WEBSITE REFERENCE

Command-Line Tool Doc Tables

Throughout *Mac OS X Unleashed*, you learned a number of different commands that can be used to interact with your system. There are literally hundreds of shell commands and utilities that can be used with the Mac OS X distribution, and, unfortunately, there simply isn't enough space to provide information on them all. This appendix provides an alphabetical reference to some of the more useful and/or interesting BSD commands.

As Apple updates Mac OS X, it's likely to add additional functionality behind the scenes. In the first few revisions, Apple added a number of security features accessible only from the command line. Keep track of the files added during each update to find out whether there are any new utilities available for your use. For more information on viewing the contents of system updates, see Chapter 32, "System Maintenance."

If you can't find what you're looking for here, remember that you have instant access to command documentation through the use of the `apropos` and `man` utilities.

apropos

`apropos`	Displays a list of manual pages by keyword lookup.
`apropos <keyword1> <keyword2> ...`	Looks up commands with any `<keywords>` in their description.

at, atq, atrm, batch

`at`	Executes commands at a specified time.
`atq`	Lists the user's pending jobs, unless the user is superuser. If the user is superuser, lists all users' jobs.
`atrm`	Deletes jobs.
`batch`	Executes commands as soon as system load levels permit. This is either when the average load drops to below 1.5 or the value specified at the invocation of `atrun`.

Using any of these commands requires the configuration of the `atrun` command in root's `crontab`. Read the `atrun` man page (following) for setup information.

```
at [-q <queue>] [-f <file>] [-m] <time>
atq [-q <queue>] [-v]
atrm [-q <queue>] <job> [<job2>...]
batch [-f <file>] [-m]
```

Both `at` and `batch` take input from either standard input or the file specified by `-f` option. The working directory, environment (except for variables TERM, TERMCAP, DISPLAY, and _), and `umask` are retained from the time of invocation. Any `at` or `batch` command invoked from an `su` shell retains the current user ID.

Permission to use these commands depends on the files `/var/at/at.allow` and `/var/at/at.deny`. The superuser may use these commands. If `/var/at/at.allow` exists, only the users (one per line) listed in the file may use these commands. If `/var/at/at.allow` does not exist, `/var/at/at.deny` is checked. Only users listed in `/var/at/at.deny` may not use these commands. If an empty `/var/at/at.deny` exists, all users may use these commands. If neither file exists, only the superuser may use these commands.

`-q <queue>`	Uses the specified queue. A queue consists of a single letter. Valid queue ranges are `a` to `l`. The `a` queue is the default and `b` is the batch queue. Queues with higher letters run with increased niceness. If `atq` is given a specific queue, it shows only the pending jobs in the specified queue.
`-f <file>`	Executes commands in the specified `<file>` rather than from standard input.
`-m`	Sends mail to the user when the job is complete, whether or not there was any output.
`-v`	For `atq`, shows completed, but not yet deleted, jobs in the queue. Otherwise, shows the time the job will be executed.
`<time>`	`<time>` may be given in a variety of formats. Times may be of the form `<HHMM>` or `<HH:MM>` for a specific time of day. If the time has already passed, the next day is assumed. You can also specify midnight, noon, or teatime (teatime for 4:00pm). You can append AM, am, PM, or pm to a specific time. A time may also include a date in any of the following forms: `<month-name>` `<day>` [`<year>`] or `MMDDYY` or `MM/DD/YY` or `DD.MM.YY`. The date must follow the time specification. Time may also be given in increments, such as `<now>` + `<count><time_units>`, where `<time_units>` can be minutes, hours, days, or weeks. Terms *today* and *tomorrow* may also be used.

atrun

`atrun` Runs jobs queued by `at`.

```
atrun [-l <load_avg>] [-d]
```

`atrun` runs commands queued by `at`. `root`'s crontab (`/etc/crontab`) must contain this line:

```
*/10 * * * * root /usr/libexec/atrun
```

so that `atrun` is called every 10 minutes. By default, the `atrun` line isn't installed to prevent disk access every 10 minutes. Enter the `atrun` line in order to use `at`.

At every invocation, every job whose start time has passed is started. A maximum of one batch job is started.

For `atrun` to work, a `cron` daemon must also be running.

`-l <load_avg>` Specifies a limiting load average, over which batch runs should not be run, instead of the default value of 1.5.

`-d` Debug mode. Prints error messages to standard error instead of using syslog.

automount

`automount` Automatic NFS mount/unmount daemon.

`automount` is a daemon that automatically mounts NFS file systems when they're first accessed and later unmounts them when they're idle.

`automount` creates a virtual file system mounted at one or more places on the client's file and directory hierarchy. Actual NFS mount points within this virtual file system appear as symbolic links. Reading a symbolic link triggers `automount` to mount the associated remote file system.

To make the trigger symbolic links used by `automount` distinguishable from normal symbolic links, the sticky bit is set in the mode flags for the link. Programs that would normally traverse symbolic links can test for this bit and avoid triggering the mount. Workspace Manager and `ls` have been modified in this way.

Each virtual file system created by `automount` is governed by a corresponding map. One or more maps may be specified on the command line.

A map may be a file or a special map. A file map is a regular file containing a list of entries of the following form:

```
location mount_options server:path
```

`mount_options` is a comma-separated list of options from the options known to `mount` and `mount_nfs` programs.

In addition to reading files specifying mount maps, `automount` supports the `-fstab` map. This causes `automount` to read the `fstab` database. All mounts with the `net` option are mounted within the `-fstab` map's file system using a path of the form:

```
server/path
```

If the `fstab` database contains an entry for `polaris:/Library/Fonts`, and if `automount` is started as

```
automount -m /Useful -fstab
```

the mount appears as `/Useful/polaris/Library/Fonts`.

`-m <directory> <map>` Associates the specified `<map>` with the given `<directory>`. This directory is created if it doesn't exist. `<map>` may be the name of a file, or it may be the name of a special map.

`-d` Runs `automount` in debug mode. The program remains attached to the command line and sends debugging information to standard output.

`-tm <secs>` Sets the timeout for NFS mounts to `<secs>` seconds. Default is 20 seconds.

`-tl <secs>` Sets the time-to-live for NFS mounts to `<secs>` seconds. Default is 3600 seconds.

biff

`biff` Notification of incoming mail and who it's from during the current terminal session.

```
biff [ny]
```

`n` Disables notification.

`y` Enables notification.

bsdmake

`bsdmake` Maintains program dependencies.

```
bsdmake [-BPSeiknqrstv] [-D <variable>] [-d <flags>] [-E <variable>] [-f <makefile>] [-I <directory>] [-j <max_jobs>]
[-m <directory>] [-V <variable>] [<variable>=<value>] [<target> ...]
```

The `bsdmake` program simplifies the maintenance of other programs. Its input is a list of specifications describing the dependency relationships between the generation of files and programs. The first `makefile` or `Makefile` that can be found in either the current directory or a special object directory is read for the list of specifications. If the file `.depend` can be found, it's also read.

`-B` Tries backward compatibility by executing a single shell per command and by executing the commands to make the sources of a dependency line in sequence. This option is turned on by default unless `-j` is used.

`-P` Collates the output of a given job and displays it only when the job finishes, instead of mixing the output of parallel jobs together. This option has no effect unless `-j` is also used.

`-S` Stops processing when an error is encountered. Default behavior. This is needed to negate the `-k` option during recursive builds.

`-e` Specifies that environment values override macro assignments within makefiles for all variables.

`-i` Ignores nonzero exit of shell commands in the makefile. Equivalent to specifying `-` before each command line in the `makefile`.

`-k` Continues processing after errors are encountered, but only on those targets that don't depend on the target whose creation caused the error.

`-n` Displays the commands that would have been executed, but doesn't actually execute them.

`-q` Doesn't execute any commands, but exits `0` if the specified targets are up-to-date, and `1` otherwise.

`-r` Doesn't use built-in rules specified in the system makefile.

`-s` Doesn't echo any commands as they're executed. Equivalent to specifying `@` before each command line in the makefile.

`-t` Rather than rebuilding a target as specified in the `makefile`, creates it or updates its modification time to make it appear up-to-date.

`-v` Is extra verbose. For multijob makes, this causes file banners to be generated.

`-D <variable>` Defines `<variable>` to be `1`, the global context.

`-E <variable>` Specifies a variable whose environment value (if any) will override macro assignments within makefiles.

`-f <makefile>` Uses `<makefile>` as the `makefile`. If `<makefile>` is `-`, standard input is read. Multiple makefiles may be specified, and are read in the order specified.

`-I <directory>` Specifies a directory in which to search for makefiles and included makefiles. The system makefile directory (or directories; see the `-m` option) is automatically included as part of this list.

`-j <max_jobs>` Specifies the maximum number of jobs that `bsdmake` may have running at any one time. Turns compatibility mode off, unless the `-B` flag is also specified.

`-m <directory>` Specifies a directory in which to search for sys.mk and `makefiles` included via the `<...>` style. Multiple directories can be added to form a search path. This path will override the default system include path: `/usr/share/mk`. Furthermore, the system `include` path will be appended to the search path used for `...-style` inclusions (see `-I` option).

`-V <variable>` Prints `gnumake`'s idea of the value of `<variable>`, in the global context. Doesn't build any targets. Multiple instances of this option may be specified; the variables will be printed one per line, with a blank line for each null or undefined variable.

`<variable>=<value>` Sets the value of the variable `<variable>` to `<value>`.

`-d <flags>` Turns on debugging, and specifies which portions of `make` are to print debugging information.

Argument `<flags>` is one or more of the following:

`A` Prints all possible debugging information; equivalent to specifying all the debugging flags.

`a` Prints debugging information about archive searching and caching.

`c` Prints debugging information about conditional evaluation.

`d` Prints debugging information about directory searching and caching.

`f` Prints debugging information about the execution of loops. Currently at no-op.

g1	Prints the input graph before making anything.
g2	Prints the input graph after making everything, or before exiting on error.
j	Prints debugging information about running multiple shells.
l	Prints commands in makefiles regardless of whether they're prefixed by @ or other quiet flags. Also known as *loud behavior*.
m	Prints debugging information about making targets, including modification dates.
s	Prints debugging information about suffix-transformation rules.
t	Prints debugging information about target list maintenance.
v	Prints debugging information about variable assignment.

bzip2, bunzip2, bzcat, bzip2recover

bzip2, bunzip2

bzcat

bzip2recover Block-sorting file compressor, v1.0.2.

Decompresses files to stdout.

Recovers data from damaged bzip2 files.

bzip2 [-cdfkqstvzVL123456789] [<*filename1*> <*filename2*> ...]

bunzip2 [-fkvsVL] [<*filename1*> <*filename2*> ...]

bzcat [-s] [<*filename1*> <*filename2*> ...]

bzip2recover <*filename*>

bzip2, bunzip2, and bzcat are really the same program. The decision about what actions to take is done on the basis of which name is used.

bzip2 compresses files using the Burrows-Wheeler block-sorting text compression algorithm, and Huffman coding.

bzip2 expects a list of filenames to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name <*original_name*>.bz2. Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time.

If no filenames are specified, bzip2 compresses from standard input to standard output.

bzip2 reads arguments from the environment variables BZIP2 and BZIP, in that order, and processes them before reading any arguments from the command line. Compression is always performed, even if the compressed file is slightly larger than the original.

bunzip2 (or bzip2 -d) decompresses files. Files not created by bzip2 are detected and ignored and a warning is issued. Filenames are restored as follows:

<*filename*>.bz2	<*filename*>
<*filename*>.bz	<*filename*>
<*filename*>.tbz2	<*filename*>.tar
<*filename*>.tbz	<*filename*>.tar
<*anyothername*>	<*anyothername*>.out

Supplying no filenames causes decompression from standard input to standard output.

`bzcat` (or `bzip2 -dc`) decompresses all specified files to standard output.

`bzip2recover` is a simple program whose purpose is to search for blocks in `.bz2` files, and write each block out into its own `.bz2` file. You can then use `bzip2 -t` to test the integrity of the resulting files, and decompress those which are undamaged.

`bzip2recover` takes a single argument, the name of the damaged file, and writes a number of files `rec00001file.bz2`, `rec00002file.bz2`, and so on, containing the extracted blocks. The output filenames are designed so that the use of wildcards in subsequent processing: For example, `bzip2 -dc rec*file.bz2 > recovered_data` processes the files in the correct order.

`-h`
`--help` Displays a help menu.

`-c`
`--stdout` Compresses or decompresses to standard output.

`-d`
`--decompress` Forces decompression.

`-f`
`--force` Forces overwrite of output files.

Normally, `bzip2` doesn't overwrite existing output files. Also forces `bzip2` to break hard links to files, which it otherwise doesn't do.

`bzip2` normally declines to decompress files that don't have the correct magic header bytes. If forced (`-f`), however, it passes such files through unmodified. This is how GNU `gzip` behaves.

`-k`
`--keep` Keeps (doesn't delete) input files during compression or decompression.

`-q`
`--quiet` Suppresses non-essential warning messages. Messages pertaining to I/O errors and other critical events are no't suppressed.

`-s`
`--small` Reduces memory usage, for compression, decompression, and testing. If your machine is low on memory (8 megabytesMB or less), use `-s` for everything.

`-t`
`--test` Checks integrity of the specified file(s), but doesn't decompress them. This really performs a trial decompression and throws away the result.

`-v`
`--verbose` Verbose mode. Shows the compression ratio for each file processed. Further `-v`s increase the verbosity level.

`-z`
`--compress` Forces compression, regardless of the invocation name.

`-V`
`--version` Displays the software version, license terms, and conditions.

`-L`
`--license` Displays the software version, license terms, and conditions.

`-1 (or –fast) .. –9 (or –best)`	Sets block size to 100k .. –900k. The `--fast` and `–best` aliases are primarily for GNU `gzip` compatibility. In particular, `--fast` doesn’t make things significantly faster. `--best` merely selects the default behavior.
`--`	Treats all subsequent arguments as filenames, even if they start with a dash. This is so that you can handle files with names beginning with a dash; for example: `bzip2 -- -myfilename`.

cancel

`cancel`	Removes print jobs from the queue.

`cancel  [ -a ] [ -h <server> ] [ <id> ] [ <destination> ] [ <destination-id> ]`

`-a`	Removes all jobs from the specified destination.
`-h <server>`	Specifies the print server hostname. The default is `localhost` or the value of the `CUPS_SERVER` environment variable.

cat

`cat`	Concatenates and prints files.

`cat [-nbsvetu] <file1> <file2> ...`

`cat [-nbsvetu] [-]`

`cat` reads files in sequential, command-line order and writes them to standard output. A single dash represents standard input.

`-n`	Numbers all output lines.
`-b`	Numbers all output lines, except `b` or blank lines.
`-s`	Squeezes multiple adjacent empty lines, causing single-spaced output.
`-v`	Displays nonprinting characters. Control characters print as `^X` for Control+X; delete (octal 0177) prints as `^?`; non-ASCII characters with the high bit set are printed as `M-` (for meta) followed by the character for the low 7 bits.
`-e`	Implies `-v` option. Displays a dollar sign (`$`) at the end of each line as well.
`-t`	Implies `-v` option. Displays tab characters as `^I` as well.
`-u`	Guarantees unbuffered output.

catman

`catman` Creates formatted files for the online manual reference.

`catman [-knpsw] [-M <directory>] [<sections>]`

`-k` Ignores errors from `nroff` when building man pages.

`-n` Doesn't create the `whatis` database.

`-p` Prints what would have been done, but doesn't actually do it.

`-s` Works silently. Doesn't echo commands as they're executed. Ignored if `-p` is specified.

`-w` Creates only the `whatis` database.

`-M <directory>` Updates the manual pages in the `<directory>` specified.

The optional `<sections>` argument is a string containing the numbers of the sections to be regenerated. For example, if `<sections>` is 138, sections 1, 3, and 8 are regenerated.

When manual pages are regenerated, `catman` also rebuilds the `whatis` database.

cd

`cd` Changes working directory.

`cd [-p] [-l] [-n | -v] [<directory>]`

`cd`

`<directory>` is an absolute or relative pathname. The interpretation of the relative pathname depends on the `CDPATH` environment variable.

`-p` Prints the final directory stack.

`-l` Expands ~ or `~<name>` to the pathname of the user's home directory. Prints output in long form.

`-n` Wraps entries before they reach the end of the screen.

`-v` Prints one entry per line, preceded by their stack position. If both `-n` and `-v` are specified, `-v` takes precedence.

The following environment variables affect the execution of `cd`:

`HOME` If `cd` is invoked without any arguments and the `$HOME` exists, `$HOME` becomes the new working directory.

`CDPATH` If `<directory>` does not begin with `/`, `.`, or `..`, `cd` searches for the directory relative to each directory named in the `CDPATH` variable, in the order listed. If the new working directory is derived from `$CDPATH`, it's printed to standard output.

chflags

`chflags` Changes file flags.

`chflags [-R [-H | -L | -P]] <flags> <file1> <file2> ...`

`-R` Recursively descends through directory arguments to change file flags.

`-H` If `-R` is specified, symbolic links on the command line are followed. Symbolic links encountered in tree traversal aren't followed.

-L	If -R is specified, all symbolic links are followed.
-P	If -R is specified, no symbolic links are followed.

Symbolic links don't have flags. Unless -H or -L is specified, chflags on a symbolic link always succeeds and has no effect. -H, -L, and -P options are ignored unless -R is specified. Furthermore, -H, -L, and -P override each other. The last option specified determines the action that's taken.

<*flags*> is a comma-separated list of keywords. Currently available keywords are as follows:

arch	Sets the archived flag (superuser only)
opaque	Sets the opaque flag (owner or superuser only)
nodump	Sets the nodump flag (owner or superuser only)
sappnd	Sets the system append-only flag (superuser only)
schg	Sets the system immutable flag (superuser only)
uappnd	Sets the user append-only flag (owner or superuser only)
uchg	Sets the user immutable flag (owner or superuser only)

Prepending the letters no to a flag turns the flag off.

chgrp

chgrp	Changes group.

```
chgrp [-R [-H | -L | -P]] [-fh] <group> <file1> <file2> ...
```

-R	Recursively descends through directory arguments to change the group ID.
-H	If -R is specified, symbolic links on the command line are followed. Symbolic links encountered in tree traversal aren't followed.
-L	If -R is specified, all symbolic links are followed.
-P	If -R is specified, no symbolic links are followed.
-f	Forces an attempt to change group ID without reporting any errors.
-h	If the file is a symbolic link, the group ID of the link is changed.

Unless -h, -H, or -L is specified, chgrp on symbolic links always succeeds and has no effect.

The -H, -L, and -P options are ignored unless -R is specified. Because they also override each other, the last one specified determines the action that's taken.

The group may be either a numeric group ID or a group name. If a group name exists for a group ID, the associated group name is used for the group.

The user invoking chgrp must belong to the specified group and be the owner of the file, or be the superuser.

Unless invoked by the superuser, chgrp clears the set-user-id and set-group-id bits.

chmod

chmod	Changes file modes.

```
chmod [-R [-H | -L | -P]] [-h] <absolute_mode> <file1> <file2> ...
chmod [-R [-H | -L | -P]] [-h] <symbolic_mode> <file1> <file2> ...
```

-R	Recursively descends through directory arguments to change file modes.
-H	If -R is specified, symbolic links on the command line are followed. Symbolic links encountered in tree traversal aren't followed.

`-L` If `-R` is specified, all symbolic links are followed.

`-P` If `-R` is specified, no symbolic links are followed.

Unless `-H` or `-L` is specified, `chmod` on a symbolic link always succeeds and has no effect. The `-H`, `-L`, and `-P` options are ignored unless `-R` is specified. Furthermore, `-H`, `-L`, and `-P` override each other. The last option specified determines the action that's taken.

Permissions are described by three sequences of letters in the order listed here. Each sequence describes the permissions for user, group, and other. If a certain permission hasn't been granted, a `-` (dash) appears in its place.

User	Group	Other
`rwx`	`rwx`	`rwx`

The permissions on a file can be viewed using `ls -l` and changed using `chmod`.

Absolute Mode

Absolute mode is constructed by `OR`ing any of the following modes:

`4000` Sets user ID on execution—If this is a program, causes it to run as though the user who owns it were actually running it, regardless of who executes it.

`2000` Sets group ID on execution—If this is a program, causes it to run with the group ID of the program, regardless of group memberships of the user who executes it.

`1000` Turns on sticky bit—Has different meanings in different contexts: for directories, protects files from modification by other than owner, even if the user has permissions to write in the directory—overridden by directory ownership.

`0400` Allows read by owner.

`0200` Allows write by owner.

`0100` Allows execute (search in a directory) by owner.

`0600` Allows read, write by owner.

`0500` Allows read, execute by owner.

`0300` Allows write, execute by owner.

`0700` Allows read, write, execute by owner.

`0040` Allows read by group.

`0020` Allows write by group.

`0010` Allows execute (search in a directory) by group.

`0060` Allows read, write by group.

`0050` Allows read, execute by group.

`0030` Allows write, execute by group.

`0070` Allows read, write, execute by group.

`0004` Allows read by other.

`0002` Allows write by other.

`0001` Allows execute (search in a directory) by other.

`0006` Allows read, write by other.

`0005` Allows read, execute by other.

`0003` Allows write, execute by other.

`0007` Allows read, write, execute by other.

Symbolic Mode

Symbolic mode is a comma-separated list, with no intervening white space, of the form

`[<who>]<operator>[<permissions>]`

`<who>` has the following form:

`< u | g | o | a>`

`u`	User's permissions
`g`	Group's permissions
`o`	Other's permissions
`a`	All permissions (user, group, other); equivalent to `ugo`

`<operator>` has the following form:

`< + | - | =>`

`+`	Adds `<permissions>`.

If `<permissions>` isn't specified, no changes occur.

If `<who>` isn't specified, `<who>` defaults to `a`, and `<permissions>` are added as specified, except that `chmod` doesn't override the file mode creation mask.

If `<who>` is specified, `<permissions>` are added as specified.

`-`	Removes `<permissions>`.

If `<permissions>` isn't specified, no changes occur.

If `<who>` isn't specified, `<who>` defaults to `a`, and `<permissions>` are removed as specified, except that `chmod` doesn't override the file mode creation mask.

If `<who>` is specified, `<permissions>` are removed as specified.

`=`	Assigns the absolute `<permissions>` specified.

If `<who>` isn't specified, `<who>` defaults to `a`.

If `<permissions>` isn't specified, `<permissions>` defaults to `remove`.

If `<who>` is specified and `<permissions>` isn't, all permissions for `<who>` are removed.

If `<who>` isn't specified and `<permissions>` is specified, `<permissions>` for all are set to `<permissions>`, except that `chmod` doesn't override the file creation mask.

If `<who>` is specified and `<permissions>` is specified, `<permissions>` for `<who>` are set as specified.

`<permissions>` has the following form:

`<r | w | x | X | s | t | u | g | o>`

`r`	Sets read bits.
`w`	Sets write bits.
`x`	Sets execute/search bits.
`X`	Sets execute/search bits if the file is a directory, or if any execution/search bits are already set in the file before `X` would act upon the file. `X` is used only with `+`, and is ignored in all other cases.
`s`	Sets the `set-user-ID-on-execution` and `set-group-ID-on-execution` bits. A process runs as the user or group specified by `s`.
`t`	Sets the sticky bit.
`u`	User permission bit in the mode of the original file.
`g`	Group permission bits in the mode of the original file.
`o`	Other permission bits in the mode of the original file.

Operations on `<who>` `o` in combination with `<permissions>` `s` or `t` are ignored.

chown

`chown` Changes file owner and group.

`chown [-R [-H | -L | -P]] [-fh] <owner> <file1> <file2> ...`

`chown [-R [-H | -L | -P]] [-fh] :<group> <file1> <file2> ...`

`chown [-R [-H | -L | -P]] [-fh] <owner>:<group> <file1> <file2> ...`

`-R` Recursively descends through directory arguments to change the user ID and/or group ID.

`-H` If `-R` is specified, symbolic links on the command line are followed. Symbolic links encountered in tree traversal aren't followed.

`-L` If `-R` is specified, all symbolic links are followed.

`-P` If `-R` is specified, no symbolic links are followed.

`-f` Forces an attempt to change user ID and/or group ID without reporting any errors.

`-h` If the file is a symbolic link, the user ID and/or group ID of the link is changed.

The `-H`, `-L`, and `-P` options are ignored unless `-R` is specified. Because they also override each other, the last option specified determines the action that's taken.

The `-L` option cannot be used with the `-h` option.

It isn't necessary to provide both <*owner*> and <*group*>; however, one must be specified. If group is specified, it must be preceded with a colon (`:`).

The owner may be either a numeric user ID or a username. If a username exists for a numeric user ID, the associated username is used as for the owner. Similarly, the group may be either a numeric group ID or a group name. If a group name exists for a group ID, the associated group name is used for the group.

Unless invoked by the superuser, `chown` clears `set-user-id` and `set-group-id` bits.

cmp

`cmp` Compares two files.

`cmp [-l | -s] <file1> <file2> [<skip1> <skip2>]`

`cmp` compares two files of any type and writes the results to the standard output. By default, `cmp` is silent if the files are the same; if they differ, the byte and line number where the first difference occurs is reported.

Bytes and line are numbered beginning with 1.

`-l` Lists the byte number (decimal) and differing byte values (octal) for each difference.

`-s` Prints nothing for differing files; returns exit status only.

The optional arguments <*skip1*> and <*skip2*> are the byte offsets, from the beginning of <*file1*> and <*file2*>, respectively, where the comparison will begin. The offset is decimal by default, but may be expressed as a hexadecimal or octal value by preceding it with a leading 0x or 0.

`cmp` exits with one of the following values:

`0`	Files are identical.
`1`	Files are different; this includes the case where one file is identical to the first part of the other. In the latter case, if `-s` hasn't been specified, `cmp` writes to standard output that EOF was reached in the shorter file before any differences were found.
`>1`	An error occurred.

compress

`compress`	Compresses data.
`uncompress`	Expands data.

`compress [-cfv] [-b <bits>] <file1> <file2> ...`

`uncompress [-cfv] <file1> <file2> ...`

`compress` reduces the size of a file and renames the file by adding a `.Z` extension. As much of the original file characteristics (modification time, access time, file flags, file mode, user ID, and group ID) are retained as permissions allow. If compression wouldn't reduce a file's size, the file is ignored.

`uncompress` restores a file reduced by `compress` to its original form, and renames the file by removing the `.Z` extension.

`-c`	Writes compressed or uncompressed output to standard output without modifying any files.
`-f`	Forces compression of a file, even when compression wouldn't reduce its size. Additionally, forces files to be overwritten without prompting for confirmation.
`-v`	Prints the percentage reduction of each file.
`-b <bits>`	Specifies the upper bit code limit. Default is 16. Bits must be between 9 and 16. Lowering the limit results in larger, less-compressed files.

cp

`cp`	Copies files.

`cp [-R (-H | -L | -P)] [-f | -i] [-p] <source> <target>`

`cp [-R (-H | -L | -P)] [-f | -i] [-p] <source1> <source2> ... <directory>`

In its first form, `cp` copies the contents of `<source>` to `<target>`.

In its second form, `cp` copies the contents of the list enumerated by `<source1> <source2> ...` to the directory named by `<directory>`. The names of the files themselves aren't changed. If `cp` detects an attempt to copy to itself, that attempt fails.

`-R`	If `<source>` is a directory, `cp` recursively copies the directory. This option also causes symbolic links to be copied, rather than directed through. Created directories have the same mode as the corresponding source directory.
`-H`	If `-R` is specified, symbolic links on the command line are followed, but symbolic links in the tree traversal aren't.

`-L` If `-R` is specified, all symbolic links are followed.

`-P` If `-R` is specified, no symbolic links are followed.

`-f` Forces an existing file to be overwritten. If permissions don’t allow the copy to succeed, this forces the existing file to be removed and a new file to be created without prompting for confirmation. The `-i` option is ignored if the `-f` option is specified.

`-i` Invokes an interactive mode that prompts for a confirmation before overwriting an existing file.

`-p` Causes `cp` to retain as much of the modification time, access time, file flags, file mode, user ID, and group ID information as permissions allow.

crontab

`crontab` Maintains crontab files for individual users.

```
crontab [-u <user>] <file>
crontab [-u <user>] [-l | -r | -e]
```

`crontab` is the program that installs, removes, or lists the tables the `cron` (8) executes for users. Each user can have his own `crontab`, which is stored in `/var/cron/tabs/`. The `crontab` isn’t edited directly. If `/var/cron/allow` exists, the *<user>* must be listed in the file to be able to use `cron`. If `/var/cron/allow` doesn’t exist, but `/var/cron/deny` exists, *<user>* mustn’t be listed in this file to use this command. If neither file exists (depending on site-dependent configuration), either only the superuser may use this command or all users may be able to use this command.

The first form of the command installs a `crontab` from *<file>* or standard input, if `-` is given instead of *<file>*. The second form of the command displays, removes, or edits the installed `crontab`.

`-u <user>` Specifies the name of the user. If not specified, the user issuing the command is assumed. If `crontab` is being used inside an `su` command, `-u` should be used.

`-l` Lists the current `crontab` on standard output.

`-r` Removes the current `crontab`.

`-e` Edits the current `crontab` using the editor specified by the environment variables `VISUAL` or `EDITOR`. On exiting the editor, the modified `crontab` is automatically installed.

Basic format of a `crontab` statement, with value ranges shown here:

```
minute hour day_of_month month day_of_week [<user>] <command>
0-59 0-23 1-31 1-12 0-7 (Sunday may be 0 or 7)
```

Fields may be separated by spaces or tabs. `*` may be used as the value of a field to mean all possible values for that field. A field value may be further specified by providing a single value, a comma-separated list of values, a range of values, or a comma-separated list of single values or ranges of values. Step values may be specified by use of *<range>*/*<number>*. For example, `0-23/2` would be every other hour. `0-23/2` is equivalent to the value list 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22. Step values may also be specified by `*/`*<number>*. For example, every other hour could also be specified by `*/2` in that field. Names may also be used for the `month` and `day_of_week` fields. Names are the first three characters of the actual name. Case doesn’t matter. Lists or ranges of names, however, may not be used.

The *<user>* field is specified only in a system `crontab`.

curl

`curl`

A utility for getting a URL with FTP, Telnet, LDAP, Gopher, DICT, FILE, HTTP, or HTTPS syntax.

`curl [<`*`options`*`>] [<URL>...]`

`-a`
`--append`

(FTP) When used in an FTP upload, this tells `curl` to append to the target file instead of overwriting it. If the file doesn't exist, it's created.

If this option is used twice, the second one disables append mode again.

`-A <`*`agent string`*`>`
`--user-agent <`*`agent string`*`>`

(HTTP) Specifies the User-Agent string to send to the HTTP server. Some badly done CGIs fail if it's it not isn't set to `"Mozilla/4.0"`. To encode blanks in the string, surround the string with single quote marks. This can also be set with the `-H/--header` flag, of course.

If this option is used more than once, the last one is the one to be used.

`-b <`*`name=data`*`>`
`--cookie <`*`name=data`*`>`

(HTTP) Passes the data to the HTTP server as a cookie. The data is supposedly the data previously received from the server in a `Set-Cookie:` line. The data should be in the format *`NAME1=VALUE1`*`;` *`NAME2=VALUE2`*, but there's nothing to say you can't change it.

If no `=` is used in the line, it's treated as a filename to use to read previously stored cookie lines from, which should be used in this session if they match. Using this method also activates the cookie parser, which makes `curl` record incoming cookies too, which may be handy for using this in combination with the `-L/--location` option. The file format of the file to read cookies from should be plain HTTP headers or the Netscape cookie file format.

Note that the file specified with `-b/--cookie` is only used as input. No cookies are stored in the file. To store cookies, save the HTTP headers to a file using `-D/--dump-header`.

If this option is used more than once, the last one is the one to be used.

`-B`
`--use-ascii`

Uses ASCII transfer when getting an FTP file or LDAP info. For FTP, this can also be enforced by using an URL that ends with `;type=A`.

If this option is used twice, the second one disables ASCII usage.

`-C <offset>`

`--continue-at <offset>` Continues/resumes a previous file transfer at the given offset. The given offset is the exact number of bytes that are skipped counted from the beginning of the source file before it is transferred to the destination. If used with uploads, the FTP server command `SIZE` isn't be used by `curl`. Upload resume is for FTP only. HTTP resume is only possible with HTTP/1.1 or later servers.

If this option is used several times, the last one is used.

`-d <data>`

`--data <data>` (HTTP) Sends the specified data in a `POST` request to the HTTP server, in a way that can emulate as if a user has filled in an HTML form and clicked the submit button. Note that the data is sent exactly as specified with no extra processing (with all newlines cut off). The data is expected to be URL-encoded. This causes `curl` to pass the data to the server using the content-type `application/x-www-form-urlencoded`. Compare to `-F`. If more than one `-d/--data` option is used on the same command line, the data pieces specified are merged together with a separating `&` character. Thus, using `-d name=daniel -d skill=lousy` generates a post chunk that looks like `name=daniel&skill=lousy`.

If this option is used several times, the ones following the first append data.

`--data-ascii <data>` (HTTP) An alias for the `-d/--data` option.

If this option is used several times, the ones following the first append data.

`--data-binary <data>` (HTTP) Posts data in a similar manner as `--data-ascii` does, although when using this option the entire context of the posted data is kept as- is. If you want to post a binary file without the `strip-newlines` feature of the `--data-ascii` option, this is for you.

`-D <file>`

`--dump-header <file>` (HTTP/FTP) Writes the HTTP headers to this `<file>`. Writes the FTP file info to this `<file>` if `-I/--head` is used.

Handy for storing the cookies that an HTTP site sends to you. The cookies could then be read in a second `curl` invoke by using the `-b/--cookie` option.

If this option is used several times, the last one is used.

`-e <URL>`

`--referer <URL>` (HTTP) Sends the referer page information to the HTTP server. This can also be set with the `-H/--header` flag, of course. When used with `-L/--location`, you can append `;auto` to the referer URL to make `curl` automatically set the previous URL when it follows a Location: header. The `;auto` string can be used alone, even if you don't set an initial referer.
If this option is used several times, the last one is used.

`--egd-file <file>` (HTTPS) Specifies the pathname to the Entropy Gathering Daemon socket. The socket is used to seed the random engine for SSL connections. See also the `--random-file` option.

`-E`

`--cert <certificate[:password]>` (HTTPS) Tells `curl` to use the specified certificate file when getting a file with HTTPS. The certificate must be in PEM format. If the optional password isn't specified, it will be queried for on the terminal. Note that this certificate is the private key and the private certificate concatenated!
If this option is used several times, the last one is used.

`--cacert <CA certificate>` (HTTPS) Tells `curl` to use the specified certificate file to verify the peer. The certificate must be in PEM format.
If this option is used several times, the last one is used.

`-f`

`--fail` (HTTP) Fails silently (no output at all) on server errors. This is mostly done to better enable scripts and so on to deal with failed attempts. In normal cases when an HTTP server fails to deliver a document, it returns an HTML document stating so (which often also explains the failure). This flag will prevent `curl` from outputting that and fails silently instead.
If this option is used twice, the second again disables silent

failure.

`-F`

`--form <name=content>` (HTTP) This lets `curl` emulate a filled-in form in which a user has clicked the submit button. This causes `curl` to `POST` data using the content-type `multipart/form-data` according to RFC1867. This enables uploading of binary files and so on. To force the content part to be a file, prefix the filename with an `@` sign. To just get the content part from a file, prefix the filename with the character <. The difference between `@` and < is that `@` makes a file get attached in the post as a file upload, whereas the < makes a text field and just gets the contents for that text field from a file.
This option can be used multiple times.

`-g`
`--globoff` Switches off the URL globbing parser. When you set this option, you can specify URLs that contain the characters `{}[]` without having them interpreted by `curl` itself. Note that these characters aren't normal legal URL contents, but they should be encoded according to the URI standard.

`-h`
`--help` Displays help.

`-H`
`--header <`*`header`*`>` (HTTP) Extra header to use when getting a Web page. You may specify any number of extra headers. Note that if you should add a custom header that has the same name as one of the internal ones curl would use, your externally set header will be used instead of the internal one. This allows you to make even trickier stuff than `curl` would normally do. You shouldn't replace internally set headers without knowing perfectly well what you're doing. Replacing an internal header with one without content on the right side of the colon will prevent that header from appearing.
This option can be used multiple times.

`-i`
`--include` (HTTP) Includes the HTTP-header in the output. The HTTP-header includes things like server-name, date of the document, HTTP-version and more.
If this option is used twice, the second again disables header include.

`--interface <`*`name`*`>` Performs an operation using a specified interface. You can enter interface name, IP address, or host name. An example could look like this:

```
curl --interface eth0:1 http://www.netscape.com/
```

If this option is used several times, the last one is used.

`-H`
`--head` (HTTP/FTP) Fetches the HTTP- header only! HTTP- servers feature the command `HEAD`, which this uses to get nothing but the header of a document. When used on an FTP file, `curl` displays the file size only.
If this option is used twice, the second again disables header only.

`--krb4 <`*`level`*`>` (FTP) Enables kerberos4 authentication and use. The level must be entered and should be one of `clear`, `safe`, `confidential`, or `private`. Should you use a level that isn't one of these, `private` is used instead.
If this option is used several times, the last one is used.

`-K`
`--config` *`<config file>`*

Specifies which config file to read `curl` arguments from. The config file is a text file in which command-line arguments can be written, which then will be used as if they were written on the actual command line. Options and their parameters must be specified on the same config file line. If the parameter is to contain white spaces, the parameter must be enclosed within quotes. If the first column of a config line is a `#` character, the rest of the line is treated as a comment.
Specify the filename as `-` to make `curl` read the file from `STDIN`.
This option can be used multiple times.

`-l`
`--list-only`

(FTP) When listing an FTP directory, this switch forces a name-only view. Especially useful if you want to machine-parse the contents of an FTP directory because the normal directory view doesn't use a standard look or format.
If this option is used twice, the second again disables list only.

`-L`
`--location`

(HTTP/HTTPS) If the server reports that the requested page has a different location (indicated with the header line `Location:`), this flag instructs `curl` to reattempt the GET on the new location. If used together with `-i` or `-I`, headers from all requested pages are shown. If this flag is used when making an `HTTP POST`, `curl` automatically switches to `GET` after the initial `POST` is done.
If this option is used twice, the second again disables location following.

`-m`
`--max-time` *`<seconds>`*

Maximum time in seconds that you allow the whole operation to take. This is useful for preventing your batch jobs from hanging for hours due to slow networks or links going down. See also the `--connect-timeout` option.
If this option is used several times, the last one is used.

`-M`
`--manual`

Manual. Displays the `curl` man page.

`-n`
`--netrc`

Makes `curl` scan the `.netrc` file in the user's home directory for login name and password. This is typically used for `ftp` on Unix. If used with HTTP, `curl` will enable user authentication. See `netrc`(4) for details on the file format. `curl` won't complain if that file doesn't have the right permissions (it shouldn't be world or group readable). The environment variable `HOME` is used to find the home directory.
If this option is used twice, the second again disables `netrc` usage.

`-N`
`--no-buffer`
Disables the buffering of the output stream. In normal work situations, `curl` uses a standard buffered output stream that has the effect that it will output the data in chunks, not necessarily exactly when the data arrives. Using this option disables that buffering.
If this option is used twice, the second use again switches on buffering.

`-o`
`--output <file>`
Writes output to `<file>` instead of STDOUT. If you're using `{}` or `[]` to fetch multiple documents, you can use `#` followed by a number in the `<file>` specifier. That variable is replaced with the current string for the URL being fetched.

`-O`
`--remote-name`
Writes output to a local file named like the remote file we get. Only the file part of the remote file is used; the path is cut off.

You may use this option as many times as you have number of URLs.

`-p`
`--proxytunnel`
When an HTTP proxy is used, this option causes non-HTTP protocols to attempt to tunnel through the proxy instead of merely using it to do HTTP-like operations. The tunnel approach is made with the HTTP proxy `CONNECT` request and requires that the proxy allows direct connect to the remote port number curl wants to tunnel through to.
If this option is used twice, the second again disables proxy tunnel.

`-P`
`--ftpport <address>`
(FTP) Reverses the initiator/listener roles when connecting with FTP. This switch makes `curl` use the `PORT` command instead of `PASV`. In practice, `PORT` tells the server to connect to the client's specified address and port, whereas `PASV` asks the server for an IP address and port to connect to. `<address>` should be one of these:
Interface, for example, "`eth0`" to specify which interface's IP address you want to use (Unix only)
IP address, for example, "`192.168.10.1`" to specify exact IP number
Host name, for example, "`my.host.domain`" to specify machine
`-` (any single-letter string) to make it pick the machine's default
If this option is used several times, the last one is used.

`-q`
If used as the first parameter on the command line, the `$HOME/.curlrc` file won't be read and used as a config file.

`-Q`
`--quote <command>`
(FTP) Sends an arbitrary command to the remote FTP server, by using the `QUOTE` command of the server. Not all servers support this command, and the set of `QUOTE` commands are server-specific! `QUOTE` commands are sent BEFORE the transfer is taking place. To make commands take place after a successful transfer, prefix them with a dash `-`. You may specify any amount of commands to be run before and after the transfer. If the server returns failure for one of the commands, the entire operation is aborted.
This option can be used multiple times.

`--random-file <file>`
(HTTPS) Specifies the pathname to a file containing what will be considered as random data. The data is used to seed the random engine for SSL connections. See also the `--edg-file` option.

`-r`
`--range <range>`
(HTTP/FTP) Retrieve a byte range (that is, a partial document) from an HTTP/1.1 or FTP server. Ranges can be specified in a number of ways.
If this option is used several times, the last one is used.

`-s`
`--silent`
Silent mode. Doesn't show progress meter or error messages.
If this option is used twice, the second again disables mute.

`-S`
`--show-error`
When used with `-s`, it makes `curl` show error message if it fails.
If this option is used twice, the second again disables show error.

`-t`
`--telnet-option <OPT=val>`
Passes options to the `telnet` protocol. Supported options are
`TTYPE=<term>` Sets the terminal type.
`XDISPLOC=<X display>` Sets the X11 display location.
`NEW_ENV=<var,val>` Sets an environment variable.

`-T`
`--upload-file <file>`
Transfers the specified local `<file>` to the remote server at `<URL>`. If there is's no file part in the specified URL, `curl` will append the local filename. Note that you must use a trailing `/` on the last directory to really prove to `curl` that you aren't providing a filename, or `curl` will think that your last directory name is the remote filename to use. That will most likely cause the upload operation to fail. If this is used on an HTTP(S) server, the `PUT` command will is be used.
If this option is used several times, the last one is used.

`-u`
`--user <`*`user:password`*`>`

Specifies user and password to use when fetching. See `README.curl` for detailed examples of how to use this. If no password is specified, `curl` asks for it interactively.
If this option is used several times, the last instance is used.

`-U`
`--proxy-user <`*`user:password`*`>`

Specifies user and password to use for proxy authentication. If no password is specified, `curl` asks for it interactively.
If this option is used several times, the last instance is used.

`--url <`*`URL`*`>`

Specifies a `URL` to fetch. This option is mostly handy when you want to specify URL(s) in a config file.
This option may be used any number of times. To control where this URL is written, use the `-o` or the `-O` options.

`-v`
`--verbose`

Makes the fetching more verbose/talkative. Mostly usable for debugging. Lines starting with > means data sent by `curl`, < means data received by `curl` that's hidden in normal cases, and lines starting with * means additional info provided by `curl`.
If this option is used twice, the second again disables verbose.

`-V`
`--version`

Displays the full version of `curl`, `libcurl`, and other third-party libraries linked with the executable.

`-w`
`--write-out <`*`format`*`>`

Defines what to display after a completed and successful operation. The format is a string that may contain plain text mixed with any number of variables. The string can be specified as `"string"`. To cause it to be read from a particular file, you specify it as `"@filename"` and to tell `curl` to read the format from STDIN, you write `"@-"`.
If this option is used several times, the last one is used.

`-x`
`--proxy <`*`proxyhost[:port]`*`>`

Uses specified proxy. If the port number isn't specified, it's assumed at port 1080.
If this option is used several times, the last one is used.

`-X`
`--request <`*`command`*`>`

(HTTP) Specifies a custom request to use when communicating with the HTTP server. The specified request is used instead of the standard `GET`. Read the HTTP 1.1 specification for details and explanations.
(FTP) Specifies a custom FTP command to use instead of `LIST` when doing file lists with FTP.
If this option is used several times, the last one is used.

`-y`
`--speed-time <time>`
	If a download is slower than speed-limit bytes per second during a speed-time period, the download gets aborted. If `speed-time` is used, the default speed limit will be `1` unless set with `-y`.
	If this option is used several times, the last one is used.

`-Y`
`--speed-limit <speed>`
	If a download is slower than this given speed, in bytes per second, for `speed-time` seconds it gets aborted. `speed-time` is set with `-Y` and is 30 if not set.
	If this option is used several times, the last one is used.

`-z`
`--time-cond <date expression>`
	(HTTP) Requests to get a file that has been modified later than the given time and date, or one that has been modified before that time. The date expression can be all sorts of date strings or if it doesn't match any internal ones, it tries to get the time from a given filename instead.
	Start the date expression with a dash (`-`) to make it request for a document that's older than the given date/time; default is a document that's newer than the specified date/time.
	If this option is used several times, the last one is used.

`-3`
`--sslv3`
	(HTTPS) Forces `curl` to use SSL version 3 when negotiating with a remote SSL server.

`-2`
`--sslv2`
	(HTTPS) Forces `curl` to use SSL version 2 when negotiating with a remote SSL server.

