Linux tree command

Updated: 12/29/2017 by Computer Hope

About tree

tree lists the contents of directories in a tree-like format. It's a really neat and useful program you can use at the command line to view the structure of your file system.

Description

tree is a recursive directory listing program that produces a depth indented listing of files (which is colorized if the **LS_COLORS** environment variable is set) and output is to tty. With no arguments, **tree** lists the files in the current directory. When directory arguments are given, **tree** lists all the files and/or directories found in the given directories each in turn. **tree** then returns the total number of files and/or directories listed.

By default, when a symbolic link is encountered, the path that the symbolic link refers to is printed after the name of the link in the format:

```
name -> real-path
```

If the `**-l**' option is given and the symbolic link refers to an actual directory, then tree will follow the path of the symbolic link as if it were a real directory.

tree syntax

```
tree [-adfgilnopqrstuxACDFNS] [-L level [-R]] [-H
baseHREF] [-T title]
     [-o file name] [--nolinks] [-P pattern] [-I pattern]
[--inodes]
     [--device] [--noreport] [--dirsfirst] [--version] [--
```

```
help]
     [--filelimit #] [--si] [--prune] [--du] [--timefmt
format]
     [directory ...]
```

Options

--help	Outputs a verbose usage listing.
--version	Outputs the version of tree.
-a	All files are printed. By default, tree does not print hidden files (those beginning with a dot `**.**'). In no event does tree print the file system constructs `**.**' (current directory) and `**..**' (previous directory).
-d	List directories only.
-f	Prints the full path prefix for each file.
-i	Tree will not print the indentation lines. Useful when used in conjunction with the **-f** option.
-l	Follows symbolic links to directories as if they were directories. Links that would result in a recursive loop are avoided.
-x	Stay on the current file system only, as with **find -xdev**.
-P *pattern*	List only those files that match the wild-card *pattern*. Note: you must use the **-a** option to also consider those files beginning with a dot `**.**' for matching. Valid wildcard operators are `*****' (any zero or more characters), `**?**' (any single character), `**[...]**' (any single character listed between brackets (optional **-** (dash) for character range may be used: ex: **[A-Z]**), and `**[^...]**'

	(any single character not listed in brackets) and `|' separates alternate patterns.
-I *pattern*	Do not list those files that match the wild-card *pattern*.
--prune	Makes tree prune empty directories from the output, useful when used in conjunction with **-P** or **-I**.
--filelimit #	Do not descend directories that contain more than # entries.
--timefmt *format*	Prints (implies **-D**) and formats the date according to the format string which uses the strftime syntax.
--noreport	Omits printing of the file and directory report at the end of the tree listing.
-p	Print the protections for each file (as per ls **-l**).
-s	Print the size of each file along with the name.
-u	Print the username, or UID # if no username is available, of the file.
-g	Print the group name, or GID # if no group name is available, of the file.
-D	Print the date of the last modification time for the file listed.
--inodes	Prints the inode number of the file or directory
--device	Prints the device number to which the file or directory belongs
-F	Append a `/' for directories, a `=' for socket files, a `**\***' for executable files and a `|' for FIFO's, as per ls **-F**
-q	Print non-printable characters in file names as question marks instead of the default carrot notation.

-N	Print non-printable characters as is instead of the default carrot notation.
-r	Sort the output in reverse alphabetic order.
-t	Sort the output by last modification time instead of alphabetically.
--dirsfirst	List directories before files.
-n	Turn colorization off always, over-ridden by the **-C** option.
-C	Turn colorization on always, using built-in color defaults if the **LS_COLORS** environment variable is not set. Useful to colorize output to a pipe.
-A	Turn on ANSI line graphics hack when printing the indentation lines.
-S	Turn on ASCII line graphics (useful when using linux console mode fonts). This option is now equivalent to `**--charset=IBM437**' and will eventually be depreciated.
-L *level*	Max display depth of the directory tree.
-R	Recursively cross down the tree each level directories (see **-L** option), and at each of them execute tree again adding `**-o 00Tree.html**' as a new option.
-H *baseHREF*	Turn on HTML output, including HTTP references. Useful for ftp sites. baseHREF gives the base ftp location when using HTML output. That is, the local directory may be `**/local/ftp/pub'**, but it must be referenced as `**ftp://host-name.organization.domain/pub'** (baseHREF should be `**ftp://hostname.organization.domain'**). Hint: don't use ANSI lines with this option, and don't give more than one directory in

	the directory list. If you want to use colors via CSS stylesheet, use the **-C** option in addition to this option to force color output.
-T *title*	Sets the title and H1 header string in HTML output mode.
--charset *charset*	Set the character set to use when outputting HTML and for line drawing.
--nolinks	Turns off hyperlinks in HTML output.
-o *file name*	Send output to file name.

tree examples

```
tree
```

Displays the contents of the current directory and subdirectories in a tree. The output takes a graphical form which will resemble the following example:

```
.
├── config.dat
├── data
│   ├── data1.bin
│   ├── data2.sql
│   └── data3.inf
├── images
│   ├── background.jpg
│   ├── icon.gif
│   └── logo.jpg
├── program.exe
└── readme.txt

2 directories, 9 files
```

```
tree -I 'example*|bin|lib'
```

Displays a tree without anything beginning with example or containing 'bin' or 'lib' as specified in the pattern.

```
tree -P 't*'
```

With a uppercase "P" you can list files with a pattern. This pattern example displays a tree only containing directories beginning with t or directories containing files that begin with t.

```
tree -p
```

With a lowercase "p" you can list a tree that also shows the file permissions.

Related commands

ls — List the contents of a directory or directories.