

Installing GT 5.0.2

Installing GT 5.0.2

Introduction

This guide is the starting point for everyone who wants to install Globus Toolkit 5.0.2. It will take you through a basic installation that installs the following Base Services: a security infrastructure (GSI), GridFTP, RLS and Execution Services (GRAM5).

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

¹ installingGT.pdf

Table of Contents

1. Before you begin	1
2. Software Prerequisites	2
1. Required software	2
2. Optional software	2
3. Platform Notes	3
1. Apple MacOS X	3
2. Debian	3
3. Fedora Core	3
4. FreeBSD	3
5. HP/UX	3
6. IBM AIX	4
7. Red Hat	4
8. SGI Altix (IA64 running Red Hat)	4
9. Sun Solaris	4
10. SuSE Linux	5
11. Tru64 Unix	5
4. Installing GT 5.0.2	6
1. Basic Installation	6
2. Advanced Installation	7
5. Basic Security Configuration	9
1. Set environment variables	9
2. Obtain host certificates	9
3. Add authorization	10
4. Verify Basic Security	11
5. Firewall configuration	11
6. Basic Setup for GT 5.0.2	12
A. Packaging details	13
1. The makefile	13
2. The Grid Packaging Toolkit	13
3. Picking a flavor for a source installation	13
B. Environmental Variables in GT 5.0.2	14
1. Common Runtime Environmental Variables	14
2. Security Environmental Variables	14
3. Data Management Environmental Variables	18
C. Installing SimpleCA	19
1. Create users	19
2. Run the setup script	19
3. Host certificates	22
4. User certificates	23
5. Verify the SimpleCA certificate installation	23
6. Configure SimpleCA for multiple machines	24
D. Troubleshooting your installation	25
E. Detailed Configuration by Component	26
F. Security Considerations in GT 5.0.2	27
1. Common Runtime	27
2. Security	27
3. Data Management	28
4. Execution Management	30
G. Usage Statistics	31
1. Data Management Usage Statistics	31
2. Execution Management Usage Statistics	33

Glossary 36

List of Tables

C.1. CA Name components	19
-------------------------------	----

Chapter 1. Before you begin

Before you start installing the Globus Toolkit 5.0.2, there are a few things you should consider. The toolkit contains many subcomponents, and you may only be interested in some of them.

There are non-web services implementations of:

- Security (GSI)
- File Transfers GridFTP
- Resource Management (GRAM5)
- and Replica Location Service

 **Important**

These all run on Unix platforms only.

Therefore, if you are new to the toolkit and want to experiment with the components, you may want to use a Unix system.

Chapter 2. Software Prerequisites

1. Required software

- Globus Toolkit installer, from Globus Toolkit [download page](#)¹
- C compiler. If `gcc`², avoid version 3.2. Versions 3.2.1 and 2.95.x are okay.
- [GNU tar](#)³
- [GNU sed](#)⁴
- [zlib 1.1.4+](#)⁵
- [GNU Make](#)⁶
- Openssl 0.9.7 or later. For linux systems, this includes the -devel version of the package.
- gpt-3.2autotools2004 (shipped with the installers, but required if building standalone GPT bundles/packages)

2. Optional software

- [iODBC](#)⁷ (compile requirement for RLS) For a more complete list of RLS prerequisites, see [Prerequisites for RLS](#).
 - A Relational Database Server (RDBMS) that supports ODBC (we provide instructions for both PostgreSQL and MySQL [olink to appendix]):
 - If you use PostgreSQL, you'll also need `psqlODBC` (the ODBC driver for PostgreSQL).
 - If you use MySQL, you'll also need the `MyODBC` (Connector/ODBC) packages. MySQL's top level installation directory must be specified. By default these are assumed to be in `$GLOBUS_LOCATION`.
 - The package is used to interface to the ODBC layer of the RDBMS. The location of `iODBC` and the `odbc.ini` file must be specified before installing the RLS server.

¹ <http://www.globus.org/toolkit/downloads/>

² <http://gcc.gnu.org>

³ <http://www.gnu.org/software/tar/tar.html>

⁴ <http://www.gnu.org/software/sed/sed.html>

⁵ <http://www.zip.org/zlib/>

⁶ <http://www.gnu.org/software/make/>

⁷ <http://www.iodbc.org/>

Chapter 3. Platform Notes

In this section, the word "flavor" refers to a combination of compiler type (gcc or other), 32 or 64 bit libraries, and debugging enabled or not.