`-#` `--progress-bar`	Displays progress information as a progress bar instead of the default statistics. If this option is used twice, the second again disables the progress bar.
`--crlf`	(FTP) Converts `LF` to `CRLF` in upload. Useful for MVS (OS/390). If this option is used twice, the second again disables `crlf` converting.
`--stderr <file>`	Redirects all writes to STDERR to the specified file instead. If the filename is a plain `-`, it's instead written to STDOUT. This option has no point when you're using a shell with decent redirecting capabilities. If this option is used several times, the last value is used.

defaults

`defaults` **Accesses the Mac OS X user defaults system.**

```
defaults [currentHost | -host <hostname>] read [<domain> [<key>]]
defaults [currentHost | -host <hostname>] read-type <domain> <key>
defaults [currentHost | -host <hostname>] write <domain> {'<plist>' | <domain> <key>
[']<value>[']}
default [currentHost | -host <hostname>] rename <domain> <old-key> <new_key>
defaults [currentHost | -host <hostname>] delete [<domain> [<key>]]
defaults [currentHost | -host <hostname>] { domains | find <word> | help }
```

`defaults` allows users to read, write, and delete Mac OS X user defaults from the command line. Applications use the defaults system to record user preferences and other information that must be maintained when applications aren't running, such as the default font for new documents. Because applications do access the defaults system while they're running, you shouldn't modify the defaults of a running application.

User defaults belong to domains, which typically correspond to individual applications. Each domain has a dictionary of keys and values to represent its defaults. Keys are always strings, but values can be complex data structures made up of arrays, dictionaries, strings, and binary data. These data structures are stored as XML property lists.

Although all applications, system services, and other programs have their own domains, they also share a domain called `NSGlobalDomain`. If a default isn't specified in the application's domain, it uses the default listed in the `NSGlobalDomain` instead.

`<`*`domain`*`>` is specified as follows:

```
<domain_name> | -app <application_name> | -globalDomain
```

Subcommands:

`read`
Prints all the user's defaults for every domain to standard output.

`read <`*`domain`*`>`
Prints all the user's defaults for the specified `<`*`domain`*`>` to standard output.

`read <`*`domain`*`> <`*`key`*`>`
Prints the value for the default of the `<`*`domain`*`>` identified by `<`*`key`*`>`.

`write <`*`domain`*` > <`*`key`*`> '<`*`value`*`>'`
Writes `<`*`value`*`>` as the value for `<`*`key`*`>` in `<`*`domain`*`>`. `<`*`value`*`>` must be a property list, and must be enclosed in single quotes. For example:

```
defaults write com.companyname.appname "Default
Color" '(255, 0, 0)'
```

Sets the default color in `com.companyname.appname` to the array containing 255, 0, 0 (red, green, blue components). Note that the key is in quotes because of the space in its name.

`write` *`<domain>`*
`'`*`<plist>`*`'`
	Overwrites the defaults information in *`<domain>`* with that specified in *`<plist>`*`.` *`<domain>`* must be a property list representation of a dictionary, and must be enclosed in single quotes. For example:
	`defaults write com.companyname.appname '{ "Default Color" = (255, 0, 0); "Default Font" = Helvetica; }'`
	Overwrites any previous defaults for `com.companyname.appname` and replaces them with the ones specified.

`delete` *`<domain>`*	Deletes all default information for *`<domain>`*.
`delete` *`<domain>`* *`<key>`*	Deletes the default named *`<key>`* in *`<domain name>`*`.`
`domains`	Prints the names of all domains in the user's defaults system.
`find` *`<word>`*	Searches for *`<word>`* in the domain names, keys, and values of the user's defaults, and prints out a list of matches.
`help`	Prints a list of possible command formats.
Options:	
`-g`	When specifying a domain, `-g` can be used as a synonym for `NSGlobalDomain`.
Specifying *`<value>`* for preference keys:	
`<value>`	Specifies *`<value>`* as a string value to use.
`'`*`<value>`*`'`	Specifies *`<value>`* as a string value to use.
`-string` *`<string_value>`*	Specifies *`<string_value>`* as the string to use.
`-data` *`<hex_digits>`*	Specifies *`<hex_digits>`* as the data to use.
`-int[eger]` *`<integer_value>`*	Specifies *`<integer_value>`* as the integer value to use.
`-bool[ean] true | false | yes | no`	Specifies the Boolean value to use.
`-date` *`<date_rep>`*	Specifies the date value to use.
`-array`	Allows the user to specify an array as the value for the given preference key. For example: `defaults write` *`<somedomain>`* *`<preferenceKey>`* `-array` *`<element1>`* *`<element2>`* *`<element3>`* The specified array overwrites the value of the key if the key was present at the time of the write. If the key wasn't present, it's created with the new value.
`-array-add`	Allows the user to add new elements to the end of an array for a key that has an array as its value. Usage is the same as `-array`.
`-dict`	Allows the user to add a dictionary to the defaults database for a domain. Keys and values are specified in order: `defaults write` *`<somedomain>`* *`<preferenceKey>`* `-dict` *`<key1>`* *`<value1>`* *`<key2>`* *`<value2>`* The specified dictionary overwrites the value of the key if the key was present at the time of the write. If the key isn't present, it's created with the new value.

`-dict-add` Allows the user to add new key/value pairs to a dictionary for a key that has a dictionary as its value. Usage is the same as `-dict`. If the key isn't present, it's created with the specified dictionary as its value.

Specifying a host for preferences:

`currentHost` Operations on the defaults database may apply only to a specific host. Specifying `currentHost` after the `defaults` command restricts the preferences operations to the host the user is currently logged in on, rather than any host the user may log in on.

diff

`diff` Finds differences between two files.

`diff [options] <`*`from-file`*`> <`*`to-file`*`>`

In its simplest form, `diff` compares the contents of two files. A filename of `-` stands for text read from the standard input. As a special case, `diff - -` compares a copy of standard input to itself.

If <*from-file*> is a directory and <*to-file*> isn't, `diff` compares the file in <*from-file*> whose filename is that of <*to-file*> and vice versa. The nondirectory file must not be `-`.

If both <*from-file*> and <*to-file*> are directories, `diff` compares corresponding files in both directories, in alphabetical order; the comparison isn't recursive unless `-r` or `-recursive` option is specified. `diff` never compares the actual content of a directory as if it were a file. The file that's fully specified may not be standard input because standard input is nameless and the notion of "file with the same name" doesn't apply.

Because `diff` options begin with `-`, normally <*from-file*> and <*to-file*> may not begin with `-`. However, `--` as an argument itself treats the remaining arguments as filenames even if they begin with `-`.

Multiple single-letter options, unless they can take an argument, can be combined into a single command-line word. Long named options can be abbreviated to any unique prefix of their name.

`-<`*`lines`*`>` Shows <*lines*> (an integer) lines of context. This option doesn't specify an output format by itself; it has no effect unless it's combined with `-c` or `-u`.

`-q`
`--brief` Reports only whether the files differ, not the details of the differences.

`-c` Uses the context output format.

`-C <`*`lines`*`>`
`--context[=<`*`lines`*`>]` Uses the context output format, showing <*lines*> (an integer) lines of context, or three lines if <*lines*> isn't given.

`--changed-group-format=` Uses <*format*> to output a line group containing differing <*format*> lines from both files in if-then-else format.

`-D <`*`name`*`>` Makes merged if-then-else format output, conditional on the preprocessor <*name*>.

`-e`
`--ed` Makes output that's valid `ed` script.

`-x <pattern>`
`--exclude=<pattern>`
	When comparing directories, ignores files and subdirectories whose base names match `<pattern>`.

`-X`
`--exclude-from=<file>`
	When comparing directories, ignores files and subdirectories whose base names match any pattern contained in `<file>`.

`-t` `--expand-tabs`	Expands tabs to spaces in the output, to preserve the alignment of tabs in the input files.
`-f`	Makes output that looks vaguely like an `ed` script but has changes in the order it appears in the file.
`-F <regexp>`	In context and unified format, for each hunk of differences, shows some of the last preceding line that matches `<regexp>`.
`--forward-ed`	Makes output that looks vaguely like an `ed` script but has changes in the order it appears in the file.
`-h`	This option has no effect; is present for Unix compatibility.
`--help`	Prints help information.
`--horizontal-lines=<lines>`	Doesn't discard the last `<lines>` lines of the common prefix and the first `<lines>` lines of the common suffix.
`-i` `--ignore-case`	Ignores changes in case; considers uppercase and lowercase letters equivalent.
`-I <regexp>`	Ignores changes that just insert or delete lines that match `<regexp>`.
`--ifdef=<name>`	Makes merged if-then-else format output, conditional on the preprocessor macro `<name>`.
`-w` `--ignore-all-space`	Ignores whitespace when comparing lines.
`-B` `--ignore-blank-lines`	Ignores changes that just insert or delete blank lines.
`--ignore-matching-lines=<regexp>`	Ignores changes that just insert or delete lines that match `<regexp>`.
`-b` `--ignore-space-change`	Ignores changes in amount of whitespace.
`-T` `--initial-tab`	Outputs a tab rather than a space before the text of a line in normal or context format. This causes the alignment of tab in the line to look normal.

`-L <label>`
`--label=<label>`
	Uses `<label>` instead of the filename in the context format and unified format headers.

`--left-column`	Prints only the left column of the two common lines in side-by-side format.
`--line-format=<format>`	Uses `<format>` to output all lines in if-then-else format.
`-d` `--minimal`	Changes the algorithm to perhaps find a smaller set of changes. This makes `diff` slower.

`-n`
`--rcs`
Outputs RCS-format diffs; like `-f` except that each command specifies the number of lines affected.

`-N`
`--new-file`
In the directory comparison, if a file is found in only one directory, treats it as present but empty in the other directory.

`--new-group-format=<format>`
Uses `<format>` to output a group of lines taken from just the second file in if-then-else format.

`--new-line-format=<format>`
Uses `<format>` to output a line taken from just the second file in if-then-else format.

`--old-group-format=<format>`
Uses `<format>` to output a group of lines taken from just the first file in if-then-else format.

`--old-line-format=<format>`
Uses `<format>` to output a line taken from just the first file in if-then-else format.

`-p`
`--show-c-function`
Shows which C function each change is in.

`-l`
`--paginate`
Passes the output through `pr` to paginate it.

`-r`
`--recursive`
When comparing directories, recursively compares any subdirectories found.

`-s`
`--report-identical-files`
Reports when two files are the same.

`-S <file>`
`--starting-file=<file>`
When comparing directories, starts with the file `<file>`. This is used for resuming an aborted comparison.

`--sdiff-merge-assist`
Prints extra information to help `sdiff`. `sdiff` uses this option when it runs `diff`. This option isn't intended for users to use directly.

`--show-function-line=<regexp>`
In context and unified format, for each hunk of differences, shows some of the last preceding line that matches `<regexp>`.

`-y`
`--side-by-side`
Uses the side-by-side output format.

`-H`
`--speed-large-files`
Uses heuristics to speed handling of large files that have numerous scattered small changes.

`--suppress-common-lines`
Doesn't print common lines in side-by-side format.

`-a`
`--text`
Treats all files as text and compares them line-by-line, even if they don't seem to be text.

`-u`
Uses the unified output format.

`-U <lines>`
`--unified[=<lines>]`
Uses the unified output format, showing `<lines>` (an integer) lines of context, or three lines if `<lines>` isn't given.

`--unchanged-group-format=`
Uses `<format>` to output a group of common lines taken `<format>` from both files in if-then-else format.

`--unchanged-line-format=`
Uses `<format>` to output a line common to both files in if-`<format>`then-else format.

`-p` `--undirectional-new-file`	When comparing directories, if a file appears only in the second directory of the two, treats it as present but empty in the other.
`-v` `--version`	Outputs the version number of `diff`.

`-W <columns>`
`--width=<columns>`
	Uses an output width of `<columns>` in side-by-side format.

df

`df`	Displays free disk space.

`df [-ikln] [-t <type>] [<file> | <filesystem> ...]`

`-i`	Includes statistics on the number of free inodes.
`-k`	Reports number in kilobyte counts. Default is 512-byte block sizes.
`-l`	Displays statistics only about mounted file systems with the `MNT_LOCAL` flag set. If a non-local file system is given as the argument, a warning is issued and no information is displayed.
`-n`	Prints out previously obtained statistics from the file system. This option should be used if it's possible that one or more file systems are in a state such that there's a long delay before they can provide statistics.
`-t <type>`	Displays information for file systems of the specified type. More than one type may be specified in a comma-separated list of the list of file system types.

If the environment variable `BLOCKSIZE` is set, and the `-k` option isn't used, the block counts are displayed according to the environment variable.

disktool

`disktool`	Disk arbitration command tool.

`disktool -[r][a][u][e][m][p][n][d][x][y][c][s][g][A][S][D][l] <deviceName> [<options>]`

Renames, ejects, mounts, or unmounts disks and volumes.

Information about disks:

`-l`	Lists the disks currently known and available on the system.

Controlling arbitration:

`-r`	Refreshes Disk Arbitration. Causes arbitration to refresh its internal tables and look for new mounts/unmounts.

Managing disks:

`-u`	Unmounts a disk (for example, `disktool -u disk2`).
`-p`	Unmounts a partition (for example, `disktool -p disk1s2`).
`-e`	Ejects a disk (for example, `disktool -e disk2`).
`-m`	Mounts a disk (for example, `disktool -m disk2`). Useful when a disk has been unmounted by hand (using `-p` or `-u` parameters).

`-a` Notifies of mount. Adds the disk to the Disk Arbitration's internal tables. Useful when you have already forced the mount and want to let applications know it. (for example, `disktool -a disk1 AFPVolName AFPFlags`).

`-d` Notifies of dismount. Removes the disk from Disk Arbitration's internal tables. Useful when you have already forced the unmount and want to let applications know it. (for example, `disktool -d disk1`).

Controlling disk parameters:

`-n` Renames volume. Renames the volume specified as the first argument. (for example, `disktool -n disk1s2 <`*`newName`*`>`).

`-g` Gets the HFS encoding on a volume. (For for example, `disktool -g disk1s2`).

`-s` Sets the HFS encoding on a volume. (For for example, `disktool -s disk1s2 4`).

`-A` Adopts the given device into the `volinfo` database.

`-D` Disowns the given device from the `volinfo` database.

`-S` Displays the status of the device in the `volinfo` database.

ditto

`ditto` Copies files and directories to a destination directory.

```
ditto [-v] [-V] [-arch <arch>] [-bom <bom>] [-rsrcFork] <src ...> <dst_directory>
ditto [-v] [-V] [-arch <arch>] [-rsrcFork] <src_file ...> <dst_file>
```

In the first synopsis form, `ditto` copies one or more source files or directories to a destination directory. If the destination directory doesn't exist, it will be created before the first source is copied. If the destination directory already exists, the source directories are merged with the previous contents of the destination.

In the second synopsis form, `ditto` copies a file to the supplied *`dst_file`*. The parent directory for *`dst_file`* must exist; otherwise, ditto will fall back to the first synopsis form.

`ditto` overwrites existing files, symbolic links, and devices in the destination when these are copied from a source. The resulting files, links, and devices will have the same mode, owner, and group as the source items from which they're copied. `ditto` doesn't modify the mode, owner, or group of existing directories in the destination. Files cannot overwrite directories or vice-versa.

`ditto` can be used to "thin" multi-architecture binaries during a copy. `ditto` can also copy files selectively based on the contents of a BOM (Bill of Materials) file. `ditto` preserves hardlinks present in the source directories and preserves setuid and setgid modes. Finally, `ditto` can also preserve resource fork and HFS meta-data information when copying files within or between file systems.

`-v` Print a line of output for each source directory copied.

`-V` Print a line of output for every file, symbolic link, and device copied.

`-arch <`*`arch`*`>` Thin multi-architecture binaries ("fat binaries") to the specified architecture. If multiple `-arch` options are specified, the resulting destination file will be multi-architectural, containing each of the specified architectures (if they're present in the source file). `arch` should be specified as `ppc`, `i386`, and so on.

`-bom <bom>`	If this option is given, only files, links, devices, and directories that are present in the specified BOM file are copied.
`-rsrcFork`	Preserve resource forks and HFS meta-data. `ditto` will store this data in AppleDouble files on file systems that don't support resource forks.

du

`du`	Displays disk usage statistics.

`du [-H | -L | -P] [-a | -s] [-ckrx] [<file> ...]`

`du` displays the file system block usage for each file argument and for each directory in the file hierarchy rooted in each directory argument. If no file is specified, the block usage of the hierarchy rooted in the current directory is displayed.

`-H`	Follows symbolic links on the command line. Symbolic links encountered during tree traversal aren't followed.
`-L`	Follows all symbolic links.
`-P`	Doesn't follow symbolic links.
`-a`	Displays an entry for each file in the file hierarchy.
`-s`	Displays only the grand total for the specified files.
`-c`	Displays the grand total after all the arguments have been processed.
`-k`	Displays the statistics in 1024-byte blocks. Default is 512-byte blocks.
`-r`	Generates a warning message about directories that cannot be read. This is the default.
`-x`	Doesn't traverse file system mount points.

`du` counts the storage used by symbolic links and not the files they reference unless `-H` or `-L` is specified. If either `-H` or `-L` is specified, the storage used by a symbolic link isn't counted or displayed. `-H`, `-L`, and `-P` override each other. The option specified last is the one executed.

Files with multiple hard links are counted and displayed once per `du` execution.

If the environment variable `BLOCKSIZE` is set and the `-k` option isn't used, the block counts are displayed according to the environment variable.

dump

`dump`	File system backup.

`dump [-0123456789cnu] [-B <records>] [-b <blocksize>] [-d <density>] [-f <file>] [-h <level>] [-s <feet>] [-T <date>] <filesystem>`

`dump [-W || -w]`

The 4.3BSD option syntax is implemented for backward compatibility, but isn't documented.

`dump` examines files on a file system and determines which files need to be backed up. These files are copied to the given disk, tape, or other storage medium for safekeeping. A dump that's larger than the output medium is broken into multiple volumes. On most media, the size is determined by writing until an end-of-media indication is returned.

On media that cannot reliably return an end-of-media, each volume is of a fixed size; the actual size is determined by the tape size and density and/or block count options. By default, the same output filename is used for each volume after prompting the operator to change media.

`-0-9` Dump levels. A level `0`, full backup, guarantees the entire file system is copied. A level number above `0`, an incremental backup, tells `dump` to copy all files new or modified since the last dump of the same or lower level. The default level is `9`.

`-c` Changes the defaults for use with a cartridge tape drive, with a density of 8000 bpi and a length of 1700 feet.

`-n` Notifies all operators in the operator group whenever `dump` requires operator attention.

`-u` Updates the file `/etc/dumpdates` after a successful dump. The format of `/etc/dumpdates` is readable by people, consisting of one free format record per line: file system name, increment level, and `ctime` (3). There may be only one entry per file system at each level.

`-B <records>` The number of kilobytes per volume, rounded down to a multiple of the block size. This option overrides the calculation of the tape size.

`-b <blocksize>` The number of kilobytes per `dump` record. Because the I/O system slices all requests into chunks of `MAXBSIZE` (typically 64KB), it isn't possible to use a larger block size without having problems later with restore. Therefore, `dump` constrains writes to `MAXBSIZE`.

`-d <density>` Sets the tape density to `<density>`. The default is 1600 bpi.

`-f <file>` Writes the backup to `<file>`. `<file>` may be a special device file, such as a tape drive or disk drive, an ordinary file, or `-` (standard output). Multiple filenames may be given as a single argument separated by commas. Each file will be used for one `dump` volume in the order listed. If the `dump` requires more volumes than the number of names listed, the last filename will be used for all remaining volumes after prompting for media changes. If the name of the file is of the form `<host>:<file>` or `<user>@<host>`: file, `dump` writes to the named file on the remote host using `rmt`.

`-h <level>` Honors the user `nodump` flag only for dumps at or above the given `<level>`. The default honor level is 1, so that incremental backups omit such files but full backups retain them.

`-s <feet>` Attempts to calculate the amount of tape needed at a particular density. If this amount is exceeded, `dump` prompts for a new tape. It's recommended to be a bit conservative on this option. The default tape length is 2300 feet.

`-T <date>` Uses the specified date as the starting time for the `dump` instead of the time determined from looking in `/etc/dumpdates`. The format of the date is the same as that of `ctime` (3). This option is useful for automated `dump` scripts that want to `dump` over a specified period of time. The `-T` flag is mutually exclusive from the `-u` flag.

`-W` Tells the operator what file systems need to be dumped. The information is gleaned from `/etc/dumpdates` and `/etc/fstab`. The `-W` flag causes dump to print out, for each file system in `/etc/dumpdates` the most recent `dump` date and level, and highlights of those file systems that should be dumped. If the `-W` flag is set, all other options are ignored, and `dump` exits immediately.

`-w` Is like `W`, but prints only those file systems that need to be dumped.

`dump` requires operator intervention on these conditions: end of tape, end of dump, tape write error, tape open error, or disk read error (if there are more than a threshold of 32). In addition to alerting all operators implied by the `-n` flag, `dump` interacts with an operator on `dump`'s control terminal at times when `dump` can no longer proceed, or if something is grossly wrong. All questions `dump` poses must be answered by typing `yes` or `no`.

Because making a `dump` involves a lot of time and effort for full `dumps`, `dump` checkpoints itself at the start of each tape volume. If writing that volume fails for some reason, `dump` will, with operator permission, restart itself from the checkpoint after the old tape has been rewound and removed, and a new tape has been mounted.

`dump` tells the operator what's going on at periodic intervals, including (usually low) estimates of the number of blocks to write, the number of tapes it will take, the time to completion, and the time to the tape change.

dumpfs

`dumpfs` Dumps file system information.

`dumpfs [<`*`filesystem`*`> | <`*`special`*`>]`

`dumpfs` prints out the super block and cylinder group information for the file system or special device specified. The listing is very long and detailed. This command is useful mostly for finding out certain file system information such as the file system block size and minimum free space percentage.

emacs

`emacs` Text editor.

`emacs [<`*`command-line switches`*`>] [<`*`file1`*`> <`*`file2`*`>...]`

`emacs` is a powerful editor that can actually do more than edit files. It has an extensive information system, which can be accessed in `emacs` with the key sequence `<Ctrl+h i>` (holding the `Control key` and `h` and then `i`). The information system can be navigated using the arrow keys to move around and pressing the Enter key to make a selection.

`emacs` has an interactive help facility, `<Ctrl+h>`. The information facility is one of the types of help available. A help tutorial is available with `<Ctrl+h t>`. Help Apropos `<Ctrl+h a>` helps the user find a command given its functionality. Help Character `<Ctrl+h c>` describes a given character's effect.

The following are `emacs` options of general interest:

`<`*`file`*`>` Edits the specified `<`*`file`*`>`.

`+<`*`number`*`>` Moves the cursor to the line number specified by `<`*`number`*`>`. (Don't include a space between + and `<`*`number`*`>`.)

`-q` Doesn’t load an init file.

`-u <user>` Loads the init file of the specified `<user>`.

`-t <file>` Uses the specified `<file>` as the terminal instead of using `stdin`/`stdout`. This must be the first argument specified in the command line.

`-nw` Tells `emacs` not to use its special X interface. If this option is given when invoking `emacs` in an `xterm(1)` window, the `emacs` display is done in that window. This must be the first option specified in the command line.

The following are basic `emacs` key sequences. Remember that two keys pressed simultaneously have a plus sign between them, and a space indicates pressing them sequentially. Most Unix documentation, including the online man pages and info pages document `Esc-x` as `M-x`, for the Meta key:

Up arrow Moves cursor up one line.

Left arrow Moves cursor to the left one character, to end of previous line if at left side of current line.

Right arrow Moves cursor to the right one character, move to the beginning of the next line if at the right side of the current line.

Down arrow Moves cursor down one line. Adds a new line to the file, if currently on the last line of the file.

Ctrl+p Moves cursor up one line.

Ctrl+b Moves cursor to the left one character, to end of previous line if at left side of current line.

Ctrl+f Moves cursor to the right one character, move to the beginning of the next line if at the right side of the current line.

Ctrl+n Moves cursor down one line. Adds a new line to the file, if currently on the last line of the file.

Ctrl+v Moves down one page in file.

Esc+v Moves up one page in file.

Ctrl+l Moves current line to the center of the page.

Ctrl+a Moves cursor to the beginning of the current line.

Ctrl+e Moves cursor to the end of the current line.

Esc+a Moves cursor to the beginning of the current sentence.

Esc+e Moves cursor to the end of the current sentence.

Ctrl+x Ctrl+h Brings up list of Ctrl+x prefixed commands. (If you do this, you’ll see that this table is a *very* abbreviated list!)

Ctrl+x Ctrl+s Saves the file.

Ctrl+x Ctrl+w Prompts for new name to save file.

Ctrl+x Ctrl+c Exits `emacs`.

Ctrl+x Ctrl+f Prompts to open file.

Ctrl+x Ctrl+b Lists current file buffers.

Ctrl+x b Prompts to switch to another buffer.

Esc+x `find-file-literally` Prompts to open file in literal mode—no Mac/Unix linefeed interpretation and so on.

Ctrl+x Ctrl+d Lists directory in `emacs` buffer (allows opening files by browsing directory rather than by typing name).

Ctrl+x Ctrl+o	Deletes blank lines in file.
Ctrl+x Ctrl+t	Transposes lines.
Ctrl+spacebar	Sets mark at the current cursor position.
Ctrl+x Ctrl+l	Downcases region. The region is the area between the cursor, and where the current mark is set.
Ctrl+x Ctrl+u	Upcases region. The region is the area between the cursor, and where the current Mark is set.
Ctrl+w	Deletes from mark to cursor. Deleted text goes to kill-ring buffer.
Esc+w	Copies from mark to cursor into kill-ring buffer.
Ctrl+k	Deletes from cursor to end of line. Place deleted text in kill-ring buffer.
Ctrl+y	Yanks top data from kill-ring buffer into the text at the current cursor position.
Ctrl+x 2	Splits current window vertically into two editing windows (two full-width windows, half the previous height).
Ctrl+x 3	Splits current window horizontally into two editing windows (two full-height windows, half the previous width).
Ctrl+x o	Switches to next editing window in split-window mode.
Ctrl+x 1	Switches to single-window mode, keeping the current window open.
Ctrl+x 0	Removes current editing window, keeping others.
Ctrl+x (	Starts recording keyboard macro.
Ctrl+x)	Stops recording keyboard macro.
Ctrl+x e	Executes recorded keyboard macro.
Ctrl+u `<####>`	Creates a numeric argument for the next command.
Ctrl+u `<####>` *`<keyseq>`*	Executes *`<keyseq>`* `####` times.
Ctrl+x f	Sets fill column for word wrap. Requires a numeric argument set with Ctrl+u `<####>`.
Esc+x `fill-region`	Word wraps region between cursor and mark.
Ctrl+h Ctrl+h	Brings up menu of help subjects.
Ctrl+h t	Brings up `emacs` tutorial.
Ctrl+h i	Brings up `emacs` info-mode manual browser. Browsing through the `emacs` info through this interface is recommended.
Esc+x `info`	Brings up `emacs` info-mode manual browser.
Esc+x `apropos`	Prompts for command or key sequence to document.
Ctrl+h h	Brings up list of ways to say hello in 34 different languages—we told you `emacs` had *everything* in it!

exports

`exports` Defines remote mount points for NFS requests.

The `exports` file specifies remote mount points for NFS mount protocol per the NFS server specification.

In a mount entry, the first field(s) specifies the directory path(s) within a server file system that clients can mount. There are two forms of this specification. The first form is to list all mount points as absolute directory paths separated by whitespace. The second form is to specify the pathname of the root of the file system followed by the -alldirs flag. This form allows hosts to mount at any point within the file system, including regular files if the -r option is used in mountd. The pathnames shouldn't have any symbolic links, or . or .. components.

The second component of a line specifies how the file system is to be exported to the host set. The options specify whether the file system is exported read-only or read-write and how the client UID is mapped to user credentials on the server.

The third component of a line specifies the client host set. The set may be specified in three ways. The first is to list the host names separated by whitespace. Standard Internet dot addresses may be used instead. The second way is to specify a netgroup, as defined in netgroup. The third way is to specify an Internet subnetwork using a network and network mask.

Export options are as follows:

-maproot=user
Credential of the specified user is used for remote access by root. The credential includes all groups to which the user is a member on the local machine. The user may be specified by name or number.

-maproot=user:group1:group2...
Specifies the precise credential to use for remote access by -root. The elements of the list may be names or numbers. Note that user: should be used to distinguish a credential containing no groups from a complete credential for the user.

-mapall=user

-mapalluser:group1:group2:...
Specifies a mapping for all client UIDs, including root, using the same semantics as -maproot.

-r
Synonym for -maproot for backward compatibility with older export file formats.

When neither -maproot nor -mapall is specified, remote accesses by root result in a credential of -2:-2. All other users are mapped to their remote credential. If a -maproot option is given, remote access by root is mapped according to the option instead of -2:-2. If -mapall is given, the credentials of all users, including root, are mapped as specified.

-kerb
Specifies that the Kerberos authentication server should be used to authenticate and map client credentials. This option requires that the kernel be built with the NFSKERB option.

-ro
Specifies that the file system should be exported read-only.

fetchmail

`fetchmail`
`fetchmail [options] [<`*`mailserver`*`>...]`
`fetchmailconf`
	Fetches mail from a POP-, IMAP-, or ETRN-capable server.

`fetchmail` fetches mail from remote servers and delivers it to your local machine. The retrieved mail can be read via conventional programs, such as `mail` or `elm`. `fetchmail` can be run in daemon mode to repeatedly poll one or more servers at a specified interval.

If `fetchmailconf` is available, it can be used to assist in setting up and editing your `.fetchmailrc` configuration file. `fetchmailconf` runs under the X Window System and requires `pythong` and `tk`. If `fetchmailconf` isn't available, you can also use a text editor to create your `.fetchmailrc` file.

`fetchmail` can be run at the command line, but the preferred way is through the `.fetchmailrc` file in your home directory. See the `fetchmail` man page for details on setting up a `.fetchmailrc` file.

If you run `fetchmail` at the command line and also have a `.fetchmailrc` file, options specified in the command-line override specifications in the `.fetchmailrc`. Each server that you specify on the command line will be queried according to the options given. If no server is specified, each poll entry of the `.fetchmailrc` is queried.

Almost all the options have a corresponding keyword that can be used to declare them in `.fetchmailrc`. The following options rarely have to be used at the command line after you have a working `.fetchmailrc`.

`-?`	Displays a list of options with brief descriptions.
`--help`	Same as `-?`.
`-V` `--version`	Displays version information.
`-c` `--check`	Checks for messages without fetching them.
`-s` `--silent`	Silent mode. Suppresses progress/status and error messages. Is overridden by `-verbose`.
`-v` `--verbose`	Verbose mode. Produces diagnostic output. Overrides `-silent`. Doubling the option (`-v -v`) causes extra diagnostic information to be printed.
`-d` `--daemon`	Runs as a daemon once per *n* seconds.
`-N` `--nodetach`	Doesn't detach daemon process.
`-q` `--quit`	Kills daemon process.
`-L` `--logfile`	Specifies logfile name.
`--syslog`	Uses `syslog`(3) for most messages when running as a daemon.
`--invisible`	Doesn't write Received and enables host spoofing.
`--postmaster`	Specifies last resort recipient.

`--nobounce`	Redirects bounces from user to postmaster.
`-a`	Retrieves all messages, both read and unread. Default is to retrieve only unread messages.
`--all`	Same as `-a`.
`-k`	Saves new messages after retrieval. Default is to delete messages from the server after they have been retrieved. This doesn't work with ETRN.
`--keep`	Same as `-k`.
`-K`	Deletes new messages after retrieval. Useful if you have specified a default of `-keep` in `.fetchmailrc`. Option is forced on with ETRN.
`--nokeep`	Same as `-K`.
`-F`	POP/IMAP only. Deletes previously retrieved messages from server. Doesn't work with ETRN. If your MTA hangs and `fetchmail` is aborted, the next time you run `fetchmail`, it will delete the messages that weren't delivered to you. The `-keep` option is recommended instead.
`--flush`	Same as `-F`.
`-p <proto>`	Specifies retrieval protocol to use when retrieving mail from the remote server. If no protocol is specified, AUTO is assumed. See man page for list of protocols.
`--protocol <proto>`	Same as `-p <proto>`.
`-U`	Forces UIDL (unique ID listing) use (POP3 only). Forces client-side tracking of newness of messages.
`--uidl`	Same as `-U`.
`-P`	Specifies TCP/IP port for making connection. Rarely necessary, as the protocols have well-established default port numbers.
`--port`	Same as `-P`.
`-t`	Specifies server nonresponse timeout in seconds.
`--timeout`	Same as `-t`.
`--plugin`	Specifies external command to open TCP connection. Useful for using socks or a special firewall setup.
`--plugout`	Specifies external command to open SMTP connection.
`-r <folder>`	Specifies remote folder name.
`--folder <folder>`	Same as `-r <folder>`.
`-S <hosts>`	Specifies a hunt list of SMTP forwarding hosts.
`--smtphost <hosts>`	Same as `-S <hosts>`.
`-D <domain>`	Sets SMTP delivery domain to use.
`--smptaddress <domain>`	Same as `-D <domain>`.
`-Z <nnn>`	
`--antispam <nnn>[,<nnn>][,<nnn>...]`	Sets antispam response values. A value of `-1` disables this option. For the command-line option, list should be a comma-separated list.

`--mda`	Forces mail to be passed to an MDA directly rather than forwarded to port 25. Not recommended unless running an SMTP listener is impossible.
`--lmtp`	Uses LMTP (Local Mail Transfer Protocol, RFC2033) for delivery. Service port must be specified because the default port 25 won't be accepted.
`--bsmtp`	Appends fetched mail to a BSMTP file. File contains the SMTP commands that would normally be generated by `fetchmail` when passing mail to an SMTP listener daemon.
`-l`	Doesn't fetch messages larger a specified size. Takes a maximum octet size as argument. Doesn't work with ETRN.
`--limit`	Same as `-l`.
`-w`	Interval in seconds between warning mail notification.
`--warnings`	Same as `-w`.
`-b`	Sets batch limit for SMTP connections. Default is `0`, no limit.
`--batchlimit`	Same as `-b`.
`-B`	Sets fetch limit for number of messages accepted from a given server in a single poll. Doesn't work with ETRN.
`--fetchlimit`	Same as `-B`.
`-e`	Makes deletions final after a given number of messages. Doesn't work with ETRN.
`--expunge`	Same as `-e`.
`-u` *<name>*	Specifies user identification to be used when logging in to the mail server. Default is login name on the client machine running `fetchmail`.
`--username` *<name>*	Same as `-u` *<name>*.
`-A`	Specifies preauthentication type. Possible values are `password`, `kerberos_v5`, or `kerberos`.
`--preauth`	Same as `-A`.
`-f` *<pathname>*	Specifies an alternative run control file. Unless `-version` is also on, the name file must have permissions no more open than 0600 or else be `/dev/null`.
`--fetchmailrc` *<pathname>*	
`-i` *<pathname>*	Specifies alternative name for the `.fetchids` file used to save POP3 UIDs.
`--iidfile` *<pathname>*	
`-n`	Doesn't rewrite header addresses. Normally fetches rewrite headers in fetched mail so that any mail IDs local to the server are expanded to full addresses so that replies on the client are addressed correctly, rather than to the local users.
`--norewrite`	Same as `-n`.
`-E`	Changes the header that `fetchmail` assumes will carry a copy of the mail's envelope address.
`--envelope`	Same as `-E`.

`-Q`	Prefix to remove from local user ID. Useful if you're using `fetchmail` to retrieve mail for an entire domain and your ISP is using `qmail`.
`--qvirtual`	Same as `-Q`.

fg

`fg` Foregrounds a job.

`fg [%<job>...]`

`%<job>`

`fg`

Brings the specified jobs (or, if no argument is given, the current job) to the foreground, continuing each as though it had stopped. `<job>` may be any acceptable form as described in `jobs`. Like backgrounding jobs, referring to a backgrounded job in % notation brings it to the foreground, that is, `%1` foregrounds background job `1`.

find

`find` Finds files.

`find [-H | -L | -P] [-EXdsx] [-f <file>] <file> .... <expression>`

`find` recursively descends the directory tree of each file listing, evaluating an `<expression>` composed of primaries and operands.

Options

`-E`	Causes `find` to interpret regular expression patterns specified with `–regex` or `–iregex` as standard modern regular expressions, rather than as basic regular expressions (BREs). See `re_format`(7) manual page for a description of each format.
`-H`	Causes the file information and file type returned for each symbolic link on the command line to be those of the file referenced, rather than those of the link itself. If the file doesn't exist, the information is for the link itself. File information of symbolic links not on the command line is that of the link itself.
`-L`	Causes the file information and file type returned for each symbolic link to be those of the referenced file, rather than those of the link itself. If the referenced file doesn't exist, the information is for the link itself.
`-P`	Causes the file information and file type returned for each symbolic link to be those of the link itself.
`-X`	Permits `find` to be safely used with `xargs`. If a filename contains any delimiting characters used by `xargs`, an error message is displayed and the file is skipped. The delimiting characters include single quote, double quote, backslash, space, tab, and newline.

`-d` Causes a depth-first traversal of the hierarchy. In other words, directory contents are visited before the directory itself. The default is for a directory to be visited before its contents.

`-s` Causes `find` to traverse the file hierarchies in lexicographical order, that is, alphabetical order within each directory. Note: `find -s` and `find | sort` may give different results.

`-x` Excludes `find` from traversing directories that have a device number different from that of the file from which the descent began.

`-h` Causes the file information and file type returned for each symbolic link to be those of the referenced file, rather than those of the link itself. If the referenced file doesn't exist, the information returned is for the link itself.

`-f` Specifies a file hierarchy for `find` to traverse. File hierarchies may also be specified as operands immediately following the options listing.

Primaries (Expressions)

All primaries that can take a numeric argument allow the number to be preceded by `+`, `-`, or nothing. *n* takes on the following meanings:

`+`*n* More than *n*

`-`*n* Less than *n*

`n` Exactly *n*

`-atime` *n* True if the file was last accessed *n* days ago. Note that `find` itself will change the access time.

`-ctime` *n* True if the file's status was changed *n* days ago.

`-mtime` *n* True if the file was last modified *n* days ago.

`-newer <`*`file`*`>` True if the current file has a more recent modification time than `<`*`file`*`>`.

`-exec <`*`command`*`>;` True if `<`*`command`*`>` returns a zero-value exit status. Optional arguments may be passed to `<`*`command`*`>`. The expression must be terminated by a semicolon. If `{}` appear anywhere in the command name or arguments, they're replaced by the current pathname.

`-follow` Follows symbolic links.

`-fstype` True if the file is contained in a file system specified by `-fstype`. Issue the command `sysctl vfs` to determine the available types of file systems on the system. There are also two pseudo types: `local` and `rdonly`. `local` matches any file system physically mounted on the system where `find` is being executed; `rdonly` matches any mounted read-only file system.

`-group <`*`gname`*`>` True if the file belongs to the specified group name. If `<`*`gname`*`>` is numeric and there's no such group name, `<`*`gname`*`>` is treated as the group ID.

`-user <uname>` True if file belongs to the user *<uname>*. If *<uname>* is numeric and there's no such user *<uname>*, it's treated as the user ID.

`-nouser` True if the file belongs to an unknown user.

`-nogroup` True if the file belongs to an unknown group.

`-inum n` True if the file has inode number *n*.

`-links n` True if the file has *n* links.

`-ls` Always true. Prints the following file statistics: inode number, size in 512-byte blocks, file permissions, number of hard links, owner, group, size in bytes, last modification time, and filename. If the file is a symbolic link, the display of the file it's linked to is preceded by `->`. The display from this `ls` is identical to that displayed by `ls -dgils`.

`-ok <command>` Same as `-exec`, except that confirmation from the user is requested before executing *<command>*.

`-name <pattern>` True if the filename contains *<pattern>*. Special shell pattern-matching characters (`[`, `]`, `*`, `?`) may be used as part of *<pattern>*. A backslash (`\`) is used to escape those characters to explicitly search for them as part of *<pattern>*.

`-path <pattern>` True if the pathname contains *<pattern>*. Special shell pattern-matching characters (`[`, `]`, `*`, `?`) may be used as part of *<pattern>*. A backslash (`\`) is used to escape those characters to explicitly search for them as part of *<pattern>*. Slashes (`/`) are treated as normal characters and don't need to be escaped.

`-perm [-]<mode>` *<mode>* may be either symbolic or octal (see `chmod`). If *<mode>* is symbolic, a starting value of `0` is assumed, and *<mode>* sets or clears permissions without regard to the process's file mode creation mask. If *<mode>* is octal, only bits 0777 of the file's mode bits are used in the comparison. If *<mode>* is preceded by a dash (`-`), this evaluates to true if at least all the bits in *<mode>* are set in the file's mode bits. If *<mode>* isn't preceded by a dash, this evaluates to true if the bits in *<mode>* match exactly the file's mode bits. If *<mode>* is symbolic, the first character may not be a dash.

`-print0` Always true. Prints the current pathname followed by a null character.

`-print` Always true. Prints the current pathname followed by a newline character. If none of `-exec`, `-ls`, `-ok`, or `-print0` is specified, `-print` is assumed.

`-prune` Always true. Doesn't descend into current file after the pattern has been matched. If `-d` is specified, `-prune` has no effect.

`-regex pattern`	True if the whole path of the file matches pattern using regular expression matching. To match a file named `./Documents/zzyzygy`, you can use the regular expression `.*/[gyz]*` or `.*/Documents/.*`, but not `zzyzygy` or `/Documents/` because the regular expression must match the *entire* filename, including the path portion (this is as opposed to the action of the `-name` argument, which would find the `zzyzygy` file with `-name zzyzygy`).
`-iregex pattern`	The same as `-regex`, except use a case-insensitive match.
`-size` *`n`*`[c]`	True if the file size, rounded up, is *n* 512-byte blocks. If `c` follows *n*, it's true if the file size is *n* bytes.
`-type` *`t`*	True if the file is of the specified type. Possible file types are `W` Whiteout `b` Block special `c` Character special `d` Directory `f` Regular file `l` Symbolic link `p` FIFO `s` Socket

Operators

Primaries may be combined using the following operators (in order of decreasing precedence).

`(`*`expression`*`)`	True if the parenthesized expression evaluates to true.
`!`*`expression`*	True if the expression is false. (`!` is the unary, not the operator.)
`expression` `[-and]` *`expression`*	
`expression expression`	True if both expressions are true. The second expression isn't evaluated if the first is false. (`-and` is the logical AND operator.)
`expression` `-or` *`expression`*	True if either expression is true. The second expression isn't evaluated if the first is true. (`-or` is the logical OR operator.)

from

`from`	Prints names of those who have sent mail.
`from [-s <`*`sender`*`>] [-f <`*`file`*`>] [<`*`user`*`>]`	
`-s <`*`sender`*`>`	Only prints entries from addresses containing the string `<`*`sender`*`>`.
`-f <`*`file`*`>`	Examines `<`*`file`*`>` instead of the invoker's mailbox. If `-f` is used, `<`*`user`*`>` shouldn't be used.
`<`*`user`*`>`	Examines `<`*`user`*`>`'s mailbox rather than the invoker's mailbox. Privileges are required.

fsck

`fsck` File system consistency check and interactive repair.

`fsck -p [-m <mode>]`

`fsck [-b <block#>] [-c <level>] [-l <maxparallel>] [-y] [-n] [-m <mode>] [<filesystem>] ...`

The first form of `fsck` preens a standard set of file systems or the specified file systems. It is's normally used in the `script /etc/rc` during automatic reboot. Here `fsck` reads the table `/etc/fstab` to determine which file systems to check. Only partitions in `fstab` that are mounted `rw`, `rq`, or `ro` and that have non-zero pass number are checked. File systems with pass number 1 (normally just the root file system) are checked one at a time. When pass 1 completes, all remaining file systems are checked, running one process per disk drive. The disk drive containing each file system is inferred from the longest prefix of the device name that ends in a digit; the remaining characters are assumed to be the partition designator. In preening mode, file systems that are marked clean are skipped. File systems are marked clean when they are unmounted, when they have been mounted read-only, or when `fsck` runs on them successfully.

The kernel takes care that only a restricted class of innocuous file system inconsistencies can happen unless hardware or software failures intervene. These are limited to the following:

Unreferenced inodes

Link counts in inodes too large

Missing blocks in the free map

Blocks in the free map also in files

Counts in the super-block wrong

These are the only inconsistencies that `fsck` with the `-p` option corrects; if it encounters other inconsistencies, it exits with an abnormal return status and an automatic reboot then fails. For each corrected inconsistency, one or more lines are printed identifying the file system on which the correction will take place, and the nature of the correction. After successfully correcting a file system, `fsck` prints the number of files on that file system, the number of used and free blocks, and the percentage of fragmentation.

If sent a QUIT signal, `fsck` finishes the file system checks, and then exits with an abnormal return status that causes an automatic reboot to fail. This is useful when you want to finish the file system checks during an automatic reboot, but don't want the machine to come up multiuser after the checks complete.

Without the `-p` option, `fsck` audits and interactively repairs inconsistent conditions for file systems. If the file system is inconsistent, the operator is prompted for concurrence before each correction is attempted. It should be noted that some of the corrective actions that aren't correctable under the `-p` option will result in some loss of data. The amount and severity of data lost may be determined from the diagnostic output. The default action for each consistency correction is to wait for the operator to respond `yes` or `no`. If the operator doesn't have write permission on the file system, `fsck` defaults to a `-n` action.

`-b <block#>` Uses the block specified immediately after the flag as the super block for the file system. Block 32 is usually an alternative super block.

`-c <level>` Converts the file system to the specified level. Note that the level of a file system can only be raised.

There are currently four levels defined:

0 The file system is in the old (static table) format.

1 The file system is in the new (dynamic table) format.

2 The file system supports 32-bit uids and gids, short symbolic links are stored in the inode, and directories have an added field showing the file type.

