

**File criteria**

```

find -name <file>
find -path <file>
find -iname <file>
find -lname <file>
find -ilname <file>
find -ipath <file>
  o "i" indicates case insensitive
  o "l" matches with symlink's target's name
  o "wholename" matches full path. wildcards
    treat all chars (i.e. "/", ".", etc.) equally.
find -regex <regex>
find -iregex <regex>
  o Defaults to emacs regex syntax.
  o To select syntax type, precede
    with -regextype <emacs | posix-awk |
    posix-basic | posix-egrep | posix-extended>
find -type <d|f|l|...>
  o d → directory
  o f → file
  o l → symbolic link
  o See man file for a few less common options.
find -samefile <file>
find -links <num_links>
  o Can use "+" (greater than) or "-"(less than)
find -mtime +1
find -ctime +1
find -atime +1
find -used +1
  o Modified/created/status_changed/used at least
    2 days ago. Note the weird rounding issue.
  o Precede with -daystart to ignore times.
  o Use 0 (i.e. "-mtime 0") for last 24 hours.
find -newer <file>
find -cnewer <file>
find -anewer <file>
  o Modified/created/status_changed more recently
    than <file>.
find -newerXY <file>
  o X and Y are in a=access time, b=birth time,
    c=status change time, m=modification time
  o Use t to pass in a specific time (formatted)
find -size +35M
  o Files more than 35 megabytes.
  o Use "-" instead of "+" for less than.
  o Other suffixes: c=bytes, k=kilobytes, G=gigs
find -empty
find -executable
find -readable
find -writable
find -perm ###
  o Permissions exactly match spec in octal.
  o Use -### for at least the specified bits (⊆).
  o Use /### for any of the specified bits (U).
  o Use ### for exact permissions (≡).
  o Can also use expressions like a=rw or u+w,g+w.
find -user <user_name>
find -uid <user_number>
find -group <group_name>
find -gid <group_num>

```

**Boolean Logic**

```

find ... expr1 -and expr2
find ... expr1 -or expr2
find ... -not expr2
find ... -true
find ... -false
find ... ( expr1 -or expr2 ) -and expr3
find ... expr1 , expr2
  o Consecutive criteria are AND'ed by default.
  o -o, -a, and ! are non-POSIX abbreviations
  o ! does not need to be quoted when using bash.
  o Actions are part of the expression, too!

```

**Tree Depth Criteria**

```

find -maxdepth 3
find -mindepth 3
find -depth
  o Process subdirs' files before dir's own files.
  o Use -d (non-POSIX) on BSD-based systems

```

**Actions**

```

find ... -delete
  o Implies -depth so cannot be used with -prune.
find ... -prune
find ... -quit
find ... -exec <command> \;
find ... -exec <command> \{\} \; +
find ... -ok <command> \;
find ... -execdir <command> \;
find ... -execdir <command> \{\} \; +
find ... -okdir <command> \;
  o Use \{\} to refer to the filename.
  o "+" versions append multiple files to one cmd.
  o -ok asks for confirmation on each file.
  o "dir" versions execute in containing dir.
  o Return code determines if this is treated as
    true or false in boolean expressions.

```

**Printing File Information**

```

find ... -ls
find ... -fls <file>
  o List in ls -dils format, including inode#,
    #/hardlinks, owner, group, size, mtime, name
find ... -print
find ... -fprint <file>
find ... -print0
find ... -fprint0 <file>
  o Print filename.
  o ~0 versions terminate lines with '\0' instead
    of '\n', for use with xargs -0 or other cmds.
find ... -printf <format>
find ... -fprintf <file> <format>
  o Print filename as specified by format string.
  o Special characters: \n, \t, \123, \b, \, %%
  o Don't forget the newline ("\n")!
  o Timestamps in ctime format:
    o creation time: %c
    o modification time: %t
    o status change time: %a
    o ("Tue Sep 24 20:05:59.0513625000 2012")
  o Timestamps in custom format: %C_, %T_, %A_
    with format codes as follows: (for creation)
    o 23:59:59 → %CT or %CX or %CH:%CM:%CS
    o 01/01/2000 → %CD or %CX or %Cm/%Cd/%CY
    o 01-Jan-00 → %Cd-%CB-%Cy
    o 11:59:59 PM → %CI:%CM:%CS %Cp or %r
    o Sat Jan 01 11:59:59 EST 2000 → %Cc
    or %Ca %Cb %Cm %CH:%CM:%CS %CZ %CY
    o 2000-Jan-01+11:59:59.0 → %C+
    o Saturday, January 1 → %CA, %CB %CD
    o day# of week → %Cw (0..6 with 0=Sunday)
    o day# of year → %Cj (001..366)
    o week# of year → %CW (00..53)
    o seconds since epoch → %C@
  o File info:
    o filename, base: %f ("xferlog")
    o relative path: %p ("./log/xferlog")
    o relative directory: %h ("./log")
    o permission: %m ("644"), %M ("-rwxr-xr-x")
    o group: %g ("hcl"), %G ("25218")
    o user: %u ("aq"), %U ("12460")
    o depth: %d
    o inode#: %i
    o size in bytes: %s
    o type: %y (d=directory, f=file, l=link)
    o type, follow links: %Y (d, f, or L=loop)