1. Apple MacOS X

MacOS binaries are provided. The Debian workaround is not needed anymore (see [bug 5481](#)¹).

For Mac OS X 10.6.x:

To build GT5.0.x on Mac OS X 10.6.x, use the following commands:

```
CFLAGS=-m32
export CFLAGS
LDFLAGS=-m32
export LDFLAGS
./configure --with-flavor=gcc32dbg --prefix="$(pwd)/gt5" \
--with-buildopts="GSI_OPENSSH_GPTMACRO=--without-openssl-header-check --verbose"
make
make install
```

2. Debian

Some kernel/libc combinations trigger a threading problem. See [bug #2194](#)². The workaround is to set LD_ASSUME_KERNEL=2.2.5 in your environment.

3. Fedora Core

Fedora Core 2 and later ship with a broken ant. Install your own ant from <http://ant.apache.org>³ and either remove the ant RPM or edit /etc/ant.conf, setting ANT_HOME to your own ant installation.

4. FreeBSD

No known issues.

5. HP/UX

5.0.2 has not been tested on HP-UX.

For HP-UX/IA64 and for additional details about GT5 on HP-UX/PA-RISC, please consult the [HP GT4 support](#)⁴ page.

¹ http://bugzilla.ncsa.uiuc.edu/show_bug.cgi?id=5481

² http://bugzilla.globus.org/globus/show_bug.cgi?id=2194

³ <http://ant.apache.org/>

⁴ <http://h71028.www7.hp.com/enterprise/cache/329379-0-0-0-121.html>

6. IBM AIX

Supported flavors are `vendorcc32dbg/vendorcc32` and `vendorcc64dbg/vendorcc64` using the Visual Age compilers (xlc). No gcc flavors are supported. Specify a flavor using `--with-flavor=flavor`.

GNU sed, tar, and make are required before the IBM ones in the PATH.

The toolkit has been tested on AIX 5.2 with:

- Visual Age C/C++ 6.0
- 32 bit version of IBM Java 1.4
- Apache Ant 1.5.4

7. Red Hat

RHEL5 has upgraded to openssl0.9.8. Our RHAS3/4 binaries are built using openssl0.9.7. You will either need to build from source on RHEL5+, or install the older RHEL4 openssl 0.9.7 RPMs.

When building from source on a Red Hat Enterprise line version 3 or 4 based OS, GPT might have a problem retrieving exit codes from subshells. You might see errors which says they were both successful and failed:

```
BUILD SUCCESSFUL Total time: 11 seconds ERROR: Build has failed make: ***
    [globus_wsrp_servicegroup] Error 10
```

The workaround is to configure with `--with-buildopts="-verbose"`

8. SGI Altix (IA64 running Red Hat)

Some extra environment variables are required for building MPI flavors. For the Intel compiler:

```
export CC=icc export CFLAGS=-no-gcc export CXX=icpc export CXXFLAGS=-no-gcc export
    LDFLAGS=-lmpi
```

For the GNU compiler:

```
export CC=gcc export CXX=g++ export LDFLAGS=-lmpi
```

In both cases, configure with `--with-flavor=mpicc64`

9. Sun Solaris

Supported flavors are `gcc32`, `gcc64`, `vendorcc32` and `vendorcc64`. The `dbg` flavors should work as well. For `gcc64`, a gcc built to target 64 bit object files is required. The `gcc32dbg` flavor will be used by default. Specify other flavors using `--with-flavor=flavor`.

For Solaris 10, you may need to use an updated GNU binutils, or the provided Sun `/usr/ccs/bin/ld` to link. See [binutils bug 1031](#)⁵ for details on Solaris 10 symbol versioning errors.

GPT has problems with the Sun provided perl and tar: <http://grid.ncsa.illinois.edu/gpt/book/latest-stable/ch01s07.html>

⁵ http://sources.redhat.com/bugzilla/show_bug.cgi?id=1031

Solaris 9 may need some environment variables set to build with vendor-provided openssl (see http://dev.globus.org/wiki/C_Security:_Vendor_OpenSSL#Known_Issues_and_Workarounds)

The toolkit has been tested on Solaris 9 with:

- Sun Workshop 6 update 2 C 5.3
- gcc 3.4.3
- Sun Java 1.4.2_02
- Apache Ant 1.5.4

10. SuSE Linux

No known issues.

11. Tru64 Unix

Specify `--with-flavor=vendorcc64` on the configure line. GNU tar, GNU sed, and GNU make are required on the PATH.

