NAME

find - walk a file hierarchy

SYNOPSIS

find [**-H** | **-L** | **-P**] [**-EXdsx**] [**-f** *path*] *path* ... [*expression*]
find [**-H** | **-L** | **-P**] [**-EXdsx**] **-f** *path* [*path* ...] [*expression*]

DESCRIPTION

The **find** utility recursively descends the directory tree for each *path* listed, evaluating an *expression* (composed of the "primaries" and "operands" listed below) in terms of each file in the tree.

The options are as follows:

-E   Interpret regular expressions following the **-regex** and **-iregex** primaries as extended (modern) regular expressions rather than basic regular expressions (BRE's). The re_format(7) manual page fully describes both formats.

-H   Cause the file information and file type (see stat(2)) returned for each symbolic link specified on the command line to be those of the file referenced by the link, not the link itself. If the referenced file does not exist, the file information and type will be for the link itself. File information of all symbolic links not on the command line is that of the link itself.

-L   Cause the file information and file type (see stat(2)) returned for each symbolic link to be those of the file referenced by the link, not the link itself. If the referenced file does not exist, the file information and type will be for the link itself.

This option is equivalent to the deprecated **-follow** primary.

-P   Cause the file information and file type (see stat(2)) returned for each symbolic link to be those of the link itself. This is the default.

-X   Permit **find** to be safely used in conjunction with xargs(1). If a file name contains any of the delimiting characters used by xargs(1), a diagnostic message is displayed on standard error, and the file is skipped. The delimiting characters include single (" ' ") and double (" " ") quotes, backslash ("\"), space, tab and newline characters.

However, you may wish to consider the **-print0** primary in conjunction with "**xargs -0**" as an effective alternative.

-d   Cause **find** to perform a depth-first traversal.

This option is a BSD-specific equivalent of the **-depth** primary specified by IEEE Std 1003.1-2001 ("POSIX.1"). Refer to its description under *PRIMARIES* for more information.

-f *path*

Add *path* to the list of paths that will be recursed into. This is useful when *path* begins with a character that would otherwise be interpreted as an *expression*, namely "!" , "(" and "-".

-s   Cause **find** to traverse the file hierarchies in lexicographical order, i.e., alphabetical order within each directory. Note: 'find -s' and 'find | sort' may give different results.

For example, 'find -s' puts a directory '*foo*' with all its contents before a directory '*foo*'. but 'find | sort' puts the directory name '*foo*'. before any string like '*foo/bar*' because '.' goes before '/' in ASCII. In locales other than *C* results may vary more due to collation differences.

-x   Prevent **find** from descending into directories that have a device number different than that of the file from which the descent began.

This option is equivalent to the deprecated **-xdev** primary.

PRIMARIES

All primaries which take a numeric argument allow the number to be preceded by a plus sign ("+") or a minus sign ("-"). A preceding plus sign means "more than n", a preceding minus sign means "less than n" and neither means "exactly n".

-Bmin *n*

True if the difference between the time of a file's inode creation and the time **find** was started, rounded up to the next full minute, is *n* minutes.

-Bnewer *file*

Same as **-newerBm**.

-Btime *n*[**smhdw**]

If no units are specified, this primary evaluates to true if the difference between the time of a file's inode creation and the time **find** was started, rounded up to the next full 24-hour period, is *n* 24-hour periods.

If units are specified, this primary evaluates to true if the difference between the time of a file's inode creation and the time **find** was started is exactly *n* units. Please refer to the **-atime** primary description for information on supported time units.

-acl   May be used in conjunction with other primaries to locate files with extended ACLs. See acl(3) for more information.

-amin [-|+]*n*
          True if the difference between the file last access time and the time **find** was started, rounded up to the next full minute, is more than *n* (+*n*), less than *n* (-*n*), or exactly *n* minutes ago.

-anewer *file*
          Same as **-neweram**.

-atime *n*[**smhdw**]
          If no units are specified, this primary evaluates to true if the difference between the file last access time and the time **find** was started, rounded up to the next full 24-hour period, is *n* 24-hour periods.

