GT 5.0.2 User's Guide

Abstract

You can download the PDF version here. This page contains information for commonly performed tasks using GT components. This assumes a default installation and covers the more basic tasks using common tools. Due to size, all GT command line clients are listed here.

Note that GT itself is typically used as middleware and not necessarily intended to be used directly by end-users. Instead, grid developers tend to use GT to develop higher-level services and systems that are then used by end-users (where GT is essentially the plumbing). However, GT Release Manuals include User's Guides for each established component that describe how the public interfaces are intended to be used - whether it is by a human or a program.
# Table of Contents

1. Setting up your environment ............................................................................................................... 1  
2. Security .......................................................................................................................................... 2  
   1. Obtaining certificates ................................................................................................................. 2  
   2. Authenticating (who are you?) ................................................................................................. 2  
   3. Authorizing (what are you allowed to do?) ............................................................................... 2  
   4. Basic procedure for using GSI C ............................................................................................. 2  
   5. Troubleshooting Certificates and GridMap Files ........................................................................ 3  
3. Data Management ............................................................................................................................ 5  
   1. File transfers with GridFTP .................................................................................................... 5  
   2. Mapping replicas with Replica Location Service (RLS) .......................................................... 7  
4. Submitting jobs to a job scheduler ...................................................................................................... 10  
   1. Preparing to use GRAM .......................................................................................................... 10  
   2. Delegating credentials .......................................................................................................... 10  
   3. Submitting jobs .................................................................................................................... 11  
A. Globus Toolkit 5.0.2 Public Interface Guides ....................................................................................... 23  
B. Globus Toolkit 5.0.2 Errors .............................................................................................................. 24  
Glossary ........................................................................................................................................... 44
## List of Tables

B.1. XIO Errors ................................................................................................................................. 25  
B.2. Credential Errors ......................................................................................................................... 27  
B.3. Gridmap Errors ........................................................................................................................... 28  
B.4. MyProxy Errors .......................................................................................................................... 29  
B.5. GSI-OpenSSH Errors ................................................................................................................... 31  
B.6. GridFTP Errors ........................................................................................................................... 32  
B.7. Replica Locator Service (RLS) Errors .......................................................................................... 34  
B.8. GRAM5 Errors .............................................................................................................................. 35
List of Examples

4.1. Generating a proxy with grid-proxy-init ............................................................... 10
4.2. Gatekeeper Service Contact Examples ................................................................. 12
4.3. Minimal job using globus-job-run ................................................................. 12
4.4. Multiprocess job using globus-job-run ................................................................. 13
4.5. Canceling an interactive job .............................................................................. 13
4.6. Setting job environment variables with globus-job-run ................................ 13
4.7. Using custom RSL clauses with globus-job-run ............................................... 13
4.8. Constructing RSL strings with globus-job-run .................................................. 14
4.9. globus-job-submit ............................................................................................ 15
4.10. Checking RSL Syntax ...................................................................................... 16
4.11. GRAM Authentication test ............................................................................. 16
4.12. GRAM version check ...................................................................................... 17
4.13. Basic Interactive Job ....................................................................................... 17
4.14. Basic Batch Job .............................................................................................. 17
4.15. Refreshing a Credential ................................................................................... 18
4.16. Proxy Expiration Example ............................................................................... 18
4.17. File stage in ..................................................................................................... 19
4.18. File stage in shared ......................................................................................... 19
4.19. File stage out .................................................................................................. 19
4.20. Staging to scratch directory ............................................................................. 20
4.21. Cleaning up a file ............................................................................................ 20
4.22. Two phase commit example ........................................................................... 21
4.23. Restart example .............................................................................................. 21
4.24. Java example .................................................................................................. 22
Chapter 1. Setting up your environment

This step is usually a prerequisite for using GT commands. Make sure you have set GLOBUS_LOCATION to the location of your Toolkit installation. There are two environment scripts called $GLOBUS_LOCATION/etc/globus-user-env.sh and $GLOBUS_LOCATION/etc/globus-user-env.csh. You should read in whichever one corresponds to the type of shell you are using.

For example, in csh or tcsh, you would run:

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

In sh, bash, ksh, or zsh, you would run:

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

Set Globus location:

$ export GLOBUS_LOCATION='/opt/globus/apps/globus-5.0.2'

Source it:

source $GLOBUS_LOCATION/etc/globus-user-env.sh
  source $GLOBUS_LOCATION/etc/globus-devel-env.sh
Chapter 2. Security

This chapter provides information about basic security tasks in GT 5.0.2.