The toolkit has been tested on Tru64 UNIX (V5.1A and V5.1B) with:

- HP C V6.4-009 and V6.5-003 compilers
- Java 1.4.2_04
- Apache Ant 1.6.2

For additional details about GT5 on Tru64 Unix, please consult the [HP GT4 support](#)⁶ page.

⁶ <http://h71028.www7.hp.com/enterprise/cache/329379-0-0-0-121.html>

Chapter 4. Installing GT 5.0.2

1. Basic Installation

Note

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

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-5.0.2
# chown globus:globus /usr/local/globus-5.0.2
```

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 [Software Prerequisites for Installing GT](#).
3. In this guide we will assume that you are installing to `/usr/local/globus-5.0.2`, but you may replace `/usr/local/globus-5.0.2` with whatever directory you wish to install to.

As the `globus` user, run:

```
globus$ export GLOBUS_LOCATION=/usr/local/globus-5.0.2
globus$ ./configure --prefix=$GLOBUS_LOCATION
```

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 Features:
--enable-il8n          Enable internationalization. Default is disabled.
[...]
Optional Packages:
[...]
--with-iodbc=dir       Use the iodbc library in dir/lib/libiodbc.so.
Required for RLS builds.
--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](#)¹. You may also be interested in subscribing to some [mailing lists](#)² 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, setup a MyProxy server, and run RLS. 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. Advanced Installation

2.1. Building from CVS

See our general instructions for building GT from CVS here: <http://www.globus.org/toolkit/docs/development/remote-cvs.html>³.

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

² http://dev.globus.org/wiki/Mailing_Lists

³ </toolkit/docs/development/remote-cvs.html>

2.3. Detailed installation instructions for these components

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

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

Chapter 5. Basic Security Configuration

1. Set environment variables

In order for the system to know the location of the Globus Toolkit commands you just installed, you must set an environment variable and source the `globus-user-env.sh` script.

1. As globus, set `GLOBUS_LOCATION` to where you installed the Globus Toolkit. This will be one of the following:

- Using Bourne shells:

```
globus$ export GLOBUS_LOCATION=/path/to/install
```

- Using csh:

```
globus$ setenv GLOBUS_LOCATION /path/to/install
```

2. Source `$GLOBUS_LOCATION/etc/globus-user-env.{sh,csh}` depending on your shell.

- Use `.sh` for Bourne shell:

```
globus$ . $GLOBUS_LOCATION/etc/globus-user-env.sh
```

- Use `.csh` for C shell.

```
globus$ source $GLOBUS_LOCATION/etc/globus-user-env.csh
```

2. Obtain host certificates

You must have X509 certificates to use the GT 5.0.2 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](#).

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:

2.1. 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](#)¹.

If you already have a CA, you will need to follow their configuration directions. If they include a CA setup package, follow the CAs 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.

2.2. 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.

2.3. Low-trust certificate

Globus offers a low-trust certificate available at <http://gcs.globus.org:8080/gcs>. This option should only be used as a last resort because it does not fulfill some of the duties of a real Certificate Authority.

This type of certificate is best suited for short term testing.

3. Add authorization

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:

```
bacon$ grid-cert-info -subject
/O=Grid/OU=GlobusTest/OU=simpleCA-mayed.mcs.anl.gov/OU=mcs.anl.gov/CN=Charles Bacon
bacon$ whoami
bacon
```

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=Charles Bacon" \
-ln bacon
```

¹ <http://www.tacar.org/>

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=Charles Bacon" bacon
```

Important

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

4. 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:

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

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

```
User Key File: /home/bacon/.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=Charles Bacon 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.

5. Firewall configuration

For information on configuring services in the presence of a firewall, see [the firewall PDF](#)².

² <http://www.globus.org/toolkit/security/firewalls/>

Chapter 6. Basic Setup for GT 5.0.2

The [Quickstart Guide](#) walks you through setting up basic services on multiple machines.

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

- `i18n`: Internationalization libraries
- `prewsgram`: GRAM5
- `gridftp`: GridFTP
- `prews`: GRAM5 and GridFTP
- `prews-test`: Tests for pre-webservices components
- `rls`: Replica Location Service

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. 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](#)¹.

3. 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.

