



Persistent Remote Shells with screen

Serge Y. Stroobandt

Copyright 2014–2022, licensed under [Creative Commons BY-NC-SA](#)

Connecting with ssh

Connect to the remote host computer using `ssh`:

```
$ ssh username@hostname.url:port
```

Evoking screen

Screen sockets and screen sessions are one and the same thing. By default, screen sessions are named according to the scheme `pid.tty.hostname`. Here are a number of commands to evoke `screen` from a normal shell.

Table 1: Commands to evoke screen from a terminal

command	function
<code>\$ screen -S sessionname</code>	Start a new screen session with a given name and attach to it.
<code>\$ screen -ls</code>	Lists all existing screen sessions.
<code>\$ screen -r</code>	Reattach* this terminal to the only existing screen session.
<code>\$ screen -r -S sessionname</code>	Reattach* this terminal to an existing screen session by name.
<code>\$ screen -r ttynumber</code>	Reattach* this terminal to an existing screen session by tty number.
<code>\$ screen -S oldsessionname -X newsessionname</code>	Rename a screen session.
<code>\$ screen -S sessionname -X quit</code>	Kill the specified screen session by executing <code>quit</code> .

Note:

* If there exists only one screen session, it is not necessary to specify any name; just type `screen -r` at the command line.

Within a session

All commands that can be issued within a screen session start by hitting the

`Ctrl` + `A` key combination, followed by another keystroke.

Table 2: Keystroke commands for use within a screen session

keystroke sequence	function
Ctrl + A ?	Show the help screen.
Ctrl + A d	Detach * the current screen session from this terminal. The screen session and its processes remain in existence.
Ctrl + A c	Create a new window and switch to it.
Ctrl + A C	Clear the screen.
Ctrl + A K	Kill the current window.
Ctrl + A p	Go to the previous screen.
Ctrl + A n	Go to the next screen.
Ctrl + A #	Go to a specific screen number .
Ctrl + A "	Select a screen from a list using the arrow keys.
Ctrl + A S	Split the current window in half horizontally.
Ctrl + A	Split the current window in half vertically.
Ctrl + A Tab	Cycle through window regions.
Ctrl + A X	Eliminate a window split.
Ctrl + A H	Log the current screen sessions in logfiles called <code>screenlog.n</code> where <code>n</code> is the number of the screen session.
Ctrl + A h	Create a screenshot of the current window in a file called <code>hardcopy.n</code> .
Ctrl + A x	Lock the screen session with the password of the current user.
Ctrl + A Esc	Enter the copy and scrollback mode.
Esc	Leave the copy and scrollback mode.

Note:

* Closing the `ssh` connection also results in a detachment of the screen session.

Typing `exit` at the command line of a screen session will destroy that screen session.

Where am I?

Experiencing an *Inception* moment? Of course, you can always hit `Ctrl` + `A` ? and see if you get the screen help information. Another way is to let the environment variable `$TERM` tell you what terminal type you are currently employing.

Inside a screen session:

```
$ echo $TERM
screen.xterm-256color
```

Inside a normal terminal:

```
$ echo $TERM
xterm-256color
```



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Other licensing available on request.

Unattended CSS typesetting with . The Prince logo is a purple rectangle with the word "Prince" in white serif font and "Print with CSS" in smaller white text above it.

This work is published at <https://hamwaves.com/screen/en/>.

Last update: Sunday, August 21, 2022.