          If units are specified, this primary evaluates to true if the difference between the file last access time and the time **find** was started is exactly *n* units. Possible time units are as follows:

          s   second
          m   minute (60 seconds)
          h   hour (60 minutes)
          d   day (24 hours)
          w   week (7 days)

          Any number of units may be combined in one **-atime** argument, for example, "-atime -1h30m". Units are probably only useful when used in conjunction with the + or **-** modifier.

-cmin [-|+]*n*
          True if the difference between the time of last change of file status information and the time **find** was started, rounded up to the next full minute, is more than *n* (+*n*), less than *n* (-*n*), or exactly *n* minutes ago.

-cnewer *file*
          Same as **-newercm**.

-ctime *n*[**smhdw**]
          If no units are specified, this primary evaluates to true if the difference between the time of last change of file status information and the time **find** was started, rounded up to the next full 24-hour period, is *n* 24-hour periods.

If units are specified, this primary evaluates to true if the difference between the time of last change of file status information and the time **find** was started is exactly *n* units. Please refer to the **-atime** primary description for information on supported time units.

-d Non-portable, BSD-specific version of **depth**. GNU find implements this as a primary in mistaken emulation of FreeBSD **find**.

-delete

Delete found files and/or directories. Always returns true. This executes from the current working directory as **find** recurses down the tree. It will not attempt to delete a filename with a "/" character in its pathname relative to "." for security reasons. Depth-first traversal processing is implied by this option. The **-delete** primary will fail to delete a directory if it is not empty. Following symlinks is incompatible with this option.

-depth

Always true; same as the non-portable **-d** option. Cause **find** to perform a depth-first traversal, i.e., directories are visited in post-order and all entries in a directory will be acted on before the directory itself. By default, **find** visits directories in pre-order, i.e., before their contents. Note, the default is *not* a breadth-first traversal.

The **-depth** primary can be useful when **find** is used with cpio(1) to process files that are contained in directories with unusual permissions. It ensures that you have write permission while you are placing files in a directory, then sets the directory's permissions as the last thing.

-depth *n*

True if the depth of the file relative to the starting point of the traversal is *n*.

-empty

True if the current file or directory is empty.

-exec *utility* [*argument ...*] ;

True if the program named *utility* returns a zero value as its exit status. Optional *arguments* may be passed to the utility. The expression must be terminated by a semicolon (";"). If you invoke **find** from a shell you may need to quote the semicolon if the shell would otherwise treat it as a control operator. If the string "{}" appears anywhere in the utility name or the arguments it is replaced by the pathname of the current file. *Utility* will be executed from the directory from which **find** was executed. *Utility* and *arguments* are not subject to the further expansion of shell patterns and constructs.

-exec *utility* [*argument ...*] {} +

Same as **-exec**, except that "{}" is replaced with as many pathnames as possible for each invocation of *utility*. This behaviour is similar to that of xargs(1). The primary always returns true; if at least one invocation of *utility* returns a non-zero exit status, **find** will return a non-zero exit status.

-execdir *utility* [*argument ...*] ;
>   The **-execdir** primary is identical to the **-exec** primary with the exception that *utility* will be executed from the directory that holds the current file. The filename substituted for the string "{}" is not qualified.

-execdir *utility* [*argument ...*] {} +
>   Same as **-execdir**, except that "{}" is replaced with as many pathnames as possible for each invocation of *utility*. This behaviour is similar to that of xargs(1). The primary always returns true; if at least one invocation of *utility* returns a non-zero exit status, **find** will return a non-zero exit status.

-flags [**-**|**+**]*flags*,*notflags*
>   The flags are specified using symbolic names (see chflags(1)). Those with the "no" prefix (except "nodump") are said to be *notflags*. Flags in *flags* are checked to be set, and flags in *notflags* are checked to be not set. Note that this is different from **-perm**, which only allows the user to specify mode bits that are set.
>
>   If flags are preceded by a dash ("-"), this primary evaluates to true if at least all of the bits in *flags* and none of the bits in *notflags* are set in the file's flags bits. If flags are preceded by a plus ("+"), this primary evaluates to true if any of the bits in *flags* is set in the file's flags bits, or any of the bits in *notflags* is not set in the file's flags bits. Otherwise, this primary evaluates to true if the bits in *flags* exactly match the file's flags bits, and none of the *flags* bits match those of *notflags*.