1. Obtaining certificates

Security is at the heart of Globus, and unless you are running without security (only recommended for testing), you will not be able to use most of Globus unless you have obtained a certificate for yourself. (Note that you may use GridFTP without certificates if you are only using ftp:// or http:// protocols.)

For basic information about obtaining certificates, see Obtaining host certificates in the Installation Guide.

⚠️ Important

Remember to keep track of when your certificates expire. If your certificates expire, you may not be able to use your services until they are refreshed.

2. Authenticating (who are you?)

2.1. Generate a valid proxy certificate

Before using many of the tools in GT, a user must generate a valid user proxy. Use grid-proxy-init. The following is an example:

```
% $GLOBUS_LOCATION/bin/grid-proxy-init
Your identity: /O=Grid/OU=GlobusTest/OU=simpleCA.mymachine/OU=mymachine/CN=John Doe
Enter GRID pass phrase for this identity:
Creating proxy ................................. Done
Your proxy is valid until: Tue Oct 26 01:33:42 2004
```

3. Authorizing (what are you allowed to do?)

Basic authorization in GT is enforced via a grid map file, a file that contains mappings of certificate subject names to local user names, like the following:

```
"/O=Grid/O=Gridbus/OU=your.domain.edu/CN=Your Name" youruser
```

For more information about gridmaps see Section 3, “Add authorization”, Section 4, “Configuring Credential Mappings” and Globus Toolkit Gridmap Processing\(^1\).

4. Basic procedure for using GSI C

In most cases, an individual will do the following:

- Acquire a user certificate from a certification authority (CA) with grid-cert-request. This certificate will typically be valid for a year or more and will be stored in a file in the individual's home directory.

\(^1\) http://dev.globus.org/wiki/Gridmap
It is important to keep in mind when your cert will expire - after your user certificate expires, you may not be able to use secure services in GT!

- Use the end-user certificate to create a proxy certificate using grid-proxy-init. This will be used to authenticate the individual to grid services. Proxy certificates typically have a much shorter lifetime than end-user certificates (usually 12 hours). Once your proxy certificate expires, simply rerun grid-proxy-init.

5. Troubleshooting Certificates and GridMap Files

For common errors, see Certificates and Gridmap errors.

5.1. Some tools to validate certificate setup

5.1.1. grid-cert-diagnostics

The grid-cert-diagnostics program checks prints diagnostics about the user's certificates, and host security environment.

% grid-cert-diagnostics -p

5.1.2. Check that the user certificate is valid

openssl verify -CApath /etc/grid-security/certificates
  --purpose sslclient ~/.globus/usercert.pem

5.1.3. Connect to the server using s_client

openssl s_client -ssl3 -cert ~/.globus/usercert.pem -key ~/.globus/userkey.pem -CApath /etc/grid-security/certificates
  --connect <host:port>

Here <host:port> denotes the server and port you connect to.

If it prints an error and puts you back at the command prompt, then it typically means that the server has closed the connection, i.e. that the server was not happy with the client's certificate and verification. Check the SSL log on the server.

If the command "hangs" then it has actually opened a telnet style (but secure) socket, and you can "talk" to the server.

You should be able to scroll up and see the subject names of the server's verification chain:

depth=2 /DC=net/DC=ES/O=ESnet/OU=Certificate Authorities/CN=ESnet Root CA 1
verify return:1
depth=1 /DC=org/DC=DOEGrids/OU=Certificate Authorities/CN=DOEGrids CA 1
verify return:1
depth=0 /DC=org/DC=doegrids/OU=Services/CN=wiggum.mcs.anl.gov
verify return:1

In this case, there were no errors. Errors would give you an extra line next to the subject name of the certificate that caused the error.
5.1.4. Check that the server certificate is valid

Requires root login on server:

    openssl verify -CApath /etc/grid-security/certificates -purpose sslserver /etc/grid-security/hostcert.pem
Chapter 3. Data Management

1. File transfers with GridFTP

1.1. Basic procedure for using GridFTP (globus-url-copy)

If you just want the "rules of thumb" on getting started (without all the details), the following options using globus-url-copy will normally give acceptable performance:

For a single file transfer:

globus-url-copy -vb -tcp-bs 1048576 -p 4 source_url destination_url

where:

- vb specifies verbose mode and displays:
  • number of bytes transferred,
  • performance since the last update (currently every 5 seconds), and
  • average performance for the whole transfer.