3 If `maxcontig` is greater than one, build the free segment maps to aid in finding contiguous sets of blocks. If `maxcontig` is equal to one, delete any existing segment maps.

In interactive mode, `fsck` lists the conversion to be made and asks whether the conversion should be done. If a negative answer is given, no further operations are done on the file system. In preen mode, the conversion is listed and done if possible without user interaction. Conversion in preen mode is best used when all the file systems are being converted at once. The format of a file system can be determined from the first line of output from `dumpfs`(8).

`-l` *`<maxparallel>`* Limits the number of parallel checks to the number specified by *`<maxparallel>`*. By default, the limit is the number of disks, running one process per disk. If a smaller limit is given, the disks are checked round-robinround robin, one file system at a time.

`-y` Assumes a yes response to all questions asked by `fsck`; this should be used with great caution as this is a free license to continue after essentially unlimited trouble has been encountered.

`-n` Assumes a no response to all questions asked by `fsck` except for CONTINUE?, which is assumed to be affirmative; doesn't open the file system for writing.

`-m` *`<mode>`* Uses the *`<mode>`* specified in octal immediately after the flag as the permission bits to use when creating the lost+found directory rather than the default 1777. In particular, systems that don't want to have lost files accessible by all users on the system should use a more restrictive set of permissions such as 700.

If no file systems are given to `fsck`, a default list of file systems is read from the file `/etc/fstab`.

Inconsistencies checked are as follows:

1. Blocks claimed by more than one inode or the free map.
2. Blocks claimed by an inode outside the range of the file system.
3. Incorrect link counts.
4. Size checks:
   Directory size not a multiple of DIRBLKSIZ.
   Partially truncated file.
5. Bad inode format.
6. Blocks not accounted for anywhere.
7. Directory checks:
   File pointing to unallocated inode.
   Inode number out of range.
   Dot or dot-dot not the first two entries of a directory or having the wrong inode number.
8. Super Block checks:
   More blocks for inodes than there are in the file system.

Bad free block map format.

Total free block and/or free inode count incorrect.

Orphaned files and directories (allocated but unreferenced) are, with the operator's concurrence, reconnected by placing them in the lost+found directory. The name assigned is the inode number. If the lost+found directory doesn't exist, it is's created. If there is's insufficient space, its size is increased. Because of inconsistencies between the block device and the buffer cache, the raw device should always be used.

fstab

`fstab`	Static information about the file systems.

The `fstab` files contain descriptive information about the various file systems. The `fstab` is only read by programs, not written by them. Each file system is described on a separate line. Fields in each line are separated by tabs or spaces. The order of records in the `fstab` is important because `fsck` (8), `mount` (8), and `umount` (8) sequentially iterate through the `fstab`.

The first field (`fs_spec`) describes the block special device or remote file system to be mounted. The `autodiskmount`(8) program supports the identification of a local file system uniquely by its UUID or by its volume name, irrespective of hardware configuration and of hardware parallelism, using the constructs UUID and LABEL.

The second field (`fs_file`) describes the mount point for the file systems. For swap partitions, the field should be specified as none.

The third field (`fs_vfstype`) describes the type of file system. The currently supported file systems are

`ufs`	A local UNIX file system.
`cd9660`	ISO 9660 CD ROM.
`fdesc`	An implementation of `/dev/fd`.
`kernfs`	Various kernel statistics.
`mfs`	A local memory-based file system.
`msdos`	MSDOS FAT file system.
`nfs`	Sun Microsystems compatible Network File System.
`procfs`	A local file system of process information.
`swap`	Disk partition to be used for swapping.
`union`	A translucent file system.

The fourth field (`fs_mntops`) describes mount options associated with the file system. It's formatted as a comma-separated list of options. It contains at least the mount type and any additional options appropriate to the file system type.

The option `auto` can be used in the `noauto` form to cause a file system not to be automatically mounted (with `mount -a` or at system boot time).

The options `userquota` and/or `groupquota` cause the file system to be automatically processed by `quotacheck` (8) command, and user and/or group disk quotas are enabled with `quotaon` (8). By default, file system quotas are maintained in the files named `quota.user` and `quota.group` located at the root of the associated file system. The defaults may be overridden by appending to the quota option `=<`*`absolute-path-to-quota-file`*`>`.

The type of the mount is extracted from the fs_mntops field and stored separately in a field named fs_type (it isn't deleted from the fs_mntops field). If fs_type is rw or ro, the file system whose name is given in the fs_file field is normally mounted read-write or read-only on the specified special file. If fs_type is sw, the special file is made available as a piece of swap space by the swapon(8) command at the end of the system reboot procedure.

The fifth field (fs_freq) is used by dump (8) to determine which file systems need to be dumped. If the field isn't present or is zero, a value of zero is returned, and dump assumes that the file system doesn't need to be dumped.

The sixth field (fs_passno) is used by fsck (8) to determine the order in which file system checks should be done at reboot time. The root file system should be specified with a value of 1. Other file systems should have a value of 2. File systems within a drive are checked sequentially, whereas file systems on different drives are checked at the same time to use parallelism available in the hardware. If the field isn't present or is zero, a value of zero is returned, and fsck assumes that the file system doesn't need to be checked.

ftp

ftp	File transfer program.

```
ftp [-AadefginpRtvV] [-o <outfile>] [-P <port>] [-r <seconds>]
          [-T <dir>,<max>[,<inc>][[<user>@]<host> [<port>]]] [<host>:<path>[/]]
          [file:///<file>] [ftp://[<user>[:<pass>]@]<host>[:<port>]/<path>[/]]
          [http://[<user>[:<pass>]@]<host>[:<port>]/<path>] [...]
ftp -u <url> <file> [...]
```

The remote host with which ftp is to communicate can be specified on the command line. Done this way, ftp immediately tries to establish a connection with the remote host. Otherwise, ftp enters its command interpreter mode, awaits commands from the user, and displays the prompt ftp>.

-A	Forces active mode ftp. By default, ftp tries to use passive mode ftp and falls back to active mode if passive isn't supported by the server.
-a	Causes ftp to bypass normal login procedure, and use an anonymous login instead.
-d	Enables debugging.
-e	Disables command-line editing.
-f	Forces a cache reload for transfers that go through the FTP or HTTP proxies.
-g	Disables filename globbing.
-i	Turns off interactive mode when transferring multiple files.
-n	Doesn't attempt auto-login upon initial connection. If auto-login isn't disabled, ftp checks for a .netrc file in the user's directory for an entry describing an account on the remote machine. If no entry is available, ftp prompts for the login name on the remote machine (defaults to the login name on the local machine), and if necessary, prompts for a password.

`-p` Enables passive mode operation for use behind connection filtering firewalls. This option has been deprecated as `ftp` now tries to use passive mode by default, falling back to active mode if the server doesn't support passive connections.

`-R` Restarts all non-proxied auto-fetches.

`-t` Enables packet tracing.

`-v` Enables verbose and progress. Default if output is to a terminal (and for progress, if `ftp` is in the foreground). Shows all responses from the remote server as well as transfer statistics.

`-V` Disables verbose and progress, overriding the default of enabled when output is to a terminal.

`-o <output>` When auto-fetching files, saves the contents in `output`. If `output` isn't `-` or doesn't start with `|`, only the first file specified is retrieved into `output`; all other files are retrieved into the basename of their remote name.

`-P <port>` Sets the port number to `<port>`.

`-r <seconds>` Retries the connection attempt if it failed, pausing for `<seconds>` seconds.

`-T <direction>, <maximum> [,<increment>]` Sets the maximum transfer rate for `<direction>` to `<maximum>` bytes/second, and if specified, the `<increment>` to `<increment>` bytes/second.

`-u <url> <file>` Uploads files on the command line to `<url>` where `<url>` is one of the `ftp` URL types as supported by auto-fetch (with an optional target filename for single file uploads), and *`file`* is one or more local files to be uploaded.

When `ftp` is in its command interpreter mode awaiting instructions from the user, there are many commands that the user might issue. Some of them include:

`ascii` Sets the file transfer type to network `ASCII`. Although this is supposed to be the default, it isn't uncommon for an FTP server to indicate that `binary` is its default.

`binary` Sets the file transfer type to support binary image transfer.

`image` Same as `binary`.

`bye` Terminates the `ftp` session and exits `ftp`. An end of file also terminates the session and exits.

`quit` Same as `bye`.

`cd <remote_directory>` Changes the current working directory on the remote host to `<remote_directory>`.

`cdup` Changes the current working directory on the remote host to the parent directory.

`lcd <directory>` Changes the working directory on the local machine. If no directory is specified, the user's home directory is used.

`close` Terminates the `ftp` session with the remote host and returns to the command interpreter.

`disconnect`	Same as `close`.
`dir [<remote-directory> [<local_file>]]`	Prints a listing of the directory on the remote machine. Most Unix systems produce an `ls -l` output. If `<remote_directory>` isn't specified, the current directory is assumed. If `<local_file>` isn't specified, or is `-`, the output is sent to the terminal.
`ftp <hostname> [<port>]`	Same as `open`.
`open <hostname> [<port>]`	Attempts to establish an `ftp` connection on `<hostname>` at `<port>`, if `<port>` is specified.
`glob`	Toggles filename expansion for `mdelete`, `mget`, and `mput`. If globbing is turned off, filename arguments are taken literally and not expanded.
`delete <remote_file>`	Deletes the specified `<remote_file>` on the remote machine.
`mdelete <remote_files>`	Deletes the specified `<remote_files>` on the remote machine.
`get <remote_file> [<local-file>]`	Downloads `<remote_file>` from the remote machine to the local machine. If `<local_file>` isn't specified, the file is also saved on the local machine with the name `<remote_file>`.
`recv <remote_file> [<local_file>]`	Same as `get`.
`mget <remote_files>`	Downloads the specified `<remote_files>`.
`put <local_file> [<remote_file>]`	Uploads the specified `<local_file>` to the remote host. If `<remote_file>` isn't specified, the file is saved on the remote host with the name `<local_file>`.
`send <local_file> [<remote_file>]`	Same as `put`.
`mput <local_files>`	Uploads the specified `<local_files>`.
`msend`	Same as `mput`.
`help [<command>]`	Displays a message describing `<command>`. If `<command>` isn't specified, a listing of known commands is displayed.
`?`	Same as `help`.
`ls [<remote_directory> [<local_file>]]`	Prints a list of the files in a directory on the remote machine. If `<remote_directory>` isn't specified, the current working directory is assumed.If `<local_file>` isn't specified, or is `-`, the output is printed to a terminal. Note that if nothing is listed, the directory might only have directories in it. Try `ls -l` or `dir` for a complete listing.
`mkdir <directory>`	Makes the specified `<directory>` on the remote machine.
`rmdir <directory>`	Removes the specified `<directory>` from the remote machine.

`passive [auto]`
Toggles passive mode if no argument is given. If `auto` is given, acts as if `FTPMODE` is set to `auto`. If passive mode is turned on (default), the `ftp` client sends a `PASV` command for data connections rather than a `PORT` command. `PASV` command requests that the remote server open a port for the data connection and return the address of that port. The remote server listens on that port and the client then sends data to it. With the `PORT` command, the client listens on a port and sends that address to the remote host, who connects back to it. Passive mode is useful when FTPing through a firewall. Not all `ftp` servers are required to support passive mode.

`pwd`
Prints the current working directory on the remote host.

`rate <direction> [<maximum> [<increment>]]`
Throttles the maximum transfer rate to `<maximum>` bytes/second. If `<maximum>` is 0, disables the throttle. Not yet implemented for `ascii` mode.

`<direction>` may be any one of: `get` (incoming transfers); `put` (outgoing transfers); `all` (both).

`<maximum>` can be modified on the fly by `<increment>` bytes (default: 1024) each time a given signal is received: `SIGUSR1` (increments `<maximum>` by `<increment>` bytes; `SIGUSR2` (decrements `<maximum>` by `<increment>` bytes—result must be a positive number).

If `<maximum>` isn't supplied, displays current throttle rates.

`verbose`
Toggles verbose mode. Default is `on`. In verbose mode, all responses from the `ftp` server are shown as well as transfer statistics.

ftpd

`ftpd`
Internet File Transfer Protocol server.

```
ftpd [-dHlqQrsuUwWX] [-a <anondir>] [-c <confdir>] [-C <user>] [-e <emailaddr>] [-h
<hostname>] [-P <dataport>] [-V <version>]
```

`ftpd` is the Internet File Transfer Protocol process. It uses the TCP protocol and runs on the port specified as `ftp` in the services directory of the NetInfo database.

`-a <anondir>`
Defines `<anondir>` as the directory to `chroot`(2) into for anonymous logins. Default is the home directory for the `ftp` user. This can also be specified with the `ftpd.conf`(5) `chroot` directive.

`-c <confdir>`
Changes the root directory of the configuration files from `/etc` to `<confdir>`. This changes the directory for the following files: `/etc/ftpchroot`, `/etc/ftpusers`, `/etc/ftpwelcome`, `/etc/motd`, and the file specified by the `ftpd.conf`(5) limit directive.

`-C <user>`	Checks whether the user would be granted access under the restrictions given in `ftpusers`(5) and exits without attempting a connection. `ftpd` exits with an exit code of `0` if access would be granted, or `1` otherwise. This can be useful for testing configurations.
`-d`	Debugging information is written to the syslog using a facility of `LOG_FTP`.
`-e <emailaddr>`	Uses `<emailaddr>` for the `%E` escape sequence (see display file escape sequences).
`-h <hostname>`	Explicitly sets the hostname to advertise as `<hostname>`. Default is the hostname associated with the IP address that `ftpd` is listening on. This ability (with or without `-h`), in conjunction with `-c <confdir>`, is useful when configuring virtual FTP servers, each listening on separate addresses as separate names.
`-H`	Equivalent to `-h <hostname>`.
`-l`	Logs each successful and failed FTP session using syslog with a facility of `LOG_FTP`. If this option is specified more than once, the retrieve (get), store (put), append, delete, make directory, remove directory, and rename operations and their filename arguments are also logged.
`-P <dataport>`	Uses `<dataport>` as the data port, overriding the default of using the port one less than the port `ftpd` is listening on.
`-q`	Enables the use of PID files for keeping track of the number of logged-in users per class. This is the default.
`-Q`	Disables the use of PID files for keeping track of the number of logged-in users per class. This might reduce the load on heavily loaded FTP servers.
`-r`	Permanently drops root privileges after the user is logged in. The use of this option could result in the server using a port other than the (listening port–1) for PORT-style commands, which is contrary to the RFC 959 specification, but in practice very few clients rely on this behavior.
`-s`	Requires a secure authentication mechanism such as Kerberos or S/Key to be used.
`-u`	Logs each concurrent FTP session to `/var/run/utmp`, making them visible to commands such as `who`(1).
`-U`	Doesn’t log each concurrent FTP session to `/var/run/utmp`. This is the default.
`-V <version>`	Uses `<version>` as the version to advertise in the login banner and in the output of `STAT` and `SYST` instead of the default version information. If version is `-` or empty, doesn’t display any version information.

`-w` Logs each FTP session to `/var/log/wtmp`, making them visible to commands such as `last`(1). This is the default.

`-W` Doesn't log each FTP session to `/var/log/wtmp`.

`-X` Logs `wu-ftpd` style xferlog entries to the syslog, prefixed with `xferlog:`, using a facility of `LOG_FTP`. These syslog entries can be converted to a `wu-ftpd` style xferlog file suitable for input into a third-party log analysis tool with a command similar to

```
grep 'xferlog: ' /var/log/xferlog | \
sed -e 's/^.*xferlog: //' > wuxferlog
```

`ftpd` supports the following FTP requests, case ignored.

`ABOR` Aborts previous command.

`ACCT` Specifies account (ignored).

`ALLO` Allocates storage (vacuously).

`APPE` Appends to a file.

`CDUP` Changes to the parent directory of the current working directory.

`CWD` Changes current working directory.

`DELE` Deletes a file.

`EPSV` Prepares for server-to-server transfer.

`EPRT` Specifies data connection port.

`FEAT` Lists extra features that aren't defined in RFC 959.

`HELP` Gives help information.

`LIST` Gives list files in a directory (`ls -lgA`).

`LPSV` Prepares for server-to-server transfer.

`LPRT` Specifies data connection port.

`MLSD` Lists contents of directory in a form that can be processed by a machine.

`MLST` Shows a pathname in a form that can be processed by a machine.

`MKD` Makes a directory.

`MDTM` Shows last modification time of file.

`MODE` Specifies data transfer mode.

`NLST` Gives name list of files in directory.

`NOOP` Does nothing.

`OPTS` Defines persistent options for a given command.

`PASS` Specifies password.

`PASV` Prepares for server-to-server transfer.

`PORT` Specifies data connection port.

`PWD` Prints current working directory.

`QUIT` Terminates session.

`REST` Restarts incomplete transfer session.

`RETR` Retrieves a file.

`RMD` Removes a directory.

`RNFR` Specifies rename-from filename.

`RNTO` Specifies rename-to filename.

SITE	Nonstandard commands (see next section).
SIZE	Returns size of file.
STAT	Returns status of server.
STOR	Stores a file.
STOU	Stores a file with a unique name.
STRU	Specifies data transfer structure.
SYST	Shows operating system type of server system.
TYPE	Specifies data transfer type.
USER	Specifies username.
XCUP	Changes to parent of current working directory (deprecated).
XCWD	Changes working directory (deprecated).
XMKD	Makes a directory (deprecated).
XPWD	Prints the current working directory (deprecated)
XRMD	Removes a directory (deprecated).

The following nonstandard commands are supported by the SITE request:

UMASK	Changes the umask; for example, SITE UMASK 002.
IDLE	Sets the idle timer; for example, SITE IDLE 60
CHMOD	Changes the mode of a file; for example, SITE CHMOD 755 <*filename*>.
HELP	Gives help information.
RATEGET	Sets maximum get rate throttle in bytes/second; for example, SITE RATEGET 5k.
RATEPUT	Sets maximum put rate throttle in bytes/second; for example, SITE RATEPUT 5k.

The following FTP requests (as specified in RFC 959) are recognized, but aren't implemented: ACCT, SMNT, and REIN. MDTM and SIZE aren't specified in RFC 959, but will appear in the next updated FTP RFC.

ftpd interprets filenames according to the globbing conventions by csh(1). This enables users to use the meta characters: *, ?, [], {}, and ~.

ftpd authenticates users according to these rules:

1. Login name must be in the password database and not have a null password.
2. Login name must be allowed based on the information in /etc/ftpusers.
3. User must have a standard shell returned by getusershell(3). If the user's shell field in the password database is empty, the shells(5), the user's shell, must be listed with full path in /etc/shells.
4. If directed by the file ftpchroot(5), the session's root directory will be changed by chroot(2) to the directory specified in the ftpd.conf(5) chroot directive (if set), or to the home directory of the user. However, the user must still supply a password. This feature is intended as a compromise between a fully anonymous account and a fully privileged account. The account should also be set up as for an anonymous account.
5. If the username is anonymous or ftp, an anonymous FTP account must be present in the password file (user ftp). In this case, the user is allowed to log in by specifying any password (by convention, an email address for the user should be used as the password).

The server performs a chroot(2) to the directory specified in the ftpd.conf(5) chroot directive (if set), the -a <*anondir*> directory (if set), or to the home directory of the ftp user.

The server then performs a chdir(2) to the directory specified in the ftpd.conf(5) homedir directive (if set); otherwise to /.

If other restrictions are required (such as disabling of certain commands and the setting of a specific umask), appropriate entries in ftpd.conf(5) are required.

If the first character of the password supplied by an anonymous user is -, the verbose messages displayed at login and on a CWD command are suppressed.

Associated files:

/etc/ftpusers	List of unwelcome/restricted users.
/etc/ftpchroot	List of normal users who should be chrooted.
/etc/ftpd.conf	Configures file conversions and other settings.
/etc/ftpwelcome	Welcome notice before login.
/etc/motd	Welcome notice after login.
/etc/nologin	If file exists, it's displayed and access is refused.
/var/run/ftpd.pids-<*CLASS*>	State file of logged-in processes for the ftpd class <*CLASS*>.
/var/run/utmp	List of logged-in users on the system.
/var/run/wtmp	Login history database.

gdb

gdb	GNU debugger.

```
gdb [-help] [-nx] [-q] [-batch] [-cd=<dir>] [-f] [-b <bps>] [-tty=<dev>] [-s <symfile>]
[-e <prog>] [-se <prog>] [-c <core>] [-x <cmds>] [-d <dir>] [<prog> [<core> | <procID>]
```

gdb can be used to debug programs written in C, C++, and Modula-2.

Arguments other than options specify an executable file and a core file or process ID. The first argument encountered with no associated option flag is equivalent to the -se option; the second, if any, is equivalent to the -c option, if it's a file. Options and command-line arguments are processed in sequential order. The order makes a difference when the -x option is specified.

-help -h	Lists all options with brief explanations.
-symbols=<*file*> -s <*file*>	Reads symbol table from file <*file*>.
-write	Enables writing into executable and core files.
-exec=<*file*> -e <*file*>	Uses <*file*> as the executable file to execute when appropriate, and for examining pure data in conjunction with a core dump.
-se=<*file*>	Reads symbol table from <*file*> and uses it as the executable file.
-core=<*file*> -c <*file*>	Uses <*file*> as a core dump to examine.
-command=<*file*> -x <*file*>	Executes gdb commands from <*file*>.
-directory=<*directory*> -d <*directory*>	Adds <*directory*> to the path to search for source files.

`-nx`	
`-n`	Doesn't execute commands from any `.gdbinit` files. Normally, commands in these files are executed after all the command options and arguments have been processed.
`-quiet`	
`-q`	Quiet mode. Doesn't print the introductory and copyright messages. Also suppresses them in batch mode.
`-batch`	Batch mode. Exits with status `0` after processing all the command files associated with the `-x` option (and `.gdbinit`, if not inhibited). Exits with nonzero status if an error occurs in executing the `gdb` commands in the command files.
`-cd=<directory>`	Runs `gdb` using `<directory>` as the working directory rather than using the current directory as the working directory.
`-fullname`	
`-f`	Outputs information used by `emacs-gdb` interface.
`-b <bps>`	Sets the line speed (baud rate or bits per second) of any serial interface used by `gdb` for remote debugging.
`-tty=<device>`	Runs using `<device>` for your program's standard input and output.
These are some of the more frequently needed `gdb` commands:	
`break [<file>]<function>`	Sets a breakpoint at `<function>` (in `<file>`).
`run [<arglist>]`	Starts your program (with `<arglist>`, if specified).
`bt`	Backtrace. Displays the program stack.
`print <expr>`	Displays the value of an expression.
`c`	Continues running your program (after stopping, such as at a breakpoint).
`next`	Executes the next program line (after stopping); steps over any function calls in the line.
`step`	Executes the next program line (after stopping); steps into any function calls in the line.
`help [<name>]`	Shows information about `gdb` command `<name>`, or general information about using `gdb`.
`quit`	Exits `gdb`.

gnumake, make

`gnumake`	GNU `make` utility to maintain groups of programs.

`gnumake [-f <makefile.] [<option>] ... [<target>] ...`

For more details, see the `make.info` file.

In Mac OS X, `/usr/bin/make` is a symbolic link to `/usr/bin/gnumake`.

The `make` utility determines automatically which pieces of a large program need to be recompiled, and issues the commands to recompile them. The `make` utility isn't limited to programs. It can be used to describe any task where some files must be updated automatically from others whenever the others change.

To prepare to use `make`, you must write a file called the `makefile` that describes the relationships among files in your program, and then states the commands for updating each file. In a program, typically the executable file is updated from object files, which are in turn made by compiling source files. After a suitable `makefile` exists, each time you change source files, this simple shell command:

```
make
```

performs all necessary recompilations. The `gnumake` program uses the makefile database and the last-modification times of files to decide which files need to be updated. For each of those files, it issues the commands recorded in the database.

`make` executes commands in the makefile to update one or more targets, where the target is typically a program. If `-f` isn't present, `make` looks for the makefiles `GNUmakefile`, `makefile`, and `Makefile`, in that order. Normally, you should call your makefile either `makefile` or `Makefile`. Note that on Mac OS X, `makefile` and `Makefile` are identical because of the case-insensitive HFS+ file system. We recommend using `Makefile`. `gnumake` isn't recommended because it would not be understood by other versions of `make`. If `makefile` is `-`, the standard input is read.

`make` updates a target if it depends on prerequisite files that have been modified since the target was last modified, or if the target does not exist.

`-b`
`-m` The options are ignored for compatibility with other versions of make.

`-C <`*`dir`*`>` Changes to directory <*dir*> before reading the makefiles or doing anything else. If multiple `C` options are specified, each is interpreted relative to the previous one. This is typically used with recursive invocations of `make`.

`-d` Prints debugging information in addition to normal processing.

`-e` Gives variables taken from the environment precedence over variables from makefiles.

`-f <`*`file`*`>` Uses <*file*> as the makefile.

`-i` Ignores all errors in commands executed to remake files.

`-I <`*`dir`*`>`
`-I<`*`dir`*`>` Specifies a directory <*dir*> to search for included makefiles. If several `-I` options are used to specify several directories, the directories are searched in the order specified. Unlike the arguments to other flags of make, the directories given with `-I` flags may come directly after the flag: `-I<`*`dir`*`>` is allowed, as well as `-I <`*`dir`*`>`. This syntax is allowed for compatibility with the C preprocessor's `-I` flag.

`-j <`*`jobs`*`>` Specifies the number of jobs (commands) to run simultaneously. If there's more than one `-j` option, the last one is effective. If the `-j` option is given without an argument, `make` doesn't limit the number of jobs that can run simultaneously.

`-k` Continues as much as possible after an error. Although the target that failed and those that depend on it can't be made, the other dependencies of these targets can be processed all the same.

`-l`	
`-l <load>`	Specifies that no new jobs (commands) should be started if there are other jobs running and the load average is at least `<load>` (a floating-point number). With no argument, removes a previous load limit.
`-n`	Prints commands that would be executed, but doesn't execute them.
`-o <file>`	Doesn't remake the file `<file>` even if it's older than its dependencies, and doesn't remake anything on account of changes in `<file>`. Essentially the file is treated as very old and its rules are ignored.
`-p`	Prints the database (rules and variable values) that results from reading the makefiles, and then executes as usual or as otherwise specified. This also prints the version information by the `-v` switch. To print the database without trying to remake any files, use make `-p -f/dev/null`.
`-q`	Question mode. Doesn't run any commands, or print anything. Just returns an exit status that's zero if the specified targets are already up to date, nonzero, or otherwise.
`-r`	Eliminates use of the built-in implicit rules. Also clears out the default list of suffixes for suffix rules.
`-s`	Silent operation. Doesn't print the commands as they're executed.
`-S`	Cancels the effect of the `-k` option. This is never necessary except in a recursive `make` when `-k` might be inherited from the top-level `make` via `MAKEFLAGS`, or if you set `-k` in `MAKEFILES` in your environment.
`-t`	Touches files (marks them up-to-date without really changing them) instead of running their commands.
`-v`	Prints the version of the `make` program plus a copyright, list of authors, and notice that there's no warranty.
`-w`	Prints a message containing the working directory before and after other processing. This might be useful for tracking down errors from complicated nests of recursive make commands.
`-W <file>`	Pretends that the target has just been modified. When used with `-n`, this shows you what would happen if you were to modify the file. Without `-n`, it's almost the same as running touch on the given file before running `make`, except that the modification time is changed only in the imagination of make.

grep

`grep` Prints lines matching a pattern.

`egrep`

`fgrep`

`grep [options] <pattern> <file1> <file2> ...`

`grep [options] [-e <pattern> | -f <file>] <file1> <file2>`

`grep` searches the list of files enumerated by `<file1> <file2>` ..., or standard input if no file is specified or if `-` is specified. By default, the matching lines are printed.

Two additional variants of the program are available as `egrep` (same as `grep -E`) or `fgrep` (same as `grep -F`).

`-A <num>` Prints `<num>` lines of trailing context after matching lines.

`--after-context=<num>` Same as `-A <num>`.

`-a` Processes a binary file as if it were a text file. Equivalent to `-binary-files=text` option.

`--text` Same as `-a`.

`-B <num>` Prints `<num>` lines of leading context before matching lines.

`--before-context=<num>` Same as `-B <num>`.

`-C <num>` Prints `<num>` lines of output context. Default is `2`.

`-<num>` Same as `-C <num>`.

`--context[=<num>]` Same as `-C <num>`.

`-b` Prints the byte offset within the input file before each line of output.

`--byte-offset` Same as `-b`.

`--binary-files=<type>` Assumes a file is type `<type>` if the first few bytes of a file contain binary data.

Default `<type>` is `binary`, and `grep` normally outputs a one-line message indicating the file is binary, or nothing if there's no match. If `<type>` is `without-match`, it's assumed that a binary file doesn't match. Equivalent to `-I` option. If `<type>` is `text`, it processes the file as though it were a text file. Equivalent to `-a` option. Warning: Using this option could result in binary garbage being output to a terminal, some of which could be interpreted by the terminal as commands, resulting in unwanted side effects.

`-I` Assumes that a binary file doesn't match. Equivalent to `-binary-files=without-match` option.

`-c` Prints a count of matching lines for each file. Combined with `-v`, counts nonmatching lines.

`--count` Same as `-c`.

`-v` Inverts matching to select nonmatching lines.

`--invert-match` Same as `-v`.

`-d <action>` If input file is a directory, uses `<action>` to process it. If `<action>` is `read`, `grep` reads directories as if they were normal files. This is the default. If `<action>` is `skip`, `grep` silently skips directories. If `<action>` is `recurse`, `grep` recursively reads files under the directory. Equivalent to `-r`.

`--directories=<action>`	Same as `-d <action>`.
`-r`	Recursively reads files under directories. Equivalent to `-d recurse` option.
`--recursive`	Same as `-r`.
`-f <file>`	Reads a list of patterns from `<file>`, which contains one pattern per line. An empty file has no patterns and matches nothing.
`--file=<file>`	Same as `-f <file>`.
`-e <pattern>`	Uses `<pattern>` as the pattern. Useful for protecting patterns beginning with `-`.
`-regexp=<pattern>`	Same as `-e <pattern>`.
`-G`	Interprets `<pattern>` as a basic regular expression. This is the default behavior.
`--basic-regexp`	Same as `-G`.
`-E`	Interprets `<pattern>` as an extended regular expression. Equivalent to `egrep`.
`-extended-regexp`	Same as `-E`.
`-F`	Interprets `<pattern>` as a list of fixed strings, separated by newlines, any of which is to be matched. Equivalent to `fgrep`.
`--fixed-strings`	Same as `-F`.
`-H`	Prints the filename for each match.
`--with-filename`	Same as `-H`.
`-h`	Suppresses filenames on output when multiple files are searched.
`--no-filename`	Same as `-h`.
`--help`	Displays a brief help message.
`-i`	Ignores case in `<pattern>` and input files.
`--ignore-case`	Same as `-i`.
`-L`	Prints a list of files that don’t have matches. Stops scanning after the first match.
`-l`	Prints a list of files that contain matches.
`--mmap`	If possible, uses `mmap(2)` system call rather than the default `read(2)` system call. Sometimes `-mmap` results in better performance. However, it can cause unexpected behavior, such as core dumps, if the file shrinks while `grep` is reading it or if an I/O error occurs.
`-n`	Output includes the line number where the match occurs.
`--line-number`	Same as `-n`.
`-q`	Quiet. Suppresses normal output. Scanning stops on the first match. Also see the `-s` and `-no-messages` options.
`--quiet`	Same as `-q`.
`--silent`	Same as `-q`.
`-s`	Suppresses error messages about nonexistent or unreadable files.
`--no-messages`	Same as `-s`.
`-V`	Prints the version number of `grep` to standard error. Includes the version number in all bug reports.

`--version`	Same as `-V`.
`-w`	Selects only lines that have matches that form whole words.
`--word-regexp`	Same as `-w`.
`-x`	Selects only those matches that exactly match the whole line.
`--line-regexp`	Same as `-x`.
`-Z`	Outputs a zero byte (the ASCII NULL character) instead of the character that normally follows a filename. This option makes the output unambiguous, even for filenames containing unusual characters such as newlines.
`--null`	Same as `-Z`.
`-y`	Obsolete equivalent for `-i`.
`-U`	Has no effect on platforms other than MS-DOS and MS Windows. On those platforms, `-U` treats files as binary files to affect how CR characters are handled.
`--binary`	Same as `-U`.
`-u`	Has no effect on platforms other than MS-DOS and MS Windows. On those platforms, reports Unix-style byte offsets; that is, with CR characters stripped off.
`--unix-byte-offsets`	Same as `-u`.

gzip, gunzip, zcat

`gzip`	Compresses or expands files.
`gunzip`	
`zcat`	

```
gzip [-acdfhlLnNrtvV19] [-S <suffix>] <file1> <file2> ...
gunzip [-acfhlLnNrtvV] [-S <suffix>] <file1> <file2> ...
zcat [-fhLV] <file1> <file2> ...
```

`gzip` reduces the size of a file and renames the file by adding the `.gz` extension. It keeps the same ownership modes, and access and modification times. If no files are specified, or if the filename `-` is specified, standard input is compressed to standard output. `gzip` compresses regular files, but ignores symbolic links.

Compressed files can be restored to their original form by using `gunzip`, `gzip -d`, or `zcat`.

`gunzip` takes a list of files from the command line, whose names end in `.gz`, `-gz`, `.z`, `-z`, `_z`, or `.Z`, and which also begin with the correct magic number, and replaces them with expanded files without the original extension. `gunzip` also recognizes the extensions `.tgz` and `.taz` as short versions of `.tar.gz` and `.tar.Z`, respectively. If necessary, `gzip` uses the `.tgz` extension to compress a `.tar` file.

`zcat` is equivalent to `gunzip -c`. It uncompresses either a list of files on the command line or from standard input and writes uncompressed data to standard output. `zcat` uncompresses files that have the right magic number, whether or not they end in `.gz`.

Compression is always formed, even if the compressed file is slightly larger than the original file.

`-a`	ASCII text mode. Converts end-of-lines using local conventions. Supported only on some non-Unix systems.
`--ascii`	Same as `-a`.
`-c`	Writes output to standard output and keeps the original files unchanged.
`--stdout`	Same as `-c`.
`--to-stdout`	Same as `-c`.
`-d`	Decompresses.
`--decompress`	Same as `-d`.
`--uncompress`	Same as `-d`.
`-f`	Forces compression or decompression, even if the file has multiple links, or if the corresponding file already exists, or if the compressed data is read from or written to a terminal. If `-f` isn't used, and `gzip` isn't working in the background, the user is prompted before a file is overwritten.
`-h`	Displays a help screen and quits.
`--help`	Same as `-h`.
`-l`	Lists the following fields for each compressed file: `compressed` (compressed size) `uncompressed` (uncompressed size) `ratio` (compression ratio; `0.0%` if unknown) `uncompressed_name` (name of uncompressed file) Uncompressed size is `-1` for files not in `gzip` format. To get an uncompressed size for such files, use `zcat <`*`file1.Z`*`> | wc -c` Combined with `-verbose`, it also displays: `method` (compression method) `crc` (32-bit CRC of the uncompressed data) `date and time` (time stamp of the uncompressed file) Compression methods supported are `deflate`, `compress`, `lzh`, and `pack`. `crc` is listed as `ffffffff` when the file isn't in `gzip` format.
`--list`	Same as `-l`.
`-L`	Displays the `gzip` license and quits.
`--license`	Same as `-L`.
`-n`	When compressing, it doesn't save the original filename and time stamp by default. (Always saves the original name if it has to be truncated.) When decompressing, it doesn't restore the original name (removes only `.gz`) and time stamp (only copies it from compressed file), if present. This is the default.
`--no-name`	Same as `-n`.

`-N`	When compressing, it always saves the original filename and time stamp. This is the default. When decompressing, it restores the original time stamp and filename, if present.
`--name`	Same as `-N`.
`-q`	Suppresses all warnings.
`--quiet`	Same as `-q`.
`-r`	Traverses the directory structure recursively. If a filename specified on the command line is a directory, `gzip`/`gunzip` descends into the directory and compresses/decompresses the files in that directory.
`--recursive`	Same as `-r`.
`-S <`*`suffix`*`>`	Uses <*suffix*> instead of `.gz`. Any suffix can be used, but we recommend that suffixes other than `.z` and `.gz` be avoided to avoid confusion when transferring the file to another system. A null suffix (`-S ""`) forces `gunzip` to try decompression on all listed files, regardless of suffix.
`--suffix <`*`suffix`*`>`	Same as `-S <`*`suffix`*`>`.
`-t`	Test. Checks the integrity of the compressed file.
`--test`	Same as `-t`.
`-v`	Verbose. Displays the name and percentage reduction for each file compressed or decompressed.
`--verbose`	Same as `-v`.
`-V`	Version. Displays the version number and compilation options and quits.
`--version`	Same as `-V`.
`-<`*`n`*`>`	
`--fast`	
`--best`	Regulates the speed of compression as specified by `-<`*`n`*`>`, where `-1` (or `--fast`) is the fastest compression method (least compression) and `-9` (or `--best`) is the slowest compression method (most compression). Default compression option is `-6`.

halt, reboot

`halt`	Stops the system.
`reboot`	Restarts the system.

`halt [-nqd]`

`reboot [-nqd]`

The `halt` and `reboot` utilities flush the system cache to disk, send all running processes a `SIGTERM` and subsequently a `SIGKILL` and, respectively, halts or restarts the system. The action is logged, including adding a shutdown record into the login accounting file.

`-n` Doesn't flush the file system cache. This option probably shouldn't be used.

`-q` Quickly and ungracefully halts/restarts the system, and only flushes the file system cache. This option probably shouldn't be used.

`-d` Creates a dump before rebooting. This option is useful for debugging system dump procedures or capturing the state of a corrupted or misbehaving system.

Normally, `shutdown` (8) is used when the system needs to be halted or restarted to warn users of their impending doom.

hdiutil

`hdiutil` Manipulates disk images.

`hdiutil <verb> [<options>]`

`hdiutil` uses the DiskImages framework to manipulate disk image files. Common verbs include `attach`, `detach`, `verify`, `create`, `convert`, and `burn`.

The rest of the verbs are `help`, `info`, `load`, `checksum`, `eject` (historical synonym for `detach`), `flatten`, `unflatten`, `imageinfo`, `mount` (historical synonym for `attach`), `mountvol`, `unmount`, `plugins`, `resize`, `segment`, and `pmap`.

Options

All `hdiutil` verbs accept these options:

`-verbose` Verbose; default is less output. This option is useful if it's unclear why a particular operation failed. At a minimum, the probing for each image type of any given files will be shown.

`-quiet` Minimizes output in most cases.

`-debug` Very verbose. This option is useful if a large amount of information about what `hdiutil` and the DiskImages framework are doing is needed.

Many `hdiutil` verbs understand the following options:

`-plist` Displays output in `plist` format.

`-srcimagekey <key>=<value>` Specifies a key/value pair for the disk image recognizer. (`-imagekey` is normally a synonym.)

`-tgtimagekey <key>=<value>` Specifies a key/value pair for any image created. (`-imagekey` is only a synonym if there's no input image.)

`-encryption [<crypto_method>]` Specifies a particular type of encryption or, if not specified, the default `CEncrypedEncoding`.

`-passphrase <`*`passphrase`*`>`
Provides a passphrase for an encrypted image. `-passphrase` is very insecure because the passphrase is visible to who can run `ps`(1).

`-shadow [<`*`shadowfile`*`>]`
Uses a shadow file in conjunction with the data in the image. This option prevents modification of the original image and allows read-only images to be used as read/write images. When blocks are being read from the image, blocks present in the shadow file override blocks in the base image. When blocks are being written, the writes will be redirected to the shadow file. If not specified, `-shadow` defaults to `<`*`imagename`*`>.shadow`. If the shadow file doesn't exist, it's created.

For the verbs that create images, it should be noted that the correct extension is added to the filenames if the extension isn't present. The creation engine also examines the filename extension of the provided filename and changes its behavior accordingly. For example, a sparse image can be created without specifying `-type SPARSE` simply by appending the .sparseimage extension to the provided filename.

Verbs

Each verb is listed with its description and individual arguments. Arguments to the verbs can be passed in almost any order. A sector is 512 bytes.

`help`
Displays the usage information for each verb.

`attach <`*`imagename`*`> [<`*`options`*`>]`
`attach` calls `hdid` with its arguments. See `hdid`(1) for options that you can pass to `attach`. `attach`, like `hdid`, will return information about an already-attached image as if it had attached it. `mount` is a synonym for `attach`.

`detach <`*`dev_name`*`> [-force]`
`detach` a disk image and terminate any associated `hdid` process. `dev_name` is a partial `/dev` node path (for example, "disk1"). If Disk Arbitration is running, `detach` uses it to unmount any partitions and detach the image. If not, the partition must be manually unmounted (by the user who mounted them or by root—if Disk Arbitration made the mount, only `root` can do a clean `umount`) using `umount`(8). Only then is

hdiutil detach able to detach the image. eject is a synonym for detach.

-force Similar to umount -f. Unmounts any file systems and detaches the image file, regardless of any open files on the image.

verify <imagename> [<options>]

Computes the checksum of a read-only (or compressed) image file, and verify it against the value stored in the image.

verify accepts the common options:

-encryption, -srcimagekey, -tgtimagekey, -passphrase, and -shadow.

create <imagename> <size_spec> [<options>]

Creates a new blank image. If imagename already exists, -ov must be specified or create will fail.

Size specifiers:

-size <??b|??k|??m|??g|??t??p|??e>

-size specifies the size of the image in the style of mkfile(8) with the addition of terabyte, petabyte, and exabyte sizes. The larger sizes are occasionally useful when creating large sparse images.

-sectors <sector_count>

Specifies the size of the image file in 512-byte sectors. Note that this quantity includes the space that might be used for a partition map or other utility partitions. The overhead for a SPUD (the default single-partition layout) is 64 sectors. The overhead for layout NONE is 0 sectors.

-megabytes <size>

Specifies the size of the image file in megabytes (1024*1024 bytes). Note that this quantity includes the space that might be used for a partition map or other utility partitions.

Common options: -plist, -tgtimagekey, -encryption, and -passphrase. -imagekey di-sparse-puma-compatible=TRUE and -imagekey di-shadow-puma-compatible=TRUE create, respectively, sparse and shadow image files that can be attached on Mac OS X 10.1.

Other options:

-align <alignment>

Specifies a size to which the final data partition will be aligned. The default is 4KB.

-type <UDIF|SPARSE>

UDIF is the default disk image format. If specified, a UDRW of the specified size is created. SPARSE creates a special type of image that grows with its use. The default is to grow one megabyte at a time, but the -imagekey option sparse-band-size can be used to specify the number of sectors that are added each time the image file is grown.

-fs <filesystem>

<filestyem> is one of HFS+, HFS, MS-DOS, or UFS. -fs causes the image to be attached, formatted with the specified file system and then detached. -fs may also change the default layout if that particular file system doesn't natively come in an Apple Partition Map.

-volname volname

Specifies the volume name (default is "untitled") of the newly created file system.

-stretch <max_stretch>

-stretch initializes HFS+ file system data such that it can later be stretched using hdiutil resize. max_stretch is specified like -size.

-fsargs newfs_options

Additional options to pass to whatever newfs program is being called.

-layout <*layout*>

Specifies the partition map layout of the image. layout can be anything specified in MediaKit.framework's MKDrivers.bundle. NONE creates an image file with no partition map. When such an image is attached, a single /dev entry is created (for example, /dev/disk1). SPUD specifies a Single Partition UDIF. This creates an image file with a DDM and an Apple partition scheme partition map, with a single entry for an Apple HFS partition. When attached, multiple /dev entries are created and the second partition is the data partition (for example, /dev/disk1, /dev/disk1s1, /dev/disk1s2; the second partition is disk1s2). Unless changed by -fs, the default is SPUD. Other layouts include UNIVERSAL HD and UNIVERSAL CD, which add appropriate Mac OS 9 driver partitions for those types of media.

-partitionType <*partition_type*>

Changes the single partition type in a SPUD. The default is Apple_HFS. The principal alternative is Apple_UFS, although the appropriate partition map is generated depending on what's passed to -fs.

-ov

Overwrites an existing file. The default isn't to overwrite existing files.

convert <*imagefile*> -format <*format*> -o <*outfile*> [<*options*>] Converts <*imagefile*> to type <*format*> and writes the result to <*outfile*>.

The correct filename extension is added only if it isn't part of the provided name. <*format*> is one of the following:

UDRW UDIF read/write image

UFBI UDIF entire image with MD5 checksum

UDRO UDIF read/only image

UDCO UDIF ADC-compressed image

UDRo UDIF read/only (obsolete format)

UDCo UDIF compressed (obsolete format)

UDTO DVD/CD-R export image

UDxx UDIF stub image

UDZO UDIF zlib-compressed image

RdWr NDIF read/write image (deprecated)

Rdxx NDIF read/only image (deprecated, but still usable on Mac OS 9 and Mac OS X)

ROCo NDIF compressed image (deprecated)

Rken NDIF compressed (obsolete format)

DC42 Disk Copy 4.2 image

In addition to the compression offered by some formats, the UDIF and NDIF non-read/write image formats completely remove unused space in HFS and UFS file systems. For UDZO, -imagekey zlib-level=<*value*> allows you to set the zlib compression level a la gzip(1). The default compression level is 1 (fastest).

<*options*> are any of these:

Common options: -shadow, -srcimagekey, -tgtimagekey, -encryption, and -passphrase.

Other options:

`-align <sector_alignment>`

Default is 4 (2KB).

`-pmap`

Add partition map. When converting a NDIF to a any variety of UDIF, or when converting a partitionless UDIF to UDIF, the default is true.

