Cool tool of the month: The *screen* command

Datto Engineering

Presented By: Fred Mora – September 2014

Agenda

- What is the screen command
- How it works
- Examples

What is screen

- GNU screen allows you to use one terminal window for multiple shells
- A similar command is tmux. We'll look only at screen tonight, but tmux uses similar concepts.
- Note: Googling “screen” alone surprisingly gives GNU screen as the first hit!
  - I expected to see an ad for LCD monitors…
  - This tells you that screen is 1. popular, 2. has a learning curve.

Why use screen?

- The console is back! Even Microsoft dropped the GUI pretense and released PowerShell.
- Locally:
  - It's often helpful to have one directory and context per project
  - This means one or more shells opened in that dir
- On the network:
  - Cheap machines and VMs means each engineer needs to manage lots of boxes
  - Often, in spite of all cloud tools, it means opening a console.

Installation and configuration

- yum install screen or apt-get install screen
- The config file is in ~/.screenrc, more on this later.
- Most urgent change: The control prefix
  - Defaults to Ctl-A, which clashes with bash line editing (Ctl-A = go to start of line)
  - Set it to an unused key combo. I used Ctl-J.
  - In ~/.screenrc:

    ```
    escape ^Jj
    ```

Basic screen use

- Type `screen` to start the screen session manager.
- This gives you a new shell.
- Create more “windows“ (in the screen sense) with `C-J c`
- List windows with `C-J “`
- Rename windows with `C-J a`
- Check existing commands with `C-J ?`

Searching in a screen window

- During tests, it helps to keep a lot of output lines in the buffer. This is option defscrollback in ~/.screenrc:

  ```
  escape ^Jj
  defscrollback 3000
  ```

- To search in a screen buffer:
  - Ctl-J Esc enters the copy mode, which allows to search and copy contents from the buffer
  - In this mode:
    - `? foobar Enter` to search backward for foobar. / to search forward
    - `Ctl-R` and `Ctl-S` for incremental Emacs-style search backward and forward.

Using screen to prepare ssh connections

- Suppose that you deal daily with a bunch of usual ~~suspects~~ machines
- You have to type ssh -o longoptions longusername@long.host.name every time you reboot, for 20 different boxes.
- If you have a shared key: Put the ssh commands in ~/.screenrc
  screen -t you@hostname sh -o longoptions longusername@long.host.name
- If you need a password prompt, ssh will time out…

My QUAD solution

- (Quick, Ugly as Ass, and Dirty)
- Create a small script, prompted_ssh

```
#! /bin/bash
# This script asks the user to confirm by pressing enter, then runs ssh with the supplied args.
# Args:
# -l: login name for ssh
# -h: host name for ssh

Progname=$(basename $0)
while getopts h:l: opt
do
    case $opt in
     h) Host=$OPTARG
         ;;
     l) Login=$OPTARG
         ;;
     ?)
         echo "Unknown option. Usage: $Progname -l username -h hostname"
         exit
      ;;
    esac
done
echo "Hit enter to connect to $Login@$Host"
read answer
echo Connecting...
ssh -l $Login $Host;
```

My QUAD solution (cont'd)

- Then, for each machine that requires an SSH password prompt:

```
screen -t me@hostname prompted_ssh -l me -h hostname
```

Sharing a screen

- You can list the screen sessions with screen -ls

- If you are logged as the same user as the session owner (or you are root, or you read the screen manual), you can share a session opened by someone else:

```
root@dlt-syncstation-13:/home/datto# screen -ls
There are screens on:
        2462.sync4      (09/02/2014 01:04:03 PM)        (Attached)
        2331.sync3      (09/02/2014 01:03:53 PM)        (Attached)
        2243.sync2      (09/02/2014 01:03:39 PM)        (Attached)
        2111.sync1      (09/02/2014 01:03:30 PM)        (Attached)
4 Sockets in /var/run/screen/S-root.

root@dlt-syncstation-13:/home/datto# screen -x sync3
```

- To detach: `Ctl-J d`

Taking over a screen session

- Scenario: You are working from home. You VPN into your work intranet, ssh to your box.

- You have a screen session with all your necessary terms open. How do you access it?
  - screen -ls to list the screen session. Look at the creation date if you have several
  - screen -D -RR sessionname to take over
  - Example:

```
fmora@fmora-dtp:~$ screen -ls
There are screens on:
    25035.pts-31.fmora-dtp   (09/03/2014 05:10:46 PM)  (Detached)
    4122.pts-1.fmora-dtp  (08/11/2014 02:49:06 PM)  (Attached)
2 Sockets in /var/run/screen/S-fmora.

fmora@fmora-dtp:~$ screen -D -RR 25035.pts-31.fmora-dtp
[detached from 25035.pts-31.fmora-dtp]
```

Questions?