

# Installing GT 5.2.1

---

# Installing GT 5.2.1

## Introduction

This guide is the starting point for everyone who wants to install Globus Toolkit 5.2.1. It will take you through a basic installation that installs the following basic services: a security infrastructure (GSI), GridFTP, and Execution Services (GRAM5).

This guide is also available as a [PDF](#)<sup>1</sup>. However, each component includes online reference material, which this guide sometimes links to.

---

<sup>1</sup> installingGT.pdf

---

# Table of Contents

1. Before you begin .....	1
2. Installing GT 5.2.1 .....	2
1. Installing from Native Linux Packages .....	2
2. Installation from Source Installer .....	4
3. Basic Security Configuration .....	8
1. Obtain host certificates .....	8
2. Add authorization .....	8
3. Verify Basic Security .....	9
4. Firewall configuration .....	10
4. Basic Setup for GT 5.2.1 .....	14
5. Platform Notes .....	15
1. Platform Notes .....	15
6. Advanced Installation for GT 5.2.1 .....	16
1. Advanced Installation .....	16
A. Packaging details .....	17
1. The makefile .....	17
2. Linking with Globus Toolkit Libraries .....	17
3. The Grid Packaging Toolkit .....	17
4. Picking a flavor for a source installation .....	18
B. Environmental Variables in GT 5.2.1 .....	19
1. Common Runtime Environmental Variables .....	19
2. Security Environmental Variables .....	19
3. Data Management Environmental Variables .....	23
C. Installing SimpleCA .....	24
1. Create users .....	24
2. Create your Simple CA .....	24
3. Host certificates .....	26
4. User certificates .....	26
5. Verify the SimpleCA certificate installation .....	27
6. Configure SimpleCA for multiple machines .....	28
D. Troubleshooting your installation .....	29
E. Detailed Configuration by Component .....	30
F. Security Considerations in GT 5.2.1 .....	31
1. Common Runtime .....	31
2. Security .....	31
3. Data Management .....	32
4. Execution Management .....	33
G. Usage Statistics .....	34
1. Data Management Usage Statistics .....	34
2. Execution Management Usage Statistics .....	35
Glossary .....	38

---

## List of Tables

3.1. Summary of Globus Toolkit Traffic .....	12
C.1. CA Name components .....	24

---

# Chapter 1. Before you begin

Before you start installing the Globus Toolkit 5.2.1, there are a few things you should consider. The toolkit contains several subcomponents, and you may only be interested in some of them.

The Globus Toolkit version 5.2.1 includes:

- GSI: security
- GridFTP: file transfer
- GRAM: job execution/resource management
- MyProxy: credential repository/certificate authority
- GSI-openssh: GSI secure single sign-on remote shell

## **Important**

These all run on Unix platforms only.

If you are new to the toolkit and want to experiment with the components, you may want to use a supported RedHat based or Debian based Linux system. With the new supported native packaging installs, they are the simplest platforms on which to install GT services.

---

# Chapter 2. Installing GT 5.2.1

## 1. Installing from Native Linux Packages

### 1.1. Enabling the Globus Repository for your distribution

The GT 5.2.1 release provides source and binary RPM packages for CentOS 5 and 6, Fedora 15 and 16, RedHat Enterprise Server 5 and 6, and Scientific Linux 5 and 6, and a set of .deb packages for several Debian and Ubuntu versions, including Debian 6.0 "squeeze", and Ubuntu 10.04LTS, 10.10, 11.04, and 11.10.

This section will show how to set up and use the Globus RPM repository. If your distribution has Globus 5.2.1 packages within its repository, you can skip to "[Installing the Toolkit](#)".

The repo-config rpms for the various binary (RPM and deb) repositories can be found [here](#).<sup>1</sup>

To install from binary RPMs, get the appropriate repo-config rpm from the link above, install it with

```
# rpm -i Globus-repo-config.<distro>.noarch.rpm
```

To install from binary debs, get the appropriate repo-config deb from the link above, install it with

```
# dpkg -i globus-repository-<distro>_0.0.2_all.deb
# apt-get update
```

### 1.2. Installing the Toolkit

The components of the toolkit can be installed separately, or all at once. This section will show how to install various components, on both RPM based and Debian based Linux systems.

#### 1.2.1. Install Toolkit components on RPM based systems

using yum:

- Install GridFTP client

```
# yum groupinstall globus-data-management-client
```

- Install GRAM client

```
# yum groupinstall globus-resource-management-client
```

- Install GridFTP server

```
# yum groupinstall globus-data-management-server
```

---

<sup>1</sup> <http://www.globus.org/ftppub/gt5/5.2/5.2.1/installers/repo/>.

- Install GRAM server

```
# yum groupinstall globus-resource-management-server
```

This will install GRAM, but only with a fork LRM. To install a PBS LRM using the scheduled event generator, for example:

```
# yum install globus-gram-job-manager-pbs-setup-seg
```

- Install GridFTP server and client

```
# yum groupinstall globus-gridftp
```

- Install GRAM server and client

```
# yum groupinstall globus-gram5
```

You can also install any given package or set of packages using

```
# yum install PACKAGENAME
```

## 1.2.2. Install Toolkit components on Debian based systems

using tasksel:

- Install GridFTP client

```
# tasksel install globus-data-management-client
```

- Install GRAM client

```
# tasksel install globus-resource-management-client
```

- Install GridFTP server

```
# tasksel install globus-data-management-server
```

- Install GRAM server

```
# tasksel install globus-resource-management-server
```

This will install GRAM, but only with a fork LRM. To install a PBS LRM using the scheduled event generator, for example:

```
# apt-get install globus-gram-job-manager-pbs-setup-seg
```

- Install GridFTP server and client

```
# tasksel install globus-gridftp
```

- Install GRAM server and client

```
# tasksel install globus-gram5
```

You can also install any given package or set of packages using

```
# apt-get install PACKAGENAME
```

### 1.2.3. Toplevel targets

The toplevel targets for a groupinstall or tasksel install are

- globus-gridftp
- globus-gram5
- globus-gsi
- globus-data-management-server
- globus-data-management-client
- globus-data-management-sdk
- globus-resource-management-server
- globus-resource-management-client
- globus-resource-management-sdk

Your next step is to setup security, which includes picking a CA to trust, getting host certificates, user certificates, and creating a grid-mapfile. The next three chapters cover these topics.

With security setup, you may start a GridFTP server, and configure GRAM5. You may also start a GSI-OpenSSH daemon, or setup a MyProxy server. The following chapters will explain how to configure these technologies. If you follow the chapters in order, you will make sure of performing tasks in dependency order.

## 2. Installation from Source Installer

### Note

Installing using the Source Installer is only recommended on platforms for which native packages are not available. If you are installing onto a RedHat or Debian based Linux system, please see the section above.

 **Note**

Make you sure you check out [Platform Notes](#) for specific installation information related to your platform.

## 2.1. Required software

To build the Globus Toolkit from the source installer, first download the source from [download page](#)<sup>2</sup>, and be sure you have all of the following prerequisites installed.

