



Using SRB Tutorial

PPARC Summer School,
NeSC Edinburgh
Guy Warner, NeSC Training Team



The aim of this tutorial is to demonstrate how srb can be used to store files that may be accessed from multiple locations. This will be achieved by running srb commands on both a remote machine and your local machine. In the tutorial please replace

- **<localname>** by your local user account name.
- **<remotename>** by your remote ngs account name.
- **<srbname>** by the your name used for accessing srb.

Remember that all three of these names will (normally) be different.

1. Open two terminal windows. One terminal will be used to run srb commands on the remote machine grid-data.rl.ac.uk whilst the other terminal will run srb commands on your local machine.

All of the commands used for this tutorial depend on GSI for authentication. Hence you must have a valid proxy with time left. You can check the state of your proxy by typing (in either terminal)

grid-proxy-info -timeleft

This shows the amount of time left in your proxy (in seconds). If this is zero or low (anything less than 10800 (3 hours) can be considered as low) then restart your proxy by running the following commands

grid-proxy-destroy
grid-proxy-init

2. In the terminal to be used for remote srb commands type the command

gsissh -p 2222 grid-data.rl.ac.uk

For clarity commands to be run on this remote machine will from here on be prefaced with "remote> ", whilst commands to be run on your local machine will be prefaced with "local> ".

3. The first stage is to create the configuration file that is used by srb to specify your default settings. This file in particular defines the default srb server location, the default user name to use and the method of securely connecting.

The NGS has provided a script to automatically generate the configuration file for you. Just use the command

remote> /home/srb/create-mdas ngs ral-ngs1 <srbname>

This will create a lot of errors, this is because the training accounts do not work on some of the NGS sites. Inspect the contents of the file using the command

remote> cat .srb/.MdasEnv

4. The simplest way to create the configuration file on your local machine is to just copy the version that has been created on the remote machine. First though the directory it needs to be stored in must be

created

```
local> cd  
local> mkdir .srb
```

5. One way to copy the file to your local machine is to use globus-url-copy

```
local> globus-url-copy gsiftp://grid-data.rl.ac.uk/home/<remotename>/.srb/.MdasEnv  
file:///home/<localname>/.srb/.MdasEnv
```

If you are not sure of your remote name it may be found by using

```
remote> whoami
```

6. The NGS uses a system of accessing different groups of software, or modules, on request. To access the srb commands (called the Scomands) it is necessary to modify your remote environment to find the relevant commands:

```
remote> module load srb
```

7. Before running srb commands it is necessary to initialise your environment, using the Sinit command. This command is needed to handle multiple simultaneous srb sessions from the same host (not done in this tutorial). Run the commands

```
remote> Sinit  
local> Sinit
```

8. Before transferring a file into srb storage a file must be first created. Create a file using the below command

```
remote> hostname > myfile1.pparc
```

9. Transfer myfile1.txt into your default (top level) directory, or collection in srb terminology.

```
remote> Sput myfile1.pparc .
```

You can check if the file is now stored in srb by using

```
remote> Sls
```

10. This file may now be easily accessed and read from your local machine

```
local> cd  
local> Sget myfile1.pparc .  
local> cat myfile1.pparc
```

11. New collections (directories) may be easily created using the Scomands and files copied into them.

```
local> Smkdir pparcdir  
local> hostname > myfile2.pparc  
local> Sput myfile2.txt pparcdir
```

12. It is also possible to read a file from srb without having to copy the file out of srb first

```
remote> Scat pparcdir/myfile2.pparc
```

13. Having now covered how to use srb, the final stage in this tutorial is to use srb as a method of staging files to and from jobs. In your local account change to the "srb" directory with the command

```
local> cd ~/srb
```

The code in this directory will search a file found in srb for a given string and then store the output back in srb. The output has meta-data associated with it containing where the search was run and what the search term was (how to view this meta data will also be shown). The files srbDb1.txt and srbDb2.txt contain an example set of data to search, in this case a fictional list of books and their authors. The file "searcher.pl" is the perl script that performs the search and handles srb.

Before running the job **create a collection in srb called "srbExampleDb" and put the files srbDB*.txt into it**. Now run the script "searcher.pl" with the command

```
local> ./searcher.pl srbExampleDb/srbDb1.txt Richard
```

If the script successfully ran then you should get output similar to below:

```
Running on pub-234.nesc.ed.ac.uk
SRB:srbDb1.txt->LOCAL:searcherTempFile | 0.005 MB | 0.060 MB/s | 0.09 s | 2005.05.06 16:45:29
Reading contents of searcherTempFile ...
Contents read. Removing searcherTempFile
removed `searcherTempFile'
Searching srbExampleDb/srbDb1.txt for Richard
Saving results temporarily to searcherTempFile
Moving results to srb
LOCAL:searcherTempFile->SRB:srbDb1.txt.out | 0.001 MB | 0.005 MB/s | 0.13 s | 2005.05.06
16:45:29
Removing local copy of searcherTempFile
removed `searcherTempFile'
```

14. Before investigating the results, **submit the script as a job to grid-data.rl.ac.uk/jobmanager-pbs, this time with the arguments "srbExampleDb/srbDb2.txt Richard"**. You will need to specify that the job needs the srb module (see the [Job Submission Tutorial](#) for help on how to do this).
15. **To investigate the results use the SIs and Scat commands**. The script also saved extra data about the search term and the host that ran the script as metadata. A limited metadata capability is provided by srb (up to 10 strings and 2 integers). The metadata associated with the output from both runs of the script (local and remote) may be viewed with the command:

```
local> Smeta srbExampleDb/*.out
```

16. When you are finished using srb it is sensible to exit the session and (optionally) unload the NGS srb module

```
local> Sexit
remote> Sexit
remote> module unload srb
remote> exit
```