find command

Updated 9/25/2009. Based on GNU find v4.4.

File criteria

find **-name** *<file>*
find **-path** *<file>*
find **-iname** *<file>*
find **-lname** *<file>*
find **-ilname** *<file>*
find **-ipath** *<file>*
- "i" indicates case insensitive
- "l" matches with symlink's target's name
- "wholename" matches full path. Wildcards treat all chars (i.e. "/", ".", etc.) equally.

find **-regex** *<regex>*
find **-iregex** *<regex>*
- Defaults to emacs regex syntax.
- To select syntax type, precede with **-regextype** <emacs | posix-awk | posix-basic | posix-egrep | posix-extended>

find **-type** <d|f|l|...>
- d → directory
- f → file
- l → symbolic link
- See man file for a few less common options.

find **-samefile** *<file>*

find **-links** *<num_links>*
- Can use "**+**" (greater than) or "**-**"(less than)

find **-mtime** +1
find **-ctime** +1
find **-atime** +1
find **-used** +1
- Modified/created/status_changed/used at least 2 days ago. Note the weird rounding issue.
- Precede with **-daystart** to ignore times.
- Use 0 (i.e. "-mtime 0") for last 24 hours.

find **-newer** *<file>*
find **-cnewer** *<file>*
find **-anewer** *<file>*
- Modified/created/status_changed more recently than *<file>*.

find **-newer*XY*** *<file>*
- X and Y are in a=access time, b=birth time, c=status change time, m=modification time
- Use t to pass in a specific time (formatted

find **-size** +35M
- Files more than 35 megabytes.
- Use "**-**" instead of "**+**" for less than.
- Other suffixes: c=bytes, k=kilobytes, G=gigs

find **-empty**

find **-executable**
find **-readable**
find **-writeable**

find **-perm** ###
- Permissions exactly match spec in octal.
- Use **-###** for at least the specified bits (⊆).
- Use **/###** for any of the specified bits (∪).
- Use **###** for exact permissions (≡).
- 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
- Consecutive criteria are AND'ed by default.
- **-o**, **-a**, and **!** are non-POSIX abbreviations
- ! does not need to be quoted when using bash.
- Actions are part of the expression, too!

Tree Depth Criteria

find **-maxdepth** 3
find **-mindepth** 3

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

Actions

find ... **-delete**
- 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>* **\;**
- Use \{\} to refer to the filename.
- "+" versions append multiple files to one cmd.
- -ok asks for confirmation on each file.
- "dir" versions execute in containing dir.
- Return code determines if this is treated as true or false in boolean expressions.

Printing File Information

find ... **-ls**
find ... **-fls** *<file>*
- List in ls -dils format, including inode#, #/hardlinks, owner, group, size, mtime, name

find ... **-print**
find ... **-fprint** *<file>*
find ... **-print0**
find ... **-fprint0** *<file>*
- Print filename.
- ~0 versions terminate lines with '\0' instead of '\n', for use with xargs -0 or other cmds.

find ... **-printf** *<format>*
find ... **-fprintf** *<file> <format>*
- Print filename as specified by format string.
- Special characters: \n, \t, \123, \b, \\, %%
- Don't forget the newline ("\n")!
- Timestamps in ctime format:
  - creation time: **%c**
  - modification time: **%t**
  - status change time: **%a**
  - ("Tue Sep 24 20:05:59.0513625000 2012")
- Timestamps in custom format: **%C_**, **%T_**, **%A_** with format codes as follows: (for creation)
  - 23:59:59 → %C**T** *or* %C**X** *or* %C**H:**%C**M:**%C**S**
  - 01/01/2000 → %C**D** *or* %C**X** *or* %C**m/**%C**d/**%C**Y**
  - 01-Jan-00 → %C**d-**%C**B-**%C**y**
  - 11:59:59 PM → %C**I:**%C**M:**%C**S** %C**p** *or* **%r**
  - Sat Jan 01 11:59:59 EST 2000 → %C**c** *or* %C**a** %C**b** %C**m** %C**H:**%C**M:**%C**S** %C**Z** %C**Y**
  - 2000-Jan-01+11:59:59.0 → %C**+**
  - Saturday, January 1 → %C**A,** %C**B** %C**D**
  - day# of week → %C**w** (0..6 with 0=Sunday)
  - day# of year → %C**j** (001..366)
  - week# of year → %C**W** (00..53)
  - seconds since epoch → %C**@**
- File info:
  - filaname, base: **%f** ("xferlog")
  - relative path: **%p** ("./log/xferlog")
  - relative directory: **%h** ("./log")
  - permission: **%m** ("644"), **%M** ("-rwxr-xr-x")
  - group: **%g** ("hcil'), **%G** ("25218")
  - user: **%u** ("aq"), **%U** ("12460")
  - depth: **%d**
  - inode#: **%i**
  - size in bytes: **%s**
  - type: **%y** (d=directory, f=file, l=link)
  - type, follow links: **%Y** (d, f, or L=loop)

find command

Updated 9/25/2009. Based on GNU find v4.4.

Examples

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

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

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

```
find /tmp -name core -type f -print0 | xargs -0 /bin/rm -f
```
- Delete files named "core".
- 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'
```
- Avoid "Permission Denied" errors while searching for all JPG image files.
- The test command means "if file is executable (and exists) and file is readable (and exists)"

```
find . \( -perm -4000 -fprintf /root/suid.txt %#m %u %p\n \) , \( -size +100M -fprintf /root/big.txt %-10s %p\n \)
```
- Do two sets of actions in one pass, thanks to the comma and parentheses.
- 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)'
```
- Files modified on September 24, 2012 (the specific date).

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

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

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

```
find . -mtime 0
```
- Modified in the last 24 hours.
- 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
```
- Grep all files for "cashews" and and print the name of each file just after the results.

```
find . -exec grep -q "cashews" '{}' \; -print
```
- Print the names of all files containing "cashews" but don't print the output of the grep command.
- 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)'
```
- Search for files modified within several date ranges.
- Takes advantage of the specific syntax used by the -ls action. Example:
  - `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 '{}' \;
```
- Change permissions to 755 on *.py files and directories and 644 on other files
- Don't touch or go into CVS directories