

# Linux Screen

Article Number: 156 | Rating: Unrated | Last Updated: Sat, Jun 2, 2018 9:42 PM

## Linux Screen

**Linux Screen** allows you to:

- Use multiple shell windows from a single SSH session.
- Keep a shell active even through network disruptions.
- Disconnect and re-connect to a shell sessions from multiple locations.
- Run a long running process without maintaining an active shell session.

## Installing Screen with Yum

Chances are that you already have screen on your system. On most Red Hat and CentOS distributions you can find Linux screen in **/usr/bin/screen**. To see if screen is in your path, you can use the which command:

```
1
                                     [root@office ~]# which screen
2
                                     /usr/bin/screen
```

If you do not have screen, then you can install it easily from an RPM or the package file for your system. For example, on CentOS you can install screen with yum:

```
1
                                     [root@office ~]# yum install screen
2
                                     ...
3
                                     Complete!
```

As you probably already have screen or can use an RPM, I am not going to cover the building of screen from source. Lets get on to how to use screen.

## Starting Linux Screen

Screen is started from the command line just like any other command:

```
1
                                     [root@office ~]# screen
```

You are now inside of a window within screen. This functions just like a normal shell except for a few special characters.

## Control Command

**Command: “Ctrl-a”**

Screen uses the command “**Ctrl-a**” that’s the control key and a lowercase “a” as a signal to send

commands to screen instead of the shell.

For example, “**Ctrl-a**” then “?”. You should now have the screen help page.

1

Screen key bindings, page 1 of 4.

2

3

Command key: ^A Literal ^A: a

4

5

break ^B b only Q

6

clear C other ^A

7

colon : pow\_break B

8

copy ^[ [ pow\_detach D

9

detach ^D d prev ^P p ^?

10

digraph ^V readbuf <

11

displays \* redisplay ^L l

12

fit F removebuf =

13

flow ^F f reset Z

14

focus ^I screen ^C c

15

hardcopy h select '

16

help ? silence \_

Key bindings are the commands the screen accepts after you hit “**Ctrl-a**”. You can reconfigure these keys to your liking using a .screenrc file, but I just use the defaults.

## Creating Windows

**Command:** “**Ctrl-a**” “**c**”.

To create a new window, you just use “**Ctrl-a**” “**c**”.

This will create a new window for you with your default prompt. Your old window is still active.

For example, I can be running top and then open a new window to do other things. Top stays running! It is still there. To try this for yourself, start up screen and then run top. (Note: I have truncated some screens to save space.)

Start top

```
1
2
3
4
5
```

```
top - 09:10:33 up 35 days, 17:26, 1 user, load
averag

Tasks: 131 total, 1 running, 130 sleeping, 0
stoppe

Cpu(s): 0.4%us, 0.2%sy, 0.0%ni, 99.4%id,
0.0%wa, 0

Mem: 12302040k total, 6363652k used,
5938388k free,

Swap: 1052248k total, 12k used, 1052236k
free,
```

Now open a new window with: "Ctrl-a" "c"

Your top window is still running you just have to switch back to it.

## Switching Between Windows

**Command: "Ctrl-a" "n"**

Screen allows you to move forward and back. In the example above, you could use "Ctrl-a "n" to get back to top. This command switches you to the next window.

The windows work like a carousel and will loop back around to your first window.

You can create several windows and toggle through them with "**Ctrl-a**" "**n**" for the next window or "**Ctrl-a**" "**p**" for the previous window.

Each process will keep running until you kill that window.

## Detaching From Screen

**Command: "Ctrl-a" "d"**

Detaching is the most powerful part of screen. Screen allows you to detach from a window and reattach later.

If your network connection fails, screen will automatically detach your session!

You can detach from the window using "**Ctrl-a**" "**d**".

This will drop you into your shell.

All screen windows are still there and you can re-attach to them later.

This is great when you are using rsync for server migration.

## Reattach to Screen

If your connection drops or you have detached from a screen, you can re-attach by just running:

```
1 [jeffh@office ~]$ screen -r
```

This will re-attach to your screen.

However, if you have multiple screens you may get this:

```
1 [jeffh@office ~]$ screen -r
2
3 There are several suitable screens on:
4 31917.pts-5.office (Detached)
5 31844.pts-0.office (Detached)
6
7 Type "screen [-d] -r [pid.]tty.host" to resume one of
8 them.
```

If you get this, just specify the screen you want.

1

```
[jeffh@office ~]$ screen -r 31844.pts-0.office
```

## Logging Your Screen Output

As a consultant, I find it important to keep track of what I do to someone's server. Fortunately, screen makes this easy.

Using "**Ctrl-a**" "**H**", creates a running log of the session.

Screen will keep appending data to the file through multiple sessions. Using the log function is very useful for capturing what you have done, especially if you are making a lot of changes. If something goes awry, you can look back through your logs.

## Getting Alerts

Screen can monitor a window for activity or inactivity. This is great if you are downloading large files, compiling, or waiting for output.

If you are waiting for output from a long running program, you can use "**Ctrl-a**" "**M**" to look for activity. Screen will then flash an alert at the bottom of the page when output is registered on that screen.

I use this when running a command that takes a long time to return data. I can just fire up the command, switch to another window and not have to keep switching back to check the status.





Alternatively, you can use "**Ctrl-a**" "**k**". You should get a message if you want to kill the screen.

Posted - Sat, Jun 2, 2018 9:42 PM. This article has been viewed 1955 times.

Online URL: <http://kb.ictbanking.net/article.php?id=156>