This table shows specific package names (where available) for systems supported by GT 5.2.1:

Prerequisite	Reason	RedHat-based Systems	Debian-based Systems	Solaris 11	Mac OS X
C Compiler	Most of the toolkit is written in C, using C99 and POSIX.1 features and libraries.	gcc	gcc	pkg:/developer/gcc-45 or <a href="#">Solaris Studio</a> <sup>3</sup> 12.3	<a href="#">XCode</a> <sup>4</sup>
GNU or BSD tar	GPT uses the <code>-z</code> option to manipulate compressed tar files.	tar	tar	pkg:/archiver/gnu-tar	(included in OS)
GNU or BSD sed	Standard sed does not support long enough lines to process autoconf-generated scripts and Makefiles	sed	sed	pkg:/text/gnu-sed	(included in OS)
GNU Make	Standard make does not support long enough lines to process autoconf-generated makefiles	make	make	pkg:/developer/build/gnu-make	(included in XCode)
libltdl	The Globus Toolkit uses this library to portably load shared libraries.	libtool-ltdl-devel	libltdl-dev	pkg:/library/libtool/libltdl	(included in xcode)
OpenSSL 0.9.7 or higher	GSI security uses OpenSSL's implementation of the SSL protocol and X.509 certificates.	openssl-devel	libssl-dev	pkg:/library/security/openssl	(included in base OS)
Perl 5.10 or higher	GPT and parts of GRAM5 are written in Perl	perl	perl	pkg:/runtime/perl-512	(included in base OS)
Archive::Tar 0.22 or higher	GPT uses Archive::Tar to manipulate packages	perl-Archive-Tar	perl-modules	pkg:/runtime/perl-512	(included in base OS)
Compress::Zlib 1.21 or higher	GPT uses Compress::Zlib to deal with compressed packages.	perl-Compress-Zlib	perl-modules	pkg:/runtime/perl-512	(included in base OS)
Digest::MD5 2.20 or higher	GPT uses Digest::MD5 to compute package digests.	perl	perl	pkg:/runtime/perl-512	(included in base OS)

<sup>2</sup> <http://www.globus.org/toolkit/downloads/5.2.1>

<sup>4</sup> <https://developer.apple.com/xcode/>

<sup>3</sup> <http://www.oracle.com/technetwork/server-storage/solarisstudio/downloads/index.html>

Prerequisite	Reason	RedHat-based Systems	Debian-based Systems	Solaris 11	Mac OS X
File::Spec 0.8 or higher	GPT uses File::Spec indirectly via Pod::Parser	perl	perl-base	pkg:/runtime/perl-512	(included in base OS)
IO::Zlib 1.1 or higher	GPT uses IO::Zlib to deal with compressed packages.	perl-IO-Zlib	perl-modules	pkg:/runtime/perl-512	(included in base OS)
Pod::Parser 1.18 or higher	GPT uses Pod::Parser to generate command-line help screens.	perl	perl-modules	pkg:/runtime/perl-512	(included in base OS)
Test::Simple	Globus Toolkit tests use this	perl-Test-Simple	perl-modules	Install <a href="#">Test::Simple</a> <sup>5</sup> from CPAN	(included in base OS)
XML::Parser	GPT uses this.	perl-XML-Parser	libxml-parser-perl	pkg:/library/perl-5/xml-parser-512	(included in base OS)

### Note

In order to use the GNU versions of sed, tar, and make on Solaris, put `/usr/gnu/bin` at the head of your path. Also, to use all of the perl executables, add `/usr/perl5/bin` to your path.

## 2.2. Installing from Source Installer

1. Create a user named `globus`. This non-privileged user will be used to perform administrative tasks, deploying services, etc. Pick an installation directory, and make sure this account has read and write permissions in the installation directory.

### Tip

You might need to create the target directory as `root`, then `chown` it to the `globus` user:

```
# mkdir /usr/local/globus-5.2.1
# chown globus:globus /usr/local/globus-5.2.1
```

### Important

If for some reason you do *not* create a user named "globus", be sure to run the installation as a *non-root* user. In that case, make sure to pick an install directory that your user account has write access to.

2. Download the required software noted in [Section 2.1, "Required software"](#).
3. The Globus Toolkit Source Installer sets the installation directory by default to `/usr/local/globus-5.2.1`, but you may replace `/usr/local/globus-5.2.1` with whatever directory you wish to install to, by setting the prefix when you configure.

As the `globus` user, run:

<sup>5</sup> <http://search.cpan.org/search?mode=all&query=Test%3A%3ASimple>

```
globus$ ./configure --prefix=<YOUR_PREFIX_DIRECTORY>
```

You can use command line arguments to `./configure` for a more custom install. Here are the lines to enable features which are disabled by default:

```
Optional Packages:
[...]
--with-gsiopensshargs="args"
Arguments to pass to the build of GSI-OpenSSH, like
--with-tcp-wrappers
```

For a full list of options, see `./configure --help`. For a list of GSI-OpenSSH options, see [Optional Build-Time Configuration for GSI-OpenSSH](#). For more information about our packaging or about choosing a flavor, see [Packaging Details for Installing GT](#).

4. Run:

```
globus$ make
```

Note that this command can take several hours to complete. If you wish to have a log file of the build, use `tee`:

```
globus$ make 2>&1 | tee build.log
```

The syntax above assumes a Bourne shell. If you are using another shell, redirect `stderr` to `stdout` and then pipe it to `tee`.



## Note

Using `make` in parallel mode (`-j`) is not entirely safe, and is not recommended.

5. Finally, run:

```
globus$ make install
```

This completes your installation. Now you may move on to the configuration sections of the following chapters.

We recommend that you install any security advisories available for your installation, which are available from the [Advisories page](#)<sup>6</sup>. You may also be interested in subscribing to some [mailing lists](#)<sup>7</sup> for general discussion and security-related announcements.

Your next step is to setup security, which includes picking a CA to trust, getting host certificates, user certificates, and creating a grid-mapfile. The next three chapters cover these topics.

With security setup, you may start a GridFTP server, and configure GRAM5. You may also start a GSI-OpenSSH daemon, or setup a MyProxy server. The following chapters will explain how to configure these technologies. If you follow the chapters in order, you will make sure of performing tasks in dependency order.

---

<sup>6</sup> <http://www.globus.org/toolkit/advisories.html>

<sup>7</sup> [http://dev.globus.org/wiki/Mailing\\_Lists](http://dev.globus.org/wiki/Mailing_Lists)

---

# Chapter 3. Basic Security Configuration

## 1. Obtain host certificates

You must have X509 certificates to use the GT 5.2.1 software securely (referred to in this documentation as *host certificates*). For an overview of certificates for *GSI* (security) see [GSI Configuration Information](#) and [GSI Environmental Variables](#).

If you will need to be interoperable with other sites, you will need to obtain certs from a trusted Certificate Authority, such as those that are included in [IGTF](#)<sup>1</sup>. If you are simply testing the software on your own resources, SimpleCA offers an easy way to create your own certificates (see section below).

Host certificates must:

- consist of the following two files: `hostcert.pem` and `hostkey.pem`
- be in the appropriate directory for secure services: `/etc/grid-security/`
- be for a machine which has a consistent name in DNS; you should *not* run it on a computer using DHCP where a different name could be assigned to your computer.

You have the following options:

### 1.1. Request a certificate from an existing CA

Your best option is to use an already existing CA. You may have access to one from the company you work for or an organization you are affiliated with. Some universities provide certificates for their members and affiliates. Contact your support organization for details about how to acquire a certificate. You may find your CA listed in the [TERENA Repository](#)<sup>2</sup>.

If you already have a CA, you will need to follow their configuration directions. If they include a CA setup package, follow the CA's instruction on how to install the setup package. If they do not, you will need to create an `/etc/grid-security/certificates` directory and include the CA cert and signing policy in that directory. See [Configuring a Trusted CA](#) for more details.

This type of certificate is best for service deployment and Grid inter-operation.

### 1.2. SimpleCA

SimpleCA provides a wrapper around the OpenSSL CA functionality and is sufficient for simple Grid services. Alternatively, you can use OpenSSL's `CA.sh` command on its own. Instructions on how to use the SimpleCA can be found in [Installing SimpleCA](#).

SimpleCA is suitable for testing or when a certificate authority is not available.

## 2. Add authorization

Installing Globus services on your resources doesn't automatically authorize your local users to use these services. Each user must have their own user certificate, and each user certificate must be mapped to a local account.

---

<sup>1</sup> <http://www.igtf.net>

<sup>2</sup> <http://www.tacar.org/>

Add authorizations for users:

Create `/etc/grid-security/grid-mapfile` as root.

You need two pieces of information:

- the subject name of a user
- the account name it should map to.

The syntax is one line per user, with the certificate subject followed by the user account name.

