
ls | xargs -p -l ar r arch
```

or many at a time:

```
ls | xargs -p -l | xargs ar r arch
```

Execute **diff** (see *diff(1)*) with successive pairs of arguments originally typed as shell arguments:

```
echo $* | xargs -n2 diff
```

**SEE ALSO**

sh(1).

**STANDARDS CONFORMANCE**

**xargs**: SVID2, SVID3, XPG2, XPG3, XPG4, POSIX.2

| **x** |

**NAME**

**xstr** - extract strings from C programs to implement shared strings

**SYNOPSIS**

```
xstr [-c] [-] [file]
```

**DESCRIPTION**

**xstr** maintains a file **strings** into which strings in component parts of a large program are hashed. These strings are replaced with references to this common area. This serves to implement shared constant strings, which are most useful if they are also read-only.

The command:

```
xstr -c name
```

extracts the strings from the C source in *name*, replacing string references with expressions of the form **(&xstr[*number*])** for some *number*. An appropriate declaration of **xstr** is placed at the beginning of the file. The resulting C text is placed in the file **x.c**, for subsequent compiling. The strings from this file are placed in the **strings** database if they are not there already. Repeated strings and strings that are suffixes of existing strings do not cause changes to the data base.

After all components of a large program have been compiled, a file **xs.c** declaring the common **xstr** space, can be created by the command:

```
xstr
```

This **xs.c** file should then be compiled and loaded with the rest of the program. If possible, the array can be made read-only (shared), saving space and swap overhead.

**xstr** can also be used on a single file. A command:

```
xstr name
```

creates files **x.c** and **xs.c** as before, without using or affecting any **strings** file in the same directory.

It may be useful to run **xstr** after the C preprocessor if any macro definitions yield strings or if there is conditional code containing strings that are not, in fact, needed. **xstr** reads from its standard input when the argument **-** is given. An appropriate command sequence for running **xstr** after the C preprocessor is:

```
cc -E name.c | xstr -c -
cc -c x.c
mv x.o name.o
```

**xstr** does not touch the file **strings** unless new items are added, thus **make** can avoid remaking **xs.o** unless truly necessary (see *make(1)*).

**WARNINGS**

If a string is a suffix of another string in the data base, but the shorter string is seen first by **xstr**, both strings are placed in the data base, when placing only the longer one there would be sufficient.

**AUTHOR**

**xstr** was developed by the University of California, Berkeley.

**FILES**

|                 |                                                          |
|-----------------|----------------------------------------------------------|
| <b>strings</b>  | Data base of strings                                     |
| <b>x.c</b>      | Massaged C source                                        |
| <b>xs.c</b>     | C source for definition of array <b>xstr</b>             |
| <b>/tmp/xs*</b> | Temp file when 'xstr name' does not touch <b>strings</b> |

**SEE ALSO**

mkstr(1).

| **X** |

**yes(1)**

**yes(1)**

**NAME**

yes - be repetitively affirmative

**SYNOPSIS**

**yes** [*expletive* ]

**DESCRIPTION**

**yes** repeatedly outputs **y**, or if *expletive* is given, the *expletive* is output repeatedly. Termination is by interrupt.

**AUTHOR**

**yes** was developed by the University of California, Berkeley.

| **y** |

**NAME**

ypcat - print all values in Network Information Service map

**SYNOPSIS**

**ypcat** [-k] [-t] [-d *domain*] *mname*

**ypcat** -x

**Remarks**

The Network Information Service (NIS) was formerly known as Yellow Pages (YP). The functionality of the two remains the same; only the name has changed.

**DESCRIPTION**

**ypcat** prints all values in a Network Information Service (NIS) map specified by *mname*, which can be either a *mapname* or a map *nickname*. A map *nickname* is a synonym by which a NIS map can be referenced. Values are listed, one per line.

**Options**

**ypcat** recognizes the following options:

- k Print the associated key preceding each value. This option is useful for examining maps in which the values are null or the keys are not part of the value, such as the *ypservers* map. The maps derived from files that have an ASCII version in */etc* (such as **passwd** and **hosts**) are not in this category.
- t Inhibit the translation of a map's *nickname* to its corresponding *mapname*. For example, **ypcat -t passwd** fails because there is no map named **passwd**, whereas **ypcat passwd** translates to **ypcat passwd.byname**.
- d Specify a *domain* other than the one returned by **domainname** (see *domainname(1)*).
- x Display the table that lists the *nickname* for each NIS map.

**EXAMPLES**

Display the network-wide password database whose *mapname* is **passwd.byname** and *nickname* is **passwd** :

```
ypcat passwd
```

**AUTHOR**

**ypcat** was developed by Sun Microsystems, Inc.

**SEE ALSO**

*domainname(1)*, *ypmatch(1)*, *ypserv(1M)*, *ypfiles(4)*.

**NAME**

ypmatch - print values of selected keys in Network Information Service map

**SYNOPSIS**