- tcp-bs specifies the size (in bytes) of the TCP buffer to be used by the underlying ftp data channels. This is critical to good performance over the WAN.
  How do I pick a value?

- p Specifies the number of parallel data connections that should be used. This is one of the most commonly used options.
  How do I pick a value?

For a directory transfer:

globus-url-copy -vb -tcp-bs 1048576 -p 4 -r -cd - cc 4 source_url destination_url

where:

- vb specifies verbose mode and displays:
  • number of bytes transferred,
  • performance since the last update (currently every 5 seconds), and
  • average performance for the whole transfer.

- tcp-bs specifies the size (in bytes) of the TCP buffer to be used by the underlying ftp data channels. This is critical to good performance over the WAN.
  How do I pick a value?

- p Specifies the number of parallel data connections that should be used. This is one of the most commonly used options.
Data Management

How do I pick a value?

-cc Specifies the number of concurrent FTP connections to use for multiple transfers.

-cc Creates destination directories, if needed.

-r Copies files in subdirectories.

The source/destination URLs will normally be one of the following:

- `file:///path/to/my/file` if you are accessing a file on a file system accessible by the host on which you are running your client.

- `gsiftp://hostname/path/to/remote/file` if you are accessing a file from a GridFTP server.

1.1.1. Putting files

One of the most basic tasks in GridFTP is to "put" files, i.e., moving a file from your file system to the server. So for example, if you want to move the file `/tmp/foo` from a file system accessible to the host on which you are running your client to a file name `/tmp/bar` on a host named `remote.machine.my.edu` running a GridFTP server, you would use this command:

```
globus-url-copy -vb -tcp-bs 2097152 -p 4 file:///tmp/foo gsiftp://remote.machine.my.edu/tmp/bar
```

Note

In theory, `remote.machine.my.edu` could be the same host as the one on which you are running your client, but that is normally only done in testing situations.

1.1.2. Getting files

A get, i.e., moving a file from a server to your file system, would just reverse the source and destination URLs:

```
globus-url-copy -vb -tcp-bs 2097152 -p 4 gsiftp://remote.machine.my.edu/tmp/bar file:///tmp/foo
```

1.1.3. Third party transfers

Finally, if you want to move a file between two GridFTP servers (a third party transfer), both URLs would use `gsiftp:` as the protocol:

```
globus-url-copy -vb -tcp-bs 2097152 -p 4 gsiftp://other.machine.my.edu/tmp/foo gsiftp://remote.machine.my.edu/tmp/bar
```

1.1.4. For more information

If you want more information and details on URLs and the command line options, the Key Concepts gives basic definitions and an overview of the GridFTP protocol as well as our implementation of it.
2. Mapping replicas with Replica Location Service (RLS)

2.1. Ping the server

To check whether your server is active you may use the `globus-rls-admin(1)` `ping` command.

```
% $GLOBUS_LOCATION/sbin/globus-rls-admin -p rls://localhost
ping rls://localhost: 0 seconds
```

2.2. Creating replica location mappings

When the RLS server is first installed its database of replica location information will be empty, as expected. To create a replica location mapping, use the `globus-rls-cli(1)` `create` command. Replica information in RLS is represented as mappings from logical names to target names. Typically, the logical name will be a unique identifier for a given replicated data set and the target name will be a URL identifying a particular replica of the data set.

```
% $GLOBUS_LOCATION/bin/globus-rls-cli create my-logical-name-1 url-for-target-name-1 rls://localhost
```

Note

The create command is intended for creating the initial replica mapping entry for a given logical name. If the user attempts to create another entry using an existing logical name, RLS will report a user error. To map additional target names to an existing logical name, see Section 4, “Adding replica location mappings”.