Run **grid-cert-info** to get your subject name, and **whoami** to get the account name:

```
gtuser$ grid-cert-info -subject
/O=Grid/OU=GlobusTest/OU=simpleCA-mayed.mcs.anl.gov/OU=mcs.anl.gov/CN=GT User
gtuser$ whoami
gtuser
```

You may add the line by running the following as root:

```
root# $GLOBUS_LOCATION/sbin/grid-mapfile-add-entry -dn \
"/O=Grid/OU=GlobusTest/OU=simpleCA-mayed.mcs.anl.gov/OU=mcs.anl.gov/CN=GT User" \
-ln gtuser
```

The corresponding line in the `grid-mapfile` should look like:

```
"/O=Grid/OU=GlobusTest/OU=simpleCA-mayed.mcs.anl.gov/OU=mcs.anl.gov/CN=GT User" gtuser
```

### Important

The quotes around the subject name are *important*, because it contains spaces.

## 3. Verify Basic Security

Now that you have installed a trusted CA, acquired a hostcert and acquired a usercert, you may verify that your security setup is complete. As your user account, run the following command:

```
gtuser$ grid-proxy-init -verify -debug

User Cert File: /home/gtuser/.globus/usercert.pem
User Key File: /home/gtuser/.globus/userkey.pem

Trusted CA Cert Dir: /etc/grid-security/certificates

Output File: /tmp/x509up_u506
Your identity: /DC=org/DC=doegrids/OU=People/CN=GT User 332900
Enter GRID pass phrase for this identity:
Creating proxy ...+++++++
.....+++++++
Done
Proxy Verify OK
Your proxy is valid until: Fri Jan 28 23:13:22 2005
```

There are a few things you can notice from this command. Your `usercert` and `key` are located in `$HOME/.globus/`. The proxy certificate is created in `/tmp/`. The "up" stands for "user proxy", and the `_u506` will be your UNIX `userid`. It also prints out your distinguished name (DN), and the proxy is valid for 12 hours.

If this command succeeds, your single node is correctly configured.

If you get an error, or if you want to see more diagnostic information about your certificates, run the following:

```
gtuser$ grid-cert-diagnostics
```

For more troubleshooting information, see the GSI [troubleshooting guide](#)

## 4. Firewall configuration

There are four possible firewall scenarios that might present themselves: restrictions on incoming and outgoing ports for both client and server scenarios.

This section divides sites into two categories: client sites, which have users that are acting as clients to Grid services, and server sites, which are running Grid services. Server sites also often act as client sites either because they also have users on site or jobs submitted by users to the site act as clients to other sites by retrieving data from other sites or spawning sub-jobs.

### 4.1. Client Site Firewall Requirements

This section describes the requirements placed on firewalls at sites containing Globus Toolkit clients. Note that often jobs submitted to sites running Globus services will act as clients (e.g. retrieving files needed by the job, spawning subjobs), so server sites will also have client site requirements.

#### 4.1.1. Allowed Outgoing Ports

Clients need to be able to make outgoing connections freely from ephemeral ports on hosts at the client site to all ports at server sites.

#### 4.1.2. Allowed Incoming Ports

As described in [Section 3, "Job State Callbacks and Polling"](#), the Globus Toolkit GRAM service uses callbacks to communicate state changes to clients and, optionally, to stage files to/from the client. If connections are not allowed back to the Globus Toolkit clients, the following restrictions will be in effect:

- You cannot do a job submission request and redirect the output back to the client. This means the `globus-job-run` command won't work. `globus-job-submit` will work, but you cannot use `globus-job-get-output`. `globusrun` with the `-o` option also will not work.
- Staging to or from the client will also not work, which precludes the `-s` and `-w` options.
- The client cannot be notified of state changes in the job, e.g. completion.

To allow these callbacks, client sites should allow incoming connection in the ephemeral port range. Client sites wishing to restrict incoming connections in the ephemeral port range should select a port range for their site. The size of this range should be approximately 10 ports per expected simultaneous user on a given host, though this may vary depending on the actual usage characteristics. Hosts on which clients run should have the `GLOBUS_TCP_PORT_RANGE` environment variable set for the users to reflect the site's chosen range.

### 4.1.3. Network Address Translation (NAT)

Clients behind NATs will be restricted as described in [Section 4.1.2, “Allowed Incoming Ports”](#) unless the firewall and site hosts are configured to allow incoming connections.

This configuration involves:

- Select a separate portion of the ephemeral port range for each host at the site on which clients will be running (e.g. 45000-45099 for host A, 45100-45199 for host B, etc.).
- Configure the NAT to direct incoming connections in the port range for each host back to the appropriate host (e.g., configure 45000-45099 on the NAT to forward to 45000-45099 on host A).
- Configure the Globus Toolkit clients on each site host to use the selected port range for the host using the techniques described in [Section 2.1, “If client is behind a firewall”](#).
- Configure Globus Toolkit clients to advertise the firewall as the hostname to use for callbacks from the server host. This is done using the GLOBUS\_HOSTNAME environment variable. The client must also have the GLOBUS\_HOSTNAME environment variable set to the hostname of the external side of the NAT firewall. This will cause the client software to advertise the firewall's hostname as the hostname to be used for callbacks causing connections from the server intended for it to go to the firewall (which redirects them to the client).

## 4.2. Server Site Firewall Requirements

This section describes firewall policy requirements at sites that host Grid services. Sites that host Grid services often host Grid clients, however the policy requirements described in this section are adequate for clients as well.

### 4.2.1. Allowed Incoming Ports

A server site should allow incoming connections to the well-known Grid Service Ports as well as ephemeral ports. These ports are 22/tcp (for gsi-enabled openssh), 2119/tcp (for GRAM) and 2811/tcp for GridFTP.

A server not allowing incoming connections in the ephemeral port range will have the following restrictions:

- If port 2119/tcp is open, GRAM will allow jobs to be submitted, but further management of the jobs will not be possible.
- While it will be possible to make GridFTP control connections if port 2811/tcp is open, it will not possible to actually get or put files.

Server sites wishing to restrict incoming connections in the ephemeral port range should select a range of port numbers. The size of this range should be approximately 20 ports per expected simultaneous user on a given host, though this may vary depending on the actual usage characteristics. While it will take some operational experience to determine just how big this range needs to be, it is suggested that any major server site open a port range of at least a few hundred ports. Grid Services should be configured as described in [Section](#) to reflect the site's chosen range.

### 4.2.2. Allowed Outgoing Ports

Server sites should allow outgoing connections freely from ephemeral ports at the server site to ephemeral ports at client sites as well as to Grid Service Ports at other sites.

### 4.2.3. Network Address Translation (NAT)

Grid services are not supported to work behind NAT firewalls because the security mechanisms employed by Globus require knowledge of the actual IP address of the host that is being connected to.

We do note there have been some successes in running GT services behind NAT firewalls.

## 4.3. Summary of Globus Toolkit Traffic

**Table 3.1. Summary of Globus Toolkit Traffic**

Application	Network Ports	Comments
GRAM Gatekeeper(to start jobs)	To 2119/tcp on server from controllable ephemeral port on client	Connections back to client (controllable ephemeral port to controllable ephemeral port) required if executable or data staged from client or output from job sent back to client. Port 2119/tcp defined by IANA
GRAM Job-Manager	From controllable ephemeral port on client to controllable ephemeral port on server.	Port on server selected when original connection made by the client to the Gatekeeper and returned to the client in a URL. May result in connection back to client from ephemeral port on server to controllable ephemeral port on client.
GridFTP	From controllable ephemeral port on client to port 2811/tcp on server for control channel.	Port 2811/tcp defined by IANA.
GSI-Enabled SSH	From ephemeral port on client to port 22/tcp on server.	Same as standard SSH. Port 22/tcp defined by IANA.
MyProxy	From ephemeral port on client to port 7512/tcp on server.	Default. Can be modified by site.