`-segmentSize [sector_count]`

Specify segmentation of `<imagename>` into `<sector_count>`-sized segments. The default `<sector_count>` when `-segmentSize` is specified is 2*1024*1024 (1GB segments) for UDTO images and 4*1024*1024 (2GB segments) for all other image types. `<sector_count>` can also be specified `<??b|??k|??m|??g|??t??p|??e>` like `mkfile`(8).

`-tasks task_count`

When converting an image into a compressed format, specifies the number of threads to use for the compression operation. The default is the number of processors active in the current system.

`burn <imagename> [<options>]`	Burns `<imagename>` to optical media in an attached drive. In all cases, a prompt for media is printed once an appropriate drive has been found.

`<options>` are any of the following:

Common options: `-shadow`, `-srcimagekey`, `-encryption`, and `-passphrase`.

Other options:

`-testburn`

Doesn’t turn on laser (laser defaults to on).

`-noeject`

Doesn’t eject disc after burning.

`-eject`

Ejects disc after burning (default).

`-verifyburn`

Verifies disc contents after burn (default).

`-noverifyburn`

Doesn’t verify disc contents after burn.

`-addpmap`

Adds partition map if necessary. Some filesystem types aren’t recognized when stored on optical media unless they’re enclosed in a partition map. This option adds a partition map to any bare file system that needs a partition map in order to be recognized when burned to optical media. This is the default behavior.

`-noaddpmap`

Doesn’t add partition map.

`-skipfinalfree`

Skips final free partition. If there’s a partition map on the image specifying an Apple Free partition as the last partition, that Apple Free partition is not burned. The burned partition map still references the empty space. This is the default behavior.

`-noskipfinalfree`

Doesn't skip any trailing Apple Free partition.

`-optimizeimage`

Optimizes file system for burning. Optimization can reduce the size of an HFS or HFS+ volume to the size of the data contained on the volume. This option changes what's burned such that the disc has a different checksum than the image it came from.

`-nooptimizeimage`

Doesn't optimize. Burns all blocks of the image (minus any blocks in trailing Apple Free partitions unless `-noskipfinalfree` is specified). This is the default behavior.

`-forceclose`

Forces the disc to be closed after burning. Further burns to the disc are impossible.

`-noforceclose`

Doesn't force disc to be closed (default)

`-speed <`*`x_factor`*`>`

`<`*`x_factor`*`>` may be `1`, `2`, `4`, `6`, …, `max`.

Specifies the desired x-factor. For example, 8 means that the drive burns at 8x speed. `max` causes the burn to proceed at the maximum speed of the drive. `max` is the default speed.

`-sizequery`

Only calculates the size of disc required (the size returned is in sectors).

`-erase`

Prompts for optical media (DVD-RW/CD-RW) and then, if the hardware supports it, quickly erases the media.

`-fullerase`

Erases all sectors of the disc (this usually takes quit a bit longer than -erase).

`info`

Displays information about the disk image driver and any image files that are currently attached. `hdiutil` info accepts `-plist`.

`load`

Manually loads the disk image driver. The disk image driver is loaded automatically by the Disk Copy application or `hdid`(8) if an image file is being attached and the driver isn't currently loaded. As of Mac OS X 10.2, the driver automatically detaches itself after use and then is unloaded after a minute or so, making `hdiutil` load something of a no-op.

`checksum <`*`imagename`*`> [<`*`options`*`>] -type <`*`type`*`>`

Calculates the specified checksum on the image data, regardless of image source or type. Common options: `-plist`, `-shadow`, `-srcimagekey`, `-encryption`, and `-passphrase`.

`<type>` is one of UDIF-CRC32—CRC-32 image checksum UDIF-MD5—MD5 image checksum DC42—Disk Copy 4.2 CRC28—CRC-32 (NDIF) CRC32—CRC-32 MD5—MD5	
`unflatten <imagename>`	Unflattens a read-only (or compressed) UDIF disk image, creating a dual-fork file in traditional format (resource-only; no XML). Common options: `-srcimagekey`, `-encryption`, and `-passphrase`.
`flatten <imagename>`	Flattens a read-only (or compressed) UDIF disk image into a single-fork file. If the image is UDZO and doesn't contain XML meta-data for in-kernel attachment, adds it. Common options are `-srcimagekey`, `-encryption`, and `-passphrase`.

`flatten` is only required if the UDIF has previously been unflattened.
Other options:
`-noxml`
Doesn't embed XML data for in-kernel attachment. The image never attaches in-kernel.
`-norsrcfork`
Doesn't embed resource fork data. The image doesn't attach on Mac OS X versions prior to Mac OS X 10.2.	

`hfsanalyze <imagename>`	Prints information about an HFS/HFS+ volume. As is often the case, `<imagename>` can be a `/dev` entry.
Common options are `-shadow`, `-srcimagekey`, `-encryption`, and `-passphrase`.	
`mountvol <devnode>`	Attempts to mount the given `/dev` node through Disk Arbitration (similar to `disktool -m`). XML output is available from `-plist`. Note that `mountvol` and `unmount` are a pair. `mount`/`attach` can be called on a `/dev` entry, but it treats the `/dev` entry as a disk image file to be attached (through another `/dev` entry). This is usually undesirable.

`unmount <`*`volume`*`>`

Unmounts a mounted volume. `<`*`volume`*`>` can be a full path to a `/dev` entry or the name of a mountpoint.

`imageinfo <`*`imagename`*`> [<`*`options`*`>]`

Prints out information about a disk image.

Common options are `-plist`, `-srcimagekey`, `-encryption`, and `-passphrase`.

Other options are

`-format`

Only prints out the image format.

`-checksum`

Only prints out the image checksum.

`plugins`

Prints information about DiskImages framework plug-ins. The user, system, local, and network domains are searched for plug-ins (that is, `~/Library/Plug-ins/DiskImages`, `/System/Library/Plug-ins/DiskImages`, `/Library/Plug-ins/DiskImages`, `/Network/Library/Plug-ins/DiskImages`). `-plist` is available.

`resize <`*`size_spec`*`> <`*`imagename`*`> [<`*`options`*`>]`

For a read/write partitioned UDIF device image, if the last partition is Apple HFS (either HFS or HFS+), attempts to resize the partition to the end of the device file, or to the last used block in the embedded HFS/HFS+ file system. This is typically used when working with a large device image file, when it's desirable to shrink the HFS/HFS+ partition before converting to CD-R/DVD-R format. Images converted to CD-R/DVD-R don't include the Apple_Free partition at the end of the device, so such conversions result in a CD-R/DVD-R master that would only write the actual data.

Common options are `-srcimagekey`, `-encryption`, and `-passphrase`.

Size specifiers:

```
-size <??b|??k|??m|??g|??t??p|??e>
-sectors <sector_count> | min | max
```

Specifies the number of 512-byte sectors to which the partition should be resized. If this falls outside the min/max values, an error is returned and the partition isn't resized. `min` automatically determines the smallest size the partition can be shortened to and uses that value. `max` automatically determines the largest size to which the partition can be grown and then uses that value.

Other options:

`-imageonly`

Only resizes the image file, not the partition(s) inside of it. This is the default for UDIF images.

`-partitiononly`

Only resizes the partition(s) in the image. This is the default for NDIF images.

`-partitionNumber <partitionNumber>`

Specifies which partition to resize (UDIF only). `<partitionNumber>` is 0-based, but, per `hdiutil pmap`, partition 0 is the partition map itself.

`-growonly`

Only allows the image to grow.

`-shrinkonly`

Only allows the image to shrink.

`-nofinalgap`

Allows resize to entirely eliminate the trailing free partition. Such an image won't boot Mac OS 9 nor does it allow Mac OS X to boot on old-world (beige) machines.

`-limits`

Displays the minimum, current, and maximum values for the size of the given volume in 512-byte sectors. Doesn't modify the image file.

`segment` Segments an NDIF or UDIF disk image.

Usage:

`segment -o <firstSegname> -segmentCount <#segs> <imagename> [<options>]`

`segment -o <firstSegname> -segmentSize <size> <imagename> [<options>]`

Common options are

`-srcimagekey`, `-tgtimagekey`,

`-encryption`, and `-passphrase`.

`<options>` include

`-segmentCount <segment_count>`

Specifies the number of segments. Only one of `-segmentCount` or `-segmentSize` is honored.

`-segmentSize <segment_size>`

Specifies the segment size in sectors. If the original image size isn't an exact multiple of the segment size, the last segment will be shorter than the others. Only one of `-segmentCount` or `-segmentSize` is honored.

`-firstSegmentSize <segment_size>`

Specifies the first segment size in sectors. Used for multi-CD restores.

`-restricted`

Makes restricted segments for use in multi-CD restores. This option is ignored for NDIF images.

`pmap <image_source> [<options>]` Displays partition map from image or device. `<image_source>` is either a plain file or special file (that is, a `/dev/disk` entry).

Common options are

`-shadow`, `-srcimagekey`, `-encryption`, and `-passphrase`

`<`*`options`*`>` defaults to `xsSgcvk` and can be any combination of the following:

`r raw`—processes all without modification

`x extended`—processes 2KB & 512 entries and merges

`s sectorize`—returns all quantities in Sectors

`S sort`—sorts all entries By Blockno

`g genfree`—accounts for all unmapped space

`c combfree`—combines adjacent freespace entries

`f fixfinal`—extends last partition to device end

`v volume synthesize`—synthesizes single volumes as a single partition entry

`k skip zero-length`—skips zero length entries

`K skip void/free`—skips all free and void partitions

`m merge free space`—merges small free partitions into a previous partition if possible

`i ignore shims`—ignores small free partitions caused by block alignment

head

`head` Displays the first lines of a file.

`head [-n <`*`number`*`>] <`*`file1`*`> <`*`file2`*`> ...`

`head [-n <`*`number`*`>]`

`-n <number>` Displays the first `<`*`number`*`>` of lines. If *`n`* isn't specified, the default is `10`.

id

`id` Returns user identity.

`id [<`*`user`*`>]`

`id -G [-n] [<`*`user`*`>]`

`id -g [-nr] [<`*`user`*`>]`

`id -u [-nr] [<`*`user`*`>]`

`id -p [<`*`user`*`>]`

The `id` utility displays the user and group names and numeric ID of the calling process to standard output. If the real and effective IDs are different, both are displayed; otherwise, only the real ID is displayed.

If a `<`*`user`*`>` (login name or user ID) is specified, the user and group IDs of that user are displayed. In this case, the real and effective IDs are assumed to be the same.

`-G` Displays the different group IDs (effective, real, and supplementary) as whitespaced numbers in no particular order.

`-g` Displays the effective group ID as a number.

`-u` Displays the effective user ID as a number.

`-n`	Displays the name of the user or group ID for the `-G`, `-g`, and `-u` options instead of the number. If any of the ID numbers cannot be mapped into names, the number will be displayed as usual.
`-r`	Displays the real ID for the `-g` and `-u` options instead of the effective ID.
`-p`	Displays the output in human-readable form. If the username returned by `getlogin`(2) is different from the login name referenced by the user ID, the name returned by `getlogin`(2) is displayed, preceded by the keyword login. The user ID as a name is displayed, preceded by the keyword `uid`. If the effective user ID is different from the real user ID, the real user ID is displayed as a name, preceded by the keyword `euid`. If the effective group ID is different from the real group ID, the real group ID is displayed as a name, preceded by the keyword `rgid`. The list of groups to which the user belongs is then displayed as names, preceded by the keyword `groups`. Each display is on a separate line.

ifconfig

`ifconfig`	Configures network interface parameters.

```
ifconfig [-L] [-m] <interface> [create] <address_family> [<address[/prefixlength]>
[<dest address>]] [<parameters>]
ifconfig <interface> destroy
ifconfig -a [-L] [-d] [-m] [-u] [<address_family>]
ifconfig -l [-d] [-u] [<address_family>]
ifconfig [-L] [-d] [-m] [-u] [-C]
```

`ifconfig` assigns an address to a network interface and/or configures network interface parameters. It must be used at boot time to define the network address of each network interface. It may also be used at a later time to redefine an interface's network address or other operating parameters.

Only the superuser can modify the configuration of a network interface.

`-m`	If passed before an interface name, `ifconfig` displays all the supported media for the specified interface.
`-L`	Displays address lifetime for IPv6 addresses, as time off-set string.
`-a`	Produces a full listing of all available interfaces.
`-l`	Produces a name-only listing of all available interfaces.
`-d`	Limits a listing to those interfaces that are down.
`-u`	Limits a listing to those interfaces that are up.

Available options for `ifconfig` are

`<address>`	For the DARPA-Internet family, the address is either a hostname in the hostname database or a DARPA-Internet address expressed in the Internet standard dot notation.
`<address family>`	Specifies the `<address family>`, which affects interpretation of the remaining parameters. The address or protocol families currently supported are `inet`, `iso`, and `ns`.

`<dest address>` Specifies the address of the correspondent on the other end of a point to point link.

`<interface>` The `<interface>` parameter is a string of the form `<name of physical unit>`, such as `en0`.

The following parameters may be set with `ifconfig`:

`add` Another name for the alias parameter. Introduced for compatibility with BSD/OS.

`alias` Establishes an additional network address for this interface. This is sometimes useful when changing network numbers, while still accepting packets for the old interface. A `<netmask>` should be used with this parameter. If the new `<alias>` address is on the same subnet as an existing address assigned to this interface, the netmask must be `255.255.255.255`. If a netmask isn't supplied, the command uses the one implied by the address itself. If the all ones netmask is used, the system handles route installation. If another is used, a route to that address might have to be added by hand; for example, `route add -host xx.xx.xx.xx -interface 127.0.0.1`, where `xx.xx.xx.xx` is the alias. In either case, the route might have to be deleted by hand when the alias is removed (`-alias` or `delete`).

`-alias` Removes the network address specified.

`anycast` (Inet6 only) Specifies that the address configured is an anycast address. Based on the current specification, only routers may configure anycast addresses. Anycast address won't be used as source address of any of outgoing IPv6 packets.

`arp` Enables the use of the Address Resolution Protocol in mapping between network-level addresses and link-level addresses (default). This is currently implemented for mapping between DARPA-Internet addresses and 10 Mb/s Ethernet addresses.

`-arp` Disables the use of the Address Resolution Protocol.

`broadcast` (Inet only) Specifies the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1s.

`debug` Enables driver-dependent bugging code. This usually turns on extra console logging.

`-debug` Disables driver-dependent debugging code.

`delete` Removes the network address specified. This is used if you incorrectly specified an alias or if it's no longer needed.

`down` Marks an interface as down. When an interface is marked as down, the system doesn't attempt to transmit messages through that interface. If possible, the interface is reset to disable reception as well. This doesn't automatically disable routes using the interface.

`ether` Another name for the `lladdr` parameter.

`lladdr <addr>` Sets the link-level address on an interface. This can be used to, for example, set a new MAC address on an ethernet interface, although the mechanism used isn't ethernet-specific. The address `<addr>` is specified as a series of colon-separated hex digits. If the interface is already up when this option is used, it's briefly brought down and then brought back up again to ensure that the receive filter in the underlying ethernet hardware is operating.

`media <type>` If the driver supports the media selection system, set the media type of the interface to *`type`*. Some interfaces support the mutually exclusive use of one of several different physical media connectors. For example, a 10Mb/s Ethernet interface might support the use of either AUI or twisted pair connectors. Setting the media type to `10base5/AUI` would change the currently active connector to the AUI port. Setting it to `10baseT/UTP` would activate twisted pair. Refer to the interfaces' driver-specific documentation or man page for a complete list of the available types.

`mediaopt <opts>` If the driver supports the media selection system, sets the specified media options on the interface. The `<opts>` argument is a comma-delimited list of options to apply to the interface. Refer to the interfaces' driver specific man page for a complete list of available options.

`-mediaopt <opts>` If the driver supports the media selection system, disables the specified media options on the interface.

`tunnel <src-addr> <dest-addr>` (IP tunnel devices only) Configure the physical source and destination address for IP tunnel interfaces (gif(4)). The arguments `<src_addr>` and `<dest_addr>` are interpreted as the outer source/destination for the encapsulating IPv4/IPv6 header.

`deletetunnel` Unconfigures the physical source and destination address for IP tunnel interfaces previously configured with `tunnel`.

`create` Create the specified network pseudo-device. If the interface is given without a unit number, try to create a new device with an arbitrary unit number. If creation of an arbitrary device is successful, the new device name is printed to standard output.

`destroy` Destroys the specified network pseudo-device.

`plumb` Another name for the `create` parameter. Included for Solaris compatibility.

`unplumb` Another name for the `destroy` parameter. Included for Solaris compatibility.

`metric <n>` Sets the routing metric of the interface to `<n>`; the default is `0`. The routing metric is used by the routing protocol. Higher metrics make a less favorable route. Metrics are counted as addition hops to the destination network or host.

`mtu <n>`	Sets the maximum transmission unit of the interface to `n`, default is interface-specific. The MTU is used to limit the size of packets that are transmitted on an interface. Not all interfaces support setting the MTU, and some interfaces have range restrictions.
`netmask <mask>`	(inet and ISO) Specifies how much of the address to reserve for subdividing networks into subnetworks. The mask includes the network part of the local address and the subnet part, which is taken from the host field of the address. The mask can be specified as a single hexadecimal number beginning with `0x`, as a dot-notation Internet address, or as a pseudo-network name listed in the network table networks. The mask contains 1s for the bit positions in the 32-bit address that are to be used for the network and subnet parts, and 0s for the host part.
`prefixlen <len>`	(Inet6 only) Specifies that *`len`* bits are reserved for subdividing networks into subnetworks. The `<len>` must be integer, and for syntactical reasons, it must be between `0` to `128`. It's almost always `64` under the current IPv6 assignment rule. If the parameter is omitted, `64` is used.
`remove`	Another name for the `-alias` parameter. Introduced for compatibility with BSD/OS.
`link[0-2]`	Enables special processing of the link level of the interface.
`-link[0-2]`	Disables special processing at the link level with the specified interface.
`up`	Marks an interface as `up`. Can be used to enable an interface after `ifconfig down` has been run. It happens automatically when setting the first address on an interface. If the interface was reset when previously marked down, the hardware is reinitialized.

ipfw

`ipfw` Controlling utility for IP firewall.

```
ipfw [-q] [-p <preproc> [-D macro[=value]] [-U macro]] pathname
     ipfw [-f | -q] flush
     ipfw [-q] {zero | resetlog | delete} [number ...]
     ipfw [-s [field]] [-aftN] {list | show} [number ...]
     ipfw [-q] add [number] rule-body
     ipfw pipe number config pipe-config-options
     ipfw pipe {delete | list | show} [number ...]
     ipfw queue number config queue-config-options
     ipfw queue {delete | list | show} [number ...]
```

`ipfw` is the user interface for controlling the `ipfirewall`(4) and the `dummynet`(4) traffic shaper in FreeBSD.

Each incoming or outgoing packet is passed through the `ipfw` rules. If host is acting as a gateway, packets forwarded by the gateway are processed by `ipfw` twice. In case a host is acting as a bridge, packets forwarded by the bridge are processed by `ipfw` once.

A firewall configuration is made of a list of numbered rules, which is scanned for each packet until a match is found and the relevant action is performed. Depending on the action and certain system settings, packets can be reinjected into the firewall at the rule after the matching one for further processing. All rules apply to all interfaces, so it's the responsibility of the system administrator to write the ruleset in such a way as to minimize the number of checks.

A configuration always includes a `DEFAULT` rule (numbered 65535) which cannot be modified by the programmer and always matches packets. The action associated with the default rule can be either deny or allow depending on how the kernel is configured.

If the ruleset includes one or more rules with the keep-state option, `ipfw` assumes a stateful behavior, that is, upon a match will create dynamic rules matching the exact parameters (addresses and ports) of the matching packet.

These dynamic rules, which have a limited lifetime, are checked at the first occurrence of a check-state or keep-state rule, and are typically used to open the firewall on-demand to legitimate traffic only.

All rules (including dynamic ones) have a few associated counters: a packet count, a byte count, a log count, and a timestamp indicating the time of the last match. Counters can be displayed or reset with `ipfw` commands.

Rules can be added with the `add` command; deleted individually with the `delete` command, and flushed globally with the `flush` command; displayed, optionally with the content of the counters, using the `show` and `list` commands.

Finally, counters can be reset with the `zero` and `resetlog` commands.

Available commands:

`add`	Adds a rule.
`delete`	Deletes the first rule with number <*number*>, if any.
`list`	Prints out the current rule set.
`show`	Equivalent to `ipfw -a list`.
`zero`	Zeroes the counters associated with rule number <*number*>.
`reset`	Zeroes the counters associated with rule number <*number*>.
`flush`	Removes all rules.

The following options are available:

`-q`	Uses quiet mode when adding, flushing, or zeroing (implies `-f`). Useful for adjusting rules by executing multiple ipfw commands in a script.
`-f`	Doesn't ask for confirmation for commands that can cause problems if misused (for example, `flush`).
`-a`	Shows counter values while listing. Also see `show`.
`-t`	Shows last match timestamp while listing.
`-N`	Tries to resolve addresses and service names in output.
`-s [`<*field*>`]`	While listing pipes, sorts according to one of the four counters (total and current packets or bytes).

To ease configuration, rules can be put into a file, which is processed using ipfw as shown in the first synopsis line. An absolute pathname must be used. The file is read line by line and applied as arguments to the ipfw utility.

Optionally, a preprocessor can be specified using -p preproc where pathname is to be piped through. Useful preprocessors include cpp(1) and m4(1). If preproc doesn't start with a slash (/) as its first character, the usual PATH name search is performed. Care should be taken in environments where not all file systems are mounted (yet) by the time ipfw is being run (for example, when they're mounted over NFS). When –p has been specified, optional -D and -U specifications can follow and are passed on to the preprocessor. This allows for flexible configuration files (like conditionalizing them on the local hostname) and the use of macros to centralize frequently required arguments like IP addresses.

The ipfw pipe commands are used to configure the traffic shaper.

Rule Format

The ipfw rule format is the following:

```
[prob <match_probability>] <action> [log [logamount <number>]] <proto> from <src> to
<dst> [<interface-spec>] [<options>]
```

Each incoming and outgoing packet is sent through the ipfw rules. In the case of a host acting as a gateway, packets forwarded by the host are processed twice: once when entering and once when leaving. Each packet can be filtered based on the following associated information:

Transmit and receive interface (by name or address)

Direction (incoming or outgoing)

Source and destination IP address (possibly masked)

Protocol (TCP, UDP, ICMP, and so on)

Source and destination port (lists, ranges or masks)

TCP flags

IP fragment flag

IP options

ICMP types

User/group ID of the socket associated with the packet

Note that it might be dangerous to filter on source IP address or source TCP/UDP port because either or both could be spoofed.

The ipfw utility works by going through the rule list for each packet until a match is found. All rules have two associated counters: a packet count and a byte count. These are updated when a packet matches the rule.

Rules are ordered by line number from 1 to 65534. Rules are tried in increasing order, with the first matching rule being the one that applies. Multiple rules may have the same number and are applied in the order they were added.

If a rule is added without a number, it's numbered 100 higher than the highest defined rule number unless the highest rule number is 65435 or greater, in which case the new rules are given that same number.

One rule is always present: 65535 deny all from any to any.

This rule, not to allow anything, is the default policy.

If the kernel option IPFIREWALL_DEFAULT_TO_ACCEPT has been enabled, the default rule is 65535 allow all from any to any.

The previous rule is the default rule in Mac OS X.

`prob <match_probability>` A match is only declared with the specified probability (floating-point number between 0 and 1). This can be useful for a number of applications such as random packet drop or (in conjunction with `dummynet`(4)) to simulate the effect of multiple paths leading to out-of-order packet delivery.

`log [logamount number]` If the kernel was compiled with `IPFIREWALL_VERBOSE`, when a packet matches a rule with the `log` keyword, a message will be logged to `syslogd`(8) with a `LOG_SECURITY` facility. Note: by default, they're appended to the `/var/log/security` file (see `syslog.conf`(5)). If the kernel was compiled with the `IPFIREWALL_VERBOSE_LIMIT` option, by default logging ceases after the number of packets specified by the option are received for that particular chain entry, and `net.inet.ip.fw.verbose_limit` will be set to that number. However, if `logamount` number is used, that number will be the logging limit rather than `net.inet.ip.fw.verbose_limit`, where the value 0 removes the logging limit. Logging may then be re-enabled by clearing the logging counter or the packet counter for that entry.

Console logging and the log limit are adjustable dynamically through the `sysctl`(8) interface in the MIB base of `net.inet.ip.fw`.

`proto` An IP protocol specified by number or name (for a complete list see `/etc/protocols`). The `ip` or `all` keywords mean any protocol will match. `tcp`, `udp`, `icmp` are commonly used ones.

Available options for `<action>`:

`allow` Allows packets that match rule. The search terminates. Aliases are `pass`, `permit`, and `accept`.

`deny` Discards packets that match rule. The search terminates. Alias is drop.

`reject` (Deprecated) Discards packets that match rule, and tries to send an ICMP host unreachable notice. The search terminates.

`unreach <code>` Discards packets that match rule, and tries to send an ICMP unreachable notice with code `<code>`, where `<code>` is a number from `0` to `255`, or one of these aliases: `net`, `host`, `protocol`, `port`, `needfrag`, `srcfail`, `net-unknown`, `host-unknown`, `isolated`, `net-prohib`, `host-prohib`, `tosnet`, `toshost`, `filter-prohib`, `host-precedence`, `precedence-cutoff`. The search terminates.

`reset` TCP packets only. Discards packets that match rule and tries to send a TCP reset (RST) notice. The search terminates.

`count` Updates counters for all packets that match rule. The search continues with the next rule.

`check-state`
Checks the packet against the dynamic ruleset. If a match is found, the search terminates; otherwise, we move to the next rule. If no `check-state` rule is found, the dynamic ruleset is checked at the first `keep-state` rule.

`divert <port>`
Diverts packets that match rule to `divert`(4) socket bound to port `<port>`. The search terminates.

`tee <port>`
Sends a copy of packets matching rule to the `divert`(4) socket bound to port `<port>`. The search terminates.

`fwd <ipaddr> [,<port>]`
Changes to the next hop on matching packets to `<ipaddr>`, which can be a dotted quad address or hostname. If `<ipaddr>` isn't directly reachable, the route as found in the local routing table for that IP address is used instead. If `<ipaddr>` is a local address, when a packet enters the system from a remote host, it's diverted to `<port>` on the local machine, keeping the local address of the socket set to the original IP address for which the packet was destined. This is intended for use with transparent proxy servers. If `<ipaddr>` isn't a local address, `<port>`, if specified, is ignored, and the rule applies only to packets leaving the system. If `<port>` isn't given, the port in the packet is used instead. The kernel must have been compiled with option `IPFIREWALL_FORWARD`.

`pipe <pip-nr>`
Passes packet to a `dummynet`(4) pipe (for bandwidth limitation, delay, and so on). The search terminates; however, on exit from the pipe and if the `sysctl`(8) variable `net.inet.ip.fw.one_pass` isn't set, the packet is passed again to the firewall code starting from the next rule.

`queue <queue-nr>`
Passes packet to a `dummynet`(4) queue (for bandwidth limitation using `WF2Q`).

`skipto <number>`
Skips subsequent rules numbered less than `<number>`. The search continues with the first rule numbered `<number>` or higher.

`src` and `dst:`

`any | me | [not] <address/mask> [<ports>]`

Specifying `any` makes the rule match any IP number.

Specifying `me` makes the rule match any IP number configured on an interface in the system. This is a computationally semiexpensive check which should be used with care.

The `<address/mask>` may be specified as

`ipno`
An IP number of the form `1.2.3.4`. Only this exact IP number will match the rule.

`ipno/bits`
An IP number with a mask width of the form `1.2.3.4/24`. In this case, all IP numbers from `1.2.3.0` to `1.2.3.255` will match.

`ipno:mask`
An IP number with a mask of the form `1.2.3.4:255.255.240.0`. In this case, all IP numbers from `1.2.0.0` to `1.2.15.255` will match.

The sense of the match can be inverted by preceding an address with the `not` modifier, causing all other addresses to be matched instead. This doesn't affect the selection of port numbers.

With the TCP and UDP protocols, optional ports may be specified as

```
{port|port-port|port:mask}[,port[,...]]
```

The `-` notation specifies a range of ports (including boundaries).

The `:` notation specifies a port and a mask, a match is declared if the port number in the packet matches the one in the rule, limited to the bits which are set in the mask.

Service names (from `/etc/services`) may be used instead of numeric port values. A range may only be specified as the first value, and the length of the port list is limited to `IP_FW_MAX_PORTS` ports (as defined in `/usr/src/sys/netinet/ip_fw.h`). A backslash (`\`) can be used to escape the dash (`-`) character in a service name:

```
ipfw add count tcp from any ftp\\-data-ftp to any
```

Fragmented packets that have a non-zero offset (that is, not the first fragment) will never match a rule which has one or more port specifications. See the `frag` option for details on matching fragmented packets.

Some combinations of the following specifiers are allowed for <*`interface-spec`*>:

`in`	Only matches incoming packets.
`out`	Only matches outgoing packets.
`via ifX`	Packet must be going through interface `ifX`.
`via if*`	Packet must be going through interface `ifX`, where `X` is any unit number.
`via any`	Packet must be going through some interface.
`via ipno`	Packet must be going through the interface having IP address `ipno`.

The `via` keyword causes the interface to always be checked. If `recv` or `xmit` is used instead of `via`, the only receive or transmit interface (respectively) is checked. By specifying both, it's possible to match packets based on both receive and transmit interface, for example:

```
ipfw add 100 deny ip from any to any out recv ed0 xmit ed1
```

The `recv` interface can be tested on either incoming or outgoing packets, whereas the `xmit` interface can only be tested on outgoing packets. So, `out` is required (and `in` is invalid) whenever `xmit` is used. Specifying via together with `xmit` or `recv` is invalid.

A packet may not have a receive or transmit interface: Packets originating from the local host have no receive interface, whereas packets destined for the local host have no transmit interface.

Options available for <*`options`*>:

`keep-state[<`*`method`*`>]`	Upon a match, the firewall creates a dynamic rule, whose default behavior is to match bidirectional traffic between source and destination IP/port using the same protocol. The rule has a limited lifetime (controlled by a set of `sysctl`(8) variables), and the lifetime is refreshed every time a matching packet is found. The actual behavior can be modified by specifying a different method, although at the moment only the default one is specified.

`bridged`
Matches only bridged packets. This can be useful for multicast or broadcast traffic, which would otherwise pass through the firewall twice: once during bridging, and a second time when the packet is delivered to the local stack.
Apart from a small performance penalty, this would be a problem when using pipes because the same packet would be accounted for twice in terms of bandwidth, queue occupation, and also counters.

`frag`
Matches if the packet is a fragment and it isn't the first fragment of the datagram. `frag` cannot be used in conjunction with either `tcpflags` or TCP/UDP port specifications.

`ipoptions <spec>`
Matches if the IP header contains the comma-separated list of options specified in <*spec*>. The supported IP options are `ssrr` (strict source route), `lsrr` (loose source route), `rr` (record packet route), and `ts` (timestamp). The absence of a particular option may be denoted with a `!`.

`tcpoptions <spec>`
Matches if the TCP header contains the comma-separated list of options specified in `spec`. The supported TCP options are `mss` (maximum segment size), `window` (tcp window advertisement), `sack` (selective ack), `ts` (rfc1323 timestamp), and `cc` (rfc1644 t/tcp connection count). The absence of a particular option may be denoted with a `!`.

`established`
TCP packets only. Matches packets that have the RST or ACK bits set.

`setup`
TCP packets only. Matches packets that have the SYN bit set but no ACK bit.

`tcpflags <spec>`
Matches if the TCP header contains the comma-separated list of flags specified in <*spec*>. The supported TCP flags are `fin`, `syn`, `rst`, `psh`, `ack`, and `urg`. The absence of a particular flag may be denoted by an `!`. A rule that contains a `tcpflags` specification can never match a fragmented packet that has a nonzero offset.

`icmptypes <types>`
Matches if the ICMP type is in the list <*types*>. The list may be specified as any combination of ranges or individual types separated by commas:
echo reply (0), destination unreachable (3), source quench (4), redirect (5), echo request (8), router advertisement (9), router solicitation (10), time-to-live exceeded (11), IP header bad (12), timestamp request (13), timestamp reply (14), information request (15), information reply (16), address mask request (17), and address mask reply (18)

`uid <user>`
Matches all TCP or UDP packets sent by or received for a user. A user may be matched by name or identification number.

`gid <group>`
Matches all TCP or UDP packets sent by or received for a group. A group may be matched by name or identification number.

Important points to consider when designing your rules:

Remember that you filter packets both going in and out. Most connections need packets going in both directions.

Remember to test very carefully. It's a good idea to be at the console at the time.

Don't forget the `loopback` interface.

jobs

`jobs`	Displays the table of current jobs.
`jobs [-l]`	
`-l`	Lists jobs in long format. This includes the job number and its associated process ID.

After you know what jobs belong to the current shell, there are several ways to refer to a job. `%` introduces a job name. Job number `1` is `%1`. An unambiguous string of characters at the beginning of the name can be used to refer to a job; the form is `%<`*`first-few-characters-of-job`*`>`. An unambiguous string of characters in the job name can also be used to refer to a job; for example, the form `%?<`*`text-string`*`>` specifies a job whose name contains `<`*`text-string`*`>`.

Output pertaining to the current job is marked with `+`; output from a previous job, `-`. `%+`, `%`, and `%%` refer to the current job. `%-` refers to the previous job.

kill

`kill`	Sends a signal to a process or terminates a process.
`kill [-<`*`signal`*`>] %<`*`job`*`> | <`*`pid`*`>`	
`kill -l [exit-status]`	
`-l [exit-status]`	With no argument, lists all the signal names; otherwise, lists the signal associated with the status *`exit-status`*.
`<`*`signal`*`>`	Specifies which signal to send to a process. If `<`*`signal`*`>` isn't specified, the `TERM` (terminate) signal is sent. `<`*`signal`*`>` may be a number or name.
`%<`*`job`*`>`	Specifies the job that should receive a signal.
`<`*`pid`*`>`	Specifies the process ID that should receive a signal. The process ID can be determined by running `ps`.

Signal `KILL (9)` is a sure way to kill a process. Signal `HUP` is another common signal to send to a process. You often can send a `HUP` signal to a process to get it to reread its configuration file.

killall

`killall`	Kills processes by name.

`killall [-d | -v] [-help] [-l] [-m] [-s] [-u <`*`user`*`>] [-t <`*`tty`*`>] [-c <`*`procname`*`>] [-<`*`SIGNAL`*`>] [<`*`procname`*`> ...]`

`Killall` kills processes selected by name, as opposed to the selection by `pid` as done by `kill`. By default, it sends a `TERM` signal to all processes with a real UID identical to the caller of `killall` that match the name <*procname*>. The root user is allowed to kill any process.

`-d | -v` Verbose mode. For a single `-d` option, a list of the processes that will be sent the signal will be printed, or a message indicating that no matching processes have been found. For `–v`, the process id is printed.

`-help` Gives help on the command usage and exits.

`-l` Lists the names of the available signals and exits, as in `kill`.

`-m` Matches the argument <*procname*> as a (case-insensitive) regular expression against the names of processes found. Caution! This is dangerous; a single dot will match any process running under the real UID of the caller.

`-s` Silent mode. Shows only what would be done, but doesn't send any signal.

`-`<*SIGNAL*> Sends the specified <*SIGNAL*> instead of the default TERM. The signal may be specified either as a name (with or without a leading SIG), or numerically.

`-u` <*user*> Limits potentially matching processes to those belonging to the specified <*user*>.

`-t` <*tty*> Limits potentially matching processes to those running on the specified <*tty*>.

`-c` <*procname*> When used with the `-u` or `-t` flags, limits potentially matching processes to those matching the specified <*procname*>.

last

`last` Indicates last logins of `users` and `ttys`.

`last [-n] [-f` <*file*>`] [-h` <*host*>`] [-t` <*tty*>`] [`<*user1*> <*user2*> `...]`

`last` lists the sessions of specified users, ttys, and hosts, in reverse time order. Each line of output contains the username, the `tty` from which the session was conducted, any hostname, the start and stop times for the session, and the duration of the session. If the session is still in progress or was cut short by a crash or shutdown, `last` indicates that.

`-n` Limits the report to `n` lines.

`-f` <*file*> Reads <*file*> instead of the default `/var/log/wtmp`.

`-h` <*host*> Lists sessions from <*host*>. <*host*> may be a name or Internet number.

`-t` <*tty*> Lists sessions on <*tty*>. <*tty*> may be given fully or abbreviated. For example, `last -t p3` is equivalent to `last -t ttyp3`.

If multiple arguments are given, the information that applies to any of the arguments is printed. For example, `last root -t console` would list all sessions of `root` as well as all sessions on the `console`. The pseudo-user `reboot` logs in at system reboot, so last reboot gives an indication of the mean time between reboots.

less

`less`	Opposite of `more` (a pager).

Pages through data or text files.

```
less -?
less --help
less -V
less --version
less [-[+]aBcCdeEfgGiImMnNqQrsSuUVwX][-b <bufs>][-h <lines>][-j <line>] [-k <keyfile>]
[--{oO} <logfile>] [-p <pattern>][-t <tag>] [-T <tagsfile>] [-x <tab>] [-y <lines>]
[-[-z] <lines>][+[+]<cmd>] [--] [<file1>...]
```

Summary of `less` Commands

Commands marked with `*` may be preceded by a number, *N*.

Notes in parentheses indicate the behavior if *N* is given.

`h`	`H`					Display this help.
`q`	`:q`	`Q`	`:Q`	`ZZ`		Exit.
MOVING						
`e`	`^E`	`j`	`^`*N*	`CR`	*	Forward one line (or *N* lines).
`y`	`^Y`	`k`	`^K`	`^P`	*	Backward one line (or *N* lines).
`f`	`^F`	`^V`	SPACE		*	Forward one window (or *N* lines).
`b`	`^B`	ESC+v			*	Backward one window (or *N* lines).
`z`					*	Forward one window (and set window to *N*).
`w`					*	Backward one window (and set window to *N*).
ESC+SPACE					*	Forward one window, but don't stop at end-of- file.
`d`	`^D`				*	Forward one half-window (and set half-window to *N*).
`u`	`^U`				*	Backward one half-window (and set half-window to *N*).
ESC+(	RightArrow				*	Left 8 character positions (or *N* positions).
ESC+)	LeftArrow				*	Right 8 character positions (or *N* positions).
`F`						Forward forever; like `tail -f`.

r	^R	^L	Repaint screen.
R			Repaint screen, discarding buffered input.

Default window is the screen height.

Default half-window is half of the screen height.

Searching

/pattern	*	Search forward for (*N*th) matching line.
?pattern	*	Search backward for (*N*th) matching line.
N	*	Repeat previous search (for *N*th occurrence).
N	*	Repeat previous search in reverse direction.
ESC+n	*	Repeat previous search, spanning files.
ESC+N	*	Repeat previous search, reverse dir. and spanning files.
ESC+u		Undo (toggle) search highlighting.

Search patterns may be modified by one or more of the following:

^N or !	Search for NON-matching lines.
^E or *	Search multiple files (pass through END OF FILE).
^F or @	Start search at FIRST file (for /) or last file (for ?).
^K	Highlight matches, but don't move (KEEP position).
^R	Don't use REGULAR EXPRESSIONS.

Jumping

g	<	ESC+<	*	Goes to first line in file (or line *N*).
G	>	ESC+>	*	Goes to last line in file (or line *N*).
p	%		*	Goes to beginning of file (or *N* percent into file).
{	(	[	*	Finds close bracket })].
}	)	]	*	Finds open bracket { ([.
ESC+^F <c1> <c2>			*	Finds close bracket <c2>.
ESC+^B <c1> <c2>			*	Finds open bracket <c1>

Each “find close bracket” command goes forward to the close bracket matching the (*N*-th) open bracket in the top line. Each “find open bracket” command goes backward to the open bracket matching the (*N*-th) close bracket in the bottom line.

`m<letter>`	Marks the current position with *`<letter>`*.
`'<letter>`	Goes to a previously marked position.
`''`	Goes to the previous position.
`^X^X`	Same as `'`.

A mark is any uppercase or lowercase letter. Certain marks are predefined:

`^`	Means beginning of the file.
`$`	Means end of the file.

Changing Files

`:e [file]`		Examines a new file.
`^X^V`		Same as `:e`.
`:n`	*	Examines the (*N*th) next file from the command line.
`:p`	*	Examines the (*N*th) previous file from the command line.
`:x`	*	Examines the first (or *N*th) file from the command line.
`:d`		Deletes the current file from the command line list.
`=` `^G` `:f`		Prints current filename.

Miscellaneous Commands

`-<flag>`	Toggles a command-line option (see OPTIONS section later).
`--<name>`	Toggles a command-line option, by name.
`_<flag>`	Displays the setting of a command-line option.
`__<name>`	Displays the setting of an option, by name.
`+cmd`	Executes the *less* cmd each time a new file is examined.
`!command`	Executes the shell command with `$SHELL`.
`|Xcommand`	Pipes file between current pos & mark X to shell command.
`V`	Edits the current file with `$VISUAL` or `$EDITOR`.
`V`	Prints version number of *less*.

Options

Most options may be changed either on the command line, or from within less by using the - or -- command.

Options may be given in one of two forms: either a single character preceded by a -, or a name preceded by --.

-?		--help	Displays help (from command line).
-a		--search-skip-screen	Forward search skips current screen.
-b [N]		--buffers=[N]	Number of buffers.
-B		--auto-buffers	Doesn't automatically allocate buffers for pipes.
-c	-C	--clear-screen --CLEAR-SCREEN	Repaint by scrolling/clearing.
-d		--dumb	Dumb terminal.
-e	-E	--quit-at-eof --QUIT-AT-EOF	Quits at end of file.
-f		--force	Forces open non-regular files.
-g		--hilite-search	Highlights only last match for searches.
-G		--HILITE-SEARCH	Doesn't highlight any matches for searches.
-h [N]		--max-back-scroll=[N]	Backward scroll limit.
-I		--ignore-case	Ignores case in searches.
-I		--IGNORE-CASE	Ignores case in searches and in search patterns.
-j [N]		--jump-target=[N]	Screen position of target lines.
-k [file]		--lesskey-file=[file]	Uses a lesskey file.
-m	-M	--long-prompt --LONG-PROMPT	Sets prompt style.
-n	-N	--line-numbers --LINE-NUMBERS	Uses line numbers.
-o [file]		--log-file=[file]	Copies to log file (standard input only).
-O [file]		--LOG-FILE=[file]	Copies to log file (unconditionally overwrite).
-p [pattern]		--pattern=[pattern]	Starts at pattern (from command line).
-P [prompt]		--prompt=[prompt]	Defines new prompt.
-q	-Q	--quiet --silent --QUIET –SILENT	Quiets the terminal bell.
-r		--raw-control-chars	Outputs raw control characters.
-s		--squeeze-blank-lines	Squeezes multiple blank lines.
-S		--chop-long-lines	Chops long lines.
-t [tag]		--tag=[tag]	Finds a tag.
-T [tagsfile]		--tag-file=[tagsfile]	Uses an alternative tags file.

`-u`	`-U`	`--underline-special` `--UNDERLINE-SPECIAL`	Changes handling of backspaces.
`-V`		`--version`	Displays the version number of *less*.
`-w`		`--hilite-unread`	Highlights first new line after forward-screen.
`-W`		`--HILITE-UNREAD`	Highlights first new line after any forward movement.
`-x [N]`		`--tabs=[N]`	Sets tab stops.
`-X`		`--no-init`	Doesn't use termcap init/deinit strings.
`-y [N]`		`--max-forw-scroll=[N]`	Forward scroll limit.
`-z [N]`		`--window=[N]`	Sets size of window.
`-" [c[c]]`		`--quotes=[c[c]]`	Sets shell quote characters.
`-~`		`--tilde`	Doesn't display tildes after end of file.

Line Editing

These keys can be used to edit text being entered on the command line at the bottom of the screen.

RightArrow	ESC+l	Moves cursor right one character.
LeftArrow	ESC+h	Moves cursor left one character.
CNTL+RightArrow	ESC+RightArrow ESC+w	Moves cursor right one word.
CNTL+LeftArrow	ESC+LeftArrow ESC+b	Moves cursor left one word.
HOME	ESC+0	Moves cursor to start of line.
END	ESC+$	Moves cursor to end of line.
BACKSPACE		Deletes char to left of cursor.
DELETE	ESC+x	Deletes char under cursor.
CNTL+BACKSPACE	ESC+BACKSPACE	Deletes word to left of cursor.
CNTL+DELETE	ESC+DELETE ESC+X	Deletes word under cursor.
CNTL+U		Deletes entire line.
UpArrow	ESC+k	Retrieves previous command line.
DownArrow	ESC+j	Retrieves next command line.
TAB		Completes filename and cycle.
SHIFT+TAB	ESC+TAB	Completes filename and reverses cycle.
CNTL+L		Completes filename, lists all.

locate

`locate` Finds files.

`locate <pattern>`

Searches a database for all pathnames that match `<pattern>`. The database is rebuilt periodically and contains the names of all publicly accessible files.

Shell and globbing characters (`*`, `?`, `\`, `[`, and `]`) may be used in *`<pattern>`*, although they must be escaped. Preceding a character by `\` eliminates any special meaning for it. No characters must be explicitly matched, including `/`.

As a special case, a pattern with no globbing characters (`foo`) is matched as (`*foo*`).

Useful files:

`/var/db/locate.database`	Database
`/usr/libexec/locate.updatedb`	Script to update database

ln

`ln`	Makes links.

```
ln [-fhns] <source> <target>
ln [-fhns] <source1> <source2> <source3> ... <directory>
```

In the first form, `ln` links *`<source>`* to *`<target>`*. If *`<target>`* is a directory, a link named *`<source>`* is placed in *`<target>`*.

