Install

Download the installer

Start downloading the installer software immediately to the machine where you wish to run the GridFTP clients and servers presented in this tutorial (your laptop or an off site UNIX machine). While GridFTP can run on many UNIX platforms, for this tutorial we only have pre-built binaries for i386 Linux and MacOS (ppc). If you plan to use a different platform please download the source bundle (note that compiling the source bundle can take over a half hour). The installers are available at:

http://gridftp.org/tutorials/

Building

Choose a directory where you wish to have Globus installed. This directory will be referred to as your GLOBUS_LOCATION. In this example we use /path/to/install. The following commands will install everything we need for GridFTP and the Simple CA software which we will use in a later exercise.

```
% tar xvfz gt-gridftp*.tar.gz
% cd gt-gridftp-installer
% ./configure  --prefix /path/to/install
    ignore any java/ant warnings.
% make gridftp  install
```

Setup your environment

In order to more easily use GridFTP various environment variables must be set based off of your GLOBUS_LOCATION. In every shell where you use Globus software you will need to perform the steps below. For convenience users often add these commands to their shell initiation files (.bashrc, .tcshrc, .login, etc).

sh

```
% export GLOBUS_LOCATION=/path/to/install
% source $GLOBUS_LOCATION/etc/globus-user-env.sh
```

csh

```
% setenv GLOBUS_LOCATION /path/to/install
% source $GLOBUS_LOCATION/etc/globus-user-env.csh
```

Environment setup steps must be repeated for every shell

Exercise 1

Getting to know the globus-gridftp-server

The Globus GridFTP server has many configuration options. All of these options are documented at: http://www.globus.org/toolkit/docs/4.0/data/gridftp/admin-index.html, but you can see them running the server with the -help command also. This will display every command line option, and will further verify that your install was successful. Take note of the *-aa* option and the *-control-interface* option because we will be using it in this exercise.

```
% globus-gridftp-server -help
```

If you have lynx installed you can get prettier output with the following command

```
% globus-gridftp-server -help -html | lynx -stdin
```

Anonymous Mode

To get a quick warm fuzzy that your server can actually do file transfers we are going to run it in anonymous mode. Anonymous mode allows any user with an ftp client to read and write (and delete) files that the server process can similarly access. To minimize potential damages we will use the *–control-interface localhost* option. This will limit access to clients running on the same machine.

```
% globus-gridftp-server -control-interface 127.0.0.1  -aa
Server listening at 127.0.0.1:58806
```

Once you run this command a contact string will be printed to stdout. In the example the contact string is localhost:58806. This is the hostname and port where FTP clients can contact the GridFTP server. Alternatively you can specify the port:

```
% globus-gridftp-server -control-interface 127.0.0.1 -aa  -p 5000
```

Standard FTP Client

Since your server is run with no special security options (GSI or SSH) you can use any FTP client that you wish to access it. Open up another shell and set up the environment for that shell as previously described. Connect to the server using the standard ftp client. When prompted for a username enter *anonymous*, and enter anything at all for a password. Once connected experiment with some FTP commands. A sample session follows *(depending on your client you may see authentication errors before being prompted for a user name):*

```
% ftp 127.0.0.1 5000
Connected to localhost.
220 localhost GridFTP Server 3.3 (gcc32dbg, 1170451261-1) ready.
Name (localhost:bresnaha): anonymous
331 Password required for ftp.
Password:
230 User ftp logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
257 "/home/bresnaha" is current directory.
ftp>
```

Two Party Transfers

Here we will do a simple two party transfer using globus-url-copy. globus-url-copy is the standard GridFTP client designed for transferring files. It is different than typical FTP clients in that it is not interactive. It is designed for batch copies. Here we will copy a known file */etc/group* to the globus-gridftp-server and store it in */tmp*. Feel free to transfer any files you wish. Once complete verify the transfer with *diff*.

```
% globus-url-copy -v file:/etc/group ftp://localhost:5000/tmp/group
Source: file:/etc/
Dest:   ftp://localhost:5000/tmp/
  group
% diff /etc/group /tmp/group
```

Third Party Transfer

In a third party transfer a client acts as an intermediary between two GridFTP servers. The client contacts both servers, tells one to send a file and the other to receive it. The data is never seen by the client. The client simply orchestrates the transfer. For simplicity sake, in this example we will contact the same server for both ends of the transfer. If we were contacting two different servers for a transfer across a real network the only difference in the command would be the hostname portion of the source and destination URL.

```
% globus-url-copy -v ftp://localhost:5000/etc/group ftp://localhost:5000/tmp/group2
Source: ftp://localhost:5000/etc/
Dest:   ftp://localhost:5000/tmp/
  group  ->  group2
```