-fstype *type*
>   True if the file is contained in a file system of type *type*. The lsvfs(1) command can be used to find out the types of file systems that are available on the system. In addition, there are two pseudo-types, "local" and "rdonly". The former matches any file system physically mounted on the system where the **find** is being executed and the latter matches any file system which is mounted read-only.

-gid *gname*
>   The same thing as **-group** *gname* for compatibility with GNU find. GNU find imposes a restriction that *gname* is numeric, while **find** does not.

-group *gname*

> True if the file belongs to the group *gname*. If *gname* is numeric and there is no such group name, then *gname* is treated as a group ID.

-ignore_readdir_race

> Ignore errors because a file or a directory is deleted after reading the name from a directory. This option does not affect errors occurring on starting points.

-ilname *pattern*

> Like **-lname**, but the match is case insensitive. This is a GNU find extension.

-iname *pattern*

> Like **-name**, but the match is case insensitive.

-inum *n*

> True if the file has inode number *n*.

-ipath *pattern*

> Like **-path**, but the match is case insensitive.

-iregex *pattern*

> Like **-regex**, but the match is case insensitive.

-iwholename *pattern*

> The same thing as **-ipath**, for GNU find compatibility.

-links *n*

> True if the file has *n* links.

-lname *pattern*

> Like **-name**, but the contents of the symbolic link are matched instead of the file name. Note that this only matches broken symbolic links if symbolic links are being followed. This is a GNU find extension.

-ls   This primary always evaluates to true. The following information for the current file is written to standard output: its inode number, size in 512-byte blocks, file permissions, number of hard links, owner, group, size in bytes, last modification time, and pathname. If the file is a block or character special file, the device number will be displayed instead of the size in bytes. If the file is a symbolic link, the pathname of the linked-to file will be displayed preceded by "->". The format is identical to that produced by "**ls -dgils**".

-maxdepth *n*

Always true; descend at most *n* directory levels below the command line arguments. If any **-maxdepth** primary is specified, it applies to the entire expression even if it would not normally be evaluated. "**-maxdepth** 0" limits the whole search to the command line arguments.

-mindepth *n*

Always true; do not apply any tests or actions at levels less than *n*. If any **-mindepth** primary is specified, it applies to the entire expression even if it would not normally be evaluated. "**-mindepth** 1" processes all but the command line arguments.

-mmin [**-**|**+**]*n*

True if the difference between the file last modification time and the time **find** was started, rounded up to the next full minute, is more than *n* (+*n*), less than *n* (-*n*), or exactly *n* minutes ago.

-mnewer *file*

Same as **-newer**.

-mount

The same thing as **-xdev**, for GNU find compatibility.

-mtime *n*[**smhdw**]

If no units are specified, this primary evaluates to true if the difference between the file last modification time and the time **find** was started, rounded up to the next full 24-hour period, is *n* 24-hour periods.

If units are specified, this primary evaluates to true if the difference between the file last modification time and the time **find** was started is exactly *n* units. Please refer to the **-atime** primary description for information on supported time units.

-name *pattern*

True if the last component of the pathname being examined matches *pattern*. Special shell pattern matching characters ("[", "]", "*", and "?") may be used as part of *pattern*. These characters may be matched explicitly by escaping them with a backslash ("\").

-newer *file*

True if the current file has a more recent last modification time than *file*.

-newer*XY file*

True if the current file has a more recent last access time (*X*=**a**), inode creation time (*X*=**B**), change time (*X*=**c**), or modification time (*X*=**m**) than the last access time (*Y*=**a**), inode creation

time (*Y*=**B**), change time (*Y*=**c**), or modification time (*Y*=**m**) of *file*. In addition, if *Y*=**t**, then *file* is instead interpreted as a direct date specification of the form understood by ISO8601 or RFC822. Note that **-newermm** is equivalent to **-newer**.

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

-noignore_readdir_race
> Turn off the effect of **-ignore_readdir_race**. This is default behaviour.

-noleaf
> This option is for GNU find compatibility. In GNU find it disables an optimization not relevant to **find**, so it is ignored.

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

-ok *utility* [*argument ...*] ;
> The **-ok** primary is identical to the **-exec** primary with the exception that **find** requests user affirmation for the execution of the *utility* by printing a message to the terminal and reading a response. If the response is not affirmative ('y' in the "POSIX" locale), the command is not executed and the value of the **-ok** expression is false.