2.3. Adding replica location mappings

To map additional target names to a logical name created by the previously described create command, use the `globus-rls-cli(1)` `add` command.

```
% $GLOBUS_LOCATION/bin/globus-rls-cli add my-logical-name-1 url-for-target-name-2 rls://localhost
```

2.4. Querying replica location mappings

Once your RLS server is populated with replica location mappings, you can query the server for useful information using the `globus-rls-cli(1)` `query` command.

```
% $GLOBUS_LOCATION/bin/globus-rls-cli query lrc lfn my-logical-name-1 rls://localhost
my-logical-name-1: url-for-target-name-1
my-logical-name-1: url-for-target-name-2
```
2.5. Deleting replica location mappings

To remove unwanted replica location mappings from your RLS server, use the `globus-rls-cli(1) delete` command. The delete operation works directly on the mapping and indirectly on the logical and target names. When the delete operation is performed by the RLS server the association between the specified logical name and the specified target name is eliminated. However, there may still be other target names associated with the logical name, and there could still be other logical names associated with the target name, though the latter scenario is less likely. Only when all mapping associations for a given logical name (or a given target name) are eliminated (i.e., the specified logical name has no target names associated with it) will the logical (or target) name be deleted from the RLS server.

% $GLOBUS_LOCATION/bin/globus-rls-cli delete my-logical-name-1 url-for-target-name-1 rls://localhost
% $GLOBUS_LOCATION/bin/globus-rls-cli query lrc lfn my-logical-name-1 rls://localhost
my-logical-name-1: url-for-target-name-2
% $GLOBUS_LOCATION/bin/globus-rls-cli delete my-logical-name-1 url-for-target-name-2 rls://localhost
globus_rls_client: LFN doesn't exist: my-logical-name-1

2.6. Using bulk operations

The `globus-rls-cli(1)` supports a variety of bulk operations that enhance productivity for users and reduce network connection overhead from making multiple, separate invocations of the client. The general pattern for bulk operation support as implemented by the client is a parameter list consisting of `bulk command-name [command-modifiers] param-1 param-2 param-N`, such as `bulk query lrc lfn my-logical-name-1 my-logical-name-2 my-logical-name-3`.

% $GLOBUS_LOCATION/bin/globus-rls-cli bulk create my-logical-name-1 url-for-target-name-1-1 my-logical-name-2 url-for-target-name-2-1 rls://localhost
% $GLOBUS_LOCATION/bin/globus-rls-cli bulk add my-logical-name-1 url-for-target-name-1-2 my-logical-name-2 url-for-target-name-2-2 my-logical-name-3: LFN doesn't exist
my-logical-name-2: url-for-target-name-2-1
my-logical-name-2: url-for-target-name-2-2
my-logical-name-1: url-for-target-name-1-1
my-logical-name-1: url-for-target-name-1-2

2.7. Using interactive mode

The `globus-rls-cli(1)` supports an interactive mode in addition to the general command-line mode. To enter the interactive mode, simply invoke the client without any command.

% $GLOBUS_LOCATION/bin/globus-rls-cli rls://localhost
rls> query lrc lfn my-logical-name-2
my-logical-name-2: url-for-target-name-2-1
my-logical-name-2: url-for-target-name-2-2
rls> query lrc lfn my-logical-name-1
my-logical-name-1: url-for-target-name-1-1
my-logical-name-1: url-for-target-name-1-2
rls> bulk delete my-logical-name-1 url-for-target-name-1-1 my-logical-name-1 url-for-target-name-1-2 my-logical-name-2 url-for-target-name-2-1
my-logical-name-2 url-for-target-name-2-2
rls> bulk query lrc lfn my-logical-name-2 my-logical-name-1
my-logical-name-1: LFN doesn't exist
my-logical-name-2: LFN doesn't exist
rls> exit
Chapter 4. Submitting jobs to a job scheduler

1. Preparing to use GRAM

The first step to being able to use GRAM5 after installation is to acquire a temporary Grid credential to use to authenticate with the GRAM5 service and any file services your job requires. Normally this is done via either grid-proxy-init or via the MyProxy service.

1.1. Proxy credentials with grid-proxy-init

To generate a proxy credential using the grid-proxy-init program, execute the command with no arguments. By default, it will generate an impersonation proxy with a lifetime of 12 hours.

Example 4.1. Generating a proxy with grid-proxy-init

This example creates a 12 hour impersonation proxy to use to authenticate with grid services such as GRAM5:

% bin/grid-proxy-init
Your identity: /O=Grid/OU=Example/CN=Joe User
Enter GRID pass phrase for this identity:
Creating proxy ........................................ Done
Your proxy is valid until: Tue Oct 26 01:33:42 2010

⚠️ Important

In order to generate a proxy credential, you must have first been issued an identity credential by some certificate authority that is trusted by the GRAM5 resource you want to use. To learn more about certificates and Grid security in general, please read Security Key Concepts.