## 4.4. Controlling The Ephemeral Port Range

Controllable ephemeral ports in the Globus Toolkit can be restricted to a given range. setting the environment variable `GLOBUS_TCP_PORT_RANGE` can restrict ephemeral ports. The value of this variable should be formatted as min,max (a comma separated pair). This will cause the GT libraries (specifically GlobusIO) to select port numbers for controllable ports in that specified range.

```
% GLOBUS_TCP_PORT_RANGE=40000 , 40010
% export GLOBUS_TCP_PORT_RANGE
% globus-gass-server
https://globicus.lbl.gov:40000
^C
%
```

This environment variable is respected by both clients and servers that are started from within the environment in which it is set. There are better ways, however, to configure a globus-job-manager or a GridFTP server to restrict its port range.

- globus-job-manager has an option, `-globus-tcp-port-range PORT_RANGE` that acts in the same manner as the environment variable. It can be specified on the command line or in the configuration file. See [here](#) for all globus-job-manager options.

- See [here](#) for GridFTP firewall information.

---

# Chapter 4. Basic Setup for GT 5.2.1

The Quickstart Guide walks you through setting up basic services on multiple machines.

---

# Chapter 5. Platform Notes

## 1. Platform Notes

None yet.

---

# Chapter 6. Advanced Installation for GT

## 5.2.1

This section introduces building from CVS, and references the advanced installation sections of component documentation.

### 1. Advanced Installation

#### 1.1. Building from CVS

See our general instructions for building GT from CVS here: <http://www.globus.org/toolkit/docs/development/remote-cvs.html><sup>1</sup>.

#### 1.2. Building a specific package from source

If you need to build a specific package from the source installer, you can use the per-package make targets that exist in the source installer's Makefile. Instead of simply running "make" in the steps above, you can, for example, run "make globus\_common" which will build the globus\_common package and its dependencies, or "make globus\_common-only" which will build exactly and only the globus\_common package. Similar targets exist for each package.

#### 1.3. Detailed installation instructions for these components

The following is a list of links to more detailed installation information available for the following components:

- [GRAM5 Installation](#)
- [Building and installing GridFTP](#)
- [Building and installing MyProxy](#)
- [Optional Build-Time Configuration for GSI-OpenSSH](#)

#### 1.4. Building an update package without an installer

If you need to build an updated package that has been released without a source installer (for example, a security update to a package, or a new version of MyProxy,) you can use "gpt-build" from your Globus Toolkit installation to do so.

GPT now installs a file similar to the 5.2 installer's config.site file as part of the GPT installation: this is necessary for libexecdir to be set correctly so that the updated files go to the same place as the original version. To build the updated package, one can, from the updated package source directory, run:

```
$gpt-build -fhs <flavor>
```

where <flavor> is the "build flavor" of the package you want to update. (This can be found by looking at the files ending with "filelist" in /usr/local/share/globus/packages/<packagename>/)

---

<sup>1</sup> /toolkit/docs/development/remote-cvs.html

---

# Appendix A. Packaging details

## 1. The makefile

You do not have to build every subcomponent of this release. The makefile specifies subtargets for different functional subpieces.

### Makefile targets

- gram: GRAM5
- gridftp: GridFTP

Note that all of these targets require the "install" target also. So, for instance, to build GridFTP alone, you would run:

```
$ ./configure --prefix=/path/to/install  
$ make gridftp install
```

## 2. Linking with Globus Toolkit Libraries

Since GT 2.0, the toolkit has included a script called `globus-makefile-header` that can be used to assemble the `cflags` and link line information when linking a program with libraries included in the toolkit. This script would walk the package metadata dependency tree to ensure that all needed flags were included, without duplicates. This method worked, and continues to work in GT 5.2.1, but we consider it to be obsolete, as we have added support for using `pkg-config`<sup>1</sup>

`Pkg-config` is very similar in concept to `globus-makefile-header`, but it has gained widespread adoption across a range of unix platforms.

To get the `cflags` and link line information to link to the `globus-ftp-client` library, for example, you could

```
$pkg-config --cflags --libs globus-ftp-client
```

Each Globus Toolkit library has a `pkg-config` metadata file that is installed as part of its `devel` package.

For more information about `pkg-config`, please see [the `pkg-config` homepage](#).<sup>2</sup>

## 3. The Grid Packaging Toolkit

The Globus Toolkit is packaged using the Grid Packaging Toolkit (GPT). The GPT provides a way for us to version packages and express dependencies between packages. The Makefile for the installer is automatically generated based on the GPT dependencies expressed in CVS. GPT versions also allow us to release update packages for small subsets of our code. For more information on the GPT, you may see its [website](#).<sup>3</sup>

---

<sup>1</sup> <http://www.freedesktop.org/wiki/Software/pkg-config>

<sup>2</sup> <http://www.freedesktop.org/wiki/Software/pkg-config>

<sup>3</sup> <http://gridpackagingtools.com/book/latest-stable/index.html>

## 4. Picking a flavor for a source installation

If you're building on a platform that is not auto-detected by the configure script, you will be prompted to specify a flavor for the `--with-flavor=` option. Typically "gcc32dbg" will work as a flavor to build 32-bit binaries using gcc. If you want to force a 64bit build, "gcc64dbg" should work.

Some platforms have better support from their native compilers, so you can use "vendorcc32dbg" to build using the native cc. Similarly, "vendorcc64dbg" will force a 64bit build instead.

---

# Appendix B. Environmental Variables in GT 5.2.1

## 1. Common Runtime Environmental Variables

### 1.1. Environmental variables for XIO

The vast majority of the environment variables that affect the Globus XIO framework are defined by the driver in use. The following are links to descriptions of the more common driver environment variables:

- [http://www.globus.org/api/c-globus-5.2.1/globus\\_xio/html/group\\_tcp\\_driver\\_envs.html](http://www.globus.org/api/c-globus-5.2.1/globus_xio/html/group_tcp_driver_envs.html)
- [http://www.globus.org/api/c-globus-5.2.1/globus\\_xio/html/group\\_file\\_driver\\_envs.html](http://www.globus.org/api/c-globus-5.2.1/globus_xio/html/group_file_driver_envs.html)
- [http://www.globus.org/api/c-globus-5.2.1/globus\\_xio/html/group\\_gsi\\_driver\\_envs.html](http://www.globus.org/api/c-globus-5.2.1/globus_xio/html/group_gsi_driver_envs.html)
- [http://www.globus.org/api/c-globus-5.2.1/globus\\_xio/html/group\\_udp\\_driver\\_envs.html](http://www.globus.org/api/c-globus-5.2.1/globus_xio/html/group_udp_driver_envs.html)

### 1.2. Environment variables for C Common Libraries

GLOBUS_HOSTNAME	Set this variable to the fully qualified name of the local machine's hostname.
GLOBUS_DOMAIN_NAME	Set this variable to the domain name to be used to qualify the local machine's hostname.
GLOBUS_ERROR_OUTPUT	Set this variable to 1 to cause Globus libraries to display error information to stderr.
GLOBUS_ERROR_VERBOSE	Set this variable to 1 to enable verbose error messages.
GLOBUS_I18N	Set this variable to 1 to attempt to use localized messages. (Currently not working)
GLOBUS_LOCATION	Set this variable to the path where the Globus Toolkit is installed, so that Globus tools can find libraries and data files. This is only needed if the Globus Toolkit was built with the source installer.
GLOBUS_THREAD_MODEL	Set to the name of a thread model to control the operation of the Globus event driver. Valid values are (depending on the platform) none for non-threaded operation (the default), pthread for POSIX threads, or windows for Windows threads.

## 2. Security Environmental Variables

### 2.1. Environmental Variables for GSI C

#### 2.1.1. Credentials

Credentials are looked for in the following order:

1. service credential

2. host credential
3. proxy credential
4. user credential

X509\_USER\_PROXY specifies the path to the *proxy credential*. If X509\_USER\_PROXY is not set, the proxy credential is created (by **grid-proxy-init**) and searched for (by client programs) in an operating-system-dependent local temporary file.

