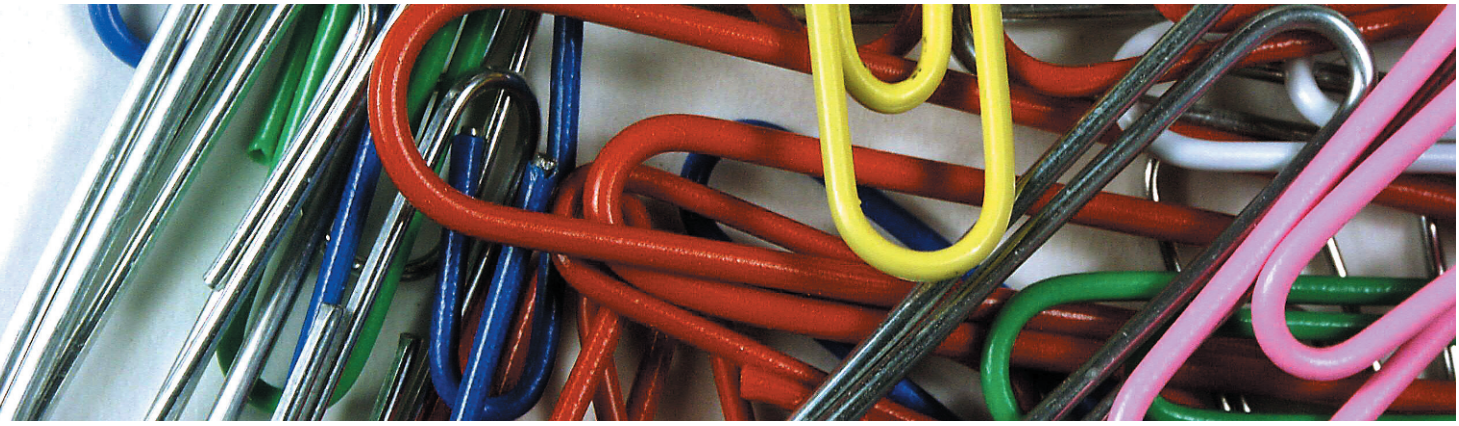
Coloring ls output with dircolors

Bright Folder Colors

The ls command outputs an overview of directory content. Colored output has become a matter of course for many users. This article describes how you can control the color palette that ls uses. BY HAGEN HÖPFNER



Any Linux user who has opened a terminal window is bound to have used the *ls* command at some time to list directory content. The major distributions are set up to color ls output, making it easier for users to read the screen at a glance (see Figure 1).

Of course, the default setup may not be perfect. The readability of the output depends on your choice of background color for the terminal; dark blue directory names on a black background are very difficult to read, as Figure 1 shows. Also, distributors have no way of knowing whether users with impaired color vision will be capable of distinguishing the colors they choose.

Opening the Paintbox

The colors that ls uses to output various file types and directories are defined in an environment variable called *LS_COLORS*. In the *Bash*-**Shell**, which most distributors use as by default, you can type *echo $LS_COLORS* to output the values of the variable. On Suse, the results can be seen in Figure 2.

dircolors helps take the pain out of handling the numerous individual values in the variable. Running the command without any options tells dircolors to output two lines. The first line shows the current content of *LS_COLORS*, and the second line has the *export LS_COLORS* command that applies the variable to any programs launched in the shell from that point onward. *dircolors -p* outputs a description of the individual values; Listing 1, next page, shows an excerpt from this list.

The supported terminal types are described right at the top of the list, followed by the colors for various file types starting with general cases: *NORMAL* applies to any file types that do not fall into another category. This is then followed by an entry for *FILE*, which specifies the display color for normal files.

DIR and *LINK* are for directories and symbolic links to other files or directories. Entries for special file types, such as the device files in */dev* then follow. The *ORPHAN* entry describes links that point to non-existent files. This is followed

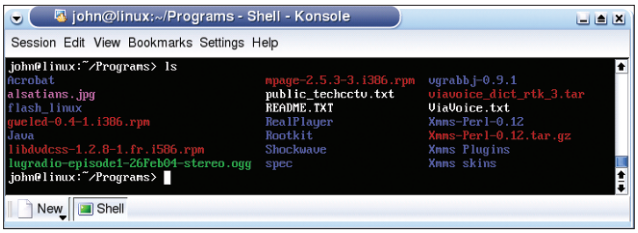


Figure 1: Most distributors configured colored ls output by default.

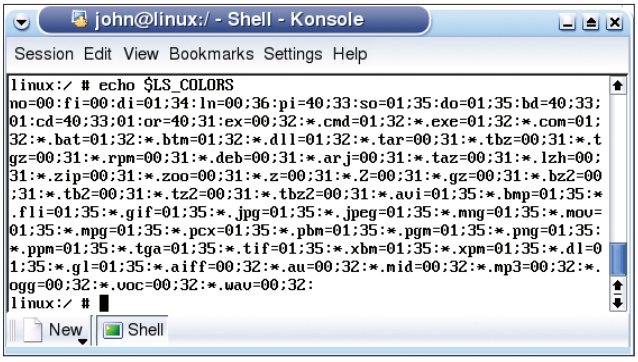


Figure 2: LS_COLORS default values.

