

ANSI escape sequences

PAIN'T YOUR BASH

Angela Parszyk, Fotolia

This month, we'll show you how to decorate your console and give it a fresh new look by adding color to the command line with ANSI escape sequences. **BY HEIKE JURZIK**

Black on white, or white on black – a terminal doesn't need to be so boring. Any modern Linux console can display colors to brighten up program output, welcome messages, and even the shell prompt. This article shows you how to add a touch of color and decorate your command line.

The Color Palette

Some programs have their own parameters for adding color to output. Both *ls* and *grep* support a `--color` option that makes command output easier to read (Figure 1). ANSI escape sequences provide the underlying mechanism for this. To get a line of colored output with *echo* in Bash, you can add an escape sequence to the text (Listing 1).

The examples show different approaches to entering the escape character (^). The octal notation is `\033`; if you use this format (or the `\e` notation), the Echo program requires the `-e` option to interpret the backslash correctly. As an alternative, you can type the character in

`Vi(m)` and press the keyboard shortcut `Ctrl + V, Esc` (or `Ctrl + V, Ctrl + [`) in Bash or `Ctrl + Q, Esc` (or `Ctrl + Q, Ctrl + [` or `Ctrl + Q, 033`) in (X)Emacs.

The escape character is followed by an opening square bracket and the color code (and closed again by the *m* character). You can add an optional attribute for the foreground color, followed by a semicolon (this is *0* for the normal font in this example) and a foreground or background color (this is *34* for blue type on a standard background in this example). The colored text follows (*blue chicken*) and `^[[0m` switches back to the standard settings for the remainder of the text.

Attributes comprise two digits (e.g., *00*). A single digit is sufficient because the first digit is always a zero. Foreground attributes include *0*, *1* (bold), *4* (underlined), *5* (blinking), or *7* (inverted). The background colors have no additional attributes.

Depending on whether you are using the console (KDE's terminal Konsole

program and the virtual console, which you typically access by pressing `Ctrl + Alt + F1` through `Ctrl + Alt + F6`) or an Xterm, the colors will look different and the attributes will have a different effect on the output. For example, *1* at the console changes the foreground brightness, whereas it sets the *bold* or *normal* font aspect in Xterm. Table 1 gives an overview of foreground colors along with attributes *0* and *1*; Table 2 shows background color options (without attributes).

To combine foreground and background colors – for example, to output the text *blue chicken* on a red background – you just add the color code for the background color to a command for the foreground color (Listing 1, line 4).

Color Your Grep

If you don't like the standard color output provided by the Grep `--color` option, you can redefine the `GREP_COLOR` variable with the escape sequences discussed previously. To replace the default

GLOSSARY

ANSI: The American National Standards Institute standardizes industrial methods – control sequences for terminals in this case. Control characters include line breaks and bells, but also colored output or prompts.

Listing 1: Escape Sequence Examples

```
01 echo -e "A \033[0;34mblue chicken\033[0m. Back to normal text."
02 echo -e "A \e[0;34mblue chicken\e[0m. And back to normal text."
03 echo "A ^[[0;34mblue chicken^[[0m. And back to normal text."
04 echo -e "A \033[0;34m\033[41mblue chicken\033[0m that likes sitting
    on red things."
```

Figure 1: The ls and grep commands will color program output if needed, giving you a clearer overview.

highlight red with an underlined turquoise font on a magenta background, just type

```
export GREP_COLOR="04;36;45"
```

The quotes prevent the shell from interpreting the semicolon. The variable is only valid in the current shell session. To recast Grep permanently, you need to add the line to your `~/.bashrc` file. The next time you log on, the new configuration will be enabled. If you don't want to wait, you can type `source ~/.bashrc` to reparse the configuration file at run time.