X509\_USER\_CERT and X509\_USER\_KEY specify the path to the end entity (user, service, or host) certificate and corresponding *private key*. The paths to the certificate and key files are determined as follows:

For *service credentials*:

1. If X509\_USER\_CERT and X509\_USER\_KEY exist and contain a valid certificate and key, those files are used.
2. Otherwise, if the files `/etc/grid-security/service/servicecert.pem` and `/etc/grid-security/service/servicekey.pem` exist and contain a valid certificate and key, those files are used.
3. Otherwise, if the files `$GLOBUS_LOCATION/etc/grid-security/service/servicecert.pem` and `$GLOBUS_LOCATION/etc/grid-security/service/servicekey.pem` exist and contain a valid certificate and key, those files are used.
4. Otherwise, if the files `service/servicecert.pem` and `service/servicekey.pem` in the user's `.globus` directory exist and contain a valid certificate and key, those files are used.

For *host credentials*:

1. If X509\_USER\_CERT and X509\_USER\_KEY exist and contain a valid certificate and key, those files are used.
2. Otherwise, if the files `/etc/grid-security/hostcert.pem` and `/etc/grid-security/hostkey.pem` exist and contain a valid certificate and key, those files are used.
3. Otherwise, if the files `$GLOBUS_LOCATION/etc/hostcert.pem` and `$GLOBUS_LOCATION/etc/hostkey.pem` exist and contain a valid certificate and key, those files are used.
4. Otherwise, if the files `hostcert.pem` and `hostkey.pem` in the user's `.globus` directory, exist and contain a valid certificate and key, those files are used.

For *user credentials*:

1. If X509\_USER\_CERT and X509\_USER\_KEY exist and contain a valid certificate and key, those files are used.
2. Otherwise, if the files `usercert.pem` and `userkey.pem` exist in the user's `.globus` directory, those files are used.
3. Otherwise, if a PKCS-12 file called `usercred.p12` exists in the user's `.globus` directory, the certificate and key are read from that file.

## 2.1.2. Gridmap file

GRIDMAP specifies the path to the *grid map file*, which is used to map distinguished names (found in certificates) to local names (such as login accounts). The location of the grid map file is determined as follows:

1. If the GRIDMAP environment variable is set, the grid map file location is the value of that environment variable.

2. Otherwise:

- If the user is root (uid 0), then the grid map file is `/etc/grid-security/grid-mapfile`.
- Otherwise, the grid map file is `$HOME/.gridmap`.

### 2.1.3. Trusted CAs directory

`X509_CERT_DIR` is used to specify the path to the trusted certificates directory. This directory contains information about which CAs are trusted (including the *CA certificates* themselves) and, in some cases, configuration information used by **grid-cert-request** to formulate certificate requests. The location of the trusted certificates directory is determined as follows:

1. If the `X509_CERT_DIR` environment variable is set, the trusted certificates directory is the value of that environment variable.
2. Otherwise, if `$HOME/.globus/certificates` exists, that directory is the trusted certificates directory.
3. Otherwise, if `/etc/grid-security/certificates` exists, that directory is the trusted certificates directory.
4. Finally, if `$GLOBUS_LOCATION/share/certificates` exists, then it is the trusted certificates directory.

### 2.1.4. GSI authorization callout configuration file

`GSI_AUTHZ_CONF` is used to specify the path to the *GSI authorization callout configuration file*. This file is used to configure authorization callouts used by both the gridmap and the authorization API. The location of the GSI authorization callout configuration file is determined as follows:

1. If the `GSI_AUTHZ_CONF` environment variable is set, the authorization callout configuration file location is the value of this environment variable.
2. Otherwise, if `/etc/grid-security/gsi-authz.conf` exists, then this file is used.
3. Otherwise, if `$GLOBUS_LOCATION/etc/gsi-authz.conf` exists, then this file is used.
4. Finally, if `$HOME/.gsi-authz.conf` exists, then this file is used.

### 2.1.5. GAA (Generic Authorization and Access control) configuration file

`GSI_GAA_CONF` is used to specify the path to the GSI *GAA (Generic Authorization and Access control) configuration file*. This file is used to configure policy language specific plugins to the GAA-API. The location of the GSI GAA configuration file is determined as follows:

1. If the `GSI_GAA_CONF` environment variable is set, the GAA configuration file location is the value of this environment variable.
2. Otherwise, if `/etc/grid-security/gsi-gaa.conf` exists, then this file is used.
3. Otherwise, if `$GLOBUS_LOCATION/etc/gsi-gaa.conf` exists, then this file is used.
4. Finally, if `$HOME/.gsi-gaa.conf` exists, then this file is used.

## 2.1.6. Grid security directory

GRID\_SECURITY\_DIR specifies a path to a directory containing configuration files that specify default values to be placed in certificate requests. This environment variable is used only by the **grid-cert-request** and **grid-default-ca** commands.

The location of the *grid security directory* is determined as follows:

1. If the GRID\_SECURITY\_DIR environment variable is set, the grid security directory is the value of that environment variable.
2. If the configuration files exist in `/etc/grid-security`, the grid security directory is that directory.
3. If the configuration files exist in `$GLOBUS_LOCATION/etc`, the grid security directory is that directory.

## 2.1.7. Using TLS

GLOBUS\_GSSAPI\_FORCE\_TLS specifies whether to use TLS by default when establishing a security context. The default behavior if this is not set is to use SSLv3.

## 2.1.8. Name Comparisons

GLOBUS\_GSSAPI\_NAME\_COMPATIBILITY specifies what name matching algorithms are supported by GSSAPI for mutual authentication and `gss_compare_name`. This variable may be set to any of the following values:

STRICT_GT2	Strictly backward-compatible with GT 2.0 name matching. X.509 subjectAltName values are ignored. Names with hyphens are treated as wildcarded as described in the <a href="#">security considerations</a> documentation. Name matching will rely on canonical host name associated with connection IP addresses.
STRICT_RFC2818	Support <a href="#">RFC 2818</a> <sup>1</sup> server identity processing. Hyphen characters are treated as normal part of a host name. DNSName and IPAddress subjectAltName extensions are matched against the host and port passed to GSSAPI. If subjectAltName is present, X.509 SubjectName is ignored.
HYBRID	Support a hybrid of the two previous name matching algorithms, liberally matching both hyphen wildcards, canonical names associated with IP addresses, and subjectAltName extensions.

If this variable is not set, the HYBRID behavior is used.

## 2.2. Environmental variables for MyProxy

Please refer to the [MyProxy Reference Manual](#)<sup>2</sup> for documentation of MyProxy environment variable interfaces.

## 2.3. Environmental variables for GSI-OpenSSH

The GSI-enabled OpenSSH needs to be able to find certain files and directories in order to properly function.

The items that OpenSSH needs to be able to locate, their default location and the environment variable to override the default location are:

<sup>1</sup> <http://www.ietf.org/rfc/rfc2818.txt>

<sup>2</sup> <http://myproxy.ncsa.uiuc.edu/man/>

- *Host key*  
Default location: /etc/grid-security/hostkey.pem  
Override with X509\_USER\_KEY environment variable
- *Host certificate*  
Default location: /etc/grid-security/hostcert.pem  
Override with X509\_USER\_CERT environment variable
- *Grid map file*  
Default location: /etc/grid-security/grid-mapfile  
Override with GRIDMAP environment variable
- *Certificate directory*  
Default location: /etc/grid-security/certificates  
Override with X509\_CERT\_DIR environment variable

## 3. Data Management Environmental Variables

### 3.1. Environment variables for GridFTP

The GridFTP *server* or *client* libraries do not read any environment variable directly, but the security and networking related variables described below may be useful.

- Non-WS (General) Authentication & Authorization Environment Variables.
- XIO Network Driver Environment Variables.

---

# Appendix C. Installing SimpleCA

The following are instructions for how to use SimpleCA to set up certificates for a GT 5.2.1 installation.