Experiment with options

One very helpful option to globus-url-copy is -dbg. This option prints the entire control channel session (along with some other information) to stdout. This allows the user to see all of the commands sent to the server and all of the replies received from the server. This is very useful when debugging or when learning about FTP communication.

```
% globus-url-copy -dbg file:/etc/group ftp://localhost:5000/tmp/group
```

Another interesting option is -vb. This will show you the current performance of a transfer. Here we will run a transfer between /dev/zero and /dev/null. Because /dev/zero has no end of file, this allows the transfer to carry on until it is manually terminated (<ctl>+<c>).

```
% globus-url-copy -vb ftp://localhost:5000/dev/zero ftp://localhost:5000/dev/null
Source: ftp://localhost:5000/dev/
Dest:   ftp://localhost:5000/dev/
  zero  ->  null
   2401501184 bytes       458.05 MB/sec avg       458.05 MB/sec inst
Canceling copy...
```

Kill The Server

Make sure the anonymous server is no longer running. Kill it with <ctl>+<c>, or killall globus-gridftp-server

Exercise 2 : password file

Not supported on mac

Create a password file

If you trust your network and want a minimal amount of security you can run the globus-gridftp-server with clear text passwords. This security model is the one originally introduced in RFC959. We do not recommend it for long running servers open to the internet. To run the server in clear text password mode we first need to create a password file dedicated to it. The format of the password file is the same as standard system password files, however it is ill advised to use system password file. To create an entry in a GridFTP password file run the following commands:

```
% touch pwfile
% gridftp-password.pl >> pwfile
Password:
```

This will ask you for a password and then create an entry in the password file for the current user name and the given password. Take a look at the file created. You will notice that the password you typed in is not in the file in a clear text form. We have run it though a one way hash algorithm before storing it in the file.

Run the server in password mode

Simply start the server pointing it at the password file you just created.

```
% globus-gridftp-server -password-file  /full/path/of/pwfile –p 5000
```

Connect With the Standard FTP Client

Password mode is still 100% backward compatible with RFC959 therefore we can communicate with any FTP client. Try with the standard program 'ftp'.

```
% ftp localhost 5000
```

Transfer with globus-url-copy

Try a few transfers with globus-url-copy as we did in the previous exercise

```
% globus-url-copy file:/etc/group ftp://username:password@localhost:5000/tmp/group
% globus-url-copy -list ftp://username:password@localhost:5000/tmp/
```

Exercise 3 : sshftp://

In this exercise we introduce the sshftp control channel protocol. This is a very simple means of obtaining strong security on the control channel only (the data channel is not authenticated). With this approach you can run a GridFTP transfer anywhere that you can ssh. sshftp:// leverages the ubiquitous ssh/sshd programs to form control channel connections much in the same way that inetd forms connections.

Configure Client Side sshftp://

Every $GLOBUS_LOCATION must be configured for client side sshftp:// connections. In order words, if we wish to use globus-url-copy with sshftp:// URLs we must first configure the $GLOBUS_LOCATION that contains globus-url-copy in the following way:

```
% $GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp
```

Configure Server Side sshftp://

Every host that wishes to run a globus-gridftp-server which can accept sshftp:// connections must run the following command as root:

```
% $GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp -server
```

In the absence of root access a user can configure the server to allow sshftp:// connections **for that user only** with the following command:

```
% $GLOBUS_LOCATION/setup/globus/setup-globus-gridftp-sshftp -server -nonroot
```

sshftp:// Transfers

In this case a globus-gridftp-server does not need to be running. The server will be started via the sshd program. Therefore the hostname and port should be that of the sshd server. Run globus-url-copy just as you have before, simply change ftp:// to sshftp://.

```
% globus-url-copy -v file:/etc/group sshftp://localhost/tmp/group
% globus-url-copy -list sshftp://localhost/tmp/
```

Exercise 4 : GSI Security

Here we introduce the gsiftp:// security protocol. GSI provides strong security for both the control and data channel. Although it is more complicated to setup than sshftp:// it provides additional functionality, such as delegation and data channel protection, that make it well worth it.

Setup Simple CA

In order to perform transfers with GSI security we must first have the proper credentials. In order to create the needed certificates we first need a trusted certificate authority (CA). In this exercise we will create a CA using the *simple CA* software in the Globus toolkit. To do this we need to run the script *setup-simple-ca*. This user of this script will be presented with many questions. In most cases default answers to these questions are fine for this exercise. When prompted for a password enter anything you like, just remember it, you will need it for later steps. A sample session follows:

```
% $GLOBUS_LOCATION/setup/globus/setup-simple-ca
Do you want to keep this as the CA subject (y/n) [y]:
...
requests will be sent to be signed by the CA): bresnaha@mcs.anl.gov
...
[default: 5 years (1825 days)]:
...
Enter PEM pass phrase:
...
***************************************************************************

Note: To complete setup of the GSI software you need to run the
following script as root to configure your security configuration
directory:

/home/globus/LCI/GL/setup/globus_simple_ca_e802b091_setup/setup-gsi

For further information on using the setup-gsi script, use the -help
option.  The -default option sets this security configuration to be
the default, and -nonroot can be used on systems where root access is
not available.
```

Note the output instructing you to run an additional setup script. If you do not have root access run the script with the *-nonroot* option:

```
% /home/globus/LCI/GL/setup/globus_simple_ca_e802b091_setup/setup-gsi -default -nonroot
```

At this point all configuration files and programs needed to create and sign certificates are installed in the $GLOBUS_LOCATION.