```
ypmatch [-k] [-t] [-d domain] key [key...] mname
ypmatch -x
```

**Remarks**

The Network Information Service (NIS) was formerly known as Yellow Pages (yp). Although the name has changed, the functionality of the service remains the same.

**DESCRIPTION**

**ypmatch** prints the values associated with one or more keys in a Network Information Service (NIS) map specified by *mname*. The *mname* can be either a *mapname* or a map *nickname*. A map *nickname* is a synonym by which a NIS map can be referenced.

If multiple keys are specified, the same map is searched for an occurrence of each key. A match is made only when the case and length of a key is the same as that stored in the database. No pattern matching is available. If a key is not matched, a diagnostic message is produced.

**Options**

**ypmatch** recognizes the following command-line options:

- k** Before printing the value associated with a key, print the key followed by a colon (:). This option is useful if the keys are not part of the values (as in a **ypservers** map), or so many keys were specified that the output could be confusing.
- t** Inhibit the translation of a map's *nickname* to its corresponding *mapname*. For example, **ypmatch -t zippy passwd** fails because there is no map named **passwd**, while **ypmatch zippy passwd** is translated to **ypmatch zippy passwd.byname**.
- d** Specify a *domain* other than the one returned by **domainname** (see *domainname*(1)).
- x** Display the table that lists the *nickname* for each NIS map.

**AUTHOR**

**ypmatch** was developed by Sun Microsystems, Inc.

**SEE ALSO**

*domainname*(1), *ypcat*(1), *ypserv*(1M), *ypfiles*(4).

|y|

**NAME**

yppasswd - change login password in Network Information System (NIS)

**SYNOPSIS**

**yppasswd** [*name*]

**Remarks**

The Network Information Service (NIS) was formerly known as Yellow Pages (YP). The functionality remains the same; only the name has changed.

**DESCRIPTION**

**yppasswd** changes or installs a password associated with the login *name* in the Network Information System (NIS). The NIS password can be different from the one on your own machine. If *name* is omitted, it defaults to the name returned by **getlogin()** (see *getlogin(3C)*).

**yppasswd** prompts for the old NIS password (even if it does not exist), then twice for the new one. The old password must be entered correctly for the change to take effect. Checks occur to ensure that the new password meets the following construction requirements.

- Only the first eight characters are significant.
- A password can be as few as four characters long if it contains
  - at least one special character or
  - a mixture of numeric, uppercase and lowercase letters.
- A password can be as few as five characters long if it contains a mixture of
  - uppercase and lowercase letters or
  - numeric and either uppercase or lowercase letters.
- A password must contain at least six characters if it contains only monospace letters.

All these rules except the first are relaxed if you try three times to enter an unacceptable new password. You cannot, however, enter a null password.

Only the owner of the *name* or the superuser can change a password.

The Network Information System password daemon, *yppasswdd(1M)*, must be running on the master NIS password server to change NIS passwords.

**WARNINGS**

The password update protocol passes the old and new passwords to the master NIS server at once. Thus, if the old NIS password is incorrect, no notification is given until the new NIS password is successfully entered.

The **yppasswd** password construction rules are different from those of the HP-UX **passwd** command (see *passwd(1)*).

User applications that call this routine must be linked with `/usr/include/librpcsvc.a`. For example,

```
cc my_source.c -lrpcsvc
```

**AUTHOR**

**yppasswd** was developed by Sun Microsystems, Inc.

**SEE ALSO**

*id(1)*, *passwd(1)*, *su(1)*, *yppasswdd(1M)*, *getlogin(3C)*, *yppasswd(3N)*, *ypfiles(4)*.

|y|

**NAME**

ypwhich - list which host is Network Information System server or map master

**SYNOPSIS**

ypwhich

ypwhich [-d *domain*] [-v1 | -v2] [*hostname*]

ypwhich [-d *domain*] [-t] [-m [*mname*]]

ypwhich -x

**Remarks**

The Network Information Service (NIS) was formerly known as Yellow Pages (yp). Although the name has changed, the functionality of the service remains the same.

**DESCRIPTION**

**ypwhich** lists the host name of the Network Information System (NIS) server that supplies NIS services to a NIS client. It can also print the NIS server that is the master for *mname*. The *mname* can be either a *mapname* or a map *nickname*. A map *nickname* is a synonym by which a NIS map can be referenced.

If invoked without arguments, **ypwhich** prints the host name of the NIS server serving the local machine. If *hostname* is specified, that machine is queried to determine which NIS server it is using.

**Options**

**ypwhich** recognizes the following command-line options and arguments:

- d** Specify a *domain* other than the one returned by *domainname*(1).
- v1** List the server that is serving Version 1 NIS protocol-speaking client processes.
- v2** List the server that is serving Version 2 NIS protocol-speaking client processes.  
If neither version is specified, **ypwhich** locates the server supplying the Version 2 (current) services. However, if no Version 2 server is found, **ypwhich** attempts to locate the server supplying the Version 1 services. Since NIS servers and NIS clients are both backward compatible, the user seldom needs to know which version is being used.
- t** Inhibit the translation of a map's *nickname* to its corresponding *mapname*. For example, **ypwhich -t -m passwd** fails because there is no map named **passwd**, whereas **ypwhich -m passwd** translates to **ypwhich -m passwd.byname**. This option is useful if a *mapname* is identical to a *nickname* (which is not true of any HP map).
- m [*mname*]** List the master NIS server for a map. No *hostname* can be specified with **-m**. The *mname* can be a *mapname* or a map *nickname*. If *mname* is omitted, a complete list of available maps and the corresponding host names of the master NIS servers is produced.
- x** Display the table that lists the *nickname* for each NIS map.

**AUTHOR**

**ypwhich** was developed by Sun Microsystems, Inc.

**SEE ALSO**

domainname(1), ypserv(1M), ypset(1M), ypfiles(4).

|y|