2. Delegating credentials

The credential created in the previous section is used to authenticate with the GRAM5 service as well as to delegate a limited proxy of that credential to the service so that it can process the job. This credential delegation occurs when the globus-gatekeeper service is first contacted when a job is to be submitted. By default, the tools provided with GT 5.0.2 delegate a limited proxy. This limited proxy can be used to authenticate with other services on the client's behalf, but with the services knowing that the proxy is not under direct control by the user.

2.1. Delegated Credential Usage

The delegated proxy can be used by the GRAM5 service and the job in a few different ways:

1. The GRAM5 service uses the credential to send job state notification messages to clients which have registered to receive them.

2. The GRAM5 service uses the credential to contact GASS and GridFTP file servers to stage files to and from the execution resource

3. The job executed by the GRAM5 service can use the delegated credential for application-specific purposes.
Note

In GRAM5, the Job Manager may manage multiple jobs simultaneously. It will use the delegated proxy with the most time left for authentication. Individual GRAM5 jobs will have separate proxies.

globusrun, globus-job-run, and globus-job-submit commands delegate credentials automatically when submitting a job. Additionally, globusrun can refresh the credentials used by the job and job manager, after the job manager is started.

3. Submitting jobs

This section describes the steps needed to submit jobs to resources managed by GRAM5 services. It describes how resources are named, tools for submitting and monitoring jobs, and the RSL language which describes requirements for jobs.

3.1. Resource Names

In GRAM5, a Gatekeeper Service Contact contains the host, port, service name, and service identity required to contact a particular GRAM service. For convenience, default values are used when parts of the contact are omitted. An example of a full gatekeeper service contact is grid.example.org:2119/jobmanager:/C=US/O=Example/OU=Grid/CN=host/grid.example.org.

The various forms of the resource name using default values follow:

- HOST
- HOST:PORT
- HOST:PORT/SERVICE
- HOST/SERVICE
- HOST:/SERVICE
- HOST:PORT:SUBJECT
- HOST/SERVICE:SUBJECT
- HOST:/SERVICE:SUBJECT
- HOST:PORT/SERVICE:SUBJECT

Where the various values have the following meaning:

HOST Network name of the machine hosting the service.
PORT Network port number that the service is listening on. If not specified, the default of 2119 is used.
SERVICE Path of the service entry in $GLOBUS_LOCATION/etc/grid-services. If not specified, the default of jobmanager is used.
SUBJECT X.509 identity of the credential used by the service. If not specified, the default of host@HOST is used.
Example 4.2. Gatekeeper Service Contact Examples

The following strings all name the service grid.example.org:2119/jobmanager:/C=US/O=Example/OU=Grid/CN=host/grid.example.org using the formats with the various defaults described above.

- grid.example.org
- grid.example.org:2119
- grid.example.org:2119/jobmanager
- grid.example.org/jobmanager
- grid.example.org:/jobmanager
- grid.example.org:2119:/C=US/O=Example/OU=Grid/CN=host/grid.example.org
- grid.example.org/jobmanager:/C=US/O=Example/OU=Grid/CN=host/grid.example.org
- grid.example.org:/jobmanager:/C=US/O=Example/OU=Grid/CN=host/grid.example.org
- grid.example.org:2119/jobmanager:/C=US/O=Example/OU=Grid/CN=host/grid.example.org

3.2. Running Jobs with globus-job-run

The `globus-job-run` provides a simple blocking command-line interface to the GRAM service. The `globus-job-run` program submits a job to a GRAM5 resource and waits for the job to terminate. After the job terminates, the output and error streams of the job are sent to the output and error streams of `globus-job-run` as if the job were run interactively. Note that input to the job must be located in a file prior to running the job; true interactive I/O is not supported by GRAM5.

The `globus-job-run` program has command-line options to control most aspects of jobs run by GRAM5. However, certain behaviors must be specified by definition of an RSL string containing various job attributes. A more detailed description about the RSL language is included on the section on running jobs with `globusrun` below.

The following examples show some of the common command-line options to `globus-job-run`. Full `globus-job-run` documentation is available in the GRAM5 public interface guide.