In the second form, `ln` makes links to the files enumerated by *`<source1>`* *`<source2.`* *`<source3>`* `...` in *`<directory>`*. The links have the same names as the sources in the list.

There are two types of links: hard links and symbolic links. The default is hard links. A hard link to a file is indistinguishable from the original directory entry. Hard links may not normally refer to directories and may not span file systems.

A symbolic link refers by name to the file to which it's linked. Symbolic links may refer to directories and may span file systems.

`-f`	Forces the link to occur by unlinking any already existing links.
`-h`	If *`<target>`* or *`<directory>`* is a symbolic link, it isn't followed. This is most useful when used with `-f`, to replace a symbolic link that might point to a directory.
`-n`	Same as `-h`. Retained for compatibility with other implementations of `ln`.
`-s`	Creates a symbolic link; this is most like the idea of aliases you're already familiar with.

lp

`lp`	Sends a job to the printer.

```
lp [ -E ] [ -c ] [ -d <printer> ] [ -h <hostname> ] [ -m ] [-n <num-copies>] [ -o
<option> ] [ -q <priority> ] [ -s  ]  [  -t <title> ] [ -H <handling> ] [ -P <page-
list> ] [ <file1> <file2> ... ]
lp  [  -E  ]  [ -c ] [ -h <server> ] [ -i <job-id> ] [ -n <num-copies>] [ -o <option>
] [ -q <priority> ] [ -t  <title>  ]  [  -H <handling> ] [ -P <page-list> ]
```

`-E`	Forces encryption when connecting to the server.
`-c`	Option is provided for backward-compatibility only. On systems that support it, this option forces the print file to be copied to the spool directory before printing. In CUPS, print files are always sent to the scheduler via IPP, which has the same effect.
`-d <printer>`	Prints to the specified `<printer>`.
`-h <server>`	Specifies the print server hostname. The default is localhost or the value of the `CUPS_SERVER` environment variable.
`-i <job-id>`	Specifies an existing job to modify.
`-m`	Sends email when the job is completed (not supported in CUPS 1.1.).
`-n <num-copies>`	Sets the number of copies to print from 1 to 100.
`-o <option>`	Sets a job option.
`-q <priority>`	Sets the job priority from 1 (lowest) to 100 (highest). The default priority is 50.
`-s`	Silent mode. Doesn't report the resulting job IDs.
`-t <title>`	Sets the job name.
`-H <handling>`	Specifies when the job should be printed. A value of immediate prints the file immediately, a value of hold holds the job indefinitely, and a time value (`HH:MM`) will hold the job until the specified time. Use a value of resume with the `-i` option to resume a held job.
`-P <page-list>`	Specifies which pages to print in the document. The list can contain a list of numbers and ranges (#-#) separated by commas (for example, 1,3-5,16).

lpadmin

`lpadmin` Configures CUPS printers and classes.

```
lpadmin [ -E ] [ -h <server> ] -d <destination>
lpadmin [ -E ] [ -h <server> ] -p <printer> <option(s)>
lpadmin [ -E ] [ -h <server> ] -x <destination>
```

`lpadmin` configures printer and class queues provided by CUPS. It can also be used to set the system default printer or class.

When specified before the `-d`, `-p`, or `-x` options, the `–E` option forces encryption when connecting to the server.

The first form of the command sets the default printer or class to `<destination>`. Subsequent print jobs submitted via the `lp(1)` or `lpr(1)` commands use this destination unless the user specifies otherwise.

The second form of the command configures the named `<printer>`.

The third form of the command deletes the printer or class `<destination>`. Any jobs that are pending for the `<destination>` are removed and any job that's is currently printing is aborted.

Printer queue configuration options:

`-c <class>` Adds the named <*printer*> to <*class*>. If <*class*> doesn't exist, it's created automatically.

`-i <interface>` Sets a System V V–style interface script for the printer. This option cannot be specified with the `–P` option (PPD file) and is intended for providing support for legacy printer drivers.

`-m <model>` Sets a standard System V interface script or PPD file from the model directory.

`-o <name>=<value>` Sets a PPD or server option for the printer. PPD options can be listed using the `-l` option with the `lpoptions`(1) command.

`-o job-k-limit=<value>` Sets the kilobyte limit for per-user quotas. The <*value*> is an integer number of kilobytes; one kilobyte is 1024 bytes.

`-o job-page-limit=<value>` Sets the page limit for per-user quotas. The <*value*> is the integer number of pages that can be printed; double-sided pages are counted as two pages.

`-o job-quota-period=<value>` Sets the accounting period for per-user quotas. The <*value*> is an integer number of seconds; 86,400 seconds are in one day.

`-r <class>` Removes the named <*printer*> from <*class*>. If the resulting class becomes empty, it's removed.

`-u allow:<user>,<user>`

`-u deny:<user>,<user>`

`-u allow:all`

`-u deny:none` Sets user-level access control on a printer. The latter two forms turn user-level access control off.

`-v <device-uri>` Sets the `device-uri` attribute of the printer queue. If <*device-uri*> is a filename, it's automatically converted to the form `file:/file/name`.

`-D <info>` Provides a textual description of the printer.

`-E` Enables the printer and accepts jobs; this is the same as running the `accept(8)` and `enable(8)` programs on the printer.

`-L <location>` Provides a textual location of the printer.

`-P <ppd-file>` Specifies a PostScript Printer Description file to use with the printer. If specified, this option overrides the `-i` option (interface script).

lpinfo

`lpinfo` Shows available printing devices and drivers.

`lpinfo [ -E ] [ -l ] [ -m ] [ -v ]`

`-E` Forces encryption when connecting to the server.

`-l` Shows a "long" listing of devices or drivers.

`-m` Shows the available printer drivers on the system. This option is useful for discovering what `-m` models are available for use with the `lpadmin` command.

`-v` Shows the available printer devices on the system.

lpoptions

`lpoptions`	Displays or sets printer options and defaults.
`lpoptions -d <printer>`	
`lpoptions [-p <printer>] -l`	
`lpoptions -p <printer> -o <option>[=<value>] ...`	
`lpoptions -x <printer>`	
`-d <printer>`	Sets the default printer to `<printer>`. Overrides the system default printer for the current user.
`-l`	Lists the printer-specific options and their current settings.
`-o <option>=<value>`	Specifies a new option for the named destination.
`-p <printer>`	Sets the destination to `<printer>`.
`-x <printer>`	Removes the options for the named destination.

If no options are specified using the `-o` option, the current options for the named printer are reported on the standard output.

Options set with the `lpoptions` command are used by the `lp`(1) and `lpr`(1) commands when submitting jobs.

lpr

`lpr`	Sends a job to the printer.

```
lpr  [ -E ] [ -P <printer> ] [ -# <num-copies> [ -l ] [ -o
   <option> ] [ -p] [ -r ] [ -C/J/T <title> ] [ <file1> <file2> ... ]
```

`lpr` submits files for printing. Files named on the command line are sent to the specified printer (or the default system printer if none is specified.). If no files are listed on the command line, `lpr` reads the print file from the standard input.

`-E`	Forces encryption when connecting to the server.
`-P <printer>`	Specifies `<printer>` as the printer. Otherwise, the site's default printer is used.
`-# <num-copies>`	Sets the number of copies to print from 1 to 100.
`-l`	Specifies that the print file is already formatted for the destination and should be sent without filtering. Option is equivalent to `-oraw`.
`-o <option>`	Sets a job option.
`-p`	Specifies that the print file should be formatted with a shaded header with the date, time, job name, and page number. Option is equivalent to `-oprettyprint` and is only useful when printing text files.
`-r`	Removes the named print files after printing them.
`-C <title>`	Sets the job name.
`-J <title>`	Sets the job name.
`-T <title>`	Sets the job name.

Options `c`, `d`, `f`, `g`, `i`, `m`, `n`, `t`, `v` and `w` aren't supported by the CUPS system and produce a warning message if used.

lprm

`lprm`	Removes print jobs from the queue.

```
lprm [ -E ] [ - ] [ -P <printer> ] [ <job#1> <job#2> ... ]
```

`-E`	Forces encryption when connecting to the server.
`-`	Removes all print jobs in the queue.
`-P<printer>`	Specifies `<printer>` as the printer. Otherwise, the site's default is used.
`<job#>`	Removes from the queue the print job specified by `<job#>`. The `<job#>` can be determined by using `lpq(1)`.

lpstat

`lpstat`	Prints CUPS status information.

```
lpstat [ -E ] [-a [<printer(s)>]] [-c [<class(es)>]] [ -d ] [ -h <server> ] [ -l ] [-o
[<destination(s)>]] [ -p [<printer(s)>]]  [  -r  ]  [ -R ] [ -s ] [ -t ] [ -u
[<user(s)>]] [ -v [<printer(s)>]]
```

`-E`	Forces encryption when connecting to the server.
`-a [<printer(s)>]`	Shows the accepting state of printer queues. If no printers are specified, all printers are listed.
`-c [<class(es)>]`	Shows the printer classes and the printers that belong to them. If no classes are specified, all classes are listed.
`-d`	Shows the current default destination.
`-h <server>`	Specifies the CUPS server to communicate with.
`-l`	Shows a long listing of printers, classes, or jobs.
`-o [<destination(s)>]`	Shows the jobs queue on the specified destinations. If no destinations are specified, all jobs are shown.
`-p [<printer(s)>]`	Shows the printers and whether or not they are're enabled for printing. If no printers are specified, all printers are listed.
`-r`	Shows whether or not the CUPS server is running.
`-R`	Shows the ranking of print jobs.
`-s`	Shows a status summary, including the default destination, a list of classes and their member printers, and a list of printers and their associated `-d`, `-c`, and `-p` options.
`-t`	Shows all status information. This is equivalent to using the `-r`, `-d`, `-c`, `-d`, `-v`, `-a`, `-p`, and `-o` options.
`-u [<user(s)>]`	Shows a list of print jobs queued by the specified users. If no users are specified, lists the jobs queued by the current user.
`-v [<printer(s)>]`	Shows the printers and what device they are're attached to. If no printers are specified, all printers are listed.

lpq

`lpq`	Displays the queue of print jobs.
`lpq [ -E ] [ -P <printer> ] [ -a ] [ -l ] [ +<interval> ]`	
`-E`	Forces encryption when connecting to the server.
`-P <printer>`	Specifies `<printer>` as the printer. Otherwise, the site's default printer is used.
`-a`	Displays the queues for all printers.
`-l`	Displays the queue information in long format. Includes the name of the host from which the job originated.
`+<interval>`	Displays a continuous report of the jobs in the queue once every `<interval>` seconds until the queue is empty.

ls

`ls`	Lists files or directory contents.
`ls [-ACFLRSTWadfgilnoqrsktcux1] <file1> <file2> ...`	
`ls [-ACFLRSTWadfgilnoqrsktcux1]`	
`-A`	Lists all entries except for "`.`" and "`..`". Always set for superuser.
`-C`	Forces multicolumn output. This is the default when output is to a terminal.
`-F`	Displays a symbol, if applicable, after each file to denote the following: slash (`/`) for a directory; asterisk (`*`) for an executable; an at sign (`@`) for a symbolic link; a percent sign (`%`) for a whiteout; an equal sign (`=`) for a socket; a vertical bar (`|`) for a FIFO.
`-L`	If the argument is a symbolic link, the file or directory the link references rather than the link itself is displayed.
`-R`	Recursively lists subdirectories.
`-S`	Sorts by size, largest file first.
`-T`	Displays complete time information, including month, day, hour, minute, second, and year.
`-W`	Displays whiteouts.
`-a`	Lists all files in the directory, including files whose names begin with a dot (`.`).
`-d`	If the argument is a directory, it's listed as a plain file, rather than listing its contents. If the argument is a symbolic link, its link information isn't displayed.
`-f`	Doesn't sort output.
`-g`	Does nothing. Kept for compatibility with older versions of `ls`.
`-i`	Lists the argument's serial number (inode number).

`-l` Lists in long format. Displays file mode, number of links, owner name, group name, size of the file in bytes, date and time file was last modified, and the file. If displayed to a terminal, the first line of output is the total number of 512-byte blocks used by the files in the directory.

`-n` Displays user and group ID as numbers rather than names in a long (`-l`) output.

`-o` Includes file flags in a long (`-l`) output.

`-q` Forces printing of nongraphic characters in filenames as character `?`. This is the default when output is to a terminal.

`-r` Reverses sort order to reverse alphabetic order; smallest first or oldest first, as appropriate.

`-s` Displays file size in 512-byte blocks, where partial units are rounded up to the next integer value. If the output is to a terminal, first line displayed is the total number of 512-byte blocks used by files in the directory.

`-k` Modifies the `-s` option to report sizes in kilobytes.

`-t` Sorts by time modified (most recently modified first) before sorting in alphabetic order.

`-c` Uses time when file status was last changed for sorting (`-t`) or printing (`-l`).

`-u` Uses time of last access for sorting (`-t`) or printing (`-l`).

`-x` Forces multicolumn output sorted across the page rather than down the page.

`-v` Forces unedited printing of nongraphic characters. This is the default when output isn't to a terminal.

`-1` Forces output to one entry per line. This is the default when output isn't to a terminal.

`-1`, `-C`, `-l`, and `-x` options override each other. The last option specified determines the format used.

`-c` and `-u` options override each other. The last option specified determines the file time used.

lsbom

`lsbom` View bill of material (bom) files.

`lsbom [-bcdflmsx] [-arch <`*`archVal`*`>] [-p <`*`parameters`*`>] <`*`bom ...`*`>`

The `lsbom` command interprets the contents of binary bom (bom(5)) files. For each file in a bom, `lsbom` prints the file path and/or requested information.

If no options are given, `lsbom` displays the output formatted such that each line contains the path of the entry, its mode (octal), and its UID/GID. There are slight differences in the output for plain files, directories, symbolic links, and device files as follows:

plain files	The UID/GID is followed by the file size and a 32-bit CRC checksum of the file's contents.
symbolic links	The UID/GID is followed by the size and checksum of the link path, and the link path itself.
device files	The UID/GID file number is followed by the device number.

The `-p` option can be used to specify a user-defined format for `lsbom's` output. The format string consists of one or more characters described following where each character represents a data type. Data types are separated by tab characters, and each line ends with a newline character. You can use this mechanism to create output similar to the `ls`(1) command.

`-b`	Lists block devices
`-c`	Lists character devices
`-d`	Lists directories
`-f`	Lists files
`-l`	Lists symbolic links
`-m`	Prints modified times (for plain files only)
`-s`	Prints only the path of each file
`-x`	Suppresses modes for directories and symlinks
`-arch`	`archVal` when displaying plain files that represent fat mach-o binaries, prints the size and checksum of the file contents for the specified `archVal` (either `ppc` or `i386`)
`-p` *`<param>`*	Prints only some of the results. Note: Each option can only be used once:
`c`	32-bit checksum
`f`	filename
`F`	filename with quotes (for example, `"/usr/bin/lsbom"`)
`g`	group id
`G`	group name
`m`	file mode (permissions)
`M`	symbolic file mode (for example, `"dr-xr-xr-x"`)
`s`	file size
`S`	formatted size
`t`	mod time
`T`	formatted mod time
`u`	user id
`U`	user name
`/`	user id/group id
`?`	user name/group name

lynx

`lynx`	Textual Web browser.

`lynx [options] [file]`

You can find out which options are available by running `lynx -help`. Here's the listing of command-line options for the current version of `lynx`:

`-`	Receives options and arguments from `STDIN`.
`-accept_all_cookies`	Accepts cookies without prompting if `Set-Cookie` handling is on (off).
`-anonymous`	Applies restrictions for anonymous account; see also `-restrictions`.
`-assume_charset=<MIMEname>`	Charset for documents that don't specify it.
`-assume_local_charset=<MIMEname>`	Charset assumed for local files.
`-assume_unrec_charset=<MIMEname>`	Use this instead of unrecognized charsets.
`-auth=<id>:<pw>`	Authentication information for protected documents.
`-base`	Prepends a request URL comment and BASE tag to `text/html` for `-source` dumps.
`-bibhost=<URL>`	Specifies a local bibp server (default `http://bibhost/`).
`-book`	Uses the bookmark page as the start file (off).
`-buried_news`	Toggles scanning of news articles for buried references (on).
`-cache=<NUMBER>`	<NUMBER> of documents cached in memory.
`-case`	Enables case-sensitive user searching (off).
`-cfg=<FILENAME>`	Specifies a `lynx.cfg` file other than the default.
`-child`	Exits on left arrow in start file, and disable save to disk.
`-cmd_log=<FILENAME>`	Logs keystroke commands to the given file.
`-cmd_script=<FILENAME>`	Reads keystroke commands from the given file.
`-connect_timeout=<N>`	Sets the <N>-second connection timeout (`18000`).
`-cookie_file=<FILENAME>`	Specifies a file to use to read cookies.
`-cookie_save_file=<FILENAME>`	Specifies a file to use to store cookies.
`-cookies`	Toggles handling of `Set-Cookie` headers (on).
`-core`	Toggles forced core dumps on fatal errors (off).
`-crawl`	With `-traversal`, outputs each page to a file. With `-dump`, formats output as with `-traversal`, but to `STDOUT`.
`-curses_pads`	Uses curses pad feature to support left/right shifting (on).
`-debug_partial`	Incrementally displays stages with `MessageSecs` delay (off).
`-display=<DISPLAY>`	Sets the display variable for X `exec`'ed programs.
`-display_charset=<MIMEname>`	Sets the charset for the terminal output.
`-dont_wrap_pre`	Inhibits wrapping of text in <pre> when `-dump`ing and `-crawl`ing, mark wrapped lines in interactive session (off).
`-dump`	Dumps the first file to `STDOUT` and exit.
`-editor=<EDITOR>`	Enables edit mode with specified editor.
`-emacskeys`	Enables `emacs`-like key movement (off).
`-enable_scrollback`	Toggles compatibility with comm programs' scrollback keys (might be incompatible with some `curses` packages) (off).

`-error_file=<FILE>`	Writes the HTTP status code here.
`-force_empty_hrefless_a`	Forces `HREF`-less `A` elements to be empty (closes them as soon as they're seen) (off).
`-force_html`	Forces the first document to be interpreted as HTML (off).
`-force_secure`	Toggles forcing of the secure flag for SSL cookies (off).
`-forms_options`	Toggles forms-based versus old-style options menu (on).
`-from`	Toggles transmission of From headers (on).
`-ftp`	Disables FTP access (off).
`-get_data`	User data for `get` forms, read from `STDIN`, terminated by `'---'` on a line.
`-head`	Sends a `HEAD` request (off).
`-help`	Prints this usage message.
`-hiddenlinks=[option]`	Hidden links options are `merge`, `listonly`, and `ignore`.
`-historical`	Toggles use of `'>'` or `'-->'` as a terminator for comments (off).
`-homepage=<URL>`	Sets home page separate from start page.
`-image_links`	Toggles inclusion of links for all images (off).
`-index=<URL>`	Sets the default index file to `<URL>`.
`-ismap`	Toggles inclusion of ISMAP links when client-side MAPs are present (off).
`-link=<NUMBER>`	Starting count for `lnk#.dat` files produced by `-crawl` (0).
`-localhost`	Disables URLs that point to remote hosts (off).
`-mime_header`	Includes MIME headers and force source dump.
`-minimal`	Toggles minimal versus valid comment parsing (off).
`-newschunksize=<NUMBER>`	Number of articles in chunked news listings.
`-newsmaxchunk=<NUMBER>`	Maximum news articles in listings before chunking.
`-nobold`	Disables bold video attribute.
`-nobrowse`	Disables directory browsing.
`-nocc`	Disables Cc: prompts for self copies of mailings (off).
`-nocolor`	Turns off color support.
`-nofilereferer`	Disables transmission of Referer headers for file URLs (on).
`-nolist`	Disables the link list feature in dumps (off).
`-nolog`	Disables mailing of error messages to document owners (on).
`-nonrestarting_sigwinch`	Makes window size change handler non-restarting (off).
`-nopause`	Disables forced pauses for status-line messages.
`-noprint`	Disables some print functions, like `-restrictions=print` (off).
`-noredir`	Doesn't follow Location: redirection (off).
`-noreferer`	Disables transmission of Referer headers (off).
`-noreverse`	Disables reverse video attribute.
`-nostatus`	Disables the miscellaneous information messages (off).
`-nounderline`	Disables underline video attribute.

`-number_fields` Forces numbering of links as well as form input fields (off).

`-number_links` Forces numbering of links (off).

`-partial` Toggles display of partial pages while downloading (on).

`-partial_thres=<NUMBER>` Number of lines to render before repainting display with partial-display logic (-1).

`-pauth=<id>:<pw>` Authentication information for protected proxy server.

`-popup` Toggles handling of single-choice `SELECT` options via pop-up windows or as lists of radio buttons (off).

`-post_data` User data for `post` forms, read from `STDIN`, terminated by `'---'` on a line.

`-preparsed` Show parsed text/HTML with `-source` and in source view to visualize how `lynx` behaves with invalid HTML (off).

`-print` Enables print functions (DEFAULT); opposite of `-noprint` (on).

`-pseudo_inlines` Toggles pseudo-`ALT`s for inlines with no `ALT` string (on).

`-raw` Toggles default setting of 8-bit character translations or CJK mode for the startup character set (off).

`-realm` Restricts access to URLs in the starting realm (off).

`-reload` Flushes the cache on a proxy server (only the first document affected) (off).

`-restrictions=[<options>]` Uses `-restrictions` to see list.

`-resubmit_posts` Toggles forced resubmissions (no cache) of forms with method `POST` when the documents they returned are sought with the `PREV_DOC` command or from the History List (off).

`-rlogin` Disables `rlogins` (off).

`-selective` Requires `.www_browsable` files to browse directories.

`-short_url` Enables examination of beginning and end of long URL in status line (off).

`-show_cursor` Toggles hiding of the cursor in the lower-right corner (on).

`-soft_dquotes` Toggles emulation of the old Netscape and Mosaic bug that treated `'>'` as a coterminator for double quotes and tags (off).

`-source` Dumps the source of the first file to `STDOUT` and exit.

`-stack_dump` Disables `SIGINT` cleanup handler (off).

`-startfile_ok` Allows non-HTTP start file and home page with `-validate` (off).

`-tagsoup` Uses `TagSoup` rather than `SortaSGML` parser (off).

`-telnet` Disables telnets (off).

`-term=TERM` Sets terminal type to `TERM`.

`-tlog` Toggles use of a `lynx` Trace Log for the current session.

`-tna` Turns on Textfields Need Activation mode (off).

`-trace` Turns on `lynx` trace mode.

-traversal	Traverses all HTTP links derived from start file.
-trim_input-fields	Trims input text/textarea fields in forms (off).
-underscore	Toggles use of _underline_ format in dumps (off).
-use_mouse	Turns on mouse support (off).
-useragent=<Name>	Sets alternative Lynx User-Agent header.
-validate	Accepts only HTTP URLs (meant for validation); implies more restrictions than -anonymous,but goto is allowed for HTTP and HTTPS.
-verbose	Toggles [LINK], [IMAGE], and [INLINE] comments with filenames of these images (on).
-version	Prints lynx version information.
-vikeys	Enables vi-like key movement (off).
-width=<Number>	Screen width for formatting of dumps (default is 80).
-with_backspaces	Omits backspaces in output if -dumping or -crawling (like man does) (off).

mail

mail	Sends and receives mail.

```
mail [-iInv] [-s <subject>] [-c <cc-addr>] [-b <bcc-addr>] <to-addr>...
mail [-iInNv] -f [<name>]
mail [-iInNv] [-u <user>]
mail
```

-i	Ignores tty interrupt signals. Especially useful for communication on noisy phone lines.
-I	Forces interactive mode, even when input isn't a terminal. Particularly useful for using the ~ character, which is only available in interactive mode.
-n	Ignores /etc/mail.rc upon startup.
-v	Verbose mode.
-s <subject>	Specifies the subject. Uses only the first argument after the flag. Be certain to use quotes for any subjects with spaces.
-c <cc-addr>	Sends a carbon copy to the users specified in <cc-addr>.
-b <bcc-addr>	Sends a blind copy to the users specified in <bcc-addr>. The list should be a comma-separated list.
-f [<name>]	Reads the contents of your mbox or the file specified by <name>. When you quit, mail writes undeleted messages back to this file.
-u <user>	Equivalent to -f /usr/mail/<user>.

Here are some of the useful options available within mail:

-<n>	Displays the previous message, if <n> isn't specified; otherwise, displays the <n>th previous message.
?	Displays a brief summary of commands.

`help`	Same as `?`.
`^D`	Sends the composed message.
`!<`*`shell_command`*`>`	Executes the shell command that follows.
<return> *n* `+`	Goes to the next message in sequence.
`Reply` `R`	Replies to the sender of the message. Doesn't reply to any other recipients of the message.
`reply` `r`	Replies to the sender and all other recipients of the message.
`respond`	Same as `reply`.

`mail <`*`user`*`>`
`m`
	Sends mail to the <*user*> specified. Takes login names and distribution group names as arguments.

`delete` `d`	Takes as its argument a list of messages and marks them to be deleted. Messages marked for deletion aren't available for most other commands.
`dp` `dt`	Deletes the current message and prints the next message.
`undelete` `u`	Takes a message list as its argument and unmarks the messages for deletion.
`edit` `e`	Takes as its argument a list of messages and points a text editor at each one in turn.
`inc`	Checks for any new incoming messages that have arrived since the session began and adds those to the message list.
`save` `s`	Takes as its argument a list of messages and a filename and saves the messages to the filename. Each message is appended to the file. If no message is given, saves the current message.
`write` `w`	Similar to `save`, except saves only the body of messages.
`unread` `U`	Takes as its argument a list of messages and marks them as not read.
`alias` `a`	With no arguments, prints out the list of currently defined aliases. With one argument, prints out the specified alias. With multiple arguments, creates a new alias or edits an old one.
`unalias`	Takes as its argument a list of names defined by alias commands and discards the remembered groups of users.

exit	
ex	
x	Exits mail without making any changes to the user's mbox, system mailbox, or the -f file that was being read.
xit	Same as exit.
quit	
q	Terminates the session, saving all undeleted messages in the user's mbox.

man

man	Formats and displays online manual pages.
man [-adfhktw] [*<section>*] [-M *<path>*] [-P *<pager>*] [-S *<list>*] [-m *<machine>*] [-p *<string>*] *<name1>* *<name2>* ...	
-a	Displays all the manual pages for a specified section and name combination. (The default is to display only the first page found.)
-d	Displays debugging information, rather than manual pages.
-f *<keyword>*	Displays a list of manual pages that contain complete word matches to the *<keyword>*. Same as whatis.
-h	Displays the help for man.
-k *<keyword>*	Displays a list of manual pages that contain the *<keyword>*. Same as apropos.
-o	Looks for original, non-localized man pages only.
-t	Uses troff to format the manual pages and outputs to stdout. The troff output can then be passed to a filter before being printed.
-w	Lists the pathnames of manual pages that would be displayed for the specified section and name combination.
-M *<path>*	Overrides the list of standard directories where man searches for manual pages. The path specified must be a colon-separated list of directories. The search path can also be specified by the MANPATH environment variable.
-P *<pager>*	Uses the specified *<pager>* to display the manual pages.
-S *<list>*	Searches the specified colon-separated section *<list>*. Overrides MANSECT environment variable.
-m *<machine>*	Searches for alternate architecture man pages.
-p *<string>*	Specifies the sequence of preprocessors to run before nroff or troff. Some of the preprocessors and the letters used to designate them are eqn (e), grap (g), pic (p), tbl (t), vgrind (v), refer (r). Overrides the MANROFFSEQ environment variable.
The optional *<section>* argument restricts man's search to the specified section.	

mkdir

`mkdir` Makes directories.

`mkdir [-p] [-m <mode>] <dir1> <dir2> ...`

`mkdir` creates the named directories in the order specified, using mode `rwxrwxrwx (0777)` as modified by the current `umask (2)`.

The user must have write permission in the parent directory.

`-p` Creates all nonexistent parent directories first. If this option isn’t specified, the full path prefix of each operand must already exist. Intermediate directories are created with permission bits `rwxrwxrwx (0777)` as modified by the current `umask (2)`, plus write and execute permission for the owner. For example, if you’re creating the directory `/Users/jray/Images/Vacation` and the `Images` directory doesn’t already exist, the `-p` option will automatically force its creation.

`-m <mode>` Sets the permission bits of the created directory to `<mode>`. `<mode>` can be in any formats specified to the `chmod (1)` utility. If a symbolic mode is specified, the operation characters `+` and `-` are interpreted relative to an initial mode of `a=rwx`.

more

`more` Pages through data or text files.

`more [-cdflsu] [-n] [+<linenumber>] [+/<pattern>] <file1> <file2> ...`

`more` pages through data a screenful at a time. When the user presses a return at the `More` prompt at the bottom of the screen, one more line is displayed. When the user presses the spacebar, another screenful of data is displayed. When `more` is invoked as `page`, each screenful is cleared before the next is displayed.

`-c` Draws each page by beginning at the top of the screen and erasing each line just before it draws on it. This option is ignored if the screen is unable to clear to the end of a line.

`-d` Prompts user with `Press space to continue, 'q' to quit.` at the end of each screenful. Responds to illegal user input with `Press 'h' for instructions.` instead of ringing the bell.

`-f` Counts logical rather than screen lines. Long lines aren’t folded. Useful when trying to display lines containing nonprinting characters or escape sequences.

`-l` Doesn’t treat `^L` (form feed) as a page break. Where form feeds occur, `more` pauses after them, as if the screen were full. Particularly recommended if piping `nroff` output through `ul`.

`-s` Squeezes multiple blank lines of output into one blank line of output. Useful for viewing `nroff` output.

-u	Suppresses underlining or stand-out mode, whichever the terminal is capable of displaying.
-n	Specifies the number of lines to use per screenful rather than the default.
+<*linenumber*>	Starts at <*linenumber*>.
+/<*pattern*>	Starts two lines before the line containing the regular expression pattern <*pattern*>.

Additional options for interacting with more when it pauses (*i* is an optional integer argument, defaulting to 1):

i<*return*>	Displays *i* more lines. Advances one line, if *i* isn't given.
i<*space*>	Displays *i* more lines. Advances another screenful if *i* isn't given.
^D	Displays 11 more lines. If *i* is given, scroll size is set to *i*.
d	Same as ^D.
*i*z	Same as typing <*space*>, except that if *i* is given, scroll size becomes *i*.
*i*s	Skips *i* lines and prints a screenful of lines.
*i*f	Skips *i* screenfuls and prints a screenful of lines.
i^F	Same as *i*f.
*i*b	Skips back *i* screenfuls and prints a screenful of lines.
i^B	Same as *i*b.
q	Exits.
Q	Exits.
=	Displays the current line number.
v	Starts the editor at the current line number, if the environment variable EDITOR is set to vi or ex. If no EDITOR is specified, vi is the default.
h	Displays the help menu.
i/<*expression*>	Searches for the *i*th occurrence of the regular expression <*expression*>. If the input is a file rather than a pipe, and there are fewer than i occurrences, the file remains unchanged. Otherwise, the display advances to two lines before the line containing <*expression*>.
*i*n	Searches for the *i*th occurrence of the last regular expression entered.
' (Single quote)	Goes to the point where the last search was started. If no search has been done on the file, it goes back to the beginning of the file.
!<*command*>	Invokes a shell that executes <*command*>. The characters % and !, when used in the <*command*>, are replaced with the current filename and the previous shell command, respectively. If there's no current filename, % isn't expanded. To escape expansion, use \% and \%, respectively.
i:n	Skips to the *i*th next file given in the command line, or to the last file if *i* is beyond range.

`i``:p`	Skips to the *`i`*th previous file in the command line, or to the first file if *`i`* is beyond range. If `more` is in the middle of displaying a file, it goes to the beginning of the file. If `more` is displaying from a pipe, the bell rings.
`:f`	Displays current filename and line number.
`:q`	Exits.
`:Q`	Exits.
`.` (Dot)	Repeats the previous command.

mount

`mount`	Mounts file systems.

```
mount
mount [-adfruvw] [-t ufs | lfs | <external_type>]
mount [-dfruvw] <special> | <node>
mount [-dfruvw] [-o <options>] [-t ufs | lfs | <external_type>] <special> | <node>
```

`mount` invokes a file system-specific program to prepare and graft the <*`special`*> device or remote node (`rhost:path`) on the file system tree at the point <*`node`*>. If neither <*`special`*> nor <*`node`*> is specified, the appropriate information is taken from the `fstab` file.

The system maintains a list of currently mounted file systems. If no arguments are given to `mount`, this list is displayed.

`-a`	All the file systems described in `fstab`(5) are mounted. Exceptions are those marked as `noauto` or are excluded by the `–t` flag.
`-d`	Causes everything to be done except for the actual system call. Useful in conjunction with the `-v` option to determine what the `mount` command is trying to do.
`-f`	Forces the revocation of write access when trying to downgrade a file system `mount` status from read-write to read-only.
`-r`	Mounts the file system read-only (even `root` may not write to it). The same as the `rdonly` option to the `-o` option.
`-u`	Indicates that the status of an already mounted file system should be changed. Any of the options available in `-o` may be changed. The file system may be changed from read-only to read-write, or vice versa. An attempt to change from read-write to read-only fails if any files on the file system are currently open for writing unless `-f` is also specified.
`-v`	Enables verbose mode.
`-w`	Sets the file system object to read-write.

`-t ufs | lfs |<external_type>` Specifies a file system type. Default is type `ufs`. The option can also be used to indicate that the actions should be performed only on the specified file system type. More than one type may be specified in a comma-separated list. The prefix `no` added to the type list may be used to specify that the actions shouldn't take place on a given type. For example, `mount -a -t nonfs,mfs` indicates that all file systems should be mounted except those of type NFS and MFS. `mount` attempts to execute a program called `mount_XXX` where `XXX` is the specified typename.

`-o` Specifies certain options. The options are specified in a comma-separated list.

The following options are available for the `-o` option:

`async` Specifies that all I/O to the file system should be done asynchronously. This is a dangerous flag to set, and shouldn't be used unless you're prepared to re-create the file system if the system crashes.

`force` Same as `-f`. Forces the revocation of write access when trying to downgrade a file system `mount` status from read-write to read-only.

`noauto` Skips this file system when `mount` is run with the `-a` flag.

`nodev` Doesn't interpret character or block special devices on the file system. The option is useful for a server that has file systems containing special devices for architectures other than its own.

`noexec` Doesn't allow the execution of any binaries on the mounted file system. This option is useful for a server containing binaries for an architecture other than its own.

`nosuid` Doesn't allow `set-user-identifier` or `set-group-identifier` bits to take effect.

`rdonly` Same as `-r`. Mounts the file system read-only. Even `root` may not write to it.

`sync` Specifies that all I/O to the file system should be done synchronously.

`update` Same as `-u`. Indicates that the status of an already mounted file system should be changed.

`union` Causes the namespace at the `mount` point to appear as the union of the mounted file system root and the existing directory. Lookups are done on the mounted file system first. If operations fail due to a nonexistent file, the underlying file system is accessed instead. All creates are done in the mounted file system.

Any additional options specific to a given file system type may be passed as a comma-separated list. The options are distinguished by a leading `-`. Options that take a value have the syntax `-<option>=<value>`.

mountd

mountd — Services remote NFS mount requests.

/sbin/mountd [-nr] [<*exportsfile*>]

mountd is the server for NFS mount requests from other client machines. mountd listens for service requests at the port indicated in the NFS server specification.

-n — Doesn't require that clients make mount requests from reserved ports (allows non-root mount requests to be served). Normally only mount requests from reserved ports are accepted. This option should be specified only if there are clients, such as PCs, that need it. The use of -n is strongly discouraged because it opens a wide variety of security problems.

-r — Allows mount RPCs requests for regular files to be served. Although this seems to violate the mount protocol specification, some diskless workstations do mount requests for their swapfiles and expect them to be regular files. Because a regular file cannot be specified in /etc/exports, the entire file system in which the swapfiles resides will have to be exported with the -alldirs flag.

<*exportsfile*> — Specifies an alternative location for the exports file.

mount_nfs

mount_nfs — Mounts NFS file systems.

mount_nfs [-23KPTUbcdilqs] [-D <*deadthresh*>] [-I <*readdirsize*>] [-L <*leaseterm*>] [-R <*retrycnt*>] [-a <*maxreadahead*>] [-g <*maxgroups*>] [-m <*realm*>] [-o <*options*>] [-r <*readsize*>] [-t <*timeout*>] [-w <*writesize*>] [-x <*retrans*>] <*rhost*>:<*path*> <*node*>

-2 — Uses NFS Version 2 protocol.

-3 — Uses NFS Version 3 protocol. Default is to try version 3 first and fall back to version 2 if the mount fails.

-K — Passes Kerberos authentication to the server for client-to-server user-credential mapping. This requires that the kernel be built with the NFSKERB option.

-P — The kernel uses a reserved port number to communicate with clients. This option is ignored and exists for compatibility with older systems.

-T — Uses TCP transport instead of UDP. This is recommended for servers that aren't on the same LAN cable as the client. This isn't supported by most non-BSD servers.

-U — Forces the mount protocol to use UDP transport, even for TCP NFS mounts. Necessary for some old BSD servers.

-b — Backgrounds the mount. If a mount fails, forks a child process that keeps trying the mount in the background. This option is useful for a file system that isn't critical to multiuser operation.

-c
Doesn't do a connect (2) for UDP mounts. This must be used for servers that don't reply to requests from the standard NFS port number 2049. It may also be required for servers with more than one IP address, if replies come from an address other than the one specified in the mount request.

-d
Turns off the dynamic retransmit timeout estimator. This may be useful for UDP mounts that exhibit high retry rates; it's possible for the dynamically estimated timeout to be too short.

-i
Makes the mount interruptible. The file system calls that are delayed due to an unresponsive server fail with EINTR when a termination signal is posted for the process.

-l
Used with NQNFS and NFSV3 to specify that the ReaddirPlus RPC should be used. This option reduces RPC traffic for cases such as ls -l, but floods the attribute and name caches with preferred entries. Probably most useful for client-to-server network interconnects with a large bandwidth * delay product.

-q
Uses the leasing extensions to NFSV3 to maintain cache consistency. This protocol version 2 revision to Not Quite NFS (NQNFS) is only supported by this updated release of NFS code. It isn't backward compatible to the version 1 NQNFS protocol that was part of the first release of 4.4 BSD-Lite.

-s
Soft mount. File system calls fail after retry round trip timeout intervals.

-D <*deadthresh*>
Used with NQNFS to set the dead server threshold to <*deadthresh*> number of round trip timeout intervals. After <*deadthresh*> retransmit timeouts, cached data for the unresponsive server is assumed to still be valid. Values may be set in the range of 1–9, with 9 being an infinite dead threshold that never assumes cached data is still valid. This option isn't generally recommended and is still experimental.

-I <*readdirsize*>
Sets the readdir read size to <*readdirsize*>. The value should normally be a multiple of DIRBLKSIZ that is <= the read size for the mount.

-L <*leaseterm*>
Used with NQNFS to set the lease term to <*leaseterm*> seconds. Only use this option for mounts with a large round-trip delay. Values are normally in the 10–30 seconds range.

-R <*retrycnt*>
Sets the retry count for doing the mount to <*retrycnt*>.

-a <*maxreadahead*>
Sets the read-ahead count to <*maxreadahead*>. This value may be in the 0–4 range, and determines how many blocks are read ahead when a large file is being read sequentially. A value larger than 1 is suggested for mounts with a large bandwidth * delay product.

`-g <maxgroups>`	Sets the maximum size of the group list for the credentials to `<maxgroups>`. This should be used for `mounts` on old servers that cannot handle a group list size of 16, as specified in RFC 1057. Try `8` if users in a log of groups cannot get a response from the `mount` point.
`-m <realm>`	Sets the Kerberos `real` to the string argument `<realm>`. Used with the `-K` option for `mounts` to other realms.
`-o <options>`	Options are specified as a comma-separated list of options. See `mount` (8) for a listing of the available options.
`-r <readsize>`	Sets the `read` data size to `<readsize>`. It should normally be a power of 2 >= 1024. This should be used for UDP mounts when the fragments dropped due to timeout value are getting large while actively using a `mount` point. Use `netstat` (1) `-s` to get the fragments dropped due to timeout value. See the `-w` option.
`-t <timeout>`	Sets the initial retransmit timeout to `<timeout>`. May be useful for fine-tuning UDP mounts over networks with high packet loss rates or an overloaded server. Try increasing the interval if `nfsstat` (1) shows high retransmit rates while the file system is active or reducing the value if there's a low retransmit rate but long response delay observed. Normally the `-d` option is also used when using this option to fine-tune the timeout interval.
`-w <writesize>`	Sets the write data size to `<writesize>`. See comments regarding the `-r` option, but using the fragments dropped due to timeout value on the server rather than the client. The `-r` and `-w` options should only be used as a last resort to improve performance when mounting servers that don't support TCP `mounts`.
`-x <retrans>`	Sets the retransmit timeout count for soft `mounts` to `<retrans>`.

mv

`mv`	Moves files.

`mv [-fi] <source> <target>`

`mv [-fi] <source1> <source2> <source3> ... <directory>`

In the first form, `mv` renames `<source>` to the name provided by `<target>`. If `<source>` is a file, a file is renamed. Likewise, if `<source>` is a directory, a directory is renamed.

In the second form, `mv` moves the list enumerated by `<source1> <source2> <source3>` ... to the directory named by `<directory>`.

`-f`	Forces an existing file to be overwritten.
`-i`	`-`Invokes an interactive mode that prompts for a confirmation before overwriting an existing file.

The last of any `-f` or `-i` options determines the behavior of `mv`.

natd

`natd` Network address translation daemon.

```
natd [-unregistered_only | -u] [-log | -l] [-proxy_only] [-reverse] [-deny_incoming |
-d] [-use_sockets | -s] [-same_ports | -m] [-verbose | -v] [-dynamic] [-in_port | -i
<port>] [-out_port | -o <port>] [-port | -p <port>] [-alias_address | -a <address>]
[-target_address | -t <address>] [-interface | -n <interface>] [-proxy_rule
<proxyspec>]  [-redirect_port <linkspec>] [-redirect_proto <linkspec>]
[-redirect_address <linkspec>] [-config | -f <configfile>] [-log_denied] [-log_facility
<facility_name>] [-punch_fw <firewall_range>]
```

This program provides a network address translation facility for use with divert(4) sockets under FreeBSD. It's intended for use with NICs—if you want to do NAT on a PPP link, use the `-nat` switch to `ppp`(8).

The `natd` normally runs in the background as a daemon. It's passed raw IP packets as they travel into and out of the machine, and will possibly change these before re-injecting them back into the IP packet stream.

It changes all packets destined for another host so that their source IP number is that of the current machine. For each packet changed in this manner, an internal table entry is created to record this fact. The source port number is also changed to indicate the table entry applying to the packet. Packets that are received with a target IP of the current host are checked against this internal table. If an entry is found, it's used to determine the correct target IP number and port to place in the packet.

`-l`
`-log` Logs various aliasing statistics and information to the file `/var/log/alias.log`. This file is truncated each time natd is started.

`-d`
`-deny_incoming` Rejects packets destined for the current IP number that have no entry in the internal translation table.

`-s`
`-use_sockets` Allocates a `socket`(2) to establish an FTP data or IRC DCC send connection. This option uses more system resources, but guarantees successful connections when port numbers conflict.

`-m`
`-same_ports` Tries to keep the same port number when allocating outgoing packets. With this option, protocols such as RPC will have a better chance of working. If it isn't possible to maintain the port number, it's silently changed as per normal.

`-v`
`-verbose` Doesn't call `fork`(2) or `daemon`(3) on startup. Instead, it stays attached to the controlling terminal and displays all packet alterations to the standard output. This option should be used only for debugging.

`-u`
`-unregistered_only` Only alters outgoing packets with an unregistered source address. According to RFC 1918, unregistered source addresses are `10.0.0.0/8`, `176.16.0.0/12`, and `192.168.0.0/16`.

`-log_denied` Logs denied incoming packets via syslog (*see also* `log_facility`).

`-log_facility` Uses specified log facility when logging information via syslog.

`<`*`facility_name`*`>` Facility names as in `syslog.conf`(5).

`-dynamic`

If the `-n` or `-interface` option is used, `natd` monitors the routing socket for alterations to the `<interface>` passed. If the interface IP number is changed, `natd` dynamically alters its concept of the alias address.

`-i <inport>`
`-in_port <inport>`

Reads from and writes to `<inport>`, treating all packets as packets coming into the machine.

`-o <output>`
`-out_port <outport>`

Reads from and writes to `<outport>`, treating all packets as packets going out of the machine.

`-p <port>`
`-port <port>`

Reads from and writes to `<port>`, distinguishing packets as incoming or outgoing using the rules specified in divert. If `<port>` isn't numeric, it's searched for in the `/etc/services` database. If this flag isn't specified, the divert port named `natd` is used as a default.

`-a <address>`
`-alias_address <address>`

Uses `<address>` as the alias address. If this option is not specified, the `-n` or `-interface` option must be used. The specified address should be the address assigned to the public-network interface.

All data passing out is rewritten with a source address equal to address. All data coming in is checked to see whether it matches any already-aliased outgoing connection. If it does, the packet is altered accordingly. If not, all `-redirect_port`, `-redirect_proto` and `-redirect_address` assignments are checked and actioned. If no other action can be made and if `-deny_incoming` isn't specified, the packet is delivered to the local machine using the rules specified in `-target_address` option below.

`-t <address>`
`-target_address <address>`

Sets the target address. When an incoming packet not associated with any pre-existing link arrives at the host machine, it is sent to the specified address.

The target address may be set to `255.255.255.255`, in which case all new incoming packets go to the alias address set by `-alias_address` or `-interface`.