SimpleCA provides a wrapper around the OpenSSL CA functionality and is sufficient for simple Grid services. Alternatively, you can use OpenSSL's `CA.sh` command on its own. SimpleCA is suitable for testing or when a certificate authority (CA) is not available. You can find other CA options in [Obtaining host certificates](#).

## 1. Create users

Make sure you have the following users on your machine:

- Your *user* account, which will be used to run the client programs.
- A generic *globus* account, which will be used to perform administrative tasks. This user will also be in charge of managing the SimpleCA. To do this, make sure this account has read and write permissions in the `$GLOBUS_LOCATION` directory.

## 2. Create your Simple CA

A script was installed to set up a new SimpleCA. You only need to run this script *once* per Grid.

Run the script:

```
/usr/bin/grid-ca-create
```

### 2.1. 2.1 Configure the subject name

This script prompts you for information about the CA you wish to create:

```
The unique subject name for this CA is:
```

```
cn=Globus Simple CA, ou=simpleCA-mayed.mcs.anl.gov, ou=GlobusTest, o=Grid
```

```
Do you want to keep this as the CA subject (y/n) [y]:
```

where:

**Table C.1. CA Name components**

cn	Represents "common name". Identifies this particular certificate as the CA certificate within the "GlobusTest/simpleCA-hostname" domain, which in this case is Globus Simple CA.
ou	Represents "organizational unit". Identifies this CA from other CAs created by SimpleCA by other people. The second "ou" is specific to your hostname (in this cases GlobusTest).
o	Represents "organization". Identifies the Grid.

Press **y** to keep the default subject name (recommended).

## 2.2. Configure the CA's email

The next prompt looks like:

```
Enter the email of the CA (this is the email where certificate
requests will be sent to be signed by the CA):
```

Enter the email address where you intend to receive certificate requests. It should be your real email address that you check, not the address of the globus user.

## 2.3. Configure the expiration date

Then you'll see:

```
The CA certificate has an expiration date. Keep in mind that
once the CA certificate has expired, all the certificates
signed by that CA become invalid. A CA should regenerate
the CA certificate and start re-issuing ca-setup packages
before the actual CA certificate expires. This can be done
by re-running this setup script. Enter the number of DAYS
the CA certificate should last before it expires.
[default: 5 years (1825 days)]:
```

This is the number of days for which the CA certificate is valid. Once this time expires, the CA certificate will have to be recreated, and all of its certificates regranted.

Accept the default (recommended).

## 2.4. Enter a passphrase

Next you'll see:

```
Enter PEM pass phrase:
```

The passphrase of the CA certificate will be used only when signing certificates (with **grid-cert-sign**). It should be hard to guess, as its compromise may compromise all the certificates signed by the CA.

Enter your passphrase.

### **Important:**

Your passphrase must *not* contain any spaces.

## 2.5. Confirm generated certificate

Finally you'll see the following:

```
Installing new CA files to /etc/grid-security/certificates... done
```

```
Creating RPM source tarball... done
globus_simple_ca_68ea3306_setup-0.17.tar.gz
```

This information will be important for setting up other machines in your grid. The number `68ea3306` in the last line is known as your *CA hash*. It will be an 8 hexadecimal digit string.

## 3. Host certificates

You must request and sign a host certificate and then copy it into the appropriate directory for secure services. The certificate must be for a machine which has a consistent name in DNS; you should not run it on a computer using DHCP where a different name could be assigned to your computer.

### 3.1. 3.1 Request a host certificate

As root, run:

```
grid-cert-request -host 'hostname'
```

This creates the following files:

- `/etc/grid-security/hostkey.pem`
- `/etc/grid-security/hostcert_request.pem`
- (an empty) `/etc/grid-security/hostcert.pem`

*Note:* If you are using your own CA, follow their instructions about creating a hostcert (one which has a commonName (CN) of your hostname), then place the cert and key in the `/etc/grid-security/` location. You may then proceed to [User certificates](#).

### 3.2. Sign the host certificate

1. As globus, run:

```
grid-ca-sign -in /etc/grid-security/hostcert_request.pem -out hostsigned.pem
```

2. A signed host certificate, named `hostsigned.pem` is written to the current directory.
3. When prompted for a passphrase, enter the one you specified in [Enter a passphrase](#) (for the private key of the CA certificate.)
4. As root, move the signed host certificate to `/etc/grid-security/hostcert.pem`.

The certificate should be owned by root, and read-only for other users.

The key should be read-only by root.

## 4. User certificates

Users also must request user certificates, which you will sign using the *globus* user.

## 4.1. Request a user certificate

As your normal user account (*not globus*), run:

```
grid-cert-request
```

After you enter a passphrase, this creates

- `~$USER/.globus/usercert.pem` (empty)
- `~$USER/.globus/userkey.pem`
- `~$USER/.globus/usercert_request.pem`

Email the `usercert_request.pem` file to the SimpleCA maintainer.

## 4.2. Sign the user certificate

1. As the SimpleCA owner *globus*, run:

```
grid-ca-sign -in usercert_request.pem -out signed.pem
```

2. When prompted for a password, enter the one you specified in [Enter a passphrase](#) (for the private key of the CA certificate).
3. Now send the signed copy (`signed.pem`) back to the user who requested the certificate.
4. As your normal user account (*not globus*), copy the signed user certificate into `~/ .globus/` and rename it as `usercert.pem`, thus replacing the empty file.

The certificate should be owned by the user, and read-only for other users.

The key should be read-only by the owner.

## 5. Verify the SimpleCA certificate installation

To verify that the SimpleCA certificate is installed in `/etc/grid-security/certificates` and that your certificate is in place with the correct permissions, run:

```
user$ grid-proxy-init -debug -verify
```

After entering your passphrase, successful output will look like:

```
[bacon@mayed schedulers]$ grid-proxy-init -debug -verify
```

```
User Cert File: /home/user/.globus/usercert.pem
```

```
User Key File: /home/user/.globus/userkey.pem
```

```
Trusted CA Cert Dir: /etc/grid-security/certificates
```

```
Output File: /tmp/x509up_u1817
```

```
Your identity: /O=Grid/OU=GlobusTest/OU=simpleCA-mayed.mcs.anl.gov/OU=mcs.anl.gov/CN=User
```

```
Enter GRID pass phrase for this identity:
Creating proxy .....+++++++
.....+++++++
Done
Proxy Verify OK
Your proxy is valid until: Sat Mar 20 03:01:46 2004
```

If you get an error, or if you want to see more diagnostic information about your certificates, run the following:

```
gtuser$ grid-cert-diagnostics
```

For more troubleshooting information, see the GSI [troubleshooting guide](#)

## 6. Configure SimpleCA for multiple machines

So far, you have a single machine configured with SimpleCA certificates. Recall that in [Confirm generated certificate](#) an RPM Source tarball CA package was created as `globus_simple_ca_HASH.tar.gz` in your current directory. If you want to use your certificates on another machine, you must install that CA package on that machine.

To create an RPM to install on a second machine, run:

```
rpmbuild -ta ./globus_simple_ca_HASH.tar.gz
```

This will create a `~/rpmbuild/RPMS/noarch/globus_simple_ca_HASH-1.0-1*.rpm` which you can copy to your second machine and install using the standard RPM commands.

Build files are also included for creating Debian packages for your SimpleCA.

If you are going to run services on the second host, it will need its own [Host certificates for SimpleCA](#) and `grid-mapfile` (as described in the basic configuration instructions in [Section 2, "Add authorization"](#)).

You may re-use your user certificates on the new host. You will need to copy the requests to the host where the SimpleCA was first installed in order to sign them.

---

# Appendix D. Troubleshooting your installation

The following is a list of links that take you to information about troubleshooting your installation by component

- Common Runtime components
  - [XIO](#)
  - [C Common Libraries](#)
- Security components
  - [GSI C](#)
  - [MyProxy](#)
  - [GSI-OpenSSH](#)
- Data Management components
  - [GridFTP](#)
- Execution Management components
  - [GRAM5](#)

---

# Appendix E. Detailed Configuration by Component

The following is a list of links that take you to information about detailed configuration for each component.