-okdir *utility* [*argument ...*] ;
> The **-okdir** primary is identical to the **-execdir** primary with the same exception as described for the **-ok** primary.

-path *pattern*
> True if the pathname being examined matches *pattern*. Special shell pattern matching characters ("[", "]", "*", and "?") may be used as part of *pattern*. These characters may be matched explicitly by escaping them with a backslash ("\"). Slashes ("/") are treated as normal characters and do not have to be matched explicitly.

-perm [**-**|+]*mode*
> The *mode* may be either symbolic (see chmod(1)) or an octal number. If the *mode* is symbolic, a starting value of zero is assumed and the *mode* sets or clears permissions without regard to the process' file mode creation mask. If the *mode* is octal, only bits 07777 (S_ISUID | S_ISGID | S_ISTXT | S_IRWXU | S_IRWXG | S_IRWXO) of the file's mode bits participate in the comparison. If the *mode* is preceded by a dash ("-"), this primary evaluates to true if at least all of the bits in the *mode* are set in the file's mode bits. If the *mode* is preceded by a plus ("+"), this

primary evaluates to true if any of the bits in the *mode* are set in the file's mode bits. Otherwise, this primary evaluates to true if the bits in the *mode* exactly match the file's mode bits. Note, the first character of a symbolic mode may not be a dash ("-").

-print   This primary always evaluates to true. It prints the pathname of the current file to standard output. If none of **-exec**, **-ls**, **-print0**, or **-ok** is specified, the given expression shall be effectively replaced by **(** *given expression* **)** **-print**.

-print0

This primary always evaluates to true. It prints the pathname of the current file to standard output, followed by an ASCII NUL character (character code 0).

-prune

This primary always evaluates to true. It causes **find** to not descend into the current file. Note, the **-prune** primary has no effect if the **-d** option was specified.

-quit   Causes **find** to terminate immediately.

-regex *pattern*

True if the whole path of the file matches *pattern* using regular expression. To match a file named "*./foo/xyzzy*", you can use the regular expression ".*/[xyz]*" or ".*/foo/.*", but not "xyzzy" or "/foo/".

-samefile *name*

True if the file is a hard link to *name*. If the command option **-L** is specified, it is also true if the file is a symbolic link and points to *name*.

-size *n*[**ckMGTP**]

True if the file's size, rounded up, in 512-byte blocks is *n*. If *n* is followed by a **c**, then the primary is true if the file's size is *n* bytes (characters). Similarly if *n* is followed by a scale indicator then the file's size is compared to *n* scaled as:

k   kilobytes (1024 bytes)
M   megabytes (1024 kilobytes)
G   gigabytes (1024 megabytes)
T   terabytes (1024 gigabytes)
P   petabytes (1024 terabytes)

-sparse

True if the current file is sparse, i.e. has fewer blocks allocated than expected based on its size in

bytes. This might also match files that have been compressed by the filesystem.

-type *t*
> True if the file is of the specified type. Possible file types are as follows:

> | | |
> |---|---|
> | **b** | block special |
> | **c** | character special |
> | **d** | directory |
> | **f** | regular file |
> | **l** | symbolic link |
> | **p** | FIFO |
> | **s** | socket |

-uid *uname*
> The same thing as *-user uname* for compatibility with GNU find. GNU find imposes a restriction that *uname* is numeric, while **find** does not.

-user *uname*
> True if the file belongs to the user *uname*. If *uname* is numeric and there is no such user name, then *uname* is treated as a user ID.

-wholename *pattern*
> The same thing as **-path**, for GNU find compatibility.

OPERATORS
The primaries may be combined using the following operators. The operators are listed in order of decreasing precedence.

(*expression* **)**
> This evaluates to true if the parenthesized expression evaluates to true.

! *expression*
-not *expression*
> This is the unary NOT operator. It evaluates to true if the expression is false.

-false   Always false.
-true   Always true.

expression **-and** *expression*
expression expression

The **-and** operator is the logical AND operator. As it is implied by the juxtaposition of two expressions it does not have to be specified. The expression evaluates to true if both expressions are true. The second expression is not evaluated if the first expression is false.

expression **-or** *expression*

The **-or** operator is the logical OR operator. The expression evaluates to true if either the first or the second expression is true. The second expression is not evaluated if the first expression is true.