If this option isn't used, or called with the argument `0.0.0.0`, all new incoming specified in the packet. This allows external machines to talk directly to internal machines if they can route packets to the machine in question.

`-n <interface>`
`-interface <interface>`

Uses `<interface>` to determine the alias address. If there is a possibility that the IP number associated with `<interface>` might change, the `-dynamic flag` should also be used. If this option isn't specified, the `-a` or `-alias_address` flag must be used. The specified `<interface>` must be the public network interface.

`-f <configfile>`
`-config <configfile>`
	Reads the configuration from `<configfile>`. `<configfile>` contains a list of options, one per line, in the same form as the long form of the command-line flags. For example, the line
	`alias_address 158.152.17.1`
	specifies an alias address of `158.152.17.1`. Options that don't take an argument are specified with an option of yes or no in the configuration file. For example, the line
	`log yes`
	is synonymous with `-log`. Empty lines and lines beginning with `#` are ignored.

`-redirect_port <proto>`
`<targetIP>:<targetPORT>`
`[<aliasIP>:]<aliasPORT>`
`[<remoteIP>[:<remotePORT>]`
	Redirects incoming connections arriving to given port to another host and port. `<proto>` is either tcp or udp; `<targetIP>` is the desired target IP number; `<targetPORT>` is the desired target port number; `<aliasPORT>` is the requested port number and `<aliasIP>` if the aliasing address. `<remoteIP>` and `<remotePORT>` can be used to specify the connection more accurately, if necessary. For example, the argument
	`tcp inside1:telnet 6666`
	means that TCP packets destined for port 6666 on this machine are sent to the telnet port on the inside1 machine.

`redirect_proto <proto>`
`<localIP> [<publicIP>`
`[<remoteIP>]]`
	Redirects incoming IP packets of protocol proto (see `protocols`(5)) destined for `publicIP` address to a `localIP` address and vice versa.
	
	If `publicIP` isn't specified, the default aliasing address is used. If `remoteIP` is specified, only packets coming from/to `remoteIP` will match the rule.

`redirect_address <localIP>`
`<publicIP>`
	Redirects traffic for public IP address to a machine on the local network. This function, known as static NAT, is normally useful if your ISP has allocated a small block of IP addresses to you, but it can be used in the case of a single address:
	`redirect_address 10.0.0.8 0.0.0.0`
	The previous command would redirect incoming traffic to machine `10.0.0.8`.
	If several address aliases specify the same public address as follows:
	`redirect_address 192.168.0.2 <public_addr>`
	`redirect_address 192.168.0.3 <public_addr>`
	`redirect_address 192.168.0.4 <public_addr>`
	The incoming traffic is directed to the last translated local address (`192.168.0.4`), but outgoing traffic to the first two addresses is aliased to the specified public address.

`-redirect_port proto targetIP:targetPORT[,targetIP:targetPORT[,...]] [aliasIP:]aliasPORT [remoteIP[:remotePORT]]`
`-redirect_address localIP[,localIP[,...]] publicIP`

These forms of `-redirect_port` and `-redirect_address` are used to transparently offload network load on a single server and distribute the load across a pool of servers. This function is known as `LSNAT` (RFC 2391). For example, the argument
`tcp www1:http,www2:http,www3:http www:http`
means that incoming HTTP requests for host `www` will be transparently redirected to one of the `www1`, `www2` or `www3`, where a host is selected simply on a round-robin basis, without regard to load on the net.

`-reverse`
Reverses operation of `natd`. This can be useful in some transparent proxying situations when outgoing traffic is redirected to the local machine and `natd` is running on the incoming interface (it usually runs on the outgoing interface).

`-proxy_only`
Forces `natd` to perform transparent proxying only. Normal address translation isn't performed.

`-proxy_rule [<`*`type`*`> encode_ip_hdr | encode_tcp_stream] port <`*`xxxx`*`> server <`*`a.b.c.d:yyyy`*`>`
Enables transparent proxying. Packets with the given port going through this host to any other host are redirected to the given server and port. Optionally, the original target address can be encoded into the packet. Use `encode_ip_header` to put this information into the IP `option` field or `encode_tcp_stream` to inject the data into the beginning of the TCP stream.

`-punch_fw <`*`basenumber`*`>:<`*`count`*`>`
This option directs `natd` to punch holes in an `ipfirewall`(4)-based firewall for FTP/IRC DCC connections. This is done dynamically by installing temporary firewall rules that allow a particular connection (and only that connection) to go through the firewall. The rules are removed when the corresponding connection terminates.
A maximum of `<`*`count`*`>` rules starting from the rule number `<`*`basenumber`*`>` will be used for punching firewall holes. The range will be cleared for all rules on startup.

netinfo

`netinfo` Network administrative information.

NetInfo is a hierarchical database of administrative information. The hierarchy is composed of directories. Each directory may have zero or more properties associated with it. Each property has a name and zero or more values.

NOTE

The information in this man page is pulled from a number of sources and we haven't been able to verify all of it.

Searching

Almost everything that uses NetInfo for lookups searches the local domain first. If the answer isn't found in the local domain, the next domain level is searched, and so on.

Database Format

The top level of the database, the `root` directory, contains a single property called `master`. This property indicates which server is the master of the database; that is, which server contains the master copy of the database.

A second property can be installed in the `root` directory to limit who can connect to the domain. By default, everyone can connect to the domain. They can read anything there, but not write. If this default is undesired, the property called `trusted_networks` can be enabled. Values for it should be the network or subnet addresses that are assumed to contain trusted machines. A name may be given instead of an address. If a name is given, that name should be listed as a subdirectory of `/networks` in the same domain and resolve to the appropriate network address.

At the second level, the following directories exist:

1	`users`
2	`groups`
3	`machines`
4	`networks`
5	`protocols`
6	`rpcs`
7	`services`
8	`aliases`
9	`mounts`
10	`printers`
55	`config`
58	`afpuser_aliases`
65	`exports`

These directories mostly contain a single property called `name`.

The directory `machines` may contain these properties having to do with automatic host installation in addition to `name`:

`promiscuous` If it exists, the `bootpd` (8) daemon is promiscuous. Has no value.

`assignable_ipaddr` A range of IP addresses to be automatically assigned, specified with two values as endpoints.

`configuration_ipaddr` Temporary IP address given to unknown machines in the process of booting.

`default_bootfile` Default bootfile to assign to a new machine.

`net_passwd` Optional property. Encrypted password for protecting automatic host installations.

The directory `/aliases` contains directories describing individual mailing addresses. The relevant properties of each directory under `aliases` are as follows:

`name` Name of the alias.

`members` List of members belonging to the alias.

The directory `/groups` contains directories that refer to individual system groups. The relevant properties of each directory under `groups` are as follows:

`name` Name of the system group.

`passwd` Password of the group.

`gid` Associated group ID.

`users` List of users belonging to the system group.

The directory `/machines` contains directories that refer to individual machines. The relevant properties of each directory under `machines` are as follows:

`name` Name of the machine. This property can have multiple values if the machine name has aliases.

`ip_address` IP address of the machine. This property can have multiple values if the machine has multiple IP addresses. This address must be stored in decimal-dot notation, with no leading zeroes.

`en_address` Ethernet address of the machine. The address must be stored in standard six-field hex ethernet notation, with no leading zeroes.

`serves` List of information about the NetInfo domain that the machine serves. Each value in the list has the format <*domain_name*>/<*domain_tag*>. The <*domain_name*> is the external domain name that the machine serves as seen by this level of the hierarchy. The <*domain_tag*> is the internal name associated with the actual process on the machine serving the domain.

`bootfile` Name of the kernel that this machine will use when `NetBooting`.

`bootparams` List of `Bootparams` protocol key-value pairs. For example, `root=parrish:/` has the `Bootparams` key `root` and the `Bootparams` value `parrish:/`.

`netgroups` List of netgroups to which the machine belongs.

The directory `/mounts` contains directories that refer to file systems. The relevant properties of each directory under `mounts` are as follows:

`name` Name of the file system. For example, `/dev/od00a` or `papazian:/`.

`dir` Name of the directory upon which the file system is mounted.

`type` File system type of the `mount`.

`opts` List of `mount` (8) options associated with the mounting of the file system.

`passno` Pass number on parallel `fsck` (8).

`freq` Dump frequency, in days.

The directory `/networks` contains directories that refer to Internet networks. The relevant properties of each directory under `networks` are as follows:

`name` Name of the network. If the network has aliases, there may be more than one value for this property.

`address` Network number of this address. This value must be in decimal-dot notation, with no leading zeroes.

The directory /printers contains directories that refer to printer entries. The relevant properties of each directory under printers are as follows:

name	Name of the printer. If the printer has aliases, this property will have multiple values.
lp, sd, and so on	Printcap (5) properties associated with the printer.

The directory /protocols contains directories that refer to transport protocols. The relevant properties of each directory under protocols are as follows:

name	Name of the protocol.
number	Associated protocol number.

The directory /services contains directories that refer to ARPA services. The relevant properties of each directory under services are as follows:

name	Name of the service. If the service has aliases, the property will have multiple values.
protocol	Name of the protocol on which the service runs. If the service runs on multiple protocols, the property will have multiple values.
port	Associated port number of the service.

The directory /users contains information that refers to users. The relevant properties of each directory under users are as follows:

name	Login name of the user.
passwd	Encrypted password of the user.
uid	User ID of the user.
gid	Default group ID of the user.
realname	Real name of the user.
home	Home directory of the user.
shell	Login shell of the user.
sharedDir	User's publicly readable directory.
picture	Image to show for user in graphical login.
hint	User's password hint.

netstat

netstat	Shows network status.

```
netstat [-AaLlnW] [-f <address_family> | -p <protocol>] [-M <core>] [-N <system>]
netstat [-gilns] [-f <address_family>] [-M <core>] [-N <system>]
netstat -i | -I <interface> [-w <wait>] [-abdgt] [-M <core>] [-N <system>]
netstat -s [-s] [-f <address_family> | -p <protocol>] [-M core] [-N <system>]
netstat -i | -I interface -s [-f <address_family> | -p <protocol>] [-M <core>] [-N
<system>]
netstat -m [-M <core>] [-N <system>]
netstat -r [-Aaln] [-f <address_family>] [-M <core>] [-N <system>]
netstat -rs [-s] [-M <core>] [-N <system>]
```

The `netstat` command symbolically displays the contents of various network-related data structures. There are a number of output formats, depending on the options for the information presented. The following forms (respectively by their order above) are available:

Displays a list of active sockets for each protocol.

Presents the contents of one of the other network data structures according to the option selected.

With a wait interval specified, `netstat` will continuously display the information regarding packet traffic on the configured network interfaces.

Displays statistics for the specified protocol or address family.

Displays per-interface statistics for the specified protocol or address family.

Displays `mbuf(9)` statistics.

Displays routing table for the specified address family.

Displays routing statistics.

`-A` With the default display, shows the address of any protocol control blocks associated with sockets; used for debugging.

`-a` With the default display, shows the state of all sockets; normally sockets used by server processes aren't shown. With the routing table display (option `-r`, as described following), shows protocol-cloned routes (routes generated by a `RTF_PRCLONING` parent route); normally these routes aren't shown.

`-b` With the interface display (option `-i`, as described following), shows the number of bytes in and out.

`-d` With either interface display (option `-i` or an interval, as described following), shows the number of dropped packets.

`-f <address-family>` Limits statistics or address control block reports to those of the specified address family. The following address families are recognized: `inet`, for AF_INET, `inet6`, for AF_INET6 and `unix`, for AF_UNIX.

`-g` Shows information related to multicast (group address) routing. By default, show the IP Multicast virtual-interface and routing tables. If the `-s` option is also present, shows multicast routing statistics.

`-I <interface>` Shows information about the specified interface; used with a wait interval as described below. If the `-s` option is present, shows per-interface protocol statistics on the interface for the specified *`<address_family>`* or *`<protocol>`*, or for all protocol families.

`-i` Shows the state of interfaces that have been auto-configured (interfaces statically configured into a system, but not located at boot time aren't shown). If the `-a` option is also present, multicast addresses currently in use are shown for each Ethernet interface and for each IP interface address. Multicast addresses are shown on separate lines following the interface address with which they're associated. If the `-s` option is present, shows per-interface statistics on all interfaces for the specified *`<address_family>`* or *`<protocol>`*, or for all protocol families.

`-L`	Shows the size of the various listen queues. The first count shows the number of unaccepted connections. The second count shows the amount of unaccepted incomplete connections. The third count is the maximum number of queued connections.
`-l`	Prints full IPv6 address.
`-M`	Extracts values associated with the name list from the specified core instead of the default `/dev/kmem`.
`-m`	Show statistics recorded by the memory management routines (the network manages a private pool of memory buffers).
`-N`	Extracts the name list from the specified system instead of the default `/kernel`.
`-n`	Shows network addresses as numbers (normally `netstat` interprets addresses and attempts to display them symbolically). This option may be used with any of the display formats.
`-p <protocol>`	Shows statistics about *<protocol>*, which is either a well-known name for a protocol or an alias for it. Some protocol names and aliases are listed in the file `/etc/protocols`. The special protocol name `bdg` is used to show bridging statistics. A null response typically means that there are no interesting numbers to report. The program complains if *<protocol>* is unknown or if there's is no statistics routine for it.
`-r`	Shows the routing tables. Use with `-a` to show protocol-cloned routes. When `-s` is also present, show routing statistics instead. When `-l` is also present, `netstat` assumes more columns are there and the maximum transmission unit (mtu) are also displayed.
`-s`	Shows per-protocol statistics. If this option is repeated, counters with a value of zero are suppressed.
`-W`	In certain displays, avoid truncating addresses even if this causes some fields to overflow.
`-w <wait>`	Shows network interface statistics at intervals of *<wait>* seconds.

Output

The default display, for active sockets, shows the local and remote addresses, send and receive queue sizes (in bytes), protocol, and the internal state of the protocol. Address formats are of the form *<host>.<port>* or *<network>.<port>* if a socket's address specifies a network but no specific host address. If known, the host and network addresses are displayed symbolically according to the databases `/etc/hosts` and `/etc/networks`, respectively. If a symbolic name for an address is unknown, or if the `-n` option is specified, the address is printed numerically, according to the address family. Unspecified, or wildcard, addresses and ports appear as `*`.

Internet Domain Socket States:

`CLOSED`	The socket isn't in use.
`LISTEN`	The socket is listening for incoming connections. Unconnected listening sockets like these are only displayed when using the `-a` option.

SYN_SENT The socket is actively trying to establish a connection to a remote peer.

SYN_RCVD The socket has passively received a connection request from a remote peer.

ESTABLISHED The socket has an established connection between a local application and a remote peer.

CLOSE_WAIT The socket connection has been closed by the remote peer, and the system is waiting for the local application to close its half of the connection.

LAST_ACK The socket connection has been closed by the remote peer, the local application has closed its half of the connection, and the system is waiting for the remote peer to acknowledge the close.

FIN_WAIT_1 The socket connection has been closed by the local application, the remote peer hasn't yet acknowledged the close, and the system is waiting for it to close its half of the connection.

FIN_WAIT_2 The socket connection has been closed by the local application, the remote peer has acknowledged the close, and the system is waiting for it to close its half of the connection.

CLOSING The socket connection has been closed by the local application and the remote peer simultaneously, and the remote peer hasn't yet acknowledged the close attempt of the local application.

TIME_WAIT The socket connection has been closed by the local application, the remote peer has closed its half of the connection, and the system is waiting to be sure that the remote peer received the last acknowledgement.

The interface display provides a table of cumulative statistics regarding packets transferred, errors, and collisions. The network addresses of the interface and the maximum transmission unit (mtu) are also displayed.

The routing table display indicates the available routes and their status. Each route consists of a destination host or network and a gateway to use in forwarding packets. The flags field shows a collection of information about the route stored as binary choices. The individual flags are discussed in more detail in the `route`(8) and `route`(4) manual pages. The mapping between letters and flags is as follows:

1	RTF_PROTO1	Protocol-specific routing flag #1
2	RTF_PROTO2	Protocol-specific routing flag #2
3	RTF_PROTO3	Protocol-specific routing flag #3
B	RTF_BLACKHOLE	Just discard packets (during updates)
b	RTF_BROADCAST	The route represents a broadcast address
C	RTF_CLONING	Generate new routes on use
c	RTF_PRCLONING	Protocol-specified generate new routes on use
D	RTF_DYNAMIC	Created dynamically (by redirect)
G	RTF_GATEWAY	Destination requires forwarding by intermediary

H	RTF_HOST	Host entry (net otherwise)
L	RTF_LLINFO	Valid protocol to link address translation
M	RTF_MODIFIED	Modified dynamically (by redirect)
R	RTF_REJECT	Host or net unreachable
S	RTF_STATIC	Manually added
U	RTF_UP	Route usable
W	RTF_WASCLONED	Route was generated as a result of cloning
X	RTF_XRESOLVE	External daemon translates proto to link address

Direct routes are created for each interface attached to the local host; the `gateway` field for such entries shows the address of the outgoing interface. The `refcnt` field gives the current number of active uses of the route. Connection-oriented protocols normally hold on to a single route for the duration of a connection, whereas connectionless protocols obtain a route while sending to the same destination. The `use` field provides a count of the number of packets sent using that route. The `interface` entry indicates the network interface utilized for the route.

When `netstat` is invoked with the `-w` option and a wait interval argument, it displays a running count of statistics related to network interfaces. An obsolete version of this option used a numeric parameter with no option, and is currently supported for backward compatibility. By default, this display summarizes information for all interfaces. Information for a specific interface may be displayed with the `-I` option.

newfs

`newfs` Constructs a new file system.

`newfs [-NO] [-S <`*`sector-size`*`>] [-T <`*`disktype`*`>] [-a <`*`maxcontig`*`>] [-b <`*`blocksize`*`>] [-c <`*`cylinders`*`>] [-d <`*`rotdelay`*`>] [-e <`*`maxbpg`*`>] [-f <`*`frag-size`*`>] [-i <`*`bytes`*`>] [-k <`*`skew`*`>] [-l <`*`interleave`*`>] [-m <`*`free-space`*`>] [-n <`*`nrpos`*`>] [-o <`*`optimization`*`>] [-p <`*`sectors`*`>] [-r <`*`revolutions`*`>] [-s <`*`size`*`>] [-u <`*`sectors`*`>] [-x <`*`sectors`*`>] [-t <`*`tracks`*`>] <`*`special`*`>`

`newfs` replaces the more obtuse `mkfs`(8) program. Before running `newfs`, the disk must be labeled using `disklabel`. `newfs` builds a file system on the specified special device basing its defaults on the information in the disk label. Typically, the defaults are reasonable; however, `newfs` has numerous options to allow the defaults to be selectively overridden.

`-N` Causes the file system parameters to be printed out without really creating the file system.

`-O` Creates a 4.3BSD-format file system. This option is primarily used to build root file systems that can be understood by older boot ROMs.

`-T <`*`disktype`*`>` Uses information for the specified disk from `/etc/disktab` instead of trying to get the information from a disklabel.

`-a <`*`maxcontig`*`>` Specifies the maximum number of contiguous blocks that will be laid out before forcing a rotational delay (see the `-d` option). The default value is `8`. See `tunefs` (8) for more details on how to set this option.

`-b <blocksize>` Specifies the block size of the file system, in bytes.

`-c <#cylinders/group>` Specifies the number of cylinders per cylinder group in a file system. The default is 16.

`-d <rotdelay>` Specifies the expected time (in milliseconds) to service a transfer completion interrupt and initiate a new transfer on the same disk. The default is 0 milliseconds. See `tunefs` (8) for more details on how to set this option.

`-e <maxbpg>` Indicates the maximum number of blocks any single file can allocate out of a cylinder group before it's forced to begin allocating blocks from another cylinder group. The default is about one quarter of the total blocks in a cylinder group. See `tunefs` (8) for more details on how to set this option.

`-f <frag-size>` Specifies the fragment size of the file system in bytes.

`-i <number-of-bytes-per-inode>` Specifies the density of inodes in the file system. The default is to create an inode for each 4096 bytes of data space. If fewer inodes are desired, a larger number should be given.

`-m <free-space-%>` Specifies the percentage of space reserved from normal users; the minimum free space threshold. The default value used is 5%. See `tunefs` (8) for more details on how to set this option.

`-n <number-of-rotational-positions>` Specifies the number of distinct rotational positions. The default is `1`.

`-o <optimization-preference>` Space or time. The file system can either be instructed to try to minimize the time spent allocating blocks or try to minimize the space fragmentation on the disk. If the value of `minfree` is less than 5%, the default is to optimize for time. See `tunefs` for more details on how to set this option.

`-s <size>` Specifies the size of the file system in sectors.

The following options override the standard sizes of the disk geometry. Their default values are taken from the disk label. Changing these defaults is useful only when using `newfs` to build a file system whose raw image will eventually be used on a different type of disk than the one on which it's initially created (for example, a write-once disk). Note that changing any of these values from their defaults will make it impossible for `fsck` to find the alternate superblocks if the standard superblock is lost.

`-S <sector-size>` Specifies the size of a sector in bytes (almost never anything but 512).

`-k <sector 0 skew, per track>` Describes perturbations in the media format to compensate for a slow controller. Track skew is the offset of sector 0 on track N relative to sector 0 on track N–1 on the same cylinder.

`-l <hardware sector interleave>` Describes perturbations in the media format to compensate for a slow controller. `<interleave>` is a physical sector interleave on each track, specified as the denominator of the ratio:

```
(sectors read)/(sectors passed over)
```

Thus, an interleave of 1/1 implies contiguous layout, whereas 1/2 implies logical sector 0 is separated by one sector from logical sector 1.

`-p <spare sectors per track>`	Spare sectors (bad sector replacements) are physical sectors that occupy space at the end of each track. They aren't counted as part of the sectors/track (`-u`) because they aren't available to the file system for data allocation.
`-r <revolutions/minute>`	Specifies the speed of the disk in revolutions per minute.
`-t <#tracks/cylinder>`	Specifies the number of tracks/cylinder available for data allocation by the file system.
`-u <sectors/track>`	Specifies the number of sectors per track available for data collection by the file system. This doesn't include sectors reserved at the end of each track for bad block replacement (see the `-p` option).
`-x <spare sectors per cylinder>`	Spare sectors (bad sector replacements) are physical sectors that occupy space at the end of the last track in the cylinder. They are deducted from the sectors/track (`-u`) of the last track of each cylinder because they aren't available to the file system for data allocation.

nfsiod

`nfsiod` Local NFS asynchronous I/O server.

`nfsiod [-n <num-servers>]`

`nfsiod` runs on an NFS client machine to service asynchronous I/O requests to its server. It improves performance, but isn't required for correct operation.

`-n <num_servers>`	Specifies the number of servers to be started. A client should run enough daemons to handle its maximum level of concurrency, typically 4 to 6.

nfsd

`nfsd` Remote NFS server.

`nfsd [-rut] [-n <num-servers>]`

`nfsd` runs on a server machine to service NFS requests from client machines. At least one `nfsd` must be running for a machine to function as a server. By default, four servers for UDP transport are started. `nfsd` listens for service requests at the port indicated in the NFS server specification.

`-r`	Registers the NFS service with `portmap` (8) without creating any servers. This option can be used along with `-u` or `-t` to re-register NFS if the `portmap` server is restarted.
`-u`	Serves UDP NFS clients.
`-t`	Serves TCP NFS clients.
`-n <num_servers>`	Specifies the number of servers to start. A server should run enough daemons to handle the maximum level of concurrency from its clients, typically 4–6.

nidump

`nidump` Extracts text or flat-file-format data from NetInfo.

`nidump [-t] { -r <directory> | <format> } <domain>`

`nidump` reads the specified NetInfo domain and dumps a portion of its contents to standard output. When a flat-file administration format is specified, `nidump` provides output in the syntax of the corresponding flat file. Allowed values for `<format>` are `aliases`, `bootparams`, `bootptab`, `exports`, `fstab`, `group`, `hosts`, `networks`, `passwd`, `printcap`, `protocols`, `rpc`, and `services`.

If `-r` is used, the first argument is interpreted as a NetInfo directory path, and its contents are dumped in a generic NetInfo format.

`-t` Interprets the domain as a tagged name.

`-r` Dumps the specified directory in raw format. Directories are delimited in curly brackets. Properties within a directory are listed in the form `property = value;`. Parentheses introduce a comma-separated list of items. The special property name `CHILDREN` is used to hold a directory's children, if any. Spacing and line breaks are significant only within double quotes, which can be used to protect any names with meta characters.

niload

`niload` Populates NetInfo directories with multiple properties at once.

`niload [-v] [-d] [-p] [-t] {-r <directory> | <format>} <domain>`

`niload` loads information from standard output into the specified NetInfo `<domain>`. If `<format>` is specified, the input is interpreted according to the flat-file format `<format>`. Acceptable values for `<format>` are `aliases`, `bootparams`, `bootptab`, `exports`, `fstab`, `group`, `hosts`, `networks`, `passwd`, `printcap`, `protocols`, `rpc`, and `services`.

If `-r <directory>` is specified instead of a flat-file format, the input is interpreted as raw NetInfo data, as generated by `nidump -r`, and is loaded into `<directory>`.

`niload` overwrites entries in the existing directory with those contained in the input. Entries that are in the directory, but not in the input, aren't deleted unless `-d` is specified. `niload` must be run as the `superuser` on the master NetInfo server for `<domain>`, unless `-p` is specified.

`-v` Verbose mode. Prints + for each entry loaded, and `-` for each entry deleted (flat-file formats only).

`-d` Deletes entries that are in the directory, but not in the input.

`-p` Prompts for the `root` password of the given domain so that the command can be run from locations other than the master.

`-u <user>` Authenticates as `<user>`. Implies `-p`.

`-P <password>` Provides `<password>` on the command line. Overrides `-p`.

`-t` Interprets the domain as a tagged domain. For example, `trotter/network` refers to the `domain` network on the machine `trotter`. Machine name can be specified as an actual name or an IP address.

`-r`	Loads entries in raw format, as generated by `nidump -r`. The first argument should be the path of a NetInfo directory into which the information is loaded. The specified directory may be renamed as a result of contents of the input, particularly if the input includes a top-level `name` property. If the specified directory doesn't exist, it's created.
`<domain>`	NetInfo `<domain>` that's receiving input. If `.` is the value for `<domain>`, it's referring to the local NetInfo database.

niutil

`niutil`	The NetInfo Utility `niutil` is used to edit the NetInfo database.

```
niutil -create [opts] <domain> <path>
niutil -destroy [opts] <domain> <path>
niutil -createprop [opts] <domain> <path> <key> [<val>...]
niutil -appendprop [opts] <domain> <path> <key> <val>...
niutil -mergeprop [opts] <domain> <path> <key> <val>...
niutil -insertval [opts] <domain> <path> <key> <val> <index>
niutil -destroyprop [opts] <domain> <path> <key>
niutil -destroyval [opts] <domain> <path> <key> <val>
niutil -renameprop [opts] <domain> <path> <oldkey> <newkey>
niutil -read [opts] <domain> <path>
niutil -list [opts] <domain> <path>
niutil -rparent [opts] <domain>
niutil -resync [opts] <domain>
niutil -statistics [opts] <domain>
```

`niutil` enables you to perform arbitrary reads and writes on the specified NetInfo `<domain>`. To perform writes, `niutil` must be run as `root` on the NetInfo master for the database, unless `-p`, `-P`, or `-u` is specified. The directory specified by `<path>` is separated by `/` characters. A numeric ID may be used for a path in place of a string. Property names may be given in a path with an `=`. The default property name is `name`. The following examples refer to a user with user ID 3:

```
/name=users/uid=3
/users/uid=3
```

`-t <host>/<tag>`	Interprets the domain as a tagged domain. For example, `parrish/network` is the domain tagged network on machine `parrish`.
`-p`	Prompts for the `root` password or the password of `<user>` if combined with `-u`.
`-u <user>`	Authenticates as `<user>`. Implies `-p`.
`-P <password>`	Provides the `root` password or the password of `<user>` if combined with `-u`. Overrides `-p`.
`-T <seconds>`	Sets the read and write timeout to `<seconds>`. Default is 30 seconds.

Operations

`-create <domain> <path>` Creates a new directory with the specified path.

`-destroy <domain> <path>` Destroys the directory with the specified path.

`-createprop <domain> <path>` Creates a new property in the directory `<path>`. `<key>` is the name of the property. Zero or more property values `<key> [<val>...]` may be specified. If the named property already exists, it's overwritten.

`-appendprop <domain> <path>` Appends new values to an existing property in directory `<path>`. `<key>` is the name of the property. Zero or more property values `<key> <val>...` may be specified. If the named property doesn't exist, it's created.

`-mergeprop <domain> <path>` Merges new values into an existing property in the directory `<path>`. `<key>` is the name of the property. Zero or more `<key> <val>...` property values may be specified. The values are appended to the property only if they don't already exist. If the named property doesn't exist, it's created.

`-insertval <domain> <path>` Inserts a new value into an existing property in the directory `<path>` at position `<propindex>`. `<key>` is the name of the `<key> <val>` property. If the named property doesn't exist, it's created. `<propindex>`

`-destroyprop <domain> <path> <key>` Destroys the property with name `<key>` in the specified `<path>`.

`-destroyval <domain> <path> <key> <val>` Destroys the specified value in the property named `<key>` in the specified `<path>`.

`-renameprop <domain> <path> <oldkey> <newkey>` Renames the property with name `<oldkey>` in the specified `<path>`.

`-read <domain> <path>` Reads the properties associated with the directory `<path>` in the specified `<domain>`.

`-list <domain> <path>` Lists the directories in the specified `<domain>` and `<path>`. Directory IDs are listed along with directory names.

`-readprop <domain> <path><path> <key>` Reads the value of the property named `<key>` in the directory of the specified `<domain>`.

`-readval <domain> <path> <key> <index>` Reads the value at the given index of the named property in the specified directory.

`-rparent <domain>` Prints the current NetInfo parent of a server. The server should be explicitly given using the `-t <host>/<tag>` option.

`-resync <domain>` Resynchronizes NetInfo. If a domain name is given, the master resynchronizes all clones. If the `-t <clone>/<tag>` option is used instead, only that clone is resynchronized. Using `-t <master>/<tag>` resynchronizes the whole domain.

`-statistics <domain>` Prints server statistics on the specified `<domain>`.

`-domainname <domain>` Prints the domain name of the given domain.

`<domain>` A value of `.` for `<domain>` refers to the local NetInfo database.

nohup

`nohup` Invokes a command immune to hangups.

```
nohup <utility> [<arg> ...]
```

`nohup` invokes `<utility>` with its arguments and at this time sets the signal `SIGHUP` to be ignored. If the standard output is a terminal, the standard output is appended to the file `nohup.out` in the current directory. If standard error is a terminal, it's directed to the same place as the standard output.

The following variable is utilized by `nohup`:

`HOME`

If the output file `nohup.out` cannot be created in the current directory, the `nohup` utility uses the directory named by `HOME` to create the file.

open

`open` Opens files and directories.

```
open [-a <application>] <file> ...
open [-e] <file> ...
```

The `open` command opens a file (or a directory or URL), just as if you had double-clicked the file's icon. If no application name is specified, the default application as determined via `LaunchServices` is used to open the specified files.

If the file is in the form of a URL, the file will be opened as a URL.

You can specify one or more filenames (or pathnames), which are interpreted relative to the shell or Terminal window's current working directory. For example, the following command would open all Word files in the current working directory:

```
open *.doc
```

`-a <application>` Specifies the application to use for opening the file.

`-e` Causes the file to be opened with /Applications/TextEdit.app.

osacompile

`osacompile` Compiles OSA scripts.

```
osacompile [-l <language>] [-e <command>] [-o <name>] [-d] [-r <type:id>] [-t <type>]
[-c <creator>] [-x] [<file ...>]
```

`osacompile` compiles the given files, or standard input if none are listed, into a single output script. Files may be plain text or other compiled scripts. The options are as follows:

`-l <language>` Overrides the language for any plain text files. Normally, plain text files are compiled as AppleScript.

`-e <command>` Enters one line of a script. Script commands given via `-e` are prepended to the normal source, if any. Multiple `-e` commands may be given to build up a multi-line script. Because most scripts use characters that are special to many shell programs (for example, AppleScript uses single and double quote marks, "(", ")", and "*"), the command must be correctly quoted and escaped to get it past the shell intact.

`-o <name>` Places the output in the filename. If `-o` isn't specified, the resulting script is placed in the file a.scpt.

`-d` Places the resulting script in the data fork of the output file.

`-r <type:id>` Places the resulting script in the resource fork of the output file, in the specified resource.

`-t <type>` Sets the output file type to type. Type is a four-character code. If this option is omitted and the output file doesn't exist, the type is set to `osas`; that is, a compiled script.

`-c <creator>` Sets the output file creator to creator. Creator is a four-character code. If this option is omitted and the output file doesn't exist, the creator is set to `ToyS`, that is, Script Editor.

`-x` Saves the resulting script as execute only.

If no options are specified, `osacompile` produces a classic Mac OS format script file, that is, type `osas` (compiled script), creator `ToyS` (Script Editor), with the script data in the scpt:128 resource and nothing in the data fork. This format is compatible with all Mac OS and Mac OS X systems.

The `-d` and `-r` options aren't exclusive. If exactly one is specified, the script is written only to that fork. If both are specified, the script is written to both forks.

osascript

`osascript` Executes OSA scripts.

`osascript [-l <language>] [-e <command>] [-s <flags>] [<programfile>]`

`osascript` executes the given script file, or standard input if none is given. Scripts may be plain text or compiled scripts. `osascript` was designed for use with AppleScript, but will work with any Open Scripting Architecture (OSA) language. To get a list of the OSA languages installed on your system, use `osalang`(1). The options are as follows:

`-e <command>` Enters one line of a script. If `-e` is given, `osascript` won't look for a filename in the argument list. Multiple `-e` commands may be given to build up a multiline script. Because most scripts use characters that are special to many shell programs (for example, AppleScript uses single and double quote marks, "(", ")", and "*"), the command must be correctly quoted and escaped to get it past the shell intact.

`-l <language>` Overrides the language for any plain text files. Normally, plain text files are compiled as AppleScript.

`-s <flags>` Modifies the output style. The *`flags`* argument is a string consisting of any of the modifier characters `e`, `h`, `o`, and `s`. Multiple modifiers can be concatenated in the same string, and multiple `-s` options can be specified. The modifiers come in exclusive pairs; if conflicting modifiers are specified, the last one takes precedence. The meanings of the modifier characters are as follows:

`h` Prints values in human-readable form (default).

`s` Prints values in recompilable source form.

`osascript` normally prints its results in human-readable form: Strings don’t have quotes around them, characters aren’t escaped, braces for lists and records are omitted, and so on. This is generally more useful, but can introduce ambiguities. For example, the lists ‘{“foo”, “bar”}’ and ‘{{“foo”, {“bar”}}}’ would both be displayed as ‘foo, bar’. To see the results in an unambiguous form that could be recompiled into the same value, use the `s` modifier.

`e` Prints script errors to stderr (default).

`o` Prints script errors to stdout.

`osascript` normally prints script errors to stderr, so downstream clients see only valid results. When running automated tests, however, using the `o` modifier lets you distinguish script errors, which you care about matching, from other diagnostic output, which you don’t.

passwd

`passwd` Modifies a user’s password.

`passwd [-l] [-k] [-y] [<user>]`

`passwd` changes the user’s local, Kerberos, or YP password. The user is first prompted for her old password. The user is next prompted for a new password, and then prompted again to retype the new password for verification.

The new password should be at least six characters in length. It should use a variety of lowercase letters, uppercase letters, numbers, and metacharacters.

`-l` Updates the user’s local password.

`-k` Updates the Kerberos database, even if the user has a local password. After the password has been verified, `passwd` transmits the information to the Kerberos authenticating host.

`-y` Updates the YP password, even if the user has a local password. The `rpc.yppasswdd` (8) daemon should be running on the YP master server.

If no flags are specified, the following occurs:

If Kerberos is active, the user’s Kerberos password is changed, even if the user has a local password.

If the password isn’t in the local database, an attempt to update the YP password occurs.

To change another user’s Kerberos password, run `kinit (1)` followed by `passwd`. The superuser isn’t required to supply the user’s password if only the local password is being modified.

pbcopy

`pbcopy` Copies data from `STDIN` into the clipboard/pasteboard.

`pbcopy [-help]`

`pbcopy` places data from its standard input (`STDIN`) into the Mac OS X clipboard/pasteboard.

`-help` Displays its only option, `-help`.

pbpaste

`pbpaste` Writes textual data from the Mac OS X clipboard/pasteboard to `STDOUT`.

`pbpaste -help`

`pbpaste [-P rtf|ps|ascii]`

`pbpaste` pasts textual data from the clipboard/pasteboard to the commandline via `STDOUT`. The `-P` option allows you to suggest a preferred output format, but isn't necessarily obeyed.

`-P rtf` Prefers output in Rich Text Format if available.

`-P ps` Prefers output in PostScript format if available.

`-P ascii` Prefers American Standard Code for Information Interchange (yup, that's what ASCII stands for) plain text format.

perldoc

`perldoc` Look up Perl documentation in pod format.

`perldoc [-h] [-v] [-t] [-u] [-m] [-l] [-F]  [-X]` *`<Page-Name|ModuleName|ProgramName>`*

`perldoc -f` *`<BuiltinFunction>`*

`perldoc -q` *`<FAQ Keyword>`*

`perldoc` looks up a piece of documentation in .pod format that's embedded in the `perl` installation tree or in a Perl script, and displays it via `pod2man | nroff -man | $PAGER`. (In addition, if running under HP-UX, `col -x` will be used.) This is primarily used for the documentation for the `perl` library modules. Your system may also have man pages installed for those modules, in which case you can probably just use the `man`(1) command.

`-h` Prints out a brief help message.

`-v` Describes search for the item in detail.

`-t` Displays docs using plain text converter, instead of nroff. This may be faster, but it won't look as nice.

`-u` Finds docs only; skips reformatting by `pod2`*.

`-m` Displays the entire module: both code and unformatted pod documentation. This may be useful if the docs don't explain a function in the detail you need, and you'd like to inspect the code directly; perldoc will find the file for you and simply hand it off for display.

`-l` Displays the filename of the module found.

`-F` Considers arguments as filenames, no search in directories will be performed.

`-f` The `-f` option followed by the name of a Perl built-in function extracts the documentation of this function from the `perlfunc` man page.

`-q` The `-q` option takes a regular expression as an argument. It searches the question headings in `perl- faq[1-9]` and print the entries matching the regular expression.

`-X`	The `-X` option looks for a entry whose basename matches the name given on the command line in the file `$Config{archlib}/pod.idx`. The `pod.idx` file should contain fully qualified filenames, one per line.
`-U`	Because `perldoc` doesn't run properly tainted, and is known to have security issues, it won't normally execute as the superuser. If you use the `-U` flag, it will do so, but only after setting the effective and real IDs to `nobody`'s or `nouser`'s account, or `-2` if unavailable. If it cannot relinquish its privileges, it won't run.
`<`*`PageName`*`|`*`ModuleName`*`|`*`ProgramName`*`>`	The item you want to look up. Nested modules (such as *`File`*`::`*`Basename`*) are specified either as *`File`*`::`*`Basename`* or *`File`*`/`*`Basename`*. You may also give a descriptive name of a page, such as `perl- func`. You may also give a partial or wrong-case name, such as *`basename`* for *`File`*`::`*`Basename`*, but this will be slower, if there's more then one page with the same partial name, you will only get the first one.

pico

`pico`	A text editor.
`pico [-f] [+<`*`n`*`>] [-n<`*`n`*`>] [-t] [-v] [-w] [-z] [<`*`file`*`>]`	
`-f`	Uses function keys for commands. This option is supported only in conjunction with UW Enhanced NCSA telnet.
`+<`*`n`*`>`	Causes `pico` to be started with the cursor located `<`*`n`*`>` lines into the file.
`-n<`*`n`*`>`	Enables new mail notification. The `<`*`n`*`>` argument is optional, and specifies how often, in seconds, your mailbox is checked for new mail. For example, `-n60` causes pico to check for new mail once every minute. The default interval is 180 seconds; minimum allowed is 30.
`-t`	Enables tool mode. Intended for when `pico` is used as the editor within other tools (for example, Elm, Pnews). Pico won't prompt for save on exit, and won't rename the buffer during the `Write Out` command.
`-v`	Views the file only, disallowing any editing.
`-w`	Disables word wrap (thus allow editing of long lines).
`-z`	Enables `^Z` suspension of `pico`.

When a running pico is disconnected (for example, receives a `SIGHUP`), pico will save the current work if needed before exiting. Work is saved under the current filename with `.save` appended. If the current work is unnamed, it's saved under the filename `pico.save`.

Commands in `pico` are given as sequences using the Control key. The online help and bottom lines of instructions denote the control key with carat character: `^`. Here's a copy of the available functions in `pico` from the online help:

`^G (F1)`	Displays this help text.
`^F`	Moves forward a character.
`^B`	Moves backward a character.
`^P`	Moves to the previous line.
`^N`	Moves to the next line.
`^A`	Moves to the beginning of the current line.
`^E`	Moves to the end of the current line.
`^V (F8)`	Moves forward a page of text.
`^Y (F7)`	Moves backward a page of text.
`^W (F6)`	Searches for (where is) text, neglecting case.
`^L`	Refreshes the display.
`^D`	Deletes the character at the cursor position.
`^^`	Marks cursor position as beginning of selected text. Note: Setting mark when already set unselects text.
`^^`	Marks cursor position as beginning of selected text. Note: Setting mark when already set unselects text.
`^K` (F9)	Cuts selected text (displayed in inverse characters). Note: The selected text's boundary on the cursor side ends at the left edge of the cursor. So, with selected text to the left of the cursor, the character under the cursor isn't selected.
`^U` (F10)	Uncuts (paste) last cut text inserting it at the current cursor position.
`^I`	Inserts a tab at the current cursor position.
`^J` (F4)	Formats (justify) the current paragraph. Note: paragraphs delimited by blank lines or indentation.
`^T` (F12)	Invokes the spelling checker.
`^C` (F11)	Reports current cursor position.
`^R` (F5)	Inserts an external file at the current cursor position.
`^O` (F3)	Outputs the current buffer to a file, saving it.
`^X` (F2)	Exits pico, saving buffer.

ping

`ping` Sends `ICMP ECHO_REQUEST` packets to network hosts.

```
ping [-Rdfnqrv] [-c <count>] [-i <wait>] [-l <preload>]
        [-p <pattern>] [-s <packetsize>] <host>
```

`ping` uses the ICMP protocol's mandatory `ECHO_REQUEST` datagram to elicit an `ICMP ECHO_RESPONSE` from a host or gateway. `ECHO_REQUEST` datagrams (pings) have an IP and ICMP header, followed by a `struct timeval` and then an arbitrary number of pad bytes used to fill out the packet.

-d	Sets the SO_DEBUG option on the socket being used.
-f	Flood ping. Outputs packets as fast as they come or one hundred times per second, whichever is more. Only root may use this option. This option can be very hard on a network and should be used with caution.
-n	Displays numeric output only. Doesn't make any attempt to lookup symbolic names for host addresses.
-q	Enables quiet output. Displays only the summary lines at startup time and when finished.
-R	Record route. Includes RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets. The IP header is large enough for only nine such routes. Many hosts ignore or discard this option.
-r	Bypasses the normal routing tables and sends directly to a host on the attached network. If the host isn't on a directly attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.
-v	Enables verbose output. Lists ICMP packets received other than ECHO_RESPONSE packets.
-c <*count*>	Stops after sending and receiving <*count*> ECHO_RESPONSE packets.
-i <*wait*>	Sets the interval between sending each packet to <*wait*> seconds. Default is to wait one second. This option is incompatible with the -f option.
-l <*preload*>	Sends <*preload*> number of packets as fast as possible before falling into its normal mode of behavior. Only root may set a preload value.
-p <*pattern*>	Up to 16 pad bytes can be specified to fill out a packet that's sent. This is useful for diagnosing data-dependent problems in a network. For example, -p ff causes the sent packet to be filled with all 1s.
-s <*packetsize*>	Specifies the number of data bytes to be sent. The default is 56, which translates to 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

popd

popd	Pops the directory stack and changes to the new top directory.

```
popd [-p] [-l] [-n | -v] [+<n>]
popd
```

Without arguments, popd pops the directory stack and returns to the new top directory. Elements in the directory stack are numbered from 0 starting at the top.

`-p`	Overrides the `pushdsilent` shell variable. (The `pushdsilent` shell variable can be set to prevent `popd` from printing the final directory stack.)
`-l`	Lists the output in long form.
`-v`	Prints one entry per line, preceded by their stack positions.
`-n`	Wraps entries before they reach the edge of the screen.
`+<`*`n`*`>`	Discards the <*n*>th directory in the stack.

If both `-n` and `-v` are specified, `-v` takes precedence.

printcap

`printcap`   Printer capability database.

`printcap` is a simplified version of the `termcap` (5) database used to describe line printers. The `printcap` format is one of the formats understood by `niload`. Each `printcap` entry describes a single printer. The default printer is normally `lp`, although the environment variable `PRINTER` may be used to override this. Each spooling utility supports an option, `-P<`*`printer`*`>` to allow a specific printer destination to be named.

