

What is covered

After reading this document you should be able to:

1. Connect to another Stats system without entering a password
2. Download (and optionally unzip) files from the command line
3. Manage jobs on a remote system

Preparation

Before you start

- i. Make sure you have added the terminal window to the Dock.
- ii. Open a terminal window

Speed things up

Don't forget that when using the command line in a terminal window, there are ways to make life easier for you:

Filename and command completion

- <tab> key completes commands and filenames

Arrow keys allow us to:

- recall previous commands
- change previous commands

1 Setting up ssh keys to speed things up

This command will be used.

Command	Purpose
ssh-keygen	Generate an ssh public/private key pair to enable moving between computers without entering a password.

Table 1: The **ssh-keygen** command

This section explains how to set up ssh keys so that you are not prompted for a password each time you move between Statistics systems. On your local desktop do the following:

- i. Enter the command

```
ssh-keygen -t rsa
```

Generate a public key/private key pair. The following output should appear

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/homes/user/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /homes/user/.ssh/id_rsa.  
Your public key has been saved in /homes/user/.ssh/id_rsa.pub.  
The key fingerprint is:  
20:30:e6:b0:5f:a0:aa:4f:d0:09:c7:b0:f4:25:45:4e  
user@gate.stats.ox.ac.uk
```

The key's randomart image is:

```
+--[ RSA 2048 ]-----+  
|o.=.oE                |  
|.X.+                  |  
|=+.o..                |  
|. = o . .             |  
|o +      S            |  
|..                     |  
|. .                    |  
| o                     |  
|. .                    |  
+-----+  
|
```

- ii. **cd**
- iii. **cd .ssh**

Note the dot before the directory name. Now check whether the file **authorized_keys** exists. If the file does use this command:

```
cat id_rsa.pub »authorized_keys
```

if the file doesn't exist use

```
cat id_rsa.pub >authorized_keys
```

Now check that this works:

```
ssh greyheron
```

The first time you connect you will see

```
The authenticity of host 'greyheron (163.1.210.96)' can't be established.
```

```
ECDSA key fingerprint is
```

```
3a:b1:d2:0d:a3:09:cf:46:e9:43:04:87:ac:f3:8e:10.
```

```
Are you sure you want to continue connecting (yes/no)?
```

Enter **yes**.

2 Download data from the web

Command	Purpose
<code>curl URL</code>	Transfer files and directories from the web address (URL) provided.
<code>unzip FILE</code>	Unpack the data from <code>.zip</code> file.

Table 2: The `scp` command

This is the simplest method for downloading data sets that are available on the web.

Use the `curl` command with the URL (web address). For example, to download the dataset 'Professional socialisation...' from the UK Data Service, to your home directory, use

```
cd
```

```
curl http://ws.ukdataservice.ac.uk/REST/Download/Download/\
DSO/1479tab_f3b6bad2bdb23b5924e346085ab27f69.zip > data.zip
```

[The command needs to be entered all on one line.]

This is a `.zip` file. To extract the contents use

```
unzip data.zip
```

Can you find the file `1479userguide.pdf`?

3 Managing jobs on remote systems

Command	Purpose
<code>screen</code>	Connect and disconnect from a session from multiple locations and allow long-running processes to persist without an active shell session.

Table 3: The `screen` command

Once you have the `R` script and any associated files on the server you are ready to submit the job.

On the remote system you should use the `screen` command. This allows you to submit `R` (and other) jobs, then disconnect from your session. Your desktop computer can then be switched

off or rebooted, without interrupting or stopping the **R** job on the remote system. To check the process of the your job you simply **ssh** again to the same server, and start the **screen** command again.

An example session would look like this.

```
ssh greyplover
screen
R CMD BATCH mandel.R &
```

Don't forget run the job in the background. This is done by appending an **&** to a command. It is also used for command which open a new window.

If you want to check that the job is running use

```
tail -f mandel.Rout
```

Once you are happy the job is running use the sequence

```
CTRL-a d
```

to detach from the screen process. You should see a message like:

```
screen
[detached from 6422.pts-0.greyplover]
```

You can then logout. To reattach the screen session log back into the server and use

```
screen -r
```

If you have multiple **screen** sessions on a server, then the command

```
screen -list
```

will display all your screen sessions. For example:

```
screen -list
There are screens on:
    7375.pts-0.greyheron      (Detached)
    6422.pts-0.greyheron      (Detached)
2 Sockets in /var/run/screen/S-jones.
```

To attach a particular session use

```
screen -r 7375.pts-0.greyplover
```

Once you have finished with a screen session reattach the session and type in

```
exit
```

You can use **screen -list** to check that it has closed. As ever, use **man screen** for full details.

There is a longer **screen** tutorial here: <http://www.rackaid.com/blog/linux-screen-tutorial-and-how-to/>.

There is an alternative to the screen command, **tmux** which is also installed on all **grey*** servers.