Listing Files and Understanding `ls` Output

Here is the output from an `ls` command using the `-l`option. The `-l`flag tells `ls` to display output in a long format. If you need to see what files or directories exist, use `ls`. However, if you need detailed information use `ls -l`.

```
$ ls -l
-rw-rw-r-- 1 bob users 10400 Sep 27 08:52 sales.data
```

On the far left of the `ls` output is a series of characters that represent the file permissions. The number that follows the permissions represents the number of links to the file. The next bit of information is the owner of the file followed by the group name. Next the file size is displayed followed by the date and time when the file was last modified. Finally, the name of the file or directory is displayed. Here is the information displayed by the `ls -l`command in table form.

Item	Value
Permissions	-rw-rw-r--
Number of links	1
Owner name	bob
Group name	users
Number of bytes in the file	10400
Last modification time	Sep 27 08:52
File name	sales.data

The meaning of `-rw-rw-r--`will be covered in detailed in the "File and Directory Permissions" lesson.

Listing All Files, Including Hidden Files

Files or directories that begin with a period (`.`) are considered hidden and are not displayed by default. To show these hidden files and directories, use the `-a`option.

```
$ ls -a
.
..
.profile
.bash_history
PerformanceReviews
sales-lecture.mp3
sales.data
tpsreports
```

Up until this point when you have used options, you have preceded each option with a hyphen (`-`). Examples are `-l`and `-a`. Options that do not take arguments can be combined. Only one hyphen is required followed by the options. If you want to show a long `ls` listing with hidden files you could run `ls -l -a`or `ls -la`. You can even change the order of the flags, so `ls -al`works too. They are all equivalent.

```
$ ls -l
total 2525
drwxr-xr-x  3  bob  users  512     Sep 28 09:20 PerformanceReviews
-rw-r--r--  1  bob  sales  2562856 Sep 27 08:54 sales-lecture.mp3
-rw-r--r--  1  bob  users  10400   Sep 27 08:52 sales.data
```

```
drwxr-xr-x  2  bob  users  512     Sep 28 14:49 tpsreports
$ ls -l -a
total 2532
drwxr-xr-x  4  bob  sales  512     Sep 28 14:56 .
drwxr-xr-x  14 root root   512     Sep 27 08:43 ..
-rw-r--r--  1  bob  users  28      Sep 28 14:22 .profile
-rw-------  1  bob  users  3314    Sep 28 14:56 .bash_history
drwxr-xr-x  3  bob  users  512     Sep 28 09:20 PerformanceReviews
-rw-r--r--  1  bob  sales  2562856 Sep 27 08:54 sales-lecture.mp3
-rw-r--r--  1  bob  users  10400   Sep 27 08:52 sales.data
drwxr-xr-x  2  bob  users  512     Sep 28 14:49 tpsreports
$ ls -la
total 2532
drwxr-xr-x  4  bob  sales  512     Sep 28 14:56 .
drwxr-xr-x  14 root root   512     Sep 27 08:43 ..
-rw-r--r--  1  bob  users  28      Sep 28 14:22 .profile
-rw-------  1  bob  users  3314    Sep 28 14:56 .bash_history
drwxr-xr-x  3  bob  users  512     Sep 28 09:20 PerformanceReviews
-rw-r--r--  1  bob  sales  2562856 Sep 27 08:54 sales-lecture.mp3
-rw-r--r--  1  bob  users  10400   Sep 27 08:52 sales.data
drwxr-xr-x  2  bob  users  512     Sep 28 14:49 tpsreports
$ ls -al
total 2532
drwxr-xr-x  4  bob  sales  512     Sep 28 14:56 .
drwxr-xr-x  14 root root   512     Sep 27 08:43 ..
-rw-r--r--  1  bob  users  28      Sep 28 14:22 .profile
-rw-------  1  bob  users  3314    Sep 28 14:56 .bash_history
drwxr-xr-x  3  bob  users  512     Sep 28 09:20 PerformanceReviews
-rw-r--r--  1  bob  sales  2562856 Sep 27 08:54 sales-lecture.mp3
-rw-r--r--  1  bob  users  10400   Sep 27 08:52 sales.data
drwxr-xr-x  2  bob  users  512     Sep 28 14:49 tpsreports
$
```