Capabilities

Name	**Type**	**Default**	**Description**
`af`	`str`	`NULL`	Name of accounting file.
`br`	`num`	`none`	If `lp` is a `tty`, sets the baud rate (`ioctl` call).
`cf`	`str`	`NULL`	`cifplot` data filter.
`ct`	`num`	`120`	TCP connection timeout in seconds.
`df`	`str`	`NULL`	Text data filter (DVI format).
`ff`	`str`	`'\f'`	String to send for a form feed.
`fo`	`bool`	`false`	Prints a form feed when device is opened.
`gf`	`str`	`NULL`	Graph data filter (plot format).
`hl`	`bool`	`false`	Prints the burst header page last.
`ic`	`bool`	`false`	Driver supports (nonstandard) `ioctl` to indent printout.
`if`	`str`	`NULL`	Name of text filter that does accounting.
`lf`	`str`	`/dev/console`	Error logging filename.
`lo`	`str`	`lock`	Name of lock file.
`lp`	`str`	`/dev/lp`	Device name to open for output.
`ms`	`str`	`NULL`	List of terminal modes to set or clear.
`mx`	`num`	`1000`	Maximum file size (in `BUFSIZ` blocks); 0=unlimited.
`nd`	`str`	`NULL`	Next directory for list of queues (unimplemented).
`nf`	`str`	`NULL`	`ditroff` data filter (device-independent `troff`).
`of`	`str`	`NULL`	Name of output filtering program.
`pc`	`num`	`200`	Price per foot or page in hundredths of cents.
`pl`	`num`	`66`	Page length in lines.

pw	num	132	Page width in characters.
px	num	0	Page width in pixels.
py	num	0	Page length in pixels.
rf	str	NULL	Filter for printing FORTRAN-style text files.
rg	str	NULL	Restricted group. Only members of group are allowed access.
rm	str	NULL	Machine name for remote printer.
rp	str	''lp''	Remote printer name argument.
rs	bool	false	Restricts remote users to those with local accounts.
rw	bool	false	Opens the printer device for reading and writing.
sb	bool	false	Short banner (one line only).
sc	bool	false	Suppresses multiple copies.
sd	str	/var/spool/lpd	Spool directory.
sf	bool	false	Suppresses form feeds.
sh	bool	false	Suppresses printing of burst page header.
sr	str	NULL	Filename to hold statistics of each data file as it's received.
ss	str	NULL	Filename to hold statistics of each data file as it's sent.
st	str	status	Status filename.
tf	str	NULL	troff data filter (cat phototypesetter).
tr	str	NULL	Trailer string to print when queue empties.
vf	str	NULL	Raster image file.

The man page Apple supplies for printcap is the 4.3 BSD man page, and it details several options and capabilities that aren't available now that Apple has switched to the CUPS printing system. We haven't included those sections here that we suspect will be ignored by the system as it now stands.

ps

ps	Displays process status report.
ps [-aCcefhjMmrSTuvwx] [-O <*fmt*>] [-o <*fmt*>] [-p <*pid*>] [-t <*tty*>] [-U <*username*>]	
ps [-L]	
-a	Includes information about processes owned by others in addition to yours.
-C	Changes the way CPU percentage is calculated by using a raw CPU calculation that ignores resident time. This normally has no effect.
-c	Changes the command column output to contain just the executable name rather than the full command line.
-e	Displays the environment.
-f	Shows command-line and environment information about swapped-out processes. This is honored only if the user's user ID is 0.

`-h` Repeats the header information so that there's one header per page of information.

`-j` Prints information associated with the following keywords: `user`, `pid`, `ppid`, `pgid`, `sess`, `jobc`, `state`, `tt`, `time`, and `command`.

`-l` Displays information associated with the following keywords: `uid`, `pid`, `ppid`, `cpu`, `pri`, `nice`, `vsz`, `rss`, `wchan`, `state`, `tt`, `time`, and `command`.

`-M` Prints the threads corresponding with each task.

`-m` Sorts by memory usage, rather than by process ID.

`-r` Sorts by current CPU usage, rather than by process ID.

`-S` Changes the way the process time is calculated by summing all exited children to their parent process.

`-T` Displays information about processes attached to the device associated with standard output.

`-u` Displays information associated with the following keywords: `user`, `pid`, `%cpu`, `%mem`, `vsz`, `rss`, `tt`, `state`, `start`, `time`, and `command`. The `-u` option implies the `-r` option.

`-v` Displays information associated with the following keywords: `pid`, `state`, `time`, `sl`, `re`, `pagein`, `vsz`, `rss`, `lim`, `tsiz`, `%cpu`, `%mem`, and `command`. The `-v` option implies the `-m` option.

`-w` Uses 132 columns to display information, instead of the default, which is your window size. If the `-w` option is specified more than once, `ps` uses as many columns as necessary, regardless of your window size.

`-x` Displays information about processes without controlling terminals.

`-O` *`<fmt>`* Adds the information associated with the space- or comma-separated list of keywords specified, after the process ID, in the default information displayed. Keywords may be further defined with an = and a string. Keywords further specified in this manner are displayed in the header as specified rather than using the standard header.

`-o` *`<fmt>`* Displays information associated with the space- or comma-separated list of keywords specified. Keywords may be further defined with an = and a string. Keywords further specified in this manner are displayed in the header as specified rather than using the standard header.

`-p` *`<pid>`* Displays information associated with the specified process ID *`<pid>`*.

`-t` *`<tty>`* Displays information about processes attached to the specified terminal device *`<tty>`*.

`-U` *`<username>`* Displays information about processes belonging to the specified *`<username>`*.

`-L` Lists the set of available keywords.

The following is a list of the definitions of the keywords that some of the options already include. There are more keywords available than are defined here.

`%cpu`	Percentage CPU usage (alias `pcpu`)
`%mem`	Percentage memory usage (alias `pmem`)
`command`	Command and arguments
`cpu`	Short-term CPU usage factor (for scheduling)
`jobc`	Job control count
`lim`	Memory use limit
`nice`	Nice value (alias to `ni`)
`pagein`	Pageins (total page faults)
`pgid`	Process group number
`pid`	Process ID
`ppid`	Parent process ID
`pri`	Scheduling priority
`re`	Core residency time (in seconds; `127` = infinity)
`rss`	Resident set size (real memory)
`rsz`	Resident set size + (text size/text use count) (alias `rs-size`)
`sess`	Session pointer
`sl`	Sleep time (in seconds; `127` = infinity)
`start`	Time started
`state`	Symbolic process state (alias `stat`)
`tsiz`	Text size (in kilobytes)
`tt`	Control terminal name (two-letter abbreviation)
`uid`	Effective user ID
`user`	Username (from `uid`)
`vsz`	Size of process in virtual memory in kilobytes (alias `vsize`)
`wchan`	Wait channel (as a symbolic name)

pushd

`pushd`	Pushes a directory onto the directory stack.

`pushd [-p] [-l] [-n | -v] [<dir> | +<n>]`

`pushd`

Without arguments, `pushd` exchanges the top two elements of the directory stack. If `pushdtohome` is set, `pushd` without arguments does `pushd ~`, like `cd`.

`-p`	Overrides the `pushdsilent` shell variable. (The `pushdsilent` shell variable can be set to prevent `pushd` from printing the final directory stack.)
`-l`	Lists the output in long form.
`-v`	Prints one entry per line, preceded by their stack positions.
`-n`	Wraps entries before they reach the edge of the screen.

`<dir>` Pushes the current directory into the stack and changes to the specified `<dir>`.

`+<n>` Rotates the `<n>`th directory to the top of the stack and changes to that directory.

If both `-n` and `-v` are specified, `-v` takes precedence.

pwd

`pwd` Prints current working directory.

```
pwd [-L|P]
```

`-L` Prints the logical path to the current working directory, as defined by the shell in the environment variable `PWD`.

`-P` Default. Prints the physical path to the current working directory, with symbolic links resolved.

restore

`restore` Restores files or file systems from backups made with `dump`.

```
restore -i [-chmvy] [-b <blocksize>] [-f <file>] [-s <fileno>]
restore -R [-cvy] [-b <blocksize>] [-f <file>] [-s <fileno>]
restore -r [-cvy] [-b <blocksize>] [-f <file>] [-s <fileno>]
restore -t [-chvy] [-b <blocksize>] [-f <file>] [-s <fileno>] [<file> ...]
restore -x [-chmvy] [-b blocksize] [-f file] [-s <fileno>] [<file> ...]
```

(The 4.3BSD option syntax is implemented for backward compatibility, but isn't documented here.)

The `restore` command performs the inverse function of `dump`(8). A full backup of a file system may be restored and subsequent incremental backups layered on top of it. Single files and directory subtrees may be restored from full or partial backups. `Restore restore` works across a network; to do this, see the `-f` flag described following. Other arguments to the command are file or directory names specifying the files that are to be restored. Unless the `-h` flag is specified (see belowlater discussion), the appearance of a directory name refers to the files and (recursively) subdirectories of that directory.

Exactly one of the following flags is required:

`-i` Interactive. After reading in the directory information from the dump, `restore` provides a shell-like interface that allows the user to move around the directory tree selecting files to be extracted. The available commands are given following; for those commands that require an argument, the default is the current directory.

`-R` Requests a particular tape of a multi volume set on which to restart a full restore (see the `-r` flag later in the table). This is useful if the restore has been interrupted.

`-r` Restore (rebuilds a file system). The target file system should be made pristine with `newfs`(8), mounted and the user cd'd into the pristine file system before starting the restoration of the initial level 0 backup. If the level 0 restores successfully, the `-r` flag may be used to restore any necessary incremental backups on top of the level 0. The `-r` flag precludes an interactive file extraction and can be detrimental to one's health if not used carefully (not to mention the disk). An example:

```
newfs /dev/rrp0g eagle
mount /dev/rp0g /mnt
cd /mnt
restore rf /dev/rst8
```

Note that `restore` leaves a file `restoresymtable` in the root directory to pass information between incremental restore passes. This file should be removed when the last incremental has been restored. `restore`, in conjunction with `newfs`(8) and `dump`(8), may be used to modify file system parameters such as size or block size.

`-t` The names of the specified files are listed if they occur on the backup. If no file argument is given, the root directory is listed, which results in the entire content of the backup being listed, unless the `-h` flag has been specified. Note that the `–t` flag replaces the function of the old `dumpdir`(8) program.

`-x` The named files are read from the given media. If a named file matches a directory whose contents are on the backup and the `–h` flag isn't specified, the directory is recursively extracted. The owner, modification time, and mode are restored (if possible). If no file argument is given, the root directory is extracted, which results in the entire content of the backup being extracted, unless the `-h` flag has been specified.

The following additional options may be specified:

`-b <`*`blocksize`*`>` The number of kilobytes per dump record. If the `-b` option isn't specified, `restore` tries to determine the block size dynamically.

`-c` Normally, `restore` tries to determine dynamically whether the dump was made from an old (pre-4.4) or new format file system. The `-c` flag disables this check, and only allows reading a dump in the old format.

`-f <`*`file`*`>` Reads the backup from file; file may be a special device file like `/dev/rmt12` (a tape drive), `/dev/rsd1c` (a disk drive), an ordinary file, or `-` (the standard input). If the name of the file is of the form `<`*`host`*`>:<`*`file`*`>`, or `user@host:file`, `restore` reads from the named file on the remote host using `rmt`(8).

`-h` Extracts the actual directory, rather than the files that it references. This prevents hierarchical restoration of complete subtrees from the dump.

`-m` Extracts by inode numbers rather than by file name. This is useful if only a few files are being extracted, and you want to avoid regenerating the complete pathname to the file.

`-s` *<fileno>* Reads from the specified *<fileno>* on a multi-file tape. File numbering starts at 1.

`-v` Normally `restore` does its work silently. The `-v` (verbose) flag causes it to type the name of each file it treats preceded by its file type.

`-y` Doesn't ask the user whether to abort the restore in the event of an error. Always tries to skip over the bad block(s) and continue.

Interactive commands:

`add` *<file>* The current directory or specified argument is added to the list of files to be extracted. If a directory is specified, it and all its descendants are added to the extraction list (unless the `-h` flag is specified on the command line). Files that are on the extraction list are prepended with a `*` when they're listed by `ls`.

`cd` *<directory>* Changes the current working directory to the specified argument.

`delete` *<file>* Deletes the current file or directory and its descendants from the list of files to be extracted (unless the `-h` flag is specified on the command line). The most expedient way to extract most of the files from a directory is to add the directory to the extraction list and then delete those files that aren't needed.

`extract` Extracts all the files that are on the extraction list for the dump. `restore` asks which volume the user wants to mount. The fastest way to extract a few files is to start with the last volume, and work towards the first volume.

`help` Lists a summary of the available commands.

`ls` *<arg>* Lists the current or specified directory. Entries that are directories are appended with a `/`. Entries that have been marked for extraction are prepended with a `*`. If the `verbose` flag is set, the inode number of each entry is also listed.

`pwd` Prints the full pathname of the current working directory.

`quit` `restore` immediately exits, even if the extraction list isn't empty.

`setmodes` All the directories that have been added to the extraction list have their owner, modes, and times set; nothing is extracted from the dump. This is useful for cleaning up after a restore has been prematurely aborted.

`verbose` The sense of the `-v` flag is toggled. When set, the `verbose` flag causes the `ls` command to list the inode numbers of all entries. It also causes `restore` to print out information about each file as it's extracted.

rm

`rm` Removes files.

`rm [-f | -i] [-dPRrW] <file1> <file2> ...`

`-f` Forces the removal of files without prompting the user for confirmation. If the file doesn't exist, no error diagnostic is displayed. The `-f` option overrides any previous `-i` options.

`-i` Invokes an interactive mode that prompts for confirmation before removing a file. The `-i` option overrides any previous `-f` options.

`-d` Attempts to remove directories as well as other types of files.

`-P` Overwrites regular files before deleting them. Files are overwritten three times before being deleted; first with byte pattern `0xff`, and then `0x00`, and then `0xff`.

`-R` Attempts to recursively remove files. Implies `-d` option.

`-r` Same as `-R`.

`-W` Attempts to undelete files. This option can be used to recover only files covered by whiteouts.

`rm` removes symbolic links, but not the files referenced by the links.

Also, attempting to remove the files `.` and `..` is an error.

rmdir

`rmdir` Removes directories.

`rmdir [-p] <directory1> <directory2> ...`

`rmdir` removes each `<directory>` argument specified, provided it's empty. Arguments are processed in the order listed on the command line. To remove a parent directory and subdirectories of the parent directory, the subdirectories must be listed first.

`-p` Attempts to remove the specified directory and its parent directories, if they're empty.

rsync

`rsync` Synchronizes files and directories; faster, flexible replacement for `rcp`.

```
rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST
rsync [OPTION]... [USER@]HOST:SRC DEST
rsync [OPTION]... SRC [SRC]... DEST
rsync [OPTION]... [USER@]HOST::SRC [DEST]
rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST
rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]
```

Primary uses for rsync

There are six different ways of using rsync:

Copying local files. This is invoked when neither source nor destination path contains a : separator.

Copying from the local machine to a remote machine using a remote shell program as the transport (such as rsh or ssh). This is invoked when the destination path contains a single : separator.

Copying from a remote machine to the local machine using a remote shell program. This is invoked when the source contains a : separator.

Copying from a remote rsync server to the local machine. This is invoked when the source path contains a :: separator or a rsync:// URL.

Copying from the local machine to a remote rsync server. This is invoked when the destination path contains a :: separator.

Listing files on a remote machine. This is done the same way as rsync transfers except that you leave off the local destination.

Note that in all cases (other than listing), at least one of the source and destination paths must be local.

Setup

rsync uses rsh for its communications, unless both the source and destination are local.

You can also specify an alternative to rsh, either by using the -e command-line option, or by setting the RSYNC_RSH environment variable.

One common substitute is to use ssh, which offers a high degree of security.

Note that rsync must be installed on both the source and destination machines.

Usage

You use rsync in the same way you use rcp. You must specify a source and a destination, one of which may be remote.

Perhaps the best way to explain the syntax is to give some examples:

```
rsync *.c foo:src/
```

This would transfer all files matching the pattern *.c from the current directory to the directory src on the machine foo. If any of the files already exist on the remote system, the rsync remote-update protocol is used to update the file by sending only the differences.

```
rsync -avz foo:src/bar /data/tmp
```

This would recursively transfer all files from the directory src/bar on the machine foo into the /data/tmp/bar directory on the local machine. The files are transferred in archive mode, which ensures that symbolic links, devices, attributes, permissions, ownerships, and so on are preserved in the transfer. Additionally, compression will be used to reduce the size of data portions of the transfer.

```
rsync -avz foo:src/bar/ /data/tmp
```

A trailing slash on the source changes this behavior to transfer all files from the directory src/bar on the machine foo into the /data/tmp/. A trailing / on a source name means "copy the contents of this directory." Without a trailing slash, it means "copy the directory." This difference becomes particularly important when using the --delete option.

You can also use rsync in local-only mode, where both the source and destination don't have a : in the name. In this case, it behaves like an improved copy command.

```
rsync somehost.mydomain.com::
```

This would list all the anonymous rsync modules available on the host somehost.mydomain.com.

Connecting to an `rsync` server

It's also possible to use `rsync` without using `rsh` or `ssh` as the transport. In this case, you will connect to a remote `rsync` server running on TCP port 873.

You may establish the connection via a Web proxy by setting the environment variable `RSYNC_PROXY` to a `hostname:port` pair pointing to your Web proxy. Note that your Web proxy's configuration must allow proxying to port 873.

Using `rsync` in this way is the same as using it with `rsh` or `ssh` except that:

You use a double colon `::` instead of a single colon to separate the hostname from the path.

The remote server may print a message of the day when you connect.

If you specify no pathname on the remote server, the list of accessible paths on the server will be shown.

If you specify no local destination, a listing of the specified files on the remote server is provided.

Some paths on the remote server may require authentication. If so, you will receive a password prompt when you connect. You can avoid the password prompt by setting the environment variable `RSYNC_PASSWORD` to the password you want to use or using the `--password-file` option. This may be useful when scripting `rsync`.

Warning: On some systems, environment variables are visible to all users. On those systems using `--password-file` is recommended.

Running an `rsync` server

An `rsync` server is configured using a config file, which by default is called `/etc/rsyncd.conf`. Please see the `rsyncd.conf`(5) man page for more information.

`-h`	
`--help`	Prints a short help page describing the available options.
`--version`	Prints `rsync` version number and exits.
`-v`	
`--verbose`	Increases verbosity.
`-q`	
`--quiet`	Decreases verbosity. Useful when invoking `rsync` from `cron`.
`-I`	
`--ignore-times`	Normally `rsync` skips any files that are already the same length and have the same timestamp. This option turns off this behavior.
`--size-only`	Normally `rsync` skips any files that are already the same length and have the same timestamp. With the `--size-only` option, files will be skipped if they have the same size, regardless of timestamp. Useful when starting to use `rsync` after using another mirroring system that may not preserve timestamps exactly.
`--modify-window`	When comparing two timestamps, `rsync` treats the timestamps as being equal if they are within the value of `modify_window`.
`-c`	
`--checksum`	Forces the sender to checksum all files using a 128-bit MD4 checksum before transfer. The checksum is then explicitly checked on the receiver, and any files of the same name that already exist and have the same checksum and size on the receiver are skipped. This option can be quite slow.

`-a`
`--archive`
Equivalent to `-rlptgoD`. It's a quick way of saying you want recursion and want to preserve almost everything.
Note, however, that `-a` doesn't preserve hardlinks, because finding multiply linked files is expensive. You must separately specify `-H`.

`-r`
`--recursive`
Copies directories recursively. If you don't specify this, `rsync` won't copy directories at all.

`-R`
`--relative`
Uses relative paths. This means that the full pathnames specified on the command line are sent to the server rather than just the last parts of the filenames. This is particularly useful when you want to send several different directories at the same time. For example, if you used the command

```
rsync foo/bar/foo.c remote:/tmp/
```

it would create a file called `foo.c` in `/tmp/` on the remote machine. If instead you used

```
rsync -R foo/bar/foo.c remote:/tmp/
```

a file called `/tmp/foo/bar/foo.c` would be created on the remote machine. The full pathname is preserved.

`-b`
`--backup`
With this option, preexisting destination files are renamed with a ~ extension as each file is transferred. You can control the backup suffix using the `--suffix` option.

`--backup-dir=<`*`DIR`*`>`
In combination with the `--backup` option, this tells `rsync` to store all backups in the specified directory. This is very useful for incremental backups.

`--suffix=<`*`SUFFIX`*`>`
This option allows you to override the default backup suffix used with the `-b` option. The default is a `~`.

`-u`
`--update`
Forces `rsync` to skip any files for which the destination file already exists and that have a date later than the source file.

`-l`
`--links`
When symlinks are encountered, re-creates the symlink on the destination.

`-L`
`--copy-links`
When symlinks are encountered, the file that they point to is copied, rather than the symlink.

`--copy-unsafe-links`
Tells `rsync` to copy the referent of symbolic links that point outside the source tree. Absolute symlinks are also treated like ordinary files, and so are any symlinks in the source path itself when `--relative` is used.

`--safe-links`	Tells `rsync` to ignore any symbolic links that point outside the destination tree. All absolute symlinks are also ignored. Using this option in conjunction with `--relative` may give unexpected results.
`-H` `--hard-links`	Tells `rsync` to re-create hard links on the remote system to be the same as the local system. Without this option, hard links are treated like regular files. Note that `rsync` can only detect hard links if both parts of the link are in the list of files being sent. This option can be quite slow, so use it only if you need it.
`-W` `--whole-file`	With this option, the incremental `rsync` algorithm isn't used and the whole file is sent as-is instead. The transfer may be faster if this option is used when the bandwidth between the source and target machines is higher than the bandwidth to disk (especially when the disk is actually a networked file system). This is the default when both the source and target are on the local machine.
`-p` `--perms`	Causes `rsync` to update the remote permissions to be the same as the local permissions.
`-o` `--owner`	Causes `rsync` to set the owner of the destination file to be the same as the source file. On most systems, only the superuser can set file ownership.
`-g` `--group`	Causes `rsync` to set the group of the destination file to be the same as the source file. If the receiving program isn't running as the superuser, only groups that the receiver is a member of will be preserved (by group name, not group id number).
`-D` `--devices`	Uses `rsync` to transfer character and block device information to the remote system to re-create these devices. This option is only available to the superuser.
`-t` `--times`	Tells `rsync` to transfer modification times along with the files and update them on the remote system. Note that if this option isn't used, the optimization that excludes files that haven't been modified cannot be effective; in other words, a missing `-t` or `-a` will cause the next transfer to behave as if it used `-I`, and all files will have their checksums compared and show up in log messages even if they haven't changed.

`-n`
`--dry-run`
Tells `rsync` to not tonot do any file transfers; instead it will just report the actions it would have taken.

`-S`
`--sparse`
Tries to handle sparse files efficiently so they take up less space on the destination.

`-x`
`--one-file-system`
Tells `rsync` not to cross file system boundaries when recursing. This is useful for transferring the contents of only one file system.

`--existing`
Tells `rsync` not to create any new files —to only update files that already exist on the destination.

`--max-delete=<NUM>`
Tells `rsync` not to delete more than NUM files or directories. This is useful when mirroring very large trees to prevent disasters.

`--delete`
Tells `rsync` to delete any files on the receiving side that aren't on the sending side. Files that are excluded from transfer are excluded from being deleted unless you use
`--delete-excluded`.

This option has no effect if directory recursion isn't selected.

This option can be dangerous if used incorrectly! It's a very good idea to run first using the dry run option (`-n`) so that you can see what files would be deleted without actually deleting them, and check to make sure important files aren't listed for deletion.

If the sending side detects any I/O errors, the deletion of any files at the destination will be automatically disabled. This is to prevent temporary file system failures (such as NFS errors) on the sending side causing a massive deletion of files on the destination. You can override this with the `--ignore-errors` option.

`--delete-excluded`
In addition to deleting the files on the receiving side that aren't on the sending side, this tells `rsync` to also delete any files on the receiving side that are excluded (see `--exclude`).

`--delete-after`
By default `rsync` does file deletions before transferring files to try to ensure that there's is sufficient space on the receiving file system. If you want to delete after transferring, use the the
`--delete-after` switch.

`--ignore-errors`
Tells `--delete` to go ahead and delete files even when there are I/O errors.

`--force`
Tells `rsync` to delete directories even if they aren't empty. This applies to both the `--delete` option and to cases where `rsync` tries to copy a normal file but the destination contains a directory of the same name.

Because this option was added, deletions were reordered to be done depth-first, so it's hardly ever needed any more except in very obscure cases.

`-B`
`--block-size=<BLOCKSIZE>` Controls the block size used in the `rsync` algorithm.

`-e`
`--rsh=COMMAND` Allows you to choose an alternative remote shell program to use for communication between the local and remote copies of `rsync`. By default, `rsync` will use `rsh`, but you might like to use `ssh` instead because of its high security.
You can also choose the remote shell program using the `RSYNC_RSH` environment variable.
See also the `--blocking-io` option, which is affected by this option.

`--rsync-path=<PATH>` Specifies the path to the copy of `rsync` on the remote machine. Useful when it ‘is no’t in your path. Note that this is the full path to the binary, not just the directory that the binary is in.

`--exclude=PATTERN` Allows you to selectively exclude certain files from the list of files to be transferred. This is most useful in combination with a recursive transfer.
You may use as many `--exclude` options on the command line as you like to build up the list of files to exclude.

`--exclude-from=<FILE>` Similar to the `--exclude` option, but instead it adds all exclude patterns listed in file `FILE` to the exclude list. Blank lines in `FILE` and lines starting with `;` or `#` are ignored.

`--include=<PATTERN>` Tells `rsync` to not exclude the specified pattern of filenames. This is useful because it allows you to build up quite complex exclude/include rules.

`--include-from=<FILE>` Specifies a list of include patterns from a file.

`-C`
`--cvs-exclude` Useful shorthand for excluding a broad range of files that you often don’t want to transfer between systems. It uses the same algorithm that CVS uses to determine whether a file should be ignored.
The exclude list is initialized to

```
RCS SCCS CVS CVS.adm RCSLOG cvslog.* tags TAGS
make.state .nse_depinfo *~ #* .#* ,* *.old *.bak *.BAK
*.orig *.rej .del-* *.a *.o *.obj *.so *.Z *.elc *.ln
core
```

Then files listed in a `$HOME/.cvsignore` are added to the list and any files listed in the `CVSIGNORE` environment variable (space delimited).
Finally, any file is ignored if it’s in the same directory as a `.cvsignore` file and matches one of the patterns listed therein.

`--csum-length=<LENGTH>` By default, the primary checksum used in `rsync` is a very strong 16-byte MD4 checksum. In most cases you will find that a truncated version of this checksum is quite efficient, and this will decrease the size of the checksum data sent over the link, making things faster.

You can choose the number of bytes in the truncated checksum using the `--csum-length` option. Any value less than or equal to 16 is valid.

`-T`
`--temp-dir=<DIR>` Instructs `rsync` to use `DIR` as a scratch directory when creating temporary copies of the files transferred on the receiving side. The default behavior is to create the temporary files in the receiving directory.

`--compare-dest=<DIR>` Instructs `rsync` to use `DIR` on the destination machine as an additional directory to compare destination files against when doing transfers. This is useful for doing transfers to a new destination while leaving existing files intact, and then doing a flash-cutover when all files have been successfully transferred (for example, by moving directories around and removing the old directory, although this requires also doing the transfer with `-I` to avoid skipping files that haven't changed). This option increases the usefulness of `--partial` because partially transferred files remain in the new temporary destination until they have a chance to be completed. If `DIR` is a relative path, it's relative to the destination directory.

`-z`
`--compress` Compresses any data from the files that it sends to the destination machine. Useful on slow links. The compression method used is the same method that `gzip` uses.

`--numeric-ids` Transfers numeric group and user ids rather than using user and group names and mapping them at both ends.

By default, `rsync` will uses the username and group name to determine what ownership to give files. The special uid 0 and the special group 0 are never mapped via user/group names even if the `--numeric-ids` option isn't specified.

If the source system is a daemon using `chroot`, or if a username or group name doesn't exist on the destination system, the numeric id ID from the source system is used instead.

`--timeout=<TIMEOUT>` Allows you to set a maximum I/O timeout in seconds. If no data is transferred for the specified time, `rsync` will exit. The default is 0, which means no timeout.

`--daemon` Tells `rsync` that it's to run as a daemon. The daemon may be accessed using the `host::module` or `rsync://host/module/ syntax`.

If standard input is a socket, `rsync` will assume that it's being run via `inetd`; otherwise, it detaches from the current terminal and becomes a background daemon. The daemon reads the config file (`/etc/rsyncd.conf`) on each connect made by a client and responds to requests accordingly.

`--no-detach`	When running as a daemon, this option instructs `rsync` not to not detach itself and become a background process. Has no effect if `rsync` is run from `inetd` or `sshd`.
`--address`	By default, `rsync` binds to the wildcard address when run as a daemon with the `--daemon` option or when connecting to an `rsync` server. The `–address` option allows you to specify a specific IP address (or hostname) to bind to. This makes virtual hosting possible in conjunction with the `--config` option.
`--config=<`*`FILE`*`>`	Specifies an alternate config file than the default `/etc/rsyncd.conf`. This is only relevant when `--daemon` is specified.
`--port=<`*`PORT`*`>`	Specifies an alternative TCP port number to use rather than the default port 873.
`--blocking-io`	Tells `rsync` to use blocking I/O when launching a remote shell transport. If `-e` or `--rsh` aren't specified or are set to the default `rsh`, this defaults to blocking I/O; otherwise, it defaults to non-blocking I/O. You may find the `--blocking-io` option is needed for some remote shells that can't handle non-blocking I/O. SSH prefers blocking I/O.
`--log-format=<`*`FORMAT`*`>`	Allows you to specify exactly what the `rsync` client logs to `stdout` on a per-file basis. The log format is specified using the same format conventions as the log format option in `rsyncd.conf`.
`--stats`	Tells `rsync` to print a verbose set of statistics on the file transfer, allowing you to tell how effective the `rsync` algorithm is for your data.
`--partial`	Deletes any partially transferred file if the transfer is interrupted. In some circumstances, it's more desirable to keep partially transferred files. Using the `--partial` option tells `rsync` to keep the partial file, which should make a subsequent transfer of the rest of the file much faster.
`--progress`	Tells `rsync` to print information showing the progress of the transfer. This gives a bored user something to watch. This option is normally combined with `-v`. Using this option without the `-v` option produces weird results on your display.
`-P`	Equivalent to `--partial --progress`.
`--password-file`	Allows you to provide a password in a file for accessing a remote `rsync` server. Note that this option is only useful when accessing an `rsync` server using the built-in transport, not when using a remote shell as the transport. The file must not be world-readable. It should contain just the password as a single line.

--bwlimit=<KBPS>	Allows you to specify a maximum transfer rate in kilobytes per second. This option is most effective when using rsync with large files (several megabytes and up). Due to the nature of rsync transfers, blocks of data are sent, and then if rsync determines that the transfer was too fast, it will wait before sending the next data block. The result is an average transfer rate equaling the specified limit. A value of zero specifies no limit.
--read-batch	Applies a previously generated change batch.
--write-batch	Generates a set of files that can be transferred as a batch update.

scp

scp	Secure remote copy.
scp [-pqrvBC46] [-F <ssh_config>] [-S <program>] [-P <port>] [-c <cipher>] [-i <identity_file>] [-o <ssh_option>] [[<user>@]<host1>:]<file1> [...] [[<user>@]<host2>:]<file2>	
-p	Preserves modification times, access times, and modes from the original file.
-q	Disables the progress meter.
-r	Recursively copies entire directories.
-v	Verbose mode. Causes scp and ssh to print debugging messages.
-B	Selects batch mode, which prevents passwords or passphrases from being requested.
-C	Enables compression. Passes the flag to ssh(1) to enable compression.
-4	Forces scp to use IPv4 addresses only.
-6	Forces scp to use IPv6 addresses only.
-F <ssh_config>	Specifies an alternative per-user configuration file for ssh. Option is directly passed to ssh(1).
-S <program>	Specifies <program> to use for the encrypted connection. Program must understand ssh(1) options.
-P <port>	Specifies the port to connect to on the remote host.
-c <cipher>	Selects the cipher to use for encrypting the data transfer. Option is passed directly to ssh(1).
-i <identity_file>	Specifies the file from which the identity (private key) for RSA authentication is read.
-o <ssh_option>	Passes specified options to ssh in the format used in ssh_config(5).

screencapture

`screencapture` Takes pictures of the current state of the screen.

`screencapture [-[i|m]wsWx] <file>`

`screencapture [-[i|m]cwsWx]`

`screencapture` takes pictures of the current state of the screen or screens present on the machine, or of windows or selectable regions of the screen. `screencapture` saves its output in `.pdf` format, or places it on the clipboard.

`screencapture` lists a `[cursor]` parameter as following the `<file>` parameter when displaying its options, but this parameter is undocumented, and an examination of the `screencapture` executable doesn't reveal any obvious candidates for parameter values. `screencapture` also accepts an undocumented `-f` option, which is apparently a placeholder option that can be used in `<file>` mode.

`-i` Captures the screen interactively, by selection or window. Pressing the spacebar toggles between
region selection (crosshair cursor)
window selection (camera cursor)
Pressing the Esc key cancels the capture.

`-c` Places the screen capture on the clipboard, instead of into a file.

`-m` Only captures the main monitor. Undefined if `-i` is present.

`-w` Only allows window selection mode.

`-s` Only allows mouse selection mode.

`-W` Starts interaction in window selection mode.

`-x` Don't play sounds.

sed

`sed` Stream editor.

`sed [-an] <command> [<file> ...]`

`sed [-an] [-e <command>] [-f <command_file>] [<file>]`

`sed` reads one or more text files, or standard input if no file is specified, makes editing changes according to a single command specified by `<command>` or by using the `-e` or `-f` options. The input is then written to standard output. All commands are applied to the input in the order they're specified, regardless of their origin.

`-a` By default, the files listed as parameters for the `w` functions are created or truncated before any processing begins. This option causes `sed` to delay opening each file until a command containing the related `w` function is applied to a line of input.

`-n` By default, each line of input is echoed to the standard output after all the commands have been applied to it. This option suppresses the default output behavior.

`-e <command>`	Appends editing commands specified by the *<command>* argument to the list of commands.
`-f <command_file>`	Appends editing commands found in the file *<command_file>* to the list of commands. The editing commands should be listed one per line.

The form of a `sed` command is as follows:

```
[address[,address]]function[arguments]
```

Whitespace may be inserted before the first address and the function portions of the command.

Normally, `sed` cyclically copies a line of input, not including its terminating newline character into a pattern space (unless there's something left after the `D` function), applies all the commands with addresses that select that pattern space, copies the resulting pattern space to the standard output (except if `-n` is used), appends a newline, and deletes the pattern space.

Some of the functions use a hold space to save all or part of the pattern space for subsequent retrieval.

`sed` Addresses

An address isn't required, but if specified, it must be a number that counts input lines cumulatively across input files, a `$` that addresses the last line of input, or a context address that consists of a regular expression preceded and followed by a delimiter.

A command line with no addresses selects every pattern space.

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 that line is selected. Starting at the first line of the selected range, `sed` starts looking again for the first address.

Editing commands can be applied to nonselected pattern spaces by use of `!` , the negation function.

`sed` Regular Expressions

`sed` regular expressions are basic regular expressions (see `regex`(3)) with these additions:

1. In context addresses, any character other than `\` or the newline character may be used to delimit a regular expression by prefixing the first use of that delimiter with `\`. Also, putting `\` before the delimiting character causes the character to be treated literally, which doesn't terminate the regular expression. For example, in the context address `\xabc\xdefx`, the second `x` stands for itself, so that the regular expression is `abcxdef`.
2. The escape sequence `\n` matches a newline character embedded in the pattern space. A literal newline character must not be used in the regular expression of a context address or in the substitute command.

One special feature of `sed` regular expressions is that they can default to the last regular expression used. If a regular expression is empty (just the delimiter characters are specified), the last regular expression encountered is used instead. The last regular expression is defined as the last regular expression used as part of an address or substitute command, and at run time, not compile time. For example, the command `/abc/s//XXX/` substitutes `XXX` for the pattern `abc`.

`sed` Functions

In the following list of commands, the maximum number of permissible addresses for each command is indicated by `[0addr]`, `[1addr]`, or `[2addr]`, representing zero, one, or two addresses.

The argument *<text>* consists of one or more lines. To embed a newline in the text, precede it with a `\`. Other backslashes in text are deleted and the following character taken literally.

The r and w functions take an optional <*file*> parameter, which should be separated from the function letter by white space. Each file given as an argument to sed is created (or its contents truncated) before any input processing begins.

The b, r, s, t, w, y, !, and : functions all accept additional arguments. The following synopses indicate which arguments have to be separated from the function letters by whitespace.

Two of the functions take a function list. This is a list of sed functions separated by newlines, as follows:

```
{ function
 function
 . . .
 }
```

The { can be preceded by whitespace and can be followed by whitespace. The function can be preceded by whitespace. The terminating } must be preceded by a newline or optional white space.

The following lists the functions:

Max # of Addresses	Command	Description
[2addr]	<*function_list*>	Executes <*function-list*> only when the pattern space is selected.
[1addr]	a\<*text*>	Writes <*text*> to standard output immediately before each attempt to read a line of output, whether by executing the N function or beginning a new cycle.
[2addr]	b <*label*>	Branches to the : function with the specified label. If the label isn't specified, it branches to the end of the script.
[2addr]	c\<*text*>	Change. Deletes the pattern space. With zero or one address or at the end of a two-address range, <*text*> is written to standard output.
[2addr]	d	Deletes the pattern space and starts the next cycle.
[2addr]	D	Deletes the initial segment of the pattern space through the first newline character and starts the next cycle.
[2addr]	g	Replaces the contents of the pattern space with the contents of the hold space.
[2addr]	G	Appends a newline character followed by the contents of the hold space to the pattern space.
[2addr]	h	Replaces the contents of the hold space with the contents of the pattern space.
[2addr]	H	Appends a newline character followed by the contents of the pattern space.
[1addr]	i\<*text*>	Insert. Writes <*text*> to the standard output.
[2addr]	l	Writes the pattern space to the standard output in a visually unambiguous form. The form is as follows: Backslash \\ Alert \a

		Form-feed `\f` Newline `\n` Carriage return `\r` Tab `\t` Vertical tab `\v` Nonprinting characters are written as three-digit octal numbers with a preceding backslash for each byte in the character (most significant byte first). Long lines are folded, with the point of folding indicated by displaying a backslash followed by a newline. The end of each line is marked with a `$`.
`[2addr]`	`n`	Writes the pattern space to the standard output if the default output hasn't been suppressed, and replaces the pattern space with the next line of input.
`[2addr]`	`N`	Appends the next line of input to the pattern space, using an embedded newline character to separate the appended material from the original contents (the current line number changes).
`[2addr]`	`p`	Writes the pattern space to standard output.
`[2addr]`	`P`	Writes the pattern space, up to the first newline character, to the standard output.
`[1addr]`	`q`	Branches to the end of the script and quits without starting a new cycle.
`[1addr]`	`r` *`<file>`*	Copies the contents of *`<file>`* to the standard output immediately before the next attempt to read a line of input. If *`<file>`* cannot be read for any reason, it's silently ignored and no error condition is set.
`[2addr]`	`t` *`<label>`*	Test. Branches to the `:` function bearing the *`<label>`* if any substitutions have been made since the most recent reading of an input line or execution of a function. If no label is specified, branches to the end of the script.
`[2addr]`	`w` *`<file>`*	Appends the pattern space to the *`<file>`*.
`[2addr]`	`x`	Exchanges the contents of the pattern and hold spaces.

`[2addr]`	`!`*`<function>`*
	`!`*`<function_list>`*
		Applies the *`<function>`* or *`<function_list>`* only to the lines that aren't selected by the address(es).

`[0addr]`	`:`*`<label>`*	This function does nothing. It bears a *`<label>`* to which the `b` and `t` commands may branch.
`[1addr]`	`=`	Writes the line number to the standard output followed by a newline character.
`[0addr]`		Empty lines are ignored.

[0addr]	#	# and the remainder of the line are ignored (treated as a comment), with the single exception that if the first two characters in the file are #n, the default output is suppressed. This is the same as specifying the -n option in the command line.
[2addr]	y/*<string1>*/*<string2>*	Replaces all occurrences of the characters in *<string1>* in the pattern space with the corresponding characters from *<string2>*. Any character other than a backslash or newline can be used instead of a slash to delimit the strings. Within *<string1>* and *<string2>*, a backslash followed by any character other than a newline is that literal character, and a backslash followed by an n is replaced by a newline character.
[2addr]	-s/*<regular-expression>*/*<replacement>*/*<flags>*	Substitutes the replacement string for the first instance of the regular expression in the pattern space. Any character other than the backslash or newline can be used instead of a slash to delimit the regular expression. Within the regular expression and the replacement, the regular expression delimiter itself can be used as a literal character if it's preceded by a backslash. An ampersand appearing in the replacement is replaced by the string matching the regular expression. The special meaning of the & in this context can be suppressed by preceding it with a backslash. The string \# where # is a digit is replaced by the text matched by the corresponding back reference expression (see re_format (7)). A line can be split by substituting a newline character into it. To specify a newline character in the replacement string, precede it with a backslash. The value of *<flags>* in the substitute function is zero or more of the following: 0 ... 9 Makes the substitution only for the nth occurrence of the regular expression in the pattern space. g Makes the substitution for all nonoverlapping matches of the regular expression, not just the first one.

p Writes the pattern space to standard output if a replacement was made. If the replacement string is identical to that which it replaces, it's still considered to have been a replacement.

w <file> Appends the pattern space to <file> if a replacement was made. If the replacement string is identical to that which it replaces, it's still considered to have been a replacement.

sftp

sftp Secure file transfer program.

```
sftp [-vC1] [-b <batchfile>] [-o <ssh_option>] [-s <subsystem> | <sftp_server>] [-B
<buffer_size>] [-F <ssh_config>] [-P <sftp_server path>] [-R <num_requests>] [-S
<program>] <host>
sftp [[<user>@]<host>[:<file1> [<file2>]]]
sftp [[<user>@]<host>[:<dir>[/]]]
```

The first usage initiates an interactive session.

The second usage retrieves files automatically if a non-interactive authentication is used. Otherwise, it retrieves the specified files after interactive authentication.

The third usage causes sftp to start in an interactive session in the specified directory.

-b <batchfile> Batch mode. Reads a series of commands from an input batchfile instead of stdin. Because it lacks user interaction, it should be used in conjunction with non-interactive authentication. sftp aborts if any of the following commands fail: get, put, rename, ln, rm, mkdir, chdir, lchdir, and or lmkdir.

-o <ssh_option> Passes options to ssh in the format used in the ssh configuration file. Useful for specifying options for which there's is no separate sftp command-line flag. For example, to specify an alternate port, use: sftp -oPort=24.

-s <subsystem> | <sftp_server> Specifies the SSH2 subsystem or the path for an sftp server on the remote host. A path is useful for using sftp over protocol version 1, or when the remote sshd doesn't have an sftp subsystem configured.

-v Raises logging level. Option is also passed to ssh.

-B <buffer_size> Specifies the size of the buffer that sftp uses when transferring files. Larger buffers require fewer round trips at the cost of higher memory consumption. Default is 32768 bytes.

-C Enables compression (via ssh's –C flag).

-F <ssh_config> Specifies an alternative per-user configuration file for ssh. Option is passed directly to ssh.

-P <sftp_server path> Connects directly to a local sftp-server (rather than via ssh). May be useful in debugging the client and server.

-R <num_requests>	Specifies how many requests may be outstanding at any one time. Increasing this may slightly improve file transfer speed but will increase memory usage. Default is 16 outstanding requests.
-S <program>	Specifies <program> as the program to use for the encrypted connection. The program must understand ssh options.
-1	Specifies the use of protocol version 1.
Interactive Commands	
bye	Quits sftp.
cd <path>	Changes remote directory to <path>.
lcd <path>	Changes local directory to <path>.
chgrp <grp> <path>	Changes group of file <path> to <grp>.
chmod <mode> <path>	Changes permissions of file <path> to <mode>.
chown <owner> <path>	Changes owner of file <path> to <owner>.
exit	Quits sftp.
get [<flags>] <remote-path> [<local-path>]	Retrieves the <remote-path> and stores it on the local machine. If the local pathname isn’t specified, it’s given the same name it has on the remote machine. If the -P flag is specified, the file’s full permission and access time are copied, too.
help	Displays help text.
lls [<ls-options> [<path>]]	Displays local directory listing of either <path> or current directory if <path> isn’t specified.
lmkdir <path>	Creates local directory specified by <path>.
ln <oldpath> <newpath>	Creates a symbolic link from oldpath to newpath.
lpwd	Prints local working directory.
ls [<path>]	Displays remote directory listing of either <path> or current directory if <path> isn’t specified.
lumask <umask>	Sets local umask to umask.
mkdir <path>	Creates remote directory specified by <path>.
put [<flags>] <local-path> [<remote-path>]	Uploads <local-path> and stores it on the remote machine. If the remote pathname isn’t specified, it’s given the same name it has on the local machine. If the -P flag is specified, the file’s full permission and access time are copied too.
pwd	Displays remote working directory.
quit	Quits sftp.
rename <oldpath> <newpath>	Renames remote file from <oldpath> to <newpath>.
rmdir <path>	Removes remote directory specified by <path>.
rm <path>	Deletes remote file specified by <path>.
symlink <oldpath> <newpath>	Creates a symbolic link from <oldpath> to <newpath>.
! <command>	Executes command in local shell.
!	Escapes to local shell.
?	Synonym for help.

showmount

`showmount`	Shows remote NFS mounts on host.
`showmount [-ade3] [<host>]`	
`-a`	Lists all mount points in the form `<host>:<dirpath>`.
`-d`	Lists directory paths of mount points instead of hosts.
`-e`	Shows the export list of `<host>`.
`-3`	Uses mount protocol version 3, compatible with NFS version 3.

shutdown

`shutdown`	Closes down the system at a given time.

`shutdown [-] [-fhkrn] <time> [<warning_message>]`

`shutdown` provides an automatic way for the superuser to politely notify users of an impending shutdown.

