

# GNU screen terminal commands

## cheat sheet

Author: Yuri Slobodyanyuk, <https://www.linkedin.com/in/yurislobodyanyuk/>

**NOTE** | **C-a** below stands for **Ctrl + a** keyboard sequence.

Command	Description
<code>~/screenrc &amp; /etc/screenrc</code>	Commands that the <b>screen</b> runs on start up.
<code>screen -ls</code>	List active screen sessions
<code>screen -Q windows</code>	List windows' names inside screen session
<code>screen -S &lt;session name&gt;</code>	Create a new screen session with the name <session name>
<code>screen -x</code>	Attach to the running session, also by its name
<code>screen -r &lt;session name&gt;</code>	
<code>screen -dRR</code>	Attach to the screen session, detach on other display if attached. If no session exists, will create a new one.
<code>C-a d</code>	Detach from the session, session keeps running. Here, and further <b>C</b> means Ctrl.
<code>C-a c</code>	Create new window in the session.
<code>C-a C-a</code>	Switch to the previous window.
<code>C-a "</code>	List all windows with option to navigate and enter any of them.
<code>C-w</code>	Show a list of active windows with their numbers.
<code>C-a &lt;number&gt;</code>	Switch to the window number <i>number</i> .
<code>C-a '</code>	Switch to the window by its name.
<code>C-a n</code>	Switch to the next window.
<code>C-a p</code>	Switch to the previous window.
<code>exit</code>	Exit and close current window. If it was the last window in a session, exits <b>screen</b> terminating the session.
<code>C-a k</code>	Kill the current window forcefully (not recommended).
<code>C-a : quit</code>	Quit screen session completely terminating it. Alternatively - exit all screen windows.

Command	Description
<b>C-a A</b>	Rename current window.
<b>C-a S</b>	Split windows display horizontally. Use <b>C-a c</b> to create a new window inside the new split or <b>C-X</b> to close this part of split.
<b>C-a  </b>	Split windows display vertically. Available starting screen 4.01, i.e. not available on Mac 2020 which still uses screen 4.00.
<b>C-a tab</b>	Jump to the next region in a split window display.
<b>C-a Q</b>	Unsplit the window, leaving the current window active.
<b>C-a [</b> or <b>C-a &lt;esc&gt;</b>	Enter buffer navigation mode to scroll output buffer, copy, edit and paste later. Navigation commands as per <b>vim</b> if Vim is set as editor. <b>&lt;esc&gt;</b> to leave the buffer mode.
<b>&lt;space&gt;</b>	Start/stop selection while in the buffer mode to select the text. Press <b>&lt;space&gt;</b> or <b>&lt;Enter&gt;</b> to copy the selected text. E.g. to select/copy the whole buffer: <b>C-a [ gg &lt;space&gt; G &lt;space&gt; &lt;esc&gt;</b>
<b>C-a ]</b>	Paste the selected text at the cursor of the terminal, or create a new window and say start Vim there and paste into it while in Insert mode.
<b>C-a h</b>	Dump the contents of the currently visible terminal to <b>hardcopy.&lt;n&gt;</b> file, where <i>n</i> is auto-incrementing number of your window.
<b>C-a H</b>	Start/end logging all output of the current window into a file <b>screenlog.N</b> where <b>N</b> is the window number. The data is appended, not overwritten if the file exists. Output printed before that is not logged.
<b>C-a a</b>	Send <b>Ctrl-a</b> sequence to the shell in the window, useful to jump to the beginning of the line.
<b>C-a M</b>	Monitor window for activity. When enabled, will notify you of any activity while you work in other window.
<b>C-a _</b>	Monitor window for 30 seconds of silence, will notify you in any other window as <b>Window 0: silence for 30 seconds</b>
<b>C-a ?</b>	Show all key bindings help.

Command	Description
<b>Save session state</b>	This is not possible. If you use the same layout each session, you can put start up commands to re-create it in <code>.screenrc</code> file in your home directory, but still - you cannot save the current session state, i.e. contents of the windows and their layout.
<b>Sharing session (e.g. for pair programming/tutoring)</b>	
<p>Original session (say <i>user1</i>):</p> <ol style="list-style-type: none"> <li>1. Set suid root bit on <code>screen</code> binary: <code>sudo chmod +s /usr/bin/screen</code></li> <li>2. Inside session you want to share: <code>C-a :</code> then <code>multiuser on</code> to enable sharing session.</li> <li>3. Add usernames to share the session with: <code>C-a : acladd &lt;username&gt;</code></li> </ol> <p>Connecting user (say <i>user2</i>):</p> <ol style="list-style-type: none"> <li>1. Run in shell: <code>screen -x &lt;sharing username&gt;/</code>, in our example <code>screen -x user1/</code></li> </ol>	<p>Sets up sharing the session. Another user connecting to the session views real-time its output, can enter and run commands himself. Also see <code>aclchg</code>, <code>acldel</code>, <code>aclgrp</code> for controlling what the connecting user can and cannot do. E.g. to remove <i>write</i> permissions from all users on all windows: <code>:aclchg * -w #</code></p>
<b>C-a *</b>	See who is connected to your shared screen session.

Follow me on <https://linkedin.com/in/yurislobodyanyuk/> for updates.