```

## Examples

```

find . -exec chmod 644 '{}' \;
  o Set all directories to 755 permissions.

find . -regex '.*/[a-z]+\.\.(py|pl|p1\)'
  o Find all Python and Perl scripts that start with a lowercase letter.

find /tmp -name core -type f -print | xargs /bin/rm -f
  o Delete files named "core".
  o Will fail if any directories contain spaces, quotes, newlines, etc.

find /tmp -name core -type f -print0 | xargs -0 /bin/rm -f
  o Delete files named "core".
  o Properly functions even in the presence of directories containing spaces, quotes, newlines, etc.

find . \( -type d ! -exec test -x {} -a -r {} \; -prune \) -o -name '*.jpg'
  o Avoid "Permission Denied" errors while searching for all JPG image files.
  o The test command means "if file is executable (and exists) and file is readable (and exists)"

find . \( -perm -4000 -fprintf /root/suid.txt %#m %u %p\n \) , \( -size +100M -fprintf /root/big.txt %-10s %p\n \)
  o Do two sets of actions in one pass, thanks to the comma and parentheses.
  o Puts names of setuid files/directories in one file and names of big files in another file.

find . -type f -newermt 2007-06-07 ! -newermt 2007-06-08
find . -type f -mtime $(( ( $(date +%s) - $(date -d '2012-09-24' +%s) ) / 60 / 60 / 24 - 1 ))
find -ls | egrep '((Feb|May) .. 2009|oct .. 2008)'
  o Files modified on September 24, 2012 (the specific date).

find . -perm /220
find . -perm /u+w,g+w
find . -perm /u=w,g=w
  o Writable by owner or group or both.
  o Other permissions are okay, too.

find . -perm -220
find . -perm -g+w,u+w
  o Writable by both owner and group, at the least.
  o Other permissions are okay, too.

find . -perm 220
find . -perm u=r
  o Readable by owner.
  o Exact match required. No other permissions allowed.

find . -mtime 0
  o Modified in the last 24 hours.
  o This works this way because the time since each file was last modified is divided by 24 hours and any remainder is discarded. That means that to match -mtime 0, a file will have to have a modification in the past which is less than 24 hours ago.

find . -exec grep "cashews" '{}' \; -print
  o Grep all files for "cashews" and and print the name of each file just after the results.

find . -exec grep -q "cashews" '{}' \; -print
  o Print the names of all files containing "cashews" but don't print the output of the grep command.
  o This works because the -exec action returns false (non-zero) and thus evaluates to false in the boolean expression. -exec and -print are essentially AND'ed together here.

find -ls | egrep '((Feb|May) .. 2009|oct .. 2008)'
  o Search for files modified within several date ranges.
  o Takes advantage of the specific syntax used by the -ls action. Example:
  o 101049516639138444 26 -rw----- 1 Alex None 53248 Mar 25 2008 ./swo
    ( inode# ??? perms lnks ownr group size(b) mdate relpath )

find . -name CVS -prune -or \( -name '*.py' -or -type d \) -exec chmod 755 '{}' \; -or -exec chmod 644
  o Change permissions to 755 on *.py files and directories and 644 on other files
  o Don't touch or go into CVS directories

```