`-f`	`shutdown` arranges for file systems not to be checked upon reboot.
`-h`	Halts the system at the specified `<time>` when shutdown executes `halt` (8).
`-k`	Kicks everybody off. The `-k` option doesn't actually halt the system, but does leave the system multiuser with logins disabled for all users except the superuser.
`-r`	Shuts the system down and executes `reboot` (8) at the specified `<time>`.
`-n`	Prevents normal `sync` (2) before stopping.
`<time>`	The time when the system is to be brought down. `<time>` can be the word `now` for immediate shutdown, or a future time in one of two formats: `<+number>` or `<yymmddhhmm>`, where the year, month, and day may be defaulted to the current system values. The first form brings the system down in `<number>` minutes and the second at the absolute time specified.
`<warning_message>`	Any other arguments comprise the warning message that's broadcast to users currently logged on the system.
`-`	Reads the warning message from standard input.

Starting at 10 hours before shutdown, the system displays the shutdown warning message. Warning messages are displayed at regular intervals, with the messages being displayed more frequently as impending shutdown approaches. Five minutes before shutdown, or immediately, if shutdown is in less than five minutes, logins are disabled by creating an `/etc/nologin` and copying the warning message there. The file is removed just before shutdown occurs.

At shutdown time, a message is written in the system log, with the time of shutdown, who initiated shutdown, and the reason.

ssh

`ssh`	Secure shell.
`slogin`	Secure shell remote login client.

```
ssh [-l <login_name>] <hostname> | <user>@<hostname> [<command>]
ssh [-aAfgknqtTvxXCNP1246] [-b <bind_address>] [-c <cipher_spec>] [-e <escap_char>] [-i
<identity_file>] [-l <login_name>] [-m <mac_spec>] [-o <option>] [-p <port>] [-F
<configfile>] [-L <port>:<host>:<hostport>] [-R <port>:<host>:<hostport>] [-D <port>]
[<hostname> | <user>@<hostname>] [<command>]
```

`-a`	Disables forwarding of the authentication agent connection.
`-A`	Enables forwarding of the authentication agent connection. This can also be specified on a per-host basis in a configuration file.
`-f`	Requests `ssh` to go to background just before command execution. Implies `-n`. The recommended way to start X11 programs at a remote site is `ssh -f` *`<host>`* `xterm`.
`-g`	Allows remote hosts to control local forwarded ports.
`-k`	Disables forwarding of Kerberos tickets and AFS tokens. This may also be specified on a per-host basis in a configuration file.
`-n`	Redirects `stdin` from `/dev/null`.
`-q`	Quiet mode. Causes warning and diagnostic messages to be suppressed.
`-t`	Forces pseudo-tty allocation. Useful for executing arbitrary screen-based programs on a remote machine.
`-T`	Disables pseudo-tty allocation.
`-v`	Verbose mode. Causes debugging messages to be printed.
`-x`	Disables X11 forwarding.
`-X`	Enables X11 forwarding. This can also be specified on a per-host basis in a configuration file.
`-C`	Requests compression of all data.
`-N`	Doesn’t execute a remote command. Useful for just forwarding ports. SSH2 only.
`-P`	Uses a nonprivileged port for outgoing connections. Useful if your firewall doesn’t permit connections from privileged ports. Turns off RhostsAuthentication and RhostsRSAAuthentication.
`-1`	Forces SSH1 protocol only.
`-2`	Forces SSH2 protocol only.
`-4`	Forces `ssh` to use IPv4 addresses only.
`-6`	Forces `ssh` to use Ipv6 addresses only.
`-b` *`<bind_address>`*	Specifies the interface to transmit from on machines with multiple interfaces or aliased addresses.
`-c blowfish|3des|des`	Selects the cipher to use for the session. 3des is the default.
`-c` *`<cipher_spec>`*	Additionally, for SSH2, a comma-separated list of ciphers.

`-e ch|^ch|none` Sets escape character for sessions with a pty (default: ~). The escape character is only recognized at the beginning of a line. Followed by a `.` closes the connection; followed by `^Z` suspends the connection; followed by itself sends the escape character once. Setting it to none disables any escapes and makes the session fully transparent.

`-i <`*`identity_file`*`>` Specifies the file from which the identity (private key) for RSA authentication is read. Default is `$HOME/.ssh/identity`.

`-l <`*`login_name`*`>` Specifies the user to log in as on the remote machine. This may also be specified on a per-host basis in a configuration file.

`-m <`*`mac_spec`*`>` Additionally, for SSH2, a comma-separated list of MAC (message authentication code) algorithms can be specified in order of preference.

`-o <`*`option`*`>` Can be used for giving options in the format used in the configuration file. Useful for specifying options that have no separate command-line flag. Option has the same format as a line in the configuration file.

`-p <`*`port`*`>` Specifies the port to connect to on the remote host. This can be specified on a per-host basis in the configuration file.

`-D <`*`port`*`>` Specifies a local dynamic application-level port forwarding. Currently the SOCKS4 protocol is supported, and `ssh` acts as a SOCKS4 server. Dynamic port forwardings can also be specified in the configuration file.

`-F <`*`configfile`*`>` Specifies an alternative per-user configuration file. If a configuration file is given on the command line, the systemwide configuration file (`/etc/ssh_config`) is ignored. Default per-user configuration file is `$HOME/.ssh/config`.

`-L <`*`port`*`>:<`*`host`*`>:<`*`hostport`*`>` Specifies that the given port on the client (local) host is to be forwarded to the given host and port on the remote side.

`-R <`*`port`*`>:<`*`host`*`>:<`*`hostport`*`>` Specifies that the given port on the remote (server) host is to be forwarded to the given host and port on the local side.

ssh-agent

`ssh-agent` Authentication agent.

`ssh-agent [-a <`*`bind_address`*`>] [-c | -s] [-d]<`*`command`*`> <`*`args`*`> ...`

`ssh-agent [-c | -s] -k`

`-a <`*`bind-address`*`>` Bind the agent to the `unix-domain` socket *`bind_address`*. The default is `/tmp/ssh-XXXXXXXX/agent.<`*`ppid`*`>`.

`-c` Generates C-shell commands on stdout. Default if `SHELL` looks like it’s a csh-style shell.

`-s` Generates Bourne shell commands on stdout. Default if `SHELL` doesn’t look like it’s a csh-style shell.

`-k` Kills the current agent (given by the `SSH_AGENT_PID` environment variable).

`-d` Debug mode. When this option is specified, `ssh-agent` doesn't fork.

`<`*`command`*`>` When given, is executed as a subprocess of the agent. When the command dies, so does the agent.

`ssh-agent` holds private keys used for public key authentication (RSA, DSA). `ssh-agent` starts at the beginning of an X session or a login session, and all other programs or windows are started as clients of the `ssh-agent` program. Through the use of environment variables, the agent can be located and automatically used for authentication when logging in to other machines using `ssh`(1).

sshd

`sshd` OpenSSH daemon.

```
sshd [-deiqtD46] [-b <bits>] [-f <config_file>] [-g <login_grace_time>] [-h
<host_key_file>] [-k <key_gen_time>] [-o <option>] [-p <port>] [-u <len>]
```

`-b <bits>` Specifies the number of bits in the ephemeral protocol version 1 server key (default 768).

`-d` Debug mode. The server sends verbose debug output to the system log, and doesn't put itself in the background. The server also doesn't fork and only processes one connection. Intended for debugging for the server. Multiple `–d` options increase the debugging level. Maximum is 3.

`-e` Sends output to standard error instead of `/var/log/system.log`.

`-f <configuration_file>` Specifies the name of the configuration file. Default is `/etc/sshd_config`. `sshd` refuses to start if there's is no configuration file.

`-g <login_grace_time>` Gives the grace time for clients to authenticate themselves. If the client fails to authenticate the user within this many seconds, the server disconnects and exits. A value of zero indicates no limit. Default is 600 seconds.

`-h <host_key_file>` Specifies a file from which a host key is read. This option must be given if `sshd` isn't run as root (as the normal host key files aren't normally not readable by anyone but root). Defaults are `/etc/ssh_host_key` for protocol version 1, and `/etc/ssh_host_rsa_key` and `/etc/ssh_host_dsa_key` for protocol version 2. It's possible to have multiple host key files for the different protocol versions and host key algorithms.

`-i` Runs `sshd` from `inetd`. `sshd` is normally not run from `inetd` because it needs to generate the server key before it can respond to the client, and this may take tens of seconds. Clients would have to wait too long if the key was regenerated every time. However, with small key sizes (for example, 512) using `sshd` from `inetd` may be feasible.

`-k <key_gen_time>` Specifies how often the ephemeral protocol version 1 server key is regenerated. The motivation for regenerating the key fairly often is that the key isn't stored anywhere, and after about an hour, it becomes impossible to recover the key for decrypting intercepted communications even if the machine is cracked into or physically seized. A value of zero indicates that the key will never be regenerated. Default is 3600 seconds or 1 hour.

`-o <option>` Can be used to give options in the format used in the configuration file. Useful for specifying options for which there's is no separate command-line flag.

`-p <port>` Specifies the port on which the server listens for connections. Multiple port options are permitted. Ports specified in the configuration file are ignored when a command-line port is specified. Default is `22`.

`-q` Quiet mode. Sends no output to `/var/log/system.log`.

`-t` Test mode. Only checks the validity of the configuration file and sanity of the keys. Useful for updating `sshd` reliably as configuration options may change.

`-u <len>` Specifies the size of the field in the `utmp` structure that holds the remote hostname. If the resolved hostname is longer than *`<len>`*, the dotted decimal value will be used instead. This allows hosts with very long hostnames that overflow this field to still be uniquely identified. Specifying `-u0` indicates that only dotted decimal addresses should be put into the `utmp` file. `-u0` is also used to prevent `sshd` from making DNS requests unless the authentication mechanism or configuration requires it. Authentication mechanisms that may require DNS include `RhostsAuthentication`, `RhostsRSAAuthentication`, `HostbasedAuthentication`, and using a `from="pattern-list"` option in a key file. Configuration options that require DNS include using a `USER@HOST` pattern in `AllowUsers` or `DenyUsers`.

`-D` `sshd` doesn't detach and doesn't become a daemon. Allows for easy monitoring of `sshd`.

`-4` Forces `sshd` to use IPv4 addresses only.

`-6` Forces `sshd` to use IPv6 addresses only.

ssh-keygen

`ssh-keygen` Tool for authentication key generation, management, and conversion.

```
ssh-keygen [-q] [-b <bits>] -t <type> [-N <new_passphrase>] [-C <comment>]  [-f
<output_keyfile>]
ssh-keygen -p [-P <old_passphrase>] [-N <new_passphrase>] [-f <keyfile>]
```

```
ssh-keygen -i [-f <input_keyfile>]
ssh-keygen -e [-f <input_keyfile>]
ssh-keygen -y [-f <input_keyfile>]
ssh-keygen -c [-P <passphrase>] [-C <comment>] [-f <keyfile>]
ssh-keygen -l [-f <input_keyfile>]
ssh-keygen -B [-f <input_keyfile>]
```

`ssh-keygen` generates, manages, and converts authentication keys for `ssh`. `ssh-keygen` can create RSA keys for use by 1, and RSA or DSA keys for use by SSH2. The type of key to be generated is specified with the `-t` option.

Normally each user who wants to use SSH with RSA or DSA authentication runs this once to create the authentication key in `$HOME/.ssh/identity`, `$HOME/.ssh/id_dsa`, or `$HOME/.ssh/id_rsa`. Additionally, the system administrator may use this to generate host keys.

`-b <`*`bits`*`>` Specifies the number of bits in the key to create. Minimum is 512 bits. Generally 1024 bits is considered sufficient, and key sizes above that no longer improve security but make things slower. Default is 1024 bits.

`-c` Requests the changing of the comment in the private and public key files. This operation is only supported for RSA1 keys.

`-e` Reads a private or public OpenSSH key file and prints the key in a SECSH Public Key File Format to `stdout`. This option allows exporting keys for use by several commercial SSH implementations.

`-f <`*`filename`*`>` Specifies the filename of the key file.

`-i` Reads an unencrypted private (or public) key file in SSH2-compatible format and prints an OpenSSH-compatible private (or public) key to STDOUT. `ssh-keygen` also reads the SECSH Public Key File Format. This option allows importing keys from several commercial SSH implementations.

`-l` Shows fingerprint of specified public key file. Private RSA1 keys are also supported. For RSA and DSA keys, `ssh-keygen` tries to find the matching public key file and prints its fingerprint.

`-p` Requests the changing of the passphrase of a private key file instead of creating a new private key.

`-q` Quiet mode. Silences `ssh-keygen`.

`-y` Reads a private OpenSSH format file and prints an OpenSSH public key to stdout.

`-t <`*`type`*`>` Specifies the type of the key to create. The possible values are `rsa1` for protocol version 1 and `rsa` or `dsa` for protocol version 2.

`-B` Show the bubblebabble digest of specified private or public key file.

`-C <`*`comment`*`>` Provides the new comment.

`-N <`*`new_passphrase`*`>` Provides the new passphrase, <*new_passphrase*>.

`-P <`*`passphrase`*`>` Provides the (old) passphrase, <*passphrase*>.

Files

$HOME/.ssh/identity	Contains the protocol version 1 RSA authentication identity of the user. This file shouldn't be readable by anyone but the user. It's possible to specify a passphrase when generating the key; that passphrase will be used to encrypt the private part of this file using 3DES. File isn't automatically accessed by ssh-keygen but is offered as the default file for the private key. ssh reads this file when a login attempt is made.
$HOME/.ssh/identity.pub	Contains the protocol version 1 RSA public key for authentication. The contents of this file should be added to $HOME/.ssh/authorized_keys on all machines where the user wants to log in using RSA authentication. There's is no need to keep the contents of this file secret.
$HOME/.ssh/id_dsa	Contains the protocol version 2 DSA authentication identity of the user. This file shouldn't be readable by anyone but the user. It's possible to specify a passphrase when generating the key; that passphrase will be used to encrypt the private part of this file using 3DES. This file isn't automatically accessed by ssh-keygen but it's offered as the default file for the private key. ssh reads this file when a login attempt is made.
$HOME/.ssh/id_dsa.pub	Contains the protocol version 2 DSA public key for authentication. The contents of this file should be added to $HOME/.ssh/authorized_keys on all machines where the user wants to log in using public key authentication. There's is no need to keep the contents of this file secret.
$HOME/.ssh/id_rsa	Contains the protocol version 2 RSA authentication identity of the user. This file shouldn't be readable by anyone but the user. It's possible to specify a passphrase when generating the key; that passphrase will be used to encrypt the private part of this file using 3DES. This file isn't automatically accessed by ssh-keygen, but it's offered as the default file for the private key. ssh reads this file when a login attempt is made.
$HOME/.ssh/id_rsa.pub	Contains the protocol version 2 RSA public key for authentication. The contents of this file should be added to $HOME/.ssh/authorized_keys on all machines where the user wants to log in using public key authentication. There's is no need to keep the contents of this file secret.

sort

`sort` Sorts lines of text.

```
sort [-cmus] [-t <separator>] [-o <output_file>] [-bdfiMnr]] [+<POS1> [-<POS2>]] [-k
<POS1> [,<POS2>]] [<file> ...]
```

`sort` sorts, merges, or compares all the lines from the given files, or the standard input if no files are given. A name of `-` means standard input. By default, `sort` writes the results to standard output.

`sort` has three modes of operation: sort (default), checking for sortedness, and merge. These options affect the mode of operation:

`-c` Checks whether given files are already sorted. If they aren't all sorted, prints an error message and exits with a status of 1.

`-m` Merges the given files by sorting them as a group. Each input file should already be sorted. It always works to sort rather than merge. merge is an option because it's faster when it's set up properly.

If any key fields are specified, `sort` compares each pair of fields, in the order specified on the command line, according to associated ordering options, until a difference is found or no fields are left.

If any global options (`Mbdfinr`) are given, but no key fields are specified, `sort` compares lines according to global options.

If all keys compare equal, or if no ordering options were specified at all, `sort` compares lines byte-by-byte in machine collating sequence. The `-s` option disables the last resort comparison, producing a stable report.

The following options affect the ordering of the output lines. They may be specified globally or as part of a specific key field.

`-b` Ignores leading blanks when finding sort keys in each line.

`-d` Sorts in dictionary order; ignores all characters except letters, digits, and blanks.

`-f` Folds lowercase characters into the equivalent uppercase characters.

`-i` Ignores characters outside the ASCII range 040–0176 (inclusive).

`-M` Compares as months. The first three nonblank characters are folded into lowercase and sorted `jan < feb < ... < dec`. Invalid names compare low to valid names. Option implies `-b`.

`-n` Compares by arithmetic value an initial numeric string consisting of any amount of whitespace, an optional `-` sign, and zero or more digits. Option implies `-b`.

`-r` Reverses the result of the comparison so that lines of greater value appear earlier rather than later in the sort.

Other available options:

`-o <output_file>` Writes to the specified `<output_file>` instead of to standard output. If `<output_file>` is one of the input files, `sort` writes to a temporary file before writing to the `<output_file>`.

`-t <separator>` Uses character `<separator>` as the field separator when finding the sort keys in each line. By default, fields are separated by the empty string between a non-whitespace character and a whitespace character.

`-u` For default case or `-m` option, outputs the first of a sequence of lines that compare equal. For `-c` option, checks that no pair of consecutive lines compare equal.

`+<POS1> [-<POS2>]` Specifies a field within each line to use as a sorting key. The field consists of the portion of the line starting with `<POS1>` and up to but not including `<POS2>`, or to the end of the line, if `<POS2>` isn't specified. The fields and character positions are numbered starting with 0.

`-k <POS1>[,<POS2>]` Alternative syntax for specifying sorting keys. The fields and character positions are numbered starting with 1.

`-s` Disables the last resort comparison.

A position has the form `f.c`, where `f` is the number of the field to use, and `c` is the number of the first character from the beginning of the field (for `+POS`) or from the end of the previous field (`-POS`). The `.c` part of a position may be omitted, in which case it's taken to be the first character in the field. If the `-b` option has been given, the `.c` part of a field specification is counted from the first nonblank character of the field (for `+POS`) or from the first nonblank character following the previous field (`-POS`).

A `+POS` or `-POS` argument may also have any of the option letters `Mbdfinr` appended to it, in which case the global ordering options aren't used for that particular field. The `-b` option may be independently attached to either or both of the `+POS` and `-POS` parts of a field specification, and if it's inherited from the global options, it will be attached to both. If a `-n` or `-M` option is used, thus implying a `-b` option, the `-b` option is taken to apply to both `+POS` and `-POS` parts of a key specification. Keys may span multiple fields.

strings

`strings` Finds the printable strings in an object or binary file.

```
strings [-] [-a] [-o] [-<number>] [<file> ...]
```

`strings` looks for ASCII strings in binary files or standard input. `strings` is useful for identifying random object files and many other things. A string is any sequence of four (the default) or more printing characters ending with a newline or a `null`. Unless the `-` flag is given, `strings` looks in all sections of the object files except the (`_TEXT`, `_text`) section. If no files are specified, standard input is read.

`-` Looks for strings in all bytes of the files (the default for non-object files).

`-a` Looks for strings in all sections of the object file (including the (`_TEXT`, `_text`) section).

`-o` Writes each string preceded by its byte offset from the start of the file.

`-<number>` The decimal `<number>` is used as the minimum string length rather than the default of four.

su

`su` Substitute user identity.

`su [-flm] [<login>] [-c <shell arguments>]`

`su` requests the password for login and switches to that user and group ID after obtaining proper authentication. A shell is then executed, and any additional shell arguments after the login name are passed to the shell.

If `su` is executed with no user name as an argument, root is assumed.

If `su` is executed by root, no password is requested and a shell with the appropriate user ID is executed.

`-c` Invokes the following command in a subshell as the specified user.

`-f` If the invoked shell is `csh`(1), this option prevents it from reading the .cshrc file.

`-l` Simulates a full login. The environment is discarded except for `HOME`, `SHELL`, `PATH`, `TERM`, and `USER`. `HOME` and `SHELL` are modified as abovementioned earlier. `USER` is set to the target login. `PATH` is set to `/bin:/usr/bin`. `TERM` is imported from your current environment. The invoked shell is the target login's, and `su` will change directory to the target login's home directory. The `-l` option is synonymous with `-`, as in `su -`.

`-m` Leaves the environment unmodified. The invoked shell is your login shell, and no directory changes are made. As a security precaution, if the target user's shell is a non-standard shell (not listed in `/etc/shells`) and the caller's real uid is non-zero (not root), `su` will fail.

`-s` Silent mode. Shows only what would be done, but doesn't send any signal.

The `-l` and `-m` options are mutually exclusive; the last one specified overrides any previous ones.

Only users in group wheel (normally gid 0) or group admin (normally gid 20) can `su` to root.

By default, unless the prompt is reset by a startup file, the superuser prompt is set to `#` to remind you of its awesome power.

sudo

`sudo` Executes a command as another user.

```
sudo -V|-h|-l|-L|-v|-k|-K|[-H][-P][-S][-b] | [-p <prompt>]  [-u <username>|<#uid>]
<command>
sudo -V|-h|-l|-L|-v|-k|-K|[-H][-P][-S][-b] | [-p <prompt>]  [-u <username>|<#uid>] -s
```

`sudo` allows a permitted user to execute a *<command>* as `root` or another user, as specified in `/etc/sudoers`. The real and effective `uid` and `gid` are set to match those of the target user as specified in the `passwd` file or Netinfo map. By default, `sudo` requires that users authenticate themselves with a password (Note: by default this is the user's password, not the `root` password.) When a user has been authenticated, a timestamp is updated and the user may then use `sudo` without a password for a short period of time after the timestamp (5 minutes unless overridden in sudoers). The timestamp is updated every time a command is executed through sudo, providing a sliding window during which the user may use commands as the alternate user without re-entering the required password.

sudo determines who is an authorized user by consulting the file /etc/sudoers. By giving sudo the -v flag, a user can update the timestamp without running a command.

If a user who isn't listed in /etc/sudoers tries to run a command via sudo, mail is sent to the proper authorities, as defined at configure time or /etc/sudoers. Note that the mail will not be sent if an unauthorized user tries to run sudo with the -l or -v flags. This allows users to determine for themselves whether or not they are're allowed to use sudo.

sudo can log attempts as well as errors to syslog(3), a log file, or both. By default sudo will log via syslog(3).

When used with the -s option instead of a <*command*>, sudo executes the target user's shell in a manner similar to the su command. The change of effective user, and executing of the shell are logged, but commands executed while in that shell aren't recorded.

-V — Causes sudo to print the version number and exit. If the invoking user is root, the -V option will print out a list of the defaults sudo was compiled with.

-l — Lists out the allowed (and forbidden) commands for the user on the current host.

-L — Lists out the parameters that may be set in a Defaults line along with a short description for each. This option is useful in conjunction with grep.

-h — Causes sudo to print a usage message and exit.

-v — Updates the user's timestamp, prompting for the user's password if necessary. This extends the sudo timeout for another 5 minutes (or whatever the timeout is set to in sudoers).

-k — Invalidates the user's timestamp by setting the time on it to the epoch. The next time sudo is run, a password will be required. This option doesn't require a password and was added to allow a user to revoke sudo permissions from a .logout file.

-K — Removes the user's timestamp entirely. Like -k, this option doesn't require a password.

-b — Tells sudo to run the given command in the background. Note that if you use the -b option, you cannot use shell job control to manipulate the process.

-p — Allows you to override the default password prompt and use a custom one. If the password prompt contains the %u escape, %u will be replaced with the user's login name. Similarly, %h will be replaced with the local hostname.

-S — Causes sudo to read the password from standard input instead of the terminal device.

-P Preserves the calling user's group vector unaltered. By default, sudo will initialize the group vector to the list of groups the target user is in. The real and effective group IDs, however, are still set to match the target user.

-H Sets the $HOME environment variable to the homedir of the target user (root by default) as specified in /etc/passwd or Netinfo. By default, sudo doesn't modify $HOME.

sudo tries to be safe when executing commands. To accomplish this, most shell variables specifying load paths for dynamically loaded libraries, user paths and similar routes by which commands may be spoofed, are ignored when searching for commands and when loading dynamic modules. This will not affect general use of the sudo command, but may result in unexpected behavior in some situations. Carefully read Apple's man page for sudo (which isn't quite in sync with the version of the command provided) if you experience difficulty with more sophisticated configurations.

SystemStarter

SystemStarter Starts, stops, and restarts system services.

SystemStarter [-gvxdDqn] [<*action*> [<*service*>]]

The SystemStarter utility may be used to start, stop, and restart the system services, which are described in the /Library/StartupItems/ and /System/Library/StartupItems/ paths.

The optional <*action*> argument specifies which action SystemStarter performs on the startup items.

The optional <*service*> argument specifies which startup items to perform the action on. If no service is specified, all startup items are acted on; otherwise, only the item providing the service, any items it requires, or any items that depend on it will be acted on.

During boot, SystemStarter is invoked by rc(8) and is responsible for starting all startup items in an order that satisfies each item's requirements.

Actions

start Starts all items, or starts the item that provides the specified <*service*> and all items providing services it requires.

stop Stops all items, or stops the item that provides the specified <*service*> and all items that depend on it.

restart Restarts all items, or restarts the item providing the specified <*service*>.

Options

-g Graphical startup.

-v Verbose (text mode) startup.

-x Safe mode startup (only runs Apple-provided items).

-d Prints debugging output.

-D Prints debugging output and dependencies.

-q Quiet (disables debugging output).

-n Doesn't actually perform action on items (no-run mode).

tail

`tail` Displays the last part of a file.

```
tail [-f | -F | -r] [-b <number> | -c <number> | -n <number>] <file>
tail [-f | -F | -r] [-b <number> | -c <number> | -n <number>]
```

`-f` Waits for and displays additional data that `<file>` receives, rather than stopping at the end of the file.

`-F` Similar to `-f`, except that every five seconds, `tail` checks whether `<file>` has been shortened or moved. If so, `tail` closes the current file, opens the filename given, displays its entire contents, and waits for more data. This option is especially useful for monitoring log files that undergo rotation.

`-r` Displays the file in reverse order, by line. The default is to display the entire file in reverse. This option also modifies the `-b`, `-c`, and `-n` options to specify the number of units to be displayed, rather than the number of units to display from the beginning or end of the input.

`-b <number>` Specifies location in number of 512-byte blocks.

`-c <number>` Specifies location in number of bytes.

`-n <number>` Specifies location in number of lines.

tar

`tar` Creates, extracts, or appends to tape archives.

```
tar [-] <c | t | x | r | u> [fbemopvwzZhHLPX014578] [<archive>] [<blocksize>] [-C
<directory>] [-s <replstr>] <file1> <file2> ...
```

`tar` saves files to and restores files from a single file. Although that single file might have originally been intended to be magnetic tape, magnetic tape isn't required.

One of the following flags is required:

`-c` Creates a new archive or overwrites an existing one.

`-t` Lists the contents of an archive. If any files are listed on the command line, only those files are listed.

`-x` Extracts files from an archive. If any files are listed on the command line, only those files are extracted. If more than one copy of a file exists in an archive, earlier copies are overwritten by later copies.

`-r` Appends the specified files to an archive. This works only on media on which an end-of-file mark can be overwritten.

`-u` Alias to `-r`.

In addition to the required flags, any of these options may be used:

`-f <archive>` Filename where the archive is stored. Default is `/dev/rmt8`.

`-b <blocksize>` Sets the blocksize to be used in the archive. Any multiple of 512 between 10240 and 32256 may be used.

-e	Stops after the first error.
-m	Doesn't preserve modification time.
-o	Doesn't create directories.
-p	Preserves user ID, group ID, file mode, and access and modification times.
-v	Verbose mode.
-w	Interactively renames files.
-z	Compresses(or uncompresses) the archive using gzip.
-Z	Compresses(or uncompresses) the archive using compress.
-h	Follows symbolic links as if they were normal files or directories.
-H	Follows symbolic links given on the command line only.
-L	Follows all symbolic links.
-P	Doesn't follow any symbolic links.
-X	Doesn't cross mount points in the file system.
[-014578]	Selects a backup device, /dev/rmtN, where N is the argument given.
-C *<directory>*	Sets the working directory for the files. When extracting, files are extracted into the specified directory. When creating, specified files are matched from the directory.
-s *<replstr>*	Modifies the filenames or archive member names specified by the pattern or file operands according to the substitution expression *<replstr>*, using the syntax of ed(1) in this format: /*old*/*new*/[gp] *old* is the old expression. *new* is the new expression. The optional trailing g applies the substitution globally. That is, it continues to apply the substitution. The first unsuccessful substitution stops the g option. The optional trailing p causes the final result of a successful substitution to be written to standard error in this format: *<original pathname>* >> *<new pathname>* Multiple -s *<replstr>* options can be specified. They're applied in the order listed.

top

top	Displays system usage statistics.
top [-u] [-w] [-k] [-s *<interval>*] [-e | -d | -a] [-l *<samples>*] [*<number>*]	
top	
-u	Sorts by CPU usage and displays usage starting with the highest usage.
-w	Generates additional columns of output data. The additional columns include VPRVT and the delta information for #PRTS, RSHRD, RSIZE, and VSIZE.

`-k` Causes `top` to traverse and report the memory object map for pid 0 (kernel task). This option is optional because it's expensive to traverse the object maps, as the kernel task may have a large number of entries.

`-s <interval>` Samples processes at the specified `<interval>`. Default is one-second intervals.

`-e` Switches to event-counting mode where counts reported are absolute counters. Options `-w` and `-k` are ignored.

`-d` Switches to an event-counting mode where counts are reported as deltas relative to the previous sample. Options `-w` and `-k` are ignored.

`-a` Switches to an event-counting mode where counts are reported as cumulative counters relative to when `top` was launched. Options `-w` and `-k` are ignored.

`-l <samples>` Switches from default screen mode to a logging mode suitable for saving the output to a file. If `<samples>` is specified, `top` samples the number of samples specified before exiting. The default is `1`.

`<number>` Limits the number of processes displayed to `<number>`.

Pressing the `Q` key causes `top` to exit immediately.

Columns displayed in default data mode:

`PID`	Unix process ID
`COMMAND`	Unix command name
`%CPU`	Percentage of CPU used (kernel and user)
`TIME`	Absolute CPU consumption (min:secs.hundredths)
`#TH`	Number of threads
`#PRTS` (delta)	Number of MACH ports
`#MERG`	Number of memory regions
`VPRVT` (`-w` only)	Private address space currently allocated
`RPRVT` (delta)	Resident shared memory (as represented by the resident page count of each shared memory object)
`RSHRD` (delta)	Total resident memory (real pages that this process currently has associated with it; some may be shared by other processes)
`VSIZE` (delta)	Total address space currently allocated (including shared)

Columns displayed in event-counting modes:

`PID`	Unix process ID
`COMMAND`	Unix command name
`%CPU`	Percentage of CPU used (kernel and user)
`TIME`	Absolute CPU consumption (min:secs.hundredths)
`FAULTS`	Number of page faults
`PAGEINS`	Number of requests for pages from a pager
`COW_FAULTS`	Number of faults that caused a page to be copied
`MSGS_SENT`	Number of mach messages sent by the process
`MSGS_RCVD`	Number of mach messages received by the process

`BSDSYSCALL`	Number of BSD system calls made by the process
`MACHSYSCALL`	Number of MACH system calls made by the process
`CSWITCH`	Number of context switches to this process

touch

`touch`	Changes file access and modification times.

`touch [-acfhm] [-r <file>] [-t [[CC]YY]MMDDhhmm[.SS]] <file> ...`

`touch` sets modification and access times of files to the current time of day. If the file doesn’t exist, it’s created with default permissions.

`-a`	Changes the access time of the file. Doesn’t change modification time unless `-m` is also specified.
`-c`	Doesn’t create the file if it doesn’t exist.
`-f`	Attempts to force the update, even if file permissions don’t currently permit it.
`-h`	If `<file>` is a symbolic link, changes access and/or modification time of the link. This option also implies `-c`.
`-m`	Changes the modification time of the file. Doesn’t change the access time unless `-a` is also specified.
`-r <file>`	Replaces access and modification time with that of `<file>`, rather than using the current time.
`-t`	Changes the access and modification time to the specified time.

The argument for `-t` should be in the form `[[CC]YY]MMDDhhmm[.SS]`, where each pair of letters represents the following:

`CC`	First two digits of the year (the century).
`YY`	Second two digits of the year. If `YY` is specified but `CC` isn’t, a value for `YY` between `69` and `99` results in a `CC` value of `19`. Otherwise, a value of `20` is used.
`MM`	The month of the year, from `1` to `12`.
`DD`	The day of the month, from `1` to `31`.
`hh`	The hour of the day, from `0` to `23`.
`mm`	The minute of the hour, from `0` to `59`.
`SS`	The second of the minute, from `0` to `61`.

If `CC` and `YY` letter pairs aren’t specified, the values default to the current year. If the `SS` letter pair isn’t specified, the value defaults to `0`.

traceroute

`traceroute`	Prints the route packets take to a network host.

`traceroute [-d] [-m <max_ttl>] [-n] [-p <port>] [-q <nqueries>] [-r] [-s <src_addr>] [-t <tos>] [-w <waittime>] <host> [<packetsize>]`

`traceroute` uses the IP protocol time-to-live field and attempts to elicit an ICMP `TIME_EXCEEDED` response from each gateway along the path to the same host.

The only mandatory parameter is *<host>*, the destination host or IP number. The default probe datagram length is 38 bytes, but this can be increased by specifying a packet size (in bytes) after the destination host name.

`-d`	Turns on socket-level debugging.
`-m` *<max_ttl>*	Sets the maximum time-to-live (maximum number of hops) used in outgoing probe packets. The default is 30 hops. The same default is used for TCP connections.
`-n`	Prints hop addresses numerically rather than symbolically and numerically (saves a nameserver address-to-name lookup for each gateway found on the path).
`-p` *<port>*	Sets the base UDP port number used in probes to *<port>*. Default is 33434. `traceroute` hopes that nothing is listening on UDP *<base>* to *<base+nohops+1>* at the destination host, so that an ICMP `PORT_UNREACHABLE` message is returned to terminate the route tracing. If something is listening on a port in the default range, this option can be used to pick an unused port range.
`-q` *<nqueries>*	Sets the number of probes per ttl to *<nqueries>*. Default is three probes.
`-r`	Bypasses the normal routing tables and sends directly to a host on an attached network. If the host isn't on a directly attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.
`-s` *<src_addr>*	Uses the following IP address (which must be given as an IP number, not a host name) as the source address in outgoing probe packets. On hosts with more than one IP address, this option can be used to force the source address to be something other than the IP address of the interface that the probe packet is sent on. If the IP address isn't one of this machine's interfaces, an error is returned and nothing is sent.
`-t` *<tos>*	Sets the type-of-service in probe packets to *<tos>*. Default is 0. Value must be a decimal integer in the range 0 to 255. This option can be used to see whether different types-of-service result in different paths. Not all values of TOS are legal or meaningful. See the IP spec for definitions. Useful values are probably `-t 16` (low delay) and `-t 8` (high throughput).
`-v`	Sets to verbose output. Lists ICMP packets received other than `TIME_EXCEEDED` and `UNREACHABLE` packets.
`-w` *<waittime>*	Sets the time to wait for a response to a probe to *<waittime>* seconds. Default is three seconds.

tunefs

`tunefs` Tunes up an existing file system.

`tunefs [-AN] [-a <maxcontig>] [-d <rotdelay>] [-e <maxbpg>] [-m <minfree>] [-o <optimize_preference>] [<special> | <filesystem>]`

`tunefs` is designed to change the dynamic parameters of a file system that affect the layout policies. The parameters that are to be changed are indicated by the flags given here:

`-A` Causes the values to be updated in all the alternate superblocks instead of just the standard superblock. If this option isn't used, use of a backup superblock by `fsck`(8) will lose anything changed by `tunefs`. The `-A` flag is ignored when the `-N` flag is specified.

`-N` Displays all the settable options (after any changes from the tuning options) but doesn't cause any of them to be changed

`-a <maxcontig>` Specifies the maximum number of contiguous blocks that will be laid out before forcing a rotational delay (see `-d`). The default value is 1 because most device drivers require an interrupt per disk transfer. Device drivers that can chain several buffers together in a single transfer should set this to the maximum chain length.

`-d <rotdelay>` Specifies the expected time (in milliseconds) to service a transfer completion interrupt and initiate a new transfer on the same disk. It's used to decide how much rotational spacing to place between successive blocks in a file.

`-e <maxbpg>` Indicates the maximum number of blocks any single file can allocate out of a cylinder group before it's forced to begin allocating blocks from another cylinder group. Typically, this value is set to about one quarter of the total blocks in a cylinder group. The intent is to prevent any single file from using up all the blocks in a single cylinder group, thus degrading access times for all files subsequently allocated in that cylinder group. The effect of this limit is to cause big files to do long seeks more frequently than if they were allowed to allocate all the blocks in a cylinder group before seeking elsewhere. For file systems with exclusively large files, this parameter should be set higher.

`-m <minfree>` Specifies the percentage of space held back from normal users; the minimum free space threshold. The default value used is 10%. This value can be set to zero, but up to a factor of three in throughput will be lost over the performance obtained at a 10% threshold. Note that if the value is raised above the current usage level, users will be unable to allocate files until enough files have been deleted to get under the higher threshold.

`-o <optimize-preference>` The file system can either try to minimize the time spent allocating blocks, or it can attempt to minimize the space fragmentation on the disk. If the value of `minfree` (see above) is less than 10%, the file system should optimize for space to avoid running out of full-sized blocks. For values of `minfree` greater than or equal to 10%, fragmentation is unlikely to be problematical, and the file system can be optimized for time.

`-p` This option shows a summary of what the current tuneable settings are on the selected file system. More detailed information can be obtained in the `dumpfs`(8) manual page.

umount

`umount` Unmounts file systems.

`umount [-fv] <special> | <node>`

`umount -a | -A [-fv] [-h <host>] [-t <type>]`

`-f` Forcibly unmounts the file system. Active special devices continue to work, but all other files return errors if further accesses are attempted. The root file system cannot be forcibly unmounted.

`-v` Enables verbose mode.

`-a` All the file systems described in `fstab` (5) are unmounted.

`-A` All the currently mounted file systems except the root are unmounted.

`-h <host>` Unmounts only file systems mounted from the specified `<host>`. This option implies the `-A` option and, unless otherwise specified with the `-t` option, unmounts only NFS file systems.

`-t <type>_` Indicates that actions should only be taken on file systems of the specified `<type>`. More than one type may be specified in a comma-separated list. The list of file system types can be prefixed with `no` to specify the file system types for which action shouldn't be taken. For example, the `umount` command `umount -a -t nfs,mfs` unmounts all file systems of the type NFS and MFS.

uptime

`uptime` Shows how long the system has been running.

`uptime`

`uptime` displays the current time, the length of time the system has been up, the number of users, and the load average of the system over the last 1, 5, and 15 minutes.

uuencode, uudecode

`uuencode`	Encodes a binary file.
`uudecode`	Decodes a binary file.

```
uuencode [<file>] <name>
uudecode [<file> ...]
```

`uuencode` and `uudecode` are used to transmit binary files over transmission mediums that only support simple ASCII data.

`uuencode` reads `<file>` (or by default the standard input) and writes an encoded version to the standard output. The encoding uses only printing ASCII characters and includes the mode of the file and the operand `<name>` for use by uudecode.

`uudecode` transforms uuencoded files (or by default the standard input) into the original form. The resulting file is named `<name>` and has the mode of the original file except that setuid and execute bits aren't retained. `uudecode` ignores any leading and trailing lines.

vi, ex, view

`vi`	Screen-oriented text editor.
`ex`	Line-oriented screen editor.
`view`	Read-only version of `vi`.

```
vi [-eFlRrSv] [-c <cmd>] [-t <tag>] [-w <size>] [<file1> <file2> ...]
ex [-eFlRrSsv] [-c <cmd>] [-t <tag>] [-w <size>] [<file1> <file2> ...]
view [-eFlRrSv] [-c <cmd>] [-t <tag>] [-w <size>] [<file1> <file2> ...]
```

`vi` is a screen-oriented text editor; `ex` is a line-oriented editor. `vi` and `ex` are different interfaces to the same program. view is equivalent to `vi -R`, the read-only option to `vi`.

The following options are available:

`-e`	Starts to edit in ex mode.
`-F`	Doesn't copy the entire file when first starting to edit. Default is to make a copy in case someone else modifies the file during your edit session.
`-l`	Starts editing with the `lisp` and `showmatch` options set.
`-R`	Starts editing in read-only mode.
`-r`	Recovers the specified file. If no file is specified, it lists the files that could be recovered. If no recoverable files with the specified name exist, `vi` starts editing as if the option hasn't been issued.
`-S`	Runs with secure edit option set, which disallows all access to external programs.
`-s`	Enters batch mode. Applicable only to `ex`. It's useful for running `ex` scripts.
`-v`	Starts editing in `vi` mode.
`-c <cmd>`	Executes `<cmd>` immediately after starting the edit session. It's especially useful for initial positioning in the file, but isn't limited to positioning commands.

`-t <tag>` Starts editing at the specified `<tag>`.

`-w <size>` Sets the initial window size to `<size>` lines.

`vi` has two modes: command mode and input mode. Command mode is the initial and normal mode. Completion of the input mode returns the user to command mode. Pressing the Esc key ends a partial command.

Input mode is required to input some types of edits. Input mode is terminated by pressing the Esc key. Upon termination of input mode, the user is returned to command mode.

Some commands for moving around in a file:

`h` Moves the cursor one character to the left.

`l` Moves the cursor one character to the right.

`j` Moves the cursor one line down.

`k` Moves the cursor one line up.

`<arrow keys>` The arrow keys often also function properly.

`<num>G` Moves the cursor to the line number specified by `<num>`. If `<num>` isn't specified, the cursor moves to the last line of the file.

Some commands for inputting text (input mode):

`i` Inserts text before the cursor.

`a` Appends new text after the cursor.

`A` Appends new text at the end of the line where the cursor is.

`o` Opens a new line below the line where the cursor is and allows the user to start entering text on the new line.

`O` Opens a new line above the line where the cursor is, and allows the user to start entering text on that new line.

Some commands for copying text:

`yy` Copies the line the cursor is on.

`p` Appends the copied line after the line the cursor is on.

Some commands for deleting text:

`dd` Deletes the line the cursor is on.

`<num>dd` Deletes `<num>` lines, starting with the line the cursor is on.

`dw` Deletes the word the cursor is on.

`x` Deletes the character the cursor is on.

Some other useful text manipulation:

`r<x>` Replaces the character the cursor is on with `<x>`.

`J` Joins the line the cursor is on with the line below.

Some commands for pattern searching:

`/<pattern>` Searches forward in the file for `<pattern>`, starting with the location of the cursor.

`?<pattern>` Searches backward in the file for `<pattern>`, starting with the location of the cursor.

`n` Repeats the last `/` or `?` pattern search.

`N` Repeats the last `/` or `?` pattern search in reverse.

Some commands to write the file:

`:w<return>`	Writes the file back to the filename originally specified when `vi` was started.
`:w <filename><return>`	Writes the file to the filename specified by `<filename>`.

Some commands to quit editing and exit `vi`:

`:q<return>`	Exits `vi`. Refuses to quit if there are any unsaved modifications, or if the file is read-only.
`:q!`	Exits `vi`, even if there are any unsaved modifications.
`ZZ`	Exits `vi`, saving changes.

w

`w`	Displays who the present users are and what they're doing.

`w [-hin] [-M <core>] [-N <system>] [<user>]`

`w` displays a summary of the current activity on the system, including what each user is doing. The first line displays the current time of day, how long the system has been running, the number of users logged in to the system, and the load averages. The load average numbers give the number of jobs in the run queue average over 1, 5, and 15 minutes.

The output fields are the user's login name, the name of the terminal where the user is logged on, the host from which the user is logged in, the time the user logged in, the time since the user last typed anything, and the name and arguments of the current process.

`-h`	Suppresses the heading.
`-i`	Sorts output by idle time.
`-n`	Shows network addresses as numbers. Normally `w` interprets addresses and attempts to display them symbolically.
`-M <core>`	Extracts values associated with the name list from the specified core instead of the default `/dev/kmem`.
`-N <system>`	Extracts the name list from the specified system instead of the default `/netbsd`.
`<user>`	If specified, restricts output to `<user>`.

which

`which`	Locates a program file including `aliases` and `paths` (`csh`(1)) only).

`which <name1> <name2> ...`

`which` displays the location of the specified commands, and displays which files would have been executed had the names been given as commands. Both aliases and paths are taken from the user's `.cshrc` file.

who

`who` Displays who is logged in.

`who [-mTuH] [<file>]`

`who am i`

`who` displays a list of all users currently logged on, showing for each user the login name, tty name, the date and time of login, and hostname, if not local.

`-m` Only prints information about the current terminal (POSIX way of saying “Who am I?”).

`-T` Prints a character after the username indicating the state of the terminal line: `+` if the terminal is writable; `-` if it isn’t writable; `?` if a bad line is encountered.

`-u` Prints the idle time for each user.

`-H` Writes column headings above the regular output.

`am i` Returns the invoker’s real username.

`<file>` Gathers information from the specified `<file>`, rather than the default `/var/run/utmp`. An alternative `<file>` is usually `/var/log/wtmp`. The `wtmp` file contains a record of every login, logout, crash, shutdown and date change since `wtmp` was truncated or created. If `/var/log/wtmp` is being used as the file, the username may be empty or one of these special characters: `|`, `}`, `~`. Logouts produce an output line without any username.

whoami

`whoami` Displays the effective user ID.

`whoami`

`whoami` has been made obsolete by the `id`(1) utility, and is equivalent to `id -un`. The command `id -p` is suggested for normal interactive use.

`whoami` displays your effective user ID as a name.