Example 4.3. Minimal job using globus-job-run

The following command line submits a single instance of the `/bin/hostname` executable to the resource named by grid.example.org:2119/jobmanager-pbs.

```
% globus-job-run grid.example.org:2119/jobmanager-pbs /bin/hostname
node1.grid.example.org
```
Example 4.4. Multiprocess job using globus-job-run

The following command line submits ten instances of an executable `a.out`, staging it from the client host to the service node using GASS. The `a.out` program prints the name of the host it is executing on.

```
% globus-job-run grid.example.org:2119/jobmanager-pbs -np 10 -s a.out
node1.grid.example.org
node3.grid.example.org
node2.grid.example.org
node5.grid.example.org
node4.grid.example.org
node8.grid.example.org
node6.grid.example.org
node9.grid.example.org
node7.grid.example.org
node10.grid.example.org
```

Example 4.5. Canceling an interactive job

This example shows how using the Control+C (or other system-specific mechanism for sending the SIGINT signal) can be used to cancel a GRAM job.

```
% globus-job-run grid.example.org:2119/jobmanager-pbs /bin/sleep 90
Control-C
GRAM Job failed because the user cancelled the job (error code 8)
```

Example 4.6. Setting job environment variables with globus-job-run

The following command line submits one instances of the executable `/usr/bin/env`, setting some environment variables in the job environment beyond those set by GRAM5.

```
% globus-job-run grid.example.org:2119/jobmanager-pbs -env TEST=1 -env GRID=1 /usr/bin/env
HOME=/home/juser
LOGNAME=juser
GLOBUS_GRAM_JOB_CONTACT=https://client.example.org:3882/16001579536700793196/52956129777485997184/GLOBUS_LOCATION=/opt/globus-5.0.2
GLOBUS_GASS_CACHE_DEFAULT=/home/juser/.globus/.gass_cache
TEST=1
X509_USER_PROXY=/home/juser/.globus/job/mactop.local/16001579536700793196.52956129777485997184/x509_user_proxy
GRID=1
```

Example 4.7. Using custom RSL clauses with globus-job-run

The following command line submits an mpi job using globus-job-run, setting the `jobtype` RSL attribute to mpi. Any RSL attribute understood by the LRM can be added to a job via this method.

```
% globus-job-run grid.example.org:2119/jobmanager-pbs -np 5 -x '&(jobtype=mpi)' a.out
Hello, MPI (rank: 0, count: 5)
Hello, MPI (rank: 3, count: 5)
Hello, MPI (rank: 1, count: 5)
Hello, MPI (rank: 4, count: 5)
Hello, MPI (rank: 2, count: 5)
```
Example 4.8. Constructing RSL strings with globus-job-run

The globus-job-run program can also generate the RSL language description of a job based on the command-line options given to it. This example combines some of the features above and prints out the resulting RSL. This RSL string can be passed to tools such as globusrun to be run later.

```bash
% globus-job-run -dump rsl grid.example.org:2119/jobmanager-pbs -np 5 -x '&(jobtype=mpi)' -
  & (jobtype=mpi)
  (executable="a.out")
  (environment= ("GRID" "1") ("TEST" "1"))
  (count=5)
```

3.3. Submitting Jobs with globus-job-submit

A related tool to globus-job-run is globus-job-submit. This command submits a job to a GRAM5 service then exits without waiting for the job to terminate. Other tools (globus-job-cancel, globus-job-clean, and globus-job-get-output) allow further interaction with the job.

⚠️ Important

When using globus-job-submit, the job output and state will remain on disk on the GRAM resource until one of globus-job-clean or globus-job-cancel is run for that job. Be sure to clean up your jobs!

The globus-job-submit program has most of the same command-line options as globus-job-run. When run, instead of displaying the output and error streams of the job, it prints the job contact, which is used with the other globus-job tools to interact with the job.
Example 4.9. globus-job-submit

This example shows the interaction of submitting a job via `globus-job-submit`, checking its status with `globus-job-status`, getting its output with `globus-job-get-output`, and then cleaning the job with `globus-job-clean`.

```
% globus-job-submit grid.example.org:2119/jobmanager-pbs /bin/hostname
https://grid.example.org:38843/16001600430615223386/5295612977486013582/
% globus-job-status https://grid.example.org:38843/16001600430615223386/5295612977486013582/
PENDING
% globus-job-status https://grid.example.org:38843/16001600430615223386/5295612977486013582/
ACTIVE
% globus-job