Painting Ls

The `ls` command also uses escape sequences to color various file types for listings: directories are blue, executables are green, and symbolic links are cyan. If you want to change the default Ls colors because you can't read blue text on a black terminal background, for example, the `Dircolors` program will help you modify the `LS_COLORS` environmental variable. Of course, you could theoretically set the variable manually; however, the output from `echo $LS_COLORS` shows how long and complicated the settings for individual file types are. `Dircolors` will handle the task for you and it knows exactly what to paint in which color. Type the name of the executable without any arguments to output a list of the current settings.

Adding the `-p` switch tells `Dircolors` to list the colors used for individual file types. It is a good idea to pipe the output

Table 1: Foreground

Color code	00	01
30	Black	Gray
31	Red	Light red
32	Green	Light green
33	Brown	Yellow
34	Blue	Light blue
35	Magenta	Pink
36	Cyan	Light cyan
37	White	Light gray

Table 2: Background

Color code	Background
40	Black
41	Red
42	Green
43	Yellow (Xterm)/Brown (console)
44	Blue
45	Magenta
46	Cyan
47	Light gray

to the `less` or `more` pagers and scroll your way through the output: `dircolors -p | less`.

To change the default color settings for `ls`, first create a configuration file for `Dircolors`.

The easiest way to do this is to store the output from `dircolors -p` in a file and then edit the file with your favorite text editor. The `dircolors -p > ~/.dircolors` command writes the output to a hidden file called `.dircolors` in your home directory. Explanations of color codes and file types appear in comment (`#`) lines.

To change the color for directories from dark blue to green, for example, locate the line `DIR 01;34 # directory` and change the color code to `DIR 01;32 # directory` for a new color combination.

Save the file and quit the editor. Now type `dircolors -b ~/.dircolors` to introduce `Dircolors` to the new configuration. Directories will be shown in new colors the next time you run `Ls`. To save these settings permanently, either export the `LS_COLORS` variable or add the `dircolors` command to your `~/.bashrc`; this is the standard approach on Debian systems, for example:

```
eval "`dircolors -b`  
~/.dircolors`"
```

Because `Dircolors` can't set the variable itself, `eval` is used to run the command.

Painting Your Front Door

To paint a colorful welcome splash on your computer, you can use escape sequences. The welcome message on the virtual console is governed by the `/etc/issue` file; the content of the file is displayed before the login prompt. On Debian systems, the file issues a mes-

sage of "Debian GNU/Linux 4.0 < host-name > tty5." If you take a look at the file, you will see that the host name is represented by the `\n` sequence and the console number is represented by `\l`. Table 3 shows more options, and the `getty` manpage has a complete list.

To paint your virtual front door, you can use the color codes shown here. An entry of `Debian GNU/Linux 4.0 ^[[0;31m\n^[[0m \l` would display the host name in red, for example. The `/etc/motd` (message of the day) file is displayed when a user logs on successfully.

If you are creative, you can use color codes to paint a picture for your welcome file. The `Linux_Logo` [1] program will help you design an attractive logo or banner. Most distributions include packages on the installation medium. As an alternative, just visit an RPM search engine. The program is easy to use at the command line; `linux_logo` on a Debian system will paint a neat banner with the Debian swirl and various information about your system.

Various command-line options are available to manipulate the image. For example, `-a` draws in ASCII and `-c` draws a penguin; type `-h` for a full list. If you like one of these masterpieces, you can easily set up a new welcome page for the shell; for example,

```
# cp /etc/motd /etc/motd.orig  
# linux_logo -b > /etc/motd
```

Users who log on will now be treated to a colorful welcome message.

Colorful Future

I still haven't talked about colorful shell prompts. Read next month's Command Line for more about coloring the prompt, using control sequences to position the cursor, and adding useful data to the prompt with various other tricks. ■

Table 3: Control Characters

<code>\d</code>	Current date in YYYY-MM-DD format
<code>\t</code>	Current time in HH:MM:SS format
<code>\s</code>	Operating system name
<code>\m</code>	CPU type
<code>\r</code>	Kernel number

INFO

[1] `Linux_Logo` website:
http://www.deater.net/weave/vmwprod/linux_logo/