Box 1: *ls* versus *dir*

Some distributors support well-known DOS commands on Linux, with an eye to making life easier for users migrating to Linux from Windows. The *dir* command is one example. DOS and the Windows command line boxes use *dir* to output directory content. This type of command is typically supported by a Linux alias. For example, if you type *dir* on the command line of a Suse 9 box, you are in fact running *ls -l*.

Table 1: Meaning of values for dircolors

Font	Font color/Background color
00: No change	30/40: black
01: Bold	31/41: red
04: Underline	32/42: green
05: Blinking	33/43: yellow
07: Switch foreground and background colors	34/44: blue
08: Hidden (not supported by some terminals)	35/45: magenta 36/46: cyan 37/47: white

by a list of normal files with specific suffixes; a color can be assigned to each suffix.

Painting by Numbers

Each entry is followed by a minimum of one, and a maximum of three numbers which specify the font color and type – e.g. bold – and the background color. If we look at an entry such as *DIR 01;34*, for example, the first value, *01*, enables bold type, and the value *34* sets the font color to blue.

The *ORPHAN 40;31;01* entry assigns three values: besides bold type *01*, and the color red (*31*), the *40* entry specifies a black background (see Figure 3). The order in which you type the values is not significant. Values between 00 and 08 always specify the font type, 30 through 37 define the foreground color, and 40 through 47 the background color. If a value is missing, ls will use the default. Table 1 shows the individual keys.

A special option is available for *LINK*. If you want to display a link in the same way as the target file, stipulate the *target* option for the entry. Instead of a numeric value, you would type:

```
LINK target
```

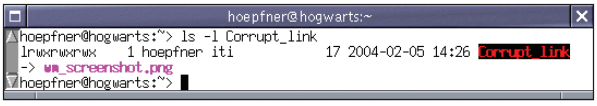


Figure 3: Colored ls output with a background color definition.

This tells ls not to distinguish between a directory and a link to a directory.

If you want to use dircolors to modify the list, you can export the set values to a file:

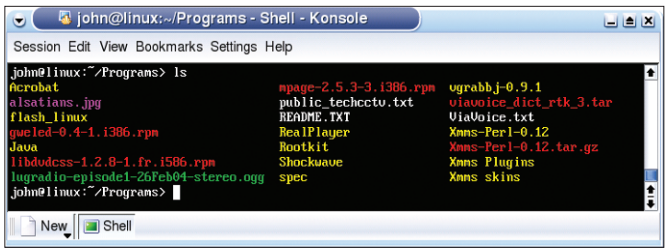


Figure 4: Directory names in yellow.

```
dircolors -p > my_colors.txt
```

You can now use your favorite text editor to modify the values in *my_colors.txt* to reflect your needs. The *dircolors my_colors.txt* command reads the values from this file and outputs the lines that have been changed by modifying the *LS_COLORS* variable. To apply the changes, use the *eval* command to execute the command output directly:

```
eval `dircolors my_colors.txt`
```

Permanent Color Schemes

After much trial and error, you are likely to hit on a color combination you like so much that you want to keep it. To tell the Bash shell to use your color scheme whenever you boot your machine, add the following lines to *~/.bashrc*. Don't forget to specify the path, to ensure that dircolors will use the right file:

```
eval `dircolors ➥
/path/to/my_colors.txt`
```

If you do not want to have to keep a *my_colors.txt* file on your hard disk, you can assign the values in the file directly by typing:

```
dircolors my_colors.txt >> ➥
~/.bashrc
```

You should remove or comment out any existing entries for *dircolors* from *~/.bashrc* to avoid any confusion in the future. But don't worry; even if you don't do this, dircolors will use the last entry that it finds in the file, so it makes sense to look at the end of the file first. ■

Listing 1: *dircolors -p* output

```
01 TERM linux
02 #...
03 TERM Eterm
04
05 NORMAL 00        # default
   setting
06 FILE 00          # default
   setting for files
07 DIR 01;34        # default
   setting for directories
08 LINK 01;36       # symbolic
   links
09 #...
10 ORPHAN 40;31;01 # orphaned
   symbolic links
11 #...
12 EXEC 01;32       # executables
13
14 .tar 01;31       # files with
   the .tar extension
15
16 .jpg 01;35
17
18 [...]
```

GLOSSARY

Shell: *The program that interprets command line input entered by a user. It allows users to run executables, stores environment variables, and can use scripting to automate recurring tasks. Besides the most popular shell on Linux, Bash (the "Bourne Again Shell"), there are many others including Bash's predecessor, SH (the original "Bourne Shell") by Steve Bourne, CSH (the "C Shell"), which resembles the C programming language, or KSH (the "Korn Shell").*

THE AUTHOR

Hagen Höpfner has a higher degree in Computer Science and is a member of scientific staff at the Otto-von-Guericke University of Magdeburg, Germany. Between time spent tinkering with his collection of Linux computers, Hagen likes to direct his creative energy into song-writing, composing lyrics, and playing the guitar for a rock band called "Gute Frage !?" (translates as "Good Question").