Listing Files by Type

When you use the `-F`option for `ls` a character is appended to the file name that reveals what type it is.

```
$ ls
dir1 link program regFile
$ ls -F
dir1/ link@ program* regFile
$ ls -lF
total 8
drwxr-xr-x 2 bob users 117 Sep 28 15:31 dir1/
lrwxrwxrwx 1 bob users 7   Sep 28 15:32 link@ -> regFile
-rwxr-xr-x 1 bob users 10  Sep 28 15:31 program*
-rw-r--r-- 1 bob users 750 Sep 28 15:32 regFile
```

Symbol	Meaning
`/`	Directory.
`@`	Link. The file that follows the `->`symbol is the target of the link.
`*`	Executable program.

A link is sometimes called a symlink, short for symbolic link. A link points to the location of the actual file or directory. You can operate on the link as if it were the actual file or directory. Symbolic links can be used to create shortcuts to long directory names. Another common use is to have a symlink point to the latest version of installed software as in this example.

```
bob@linuxsvr:~$ cd /opt/apache
bob@linuxsvr:/opt/apache ~$ ls -F
```

```
2.4.5/ 2.4.7/ current@ README
bob@linuxsvr:/opt/apache$ ls -l
drwxr-xr-x 2 root root 4096 Sep 14 12:21 2.4.5
drwxr-xr-x 2 root root 4096 Nov 27 15:43 2.4.7
lrwxrwxrwx 1 root root    5 Nov 27 15:43 current -> 2.4.7
-rw-r--r-- 1 root root 1048 Sep 14 12:58 README
```

Listing Files by Time and in Reverse Order

If you would like to sort the `ls` listing by time, use the `-t`option.

```
$ ls -t
tpsreports
PerformanceReviews
sales-lecture.mp3
sales.data
$ ls -lt
total 2532
drwxr-xr-x 2 bob users 512 Sep 28 14:49 tpsreports
drwxr-xr-x 3 bob users 512 Sep 28 09:20 PerformanceReviews
-rw-r--r-- 1 bob sales 2562856 Sep 27 08:54 sales-lecture.mp3
-rw-r--r-- 1 bob users 10400 Sep 27 08:52 sales.data
```

When you have a directory that contains many files it can be convenient to sort them by time, but in reverse order. This will put the latest modified files at the end of the `ls` output. The old files will scroll off the top of your display, but the most recent files will be right above your prompt.

```
$ ls -latr
total 2532
drwxr-xr-x 14 root root 512 Sep 27 08:43 ..
-rw-r--r-- 1 bob users 10400 Sep 27 08:52 sales.data
-rw-r--r-- 1 bob sales 2562856 Sep 27 08:54 sales-lecture.mp3
drwxr-xr-x 3 bob users 512 Sep 28 09:20 PerformanceReviews
-rw-r--r-- 1 bob users 28 Sep 28 14:22 .profile
drwxr-xr-x 2 bob users 512 Sep 28 14:49 tpsreports
drwxr-xr-x 4 bob sales 512 Sep 28 14:56 .
-rw------- 1 bob users 3340 Sep 28 15:04 .bash_history
```

Listing Files Recursively

Using the `-R`option with `ls` causes files and directories to be displayed recursively.

```
$ ls -R
.:
PerformanceReviews sales-lecture.mp3 sales.data tpsreports
./PerformanceReviews:

Fred John old

./PerformanceReviews/old:
Jane.doc
$
```

You can also use the `tree` command for more visually appealing output. If you only want to see the directory structure, use `tree -d`.