Create a User Credential

In this step a user credential is created. We will be acting as both the user requesting the credential and the CA signing the credential. In most cases these will be separate entities, but we present both sides here. First we create a certificate request:

```
% grid-cert-request
...
```

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
...
```

Again you are prompted for a password. This is the proxy password. The file ~/.globus/usercert_request.pem now exists. Ordinarily you would email this to the CA for signing, but since we are the CA, we already have it and can sign it:

```
% grid-ca-sign -in ~/.globus/usercert_request.pem  -out ~/.globus/signed.pem
please enter the password for the CA key:
The new signed certificate is at: /home/globus/.globus/simpleCA/newcerts/01.pem
```

Note that prompt for the CA password. This is the password selected in the step where we created the CA, not the proxy password. The output file 01.pem is the signed user certificate. Ordinarily the CA would mail this back to the user, but since we are both the user and the CA we simply must copy it into ~/.globus/ with the following command:

```
cp ~/.globus/simpleCA/newcerts/01.pem ~/.globus/usercert.pem
```

Now in your ~/.globus directory you should see the following files:
(these files can be removed)*

```
signed.pem *
simpleCA
usercert_request.pem *
userkey.pem
usercert.pem
```

Create a proxy

The user credentials are not valid until they are activated and a proxy is created. A proxy is created with the command *grid-proxy-init* and by using the proxy password selected above. The proxy is only valid for a short amount of time and thus provides the user with some additional security. Running *grid-proxy-init* with the *–debug* and *–verify* options will help verify that we have successful obtained a user certificate.

```
% grid-proxy-init -debug -verify
```

Create a Gridmap File

Part of the gsiftp:// authentication process with the GridFTP server involves a gridmap file. This file maps a clients certificate *distinguished name* (DN) to a local user account. If the DN is not found in the gridmap file the client cannot access the server. When the GridFTP server is run as a user, it expects to find the gridmap file at *$HOME/.gridmap*. The file has the following format:

```
“<DN>” <unix account name>
```

To get the DN of your cert run grid-cert-info:

```
% grid-cert-info -subject
/O=Grid/OU=GlobusTest/OU=simpleCA-laptroll/CN=Globus Tester
% whoami
bresnaha
% vi $HOME/.gridmap
“/O=Grid/OU=GlobusTest/OU=simpleCA-laptroll/CN=Globus Tester”  bresnaha
```

Run the GridFTP Server

Most of the time a host certificate is used by a GridFTP server. However, you can use a user certificate just as easily. Here we will run transfers using the user certificate for both the server and the client. The previous step created an active proxy in the users environment and he environment setup steps in the first exercise set all the proper variables to so that the server can find the proxy. All that is left to do is simply run the server:

```
% globus-gridftp-server –p 5000
```

Perform a Transfer

Similarly we do not need to do anything special to run the client with the user's proxy. However, we do need to tell the client what to expect as a subject from the server. By default the client assumes the server will use a host based subject name in its security hand shake, but since our server is using a user certificate the subject will be different. Therefore we must tell our client the right subject to expect. To discover our subject run:

```
% grid-cert-info -subject
/O=Grid/OU=GlobusTest/OU=simpleCA-laptroll/CN=Globus Tester
```

Now run a transfer as we have in previous exercises but using the -subject switch:

```
% globus-url-copy -subject “/O=Grid/OU=GlobusTest/OU=simpleCA-laptroll/CN=Globus Tester” -vb file:/etc/group
gsiftp://localhost:5000/tmp/group
```

Experiment with various data channel security options and observe the effects on performance:

```
% globus-url-copy –dcsafe --subject “/O=Grid/OU=GlobusTest/OU=simpleCA-laptroll/CN=Globus Tester” -vb
gsiftp://localhost:5000/dev/zero gsiftp://localhost:5000/dev/null

% globus-url-copy –dcpriv --subject “/O=Grid/OU=GlobusTest/OU=simpleCA-laptroll/CN=Globus Tester” -vb
gsiftp://localhost:5000/dev/zero gsiftp://localhost:5000/dev/null
```