All operands and primaries must be separate arguments to **find**. Primaries which themselves take arguments expect each argument to be a separate argument to **find**.

ENVIRONMENT

The LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES and LC_TIME environment variables affect the execution of the **find** utility as described in environ(7).

EXAMPLES

The following examples are shown as given to the shell:

find / \! -name "*.c" -print
    Print out a list of all the files whose names do not end in *.c*.

find / -newer ttt -user wnj -print
    Print out a list of all the files owned by user "wnj" that are newer than the file *ttt*.

find / \! \(-newer ttt -user wnj \) -print
    Print out a list of all the files which are not both newer than *ttt* and owned by "wnj".

find / \(-newer ttt -or -user wnj \) -print
    Print out a list of all the files that are either owned by "wnj" or that are newer than *ttt*.

find / -newerct '1 minute ago' -print
    Print out a list of all the files whose inode change time is more recent than the current time minus one minute.

find / -type f -exec echo {} \;
    Use the echo(1) command to print out a list of all the files.

find -L /usr/ports/packages -type l -exec rm -- {} +
    Delete all broken symbolic links in */usr/ports/packages*.

find /usr/src -name CVS -prune -o -depth +6 -print
> Find files and directories that are at least seven levels deep in the working directory */usr/src*.

find /usr/src -name CVS -prune -o -mindepth 7 -print
> Is not equivalent to the previous example, since **-prune** is not evaluated below level seven.

COMPATIBILITY

The **-follow** primary is deprecated; the **-L** option should be used instead. See the *STANDARDS* section below for details.

SEE ALSO

chflags(1), chmod(1), locate(1), lsvfs(1), whereis(1), which(1), xargs(1), stat(2), acl(3), fts(3), getgrent(3), getpwent(3), strmode(3), ascii(7), re_format(7), symlink(7)

STANDARDS

The **find** utility syntax is a superset of the syntax specified by the IEEE Std 1003.1-2001 ("POSIX.1") standard.

All the single character options except **-H** and **-L** as well as **-amin**, **-anewer**, **-cmin**, **-cnewer**, **-delete**, **-empty**, **-fstype**, **-iname**, **-inum**, **-iregex**, **-ls**, **-maxdepth**, **-mindepth**, **-mmin**, **-not**, **-path**, **-print0**, **-regex**, **-sparse** and all of the **-B*** birthtime related primaries are extensions to IEEE Std 1003.1-2001 ("POSIX.1").

Historically, the **-d**, **-L** and **-x** options were implemented using the primaries **-depth**, **-follow**, and **-xdev**. These primaries always evaluated to true. As they were really global variables that took effect before the traversal began, some legal expressions could have unexpected results. An example is the expression **-print -o -depth**. As **-print** always evaluates to true, the standard order of evaluation implies that **-depth** would never be evaluated. This is not the case.

The operator **-or** was implemented as **-o**, and the operator **-and** was implemented as **-a**.

Historic implementations of the **-exec** and **-ok** primaries did not replace the string "{}" in the utility name or the utility arguments if it had preceding or following non-whitespace characters. This version replaces it no matter where in the utility name or arguments it appears.

The **-E** option was inspired by the equivalent grep(1) and sed(1) options.

HISTORY

A simple **find** command appeared in Version 1 AT&T UNIX and was removed in Version 3 AT&T UNIX. It was rewritten for Version 5 AT&T UNIX and later be enhanced for the Programmer's

Workbench (PWB). These changes were later incorporated in Version 7 AT&T UNIX.

BUGS

The special characters used by **find** are also special characters to many shell programs. In particular, the characters "*", "[", "]", "?", "(", ")", "!", "\" and ";" may have to be escaped from the shell.

As there is no delimiter separating options and file names or file names and the *expression*, it is difficult to specify files named *-xdev* or *!*. These problems are handled by the **-f** option and the getopt(3) "**--**" construct.

The **-delete** primary does not interact well with other options that cause the file system tree traversal options to be changed.

The **-mindepth** and **-maxdepth** primaries are actually global options (as documented above). They should probably be replaced by options which look like options.