`tree` - List contents of directories in a tree-like format.

`tree -d` - List directories only.

`tree -C` - Colorize output.

```
$ tree
.
├── PerformanceReviews
│   ├── Fred
│   ├── John
│   └── old
│       └── Jane.doc
├── sales.data
├── sales-lecture.mp3
└── tpsreports

2 directories, 6 files
$ tree -d
.
└── PerformanceReviews
    └── old

2 directories
$
```

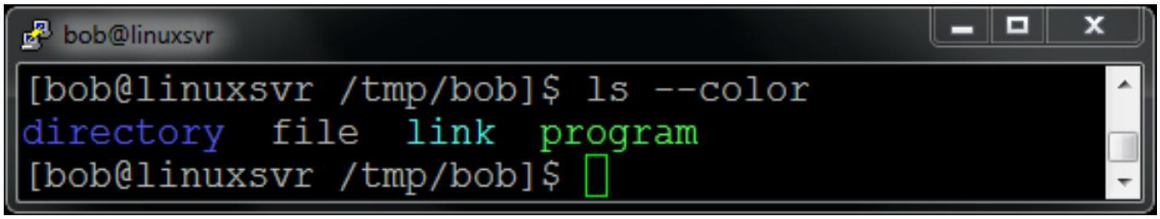
List Directories, Not Contents

Normally when you run `ls` against a directory the contents of that directory are displayed. If you want to ensure you only get the directory name, use the `-d`option.

```
$ ls -l PerformanceReviews
total 3
-rw-r--r-- 1 bob users  36 Sep 27 08:49 Fred
-rw-r--r-- 1 bob users  36 Sep 28 09:21 John
drwxr-xr-x 2 bob users 512 Sep 27 12:40 old
$ ls -ld PerformanceReviews
drwxr-xr-x 3 bob users 512 Sep 28 09:20 PerformanceReviews
$ ls -d PerformanceReviews
PerformanceReviews
```

Listing Files with Color

Earlier you used `ls -F`to help differentiate file types by adding a character to the end of their names in the `ls` output. You can also use color to distinguish file types by using `ls --color`.



Commonly Used `ls` Options

Here is a recap of the `ls` options you have learned.

Option	Description
`-a`	All files, included hidden files
`--color`	List files with colorized output
`-d`	List directory names and not their contents
`-l`	Long format

`-r`	Reverse order
`-R`	List files recursively
`-t`	Sort by time, most recently modified first

Working with Spaces in Names

If you want to make your life easier when working from the command line, do not use spaces in file and directory names. Hyphens (`-`) or underscores (`_`) can be good substitutes for spaces. CamelCase, the practice of capitalizing each word, is another good option. For example, instead of naming your latest literary attempt "the next great american novel.txt" you could use "the-next-great-american-novel.txt", "the_next_great_american_novel.txt" or even "TheNextGreatAmericanNovel.txt."

Sooner or later you will encounter a file or directory that contains a space in the name. There are two ways to deal with this. The first is to use quotation marks. Even though the file name is `a file`, operate on it using `"a file."`The second option is to escape the space. Escaping is like using quotes, but for single characters. The escape symbol is `\`, also known as a backslash. To escape a space, precede the space with the backslash (`\`) character.

```
$ ls -l
-rw-r--r-- 1 bob users 18 Oct 2 05:03 a file
$ ls -l a file
ls: a: No such file or directory
ls: file: No such file or directory
$ ls -l "a file"
-rw-r--r-- 1 bob users 18 Oct 2 05:03 a file
$ ls -l a\ file
-rw-r--r-- 1 bob users 18 Oct 2 05:03 a file
$ ls -lb a*
-rw-r--r-- 1 bob users 18 Oct 2 05:03 a\ file
$
```

The `-b`option to `ls` causes it to print escape codes. Note that quoting and escaping not only applies to spaces, but with other special characters as well including `| & ' ; ( ) < > space tab`.

.