¹ <http://gridpackagingtools.com/book/latest-stable/index.html>

Appendix B. Environmental Variables in GT 5.0.2

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-5.0.2/globus_xio/html/group_tcp_driver_envs.html
- http://www.globus.org/api/c-globus-5.0.2/globus_xio/html/group_file_driver_envs.html
- http://www.globus.org/api/c-globus-5.0.2/globus_xio/html/group_gsi_driver_envs.html
- http://www.globus.org/api/c-globus-5.0.2/globus_xio/html/group_udp_driver_envs.html

1.2. Environmental variables for C Common Libraries

- GLOBUS_ERROR_VERBOSE=1 can be set to enable verbose error messages.
- GLOBUS_ERROR_OUTPUT=1 can be set to enable output of all errors (including some that should be ignored).

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` and `/etc/grid-security/service/servicekey` exist and contain a valid certificate and key, those files are used.
3. Otherwise, if the files `$GLOBUS_LOCATION/etc/grid-security/service/servicecert` and `$GLOBUS_LOCATION/etc/grid-security/service/servicekey` exist and contain a valid certificate and key, those files are used.
4. Otherwise, if the files `service/servicecert` and `service/servicekey` 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

GLOBALBUS_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

GLOBALBUS_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 security considerations documentation. Name matching will rely on canonical host name associated with connection IP addresses.
STRICT_RFC2818	Support RFC 2818 ¹ 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](#)² 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:

- *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*

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

² <http://myproxy.ncsa.uiuc.edu/man/>

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 5.0.2 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. Run the setup script

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

Run the setup script:

```
$GLOBUS_LOCATION/setup/globus/setup-simple-ca
```

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:

```
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/home/globus/.globus/simpleCA//private/cakey.pem'
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:

A self-signed certificate has been generated
for the Certificate Authority with the subject:

```
/O=Grid/OU=GlobusTest/OU=simpleCA-mayed.mcs.anl.gov/CN=Globus Simple CA
```

If this is invalid, rerun this script

```
setup/globus/setup-simple-ca
```

and enter the appropriate fields.

The private key of the CA is stored in /home/globus/.globus/simpleCA//private/cakey.pem
The public CA certificate is stored in /home/globus/.globus/simpleCA//cacert.pem

The distribution package built for this CA is stored in

```
/home/globus/.globus/simpleCA//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.

Press any key to acknowledge this screen.

Your CA setup package finishes installing and ends the procedure with the following reminder:

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

```
/opt/gt4/setup/globus_simple_ca_68ea3306_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.

```
setup-ssl-utils: Complete
```

We'll run the setup-gsi script in the next section. For now, just notice that it refers to your `$GLOBUS_LOCATION` and the *CA Hash* from the last message.

2.6. Complete setup of GSI

To finish the setup of GSI, we'll run the script noted in the previous step.

Run the following as root (or, if no root privileges are available, add the **-nonroot** option to the command line):

```
$GLOBUS_LOCATION/setup/globus_simple_ca_CA_Hash_setup/setup-gsi -default
```

The output should look like:

```
setup-gsi: Configuring GSI security
Installing /etc/grid-security/certificates//grid-security.conf.CA_Hash...
Running grid-security-config...
Installing Globus CA certificate into trusted CA certificate directory...
Installing Globus CA signing policy into trusted CA certificate directory...
setup-gsi: Complete
```

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
```

6. Configure SimpleCA for multiple machines

So far, you have a single machine configured with SimpleCA certificates. Recall that in [Complete setup of GSI](#) a CA setup package was created in `.globus/simpleCA/globus_simple_ca_HASH_setup-0.17.tar.gz`. If you want to use your certificates on another machine, you must install that CA setup package on that machine.

To install it, copy that package to the second machine and run:

```
$GLOBUS_LOCATION/sbin/gpt-build globus_simple_ca_HASH_setup-0.17.tar.gz gcc32dbg
```

Then you will have to perform **setup-gsi -default** from [Sign the host certificate](#).

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 3, "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](#)
 - [Replica Location Service \(RLS\)](#)
- 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
 - [GSI C](#)
 - [MyProxy](#)
 - [GSI-OpenSSH](#)
- Data Management components
 - [GridFTP](#)
 - [Replica Location Service \(RLS\)](#)
- Execution Management components
 - [GRAM5](#)

Appendix F. Security Considerations in GT 5.0.2

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](#)¹ has been opened to see if this feature needs to be modified.

In GT 5.0.2, 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.

¹ 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/)² and includes full OpenSSH functionality. For more information on OpenSSH security, see the [OpenSSH Security](http://www.openssh.org/security.html)³ 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 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 outbound ports of the GridFTP server to this range. 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.

If the GridFTP *client* is behind a firewall:

² <http://www.openssh.org/>

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

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 outbound ports of the GridFTP client to this range. 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](#)⁴.