- Common Runtime components
  - [XIO](#)
- Security components
  - [GSLC](#)
  - [MyProxy](#)
  - [GSI-OpenSSH](#)
- Data Management components
  - [GridFTP](#)
- Execution Management components
  - [GRAM5](#)

---

# Appendix F. Security Considerations in GT 5.2.1

## 1. Common Runtime

### 1.1. Security considerations for XIO

Globus XIO is a framework for creating network protocols. Several existing protocols, such as TCP, come built into the framework. XIO itself introduces no known security risks. However, all network applications expose systems to the risks inherent when outsiders can connect to them. Also included in the XIO distribution is the GSI driver, which provides a driver that allows for secure connections.

## 2. Security

### 2.1. Security considerations for GSI C

- During host authorization, the toolkit treats host names of the form "hostname-*ANYTHING*.edu" as equivalent to "hostname.edu". This means that if a service was set up to do host authorization and hence accept the certificate "hostname.edu", it would also accept certificates with DNs "hostname-*ANYTHING*.edu".

The feature is in place to allow a multi-homed host following a "hostname-interface" naming convention, to have a single host certificate. For example, host "grid.test.edu" would also accept the likes of "grid-1.test.edu" or "grid-foo.test.edu".



#### Note

The string *ANYTHING* matches only the name of the host and not domain components. This means that "hostname.edu" will not match "hostname-foo.sub.edu", but will match "host-foo.edu".



#### Note

If a host was set up to accept "hostname-1.edu", it will not accept "hostname-*ANYTHING*.edu" but will accept "hostname.edu". That is, only one of the names being compared may contain the hyphen character in the host name.

A [bug<sup>1</sup>](#) has been opened to see if this feature needs to be modified.

In GT 5.2.1, it is possible to disable this behavior, by setting the environment variable `GLOBUS_GSS-API_NAME_COMPATIBILITY` to `STRICT_RFC2818`.

### 2.2. MyProxy Security Considerations

You should choose a well-protected host to run the myproxy-server on. Consult with security-aware personnel at your site. You want a host that is secured to the level of a Kerberos KDC, that has limited user access, runs limited services, and is well monitored and maintained in terms of security patches.

---

<sup>1</sup> [http://bugzilla.globus.org/bugzilla/show\\_bug.cgi?id=2969](http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=2969)

For a typical myproxy-server installation, the host on which the myproxy-server is running must have `/etc/grid-security` created and a *host certificate* installed. In this case, the myproxy-server will run as root so it can access the host certificate and key.

## 2.3. GSI-OpenSSH Security Considerations

GSI-OpenSSH is a modified version of [OpenSSH](http://www.openssh.org/)<sup>2</sup> and includes full OpenSSH functionality. For more information on OpenSSH security, see the [OpenSSH Security](http://www.openssh.org/security.html)<sup>3</sup> page.

# 3. Data Management

## 3.1. Security Considerations

### 3.1.1. Ways to configure your server

There are various ways to configure your GridFTP server that provide varying levels of security. For more information, see [System Administrator's Guide](#).

### 3.1.2. Firewall requirements

If the GridFTP *server* is behind a firewall:

1. Contact your network administrator to open up port 2811 (for GridFTP control channel connection) and a range of ports (for GridFTP data channel connections) for the incoming connections. If the firewall blocks the outgoing connections, open up a range of ports for outgoing connections as well.
2. Set the environment variable `GLOBUS_TCP_PORT_RANGE`:

```
export GLOBUS_TCP_PORT_RANGE=min,max
```

where `min,max` specify the port range that you have opened for the incoming connections on the firewall. This restricts the listening ports of the GridFTP server to this range. Recommended range is 1000 (e.g., 50000-51000) but it really depends on how much use you expect.

3. If you have a firewall blocking the outgoing connections and you have opened a range of (local) ports, set the environment variable `GLOBUS_TCP_SOURCE_RANGE`:

```
export GLOBUS_TCP_SOURCE_RANGE=min,max
```

where `min,max` specify the port range that you have opened for the outgoing connections on the firewall. This restricts the GridFTP server to bind to a local port in this range for outbound connections. Recommended range is twice the range used for `GLOBUS_TCP_PORT_RANGE`, because if parallel TCP streams are used for transfers, the listening port would remain the same for each connection but the connecting port would be different for each connection.



### Note

If the server is behind NAT, the `--data-interface <real ip/hostname>` option needs to be used on the server.

---

<sup>2</sup> <http://www.openssh.org/>

<sup>3</sup> <http://www.openssh.org/security.html>

If the GridFTP *client* is behind a firewall:

1. Contact your network administrator to open up a range of ports (for GridFTP data channel connections) for the incoming connections. If the firewall blocks the outgoing connections, open up a range of ports for outgoing connections as well.

2. Set the environment variable `GLOBUS_TCP_PORT_RANGE`

```
export GLOBUS_TCP_PORT_RANGE=min,max
```

where min,max specify the port range that you have opened for the incoming connections on the firewall. This restricts the listening ports of the GridFTP client to this range. Recommended range is 1000 (e.g., 50000-51000) but it really depends on how much use you expect.

3. If you have a firewall blocking the outgoing connections and you have opened a range of ports, set the environment variable `GLOBUS_TCP_SOURCE_RANGE`:

```
export GLOBUS_TCP_PORT_RANGE=min,max
```

where min,max specify the port range that you have opened for the outgoing connections on the firewall. This restricts the GridFTP client to bind to a local port in this range for outbound connections. Recommended range is twice the range used for `GLOBUS_TCP_PORT_RANGE`, because if parallel TCP streams are used for transfers, the listening port would remain the same for each connection but the connecting port would be different for each connection.

Additional information on Globus Toolkit Firewall Requirements is available [here](#)<sup>4</sup>.

## 4. Execution Management

### 4.1. Security Considerations

#### 4.1.1. Gatekeeper Security Considerations

GRAM5 runs different parts of itself under different privilege levels. The **globus-gatekeeper** runs as root, and uses its root privilege to access the host's private key. It uses the *grid map file* to map *Grid Certificates* to local user ids and then uses the `setuid()` function to change to that user and execute the **globus-job-manager** program

#### 4.1.2. Job Manager Security Considerations

The **globus-job-manager** program runs as a local non-root account. It receives a delegated limited *proxy certificate* from the GRAM5 client which it uses to access Grid storage resources via GridFTP and to authenticate job signals (such as client cancel requests), and send job state callbacks to registered clients. This proxy is generally short-lived, and is automatically removed by the job manager when the job completes.

The **globus-job-manager** program uses a publicly-writable directory for job state files. This directory has the *sticky* bit set, so users may not remove other users files. Each file is named by a UUID, so it should be unique.

#### 4.1.3. Fork SEG Module Security Considerations

The Fork Scheduler Event Generator module uses a globally writable file for job state change events. This is not recommended for production use.

---

<sup>4</sup> <http://www.globus.org/toolkit/security/firewalls/>

---

# Appendix G. Usage Statistics

The following components collect usage statistics as outlined here (along with information about how to opt-out): [Usage Statistics in GT](#)<sup>1</sup>

## 1. Data Management Usage Statistics

### 1.1. GridFTP-specific usage statistics

The following GridFTP-specific usage statistics are sent in a UDP packet at the end of each transfer, in addition to the standard header information described in the [Usage Stats](#)<sup>2</sup> section.

- Start time of the transfer
- End time of the transfer
- Version string of the server
- TCP buffer size used for the transfer
- Block size used for the transfer
- Total number of bytes transferred
- Number of parallel streams used for the transfer
- Number of stripes used for the transfer
- Type of transfer (STOR, RETR, LIST)
- FTP response code -- Success or failure of the transfer



#### Note

The client (`globus-url-copy`) does NOT send any data. It is the *servers* that send the usage statistics.

We have made a concerted effort to collect only data that is not too intrusive or private and yet still provides us with information that will help improve and gauge the usage of the GridFTP server. Nevertheless, if you wish to disable this feature for GridFTP only, use the `-disable-usage-stats` option of [globus-gridftp-server](#). Note that you can disable transmission of usage statistics globally for all C components by setting "GLOBUS\_USAGE\_OPTOUT=1" in your environment.

Also, please see our [policy statement](#)<sup>3</sup> on the collection of usage statistics.

---

<sup>1</sup> ../Usage\_Stats.html

<sup>2</sup> /toolkit/docs/5.0/5.2.1/Usage\_Stats.html

<sup>3</sup> /toolkit/docs/latest-stable/Usage\_Stats.html

## 2. Execution Management Usage Statistics

### 2.1. GRAM5-specific usage statistics

The following usage statistics are sent by default in a UDP packet (in addition to the GRAM component code, packet version, timestamp, and source IP address) at the end of each job.

- Job Manager Session ID
- dryrun used
- RSL Host Count
- Timestamp when job hit GLOBUS\_GRAM\_PROTOCOL\_JOB\_STATE\_UNSUBMITTED
- Timestamp when job hit GLOBUS\_GRAM\_PROTOCOL\_JOB\_STATE\_FILE\_STAGE\_IN
- Timestamp when job hit GLOBUS\_GRAM\_PROTOCOL\_JOB\_STATE\_PENDING
- Timestamp when job hit GLOBUS\_GRAM\_PROTOCOL\_JOB\_STATE\_ACTIVE
- Timestamp when job hit GLOBUS\_GRAM\_PROTOCOL\_JOB\_STATE\_FAILED
- Timestamp when job hit GLOBUS\_GRAM\_PROTOCOL\_JOB\_STATE\_FILE\_STAGE\_OUT
- Timestamp when job hit GLOBUS\_GRAM\_PROTOCOL\_JOB\_STATE\_DONE
- Job Failure Code
- Number of times status is called
- Number of times register is called
- Number of times signal is called
- Number of times refresh is called
- Number of files named in file\_clean\_up RSL
- Number of files being staged in (including executable, stdin) from http servers
- Number of files being staged in (including executable, stdin) from https servers
- Number of files being staged in (including executable, stdin) from ftp servers
- Number of files being staged in (including executable, stdin) from gsiftp servers
- Number of files being staged into the GASS cache from http servers
- Number of files being staged into the GASS cache from https servers
- Number of files being staged into the GASS cache from ftp servers
- Number of files being staged into the GASS cache from gsiftp servers
- Number of files being staged out (including stdout and stderr) to http servers

- Number of files being staged out (including stdout and stderr) to https servers
- Number of files being staged out (including stdout and stderr) to ftp servers
- Number of files being staged out (including stdout and stderr) to gsiftp servers
- Bitmask of used RSL attributes (values are  $2^{\text{id}}$  from the `gram5_rsl_attributes` table)
- Number of times `unregister` is called
- Value of the `count` RSL attribute
- Comma-separated list of string names of other RSL attributes not in the set defined in `globus-gram-job-manager.rvf`
- Job type string
- Number of times the job was restarted
- Total number of state callbacks sent to all clients for this job

The following information can be sent as well in a job status packet but it is not sent unless explicitly enabled by the system administrator:

- Value of the executable RSL attribute
- Value of the arguments RSL attribute
- IP address and port of the client that submitted the job
- User DN of the client that submitted the job

In addition to job-related status, the job manager sends information periodically about its execution status. The following information is sent by default in a UDP packet (in addition to the GRAM component code, packet version, timestamp, and source IP address) at job manager start and every 1 hour during the job manager lifetime:

- Job Manager Start Time
- Job Manager Session ID
- Job Manager Status Time
- Job Manager Version
- LRM
- Poll used
- Audit used
- Number of restarted jobs
- Total number of jobs
- Total number of failed jobs
- Total number of canceled jobs

- Total number of completed jobs
- Total number of dry-run jobs
- Peak number of concurrently managed jobs
- Number of jobs currently being managed
- Number of jobs currently in the UNSUBMITTED state
- Number of jobs currently in the STAGE\_IN state
- Number of jobs currently in the PENDING state
- Number of jobs currently in the ACTIVE state
- Number of jobs currently in the STAGE\_OUT state
- Number of jobs currently in the FAILED state
- Number of jobs currently in the DONE state

Also, please see our [policy statement](#)<sup>4</sup> on the collection of usage statistics.

---

<sup>4</sup>[/toolkit/docs/latest-stable/Usage\\_Stats.html](/toolkit/docs/latest-stable/Usage_Stats.html)

---

# Glossary

## C

CA Certificate	The CA's certificate. This certificate is used to verify signature on certificates issued by the CA. GSI typically stores a given CA certificate in <code>/etc/grid-security/certificates/&lt;hash&gt;.0</code> , where <code>&lt;hash&gt;</code> is the hash code of the CA identity.
certificate	A public key plus information about the certificate owner bound together by the digital signature of a CA. In the case of a CA certificate, the certificate is self signed, i.e. it was signed using its own private key.
client	A process that sends commands and receives responses. Note that in GridFTP, the client may or may not take part in the actual movement of data.

## G

GAA configuration file	A file that configures the Generic Authorization and Access control GAA libraries. When using GSI, this file is typically found in <code>/etc/grid-security/gsi-gaa.conf</code> .
grid map file	A file containing entries mapping certificate subjects to local user names. This file can also serve as a access control list for GSI enabled services and is typically found in <code>/etc/grid-security/grid-mapfile</code> . For more information see the Gridmap section <a href="#">here</a> .
grid security directory	The directory containing GSI configuration files such as the GSI authorization callout configuration and GAA configuration files. Typically this directory is <code>/etc/grid-security</code> . For more information see <a href="#">this</a> .
Grid Security Infrastructure (GSI)	GSI stands for Grid Security Infrastructure and is used to describe the original infrastructure of GT security, which is comprised of SSL, PKI and proxy certificates.
GSI authorization callout configuration file	A file that configures authorization callouts to be used for mapping and authorization in GSI enabled services. When using GSI this file is typically found in <code>/etc/grid-security/gsi-authz.conf</code> .

## H

host certificate	An <a href="#">EEC</a> <sup>3</sup> belonging to a host. When using GSI this certificate is typically stored in <code>/etc/grid-security/hostcert.pem</code> . For more information on possible host certificate locations see the <a href="#">GSI C Developer's Guide</a> .
host credentials	The combination of a host certificate and its corresponding private key.

---

<sup>3</sup> #EEC

## P

**private key** The private part of a key pair. Depending on the type of certificate the key corresponds to it may typically be found in `$HOME/.globus/userkey.pem` (for user certificates), `/etc/grid-security/hostkey.pem` (for host certificates) or `/etc/grid-security/<service>/<service>key.pem` (for service certificates).

For more information on possible private key locations see [this](#).

**proxy certificate** A short lived certificate issued using a EEC. A proxy certificate typically has the same effective subject as the EEC that issued it and can thus be used in its place. GSI uses proxy certificates for single sign on and delegation of rights to other entities.

For more information about types of proxy certificates and their compatibility in different versions of GT, see <http://dev.globus.org/wiki/Security/ProxyCertTypes>.

**proxy credentials** The combination of a proxy certificate and its corresponding private key. GSI typically stores proxy credentials in `/tmp/x509up_u<uid>` , where `<uid>` is the user id of the proxy owner.

## S

**server** A process that receives commands and sends responses to those commands. Since it is a server or service, and it receives commands, it must be listening on a port somewhere to receive the commands. Both FTP and GridFTP have IANA registered ports. For FTP it is port 21, for GridFTP it is port 2811. This is normally handled via `inetd` or `xinetd` on Unix variants. However, it is also possible to implement a daemon that listens on the specified port. This is described more fully in in the Architecture section of the GridFTP Developer's Guide.

**service credentials** The combination of a service certificate and its corresponding private key.

## U

**user credentials** The combination of a user certificate and its corresponding private key.