3.2. Replica Location Service (RLS) Security Considerations

Security recommendations include:

- *Dedicated User Account:* It is recommended that users create a dedicated user account for installing and running the RLS service (e.g., `globus` as recommended in the general GT installation instructions). This account may be used to install and run other services from the Globus Toolkit.
- *Key and Certificate:* It is recommended that users do not use their `hostkey` and `hostcert` for use by the RLS service. Create a `containerkey` and `containercert` with permissions 400 and 644 respectively and owned by the `globus` user. Change the `rlskeyfile` and `rlscertfile` settings in the RLS configuration file (`$GLOBUS_LOCATION/etc/globus-rls-server.conf`) to reflect the appropriate filenames.
- *LRC and RLI Databases:* Users must ensure security of the RLS data as maintained by their chosen database management system. Appropriate precautions should be made to protect the data and access to the database. Such precautions may include creating a user account specifically for RLS usage, encrypting database users' passwords, etc.
- *RLS Configuration:* It is recommended that the RLS configuration file (`$GLOBUS_LOCATION/etc/globus-rls-server.conf`) be owned by and accessible only by the dedicated user account for RLS (e.g., `globus` account per above recommendations). The file contains the database user account and password used to access the LRC and RLI databases along with important settings which, if tampered with, could adversely affect the RLS service.

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

4. Execution Management

4.1. Security Considerations

No special security considerations exist at this time.

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](#)¹

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](#)² 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](#)³ on the collection of usage statistics.

1.2. RLS-specific usage statistics

The following usage statistics are sent by RLS Server by default in a UDP packet:

- Component identifier

¹ [../Usage_Stats.html](#)

² [/toolkit/docs/5.0/5.0.2/Usage_Stats.html](#)

³ [/toolkit/docs/latest-stable/Usage_Stats.html](#)

- Usage data format identifier
- Time stamp
- Source IP address
- Source hostname (to differentiate between hosts with identical private IP addresses)
- Version number
- Uptime
- *LRC* service indicator
- *RLI* service indicator
- Number of *LFNs*
- Number of *PFNs*
- Number of Mappings
- Number of RLI LFNs
- Number of RLI LRCs
- Number of RLI Senders
- Number of RLI Mappings
- Number of threads
- Number of connections

The RLS sends the usage statistics at server startup, server shutdown, and once every 24 hours when the service is running.

If you wish to disable this feature, you can set the following environment variable before running the RLS:

```
export GLOBUS_USAGE_OPTOUT=1
```

By default, these usage statistics UDP packets are sent to `usage-stats.globus.org:4180` but can be redirected to another host/port or multiple host/ports with the following environment variable:

```
export GLOBUS_USAGE_TARGETS="myhost.mydomain:12345 myhost2.mydomain:54321"
```

You can also dump the usage stats packets to `stderr` as they are sent (although most of the content is non-ascii). Use the following environment variable for that:

```
export GLOBUS_USAGE_DEBUG=MESSAGES
```

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

⁴ /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^{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](#)⁵ on the collection of usage statistics.

⁵ /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 `/etc/grid-security/certificates/<hash>.0`, where `<hash>` is the hash code of the CA identity.
- 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 `/etc/grid-security/gsi-gaa.conf`.
- 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 `/etc/grid-security/grid-mapfile`. For more information see the Gridmap section [here](#).
- grid security directory The directory containing GSI configuration files such as the GSI authorization callout configuration and GAA configuration files. Typically this directory is `/etc/grid-security`. For more information see [this](#).
- 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 `/etc/grid-security/gsi-authz.conf`.

H

- host certificate An EEC belonging to a host. When using GSI this certificate is typically stored in `/etc/grid-security/hostcert.pem`. For more information on possible host certificate locations see the [GSI C Developer's Guide](#).
- host credentials The combination of a host certificate and its corresponding private key.

L

- Local Replica Catalog (LRC) Stores mappings between logical names for data items and the target names (often the physical locations) of replicas of those items. Clients query the LRC to discover replicas associated with a logical name. Also may associate attributes with logical or target names. Each LRC periodically sends information about its logical name mappings to one or more RLIs.

See also [RLI](#)³.

logical file name A unique identifier for the contents of a file.

P

physical file name The address or the location of a copy of a file on a storage system.

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 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.

R

Replica Location Index (RLI) Collects information about the logical name mappings stored in one or more Local Replica Catalogs (LRCs) and answers queries about those mappings. Each RLI periodically receives updates from one or more LRCs that summarize their contents.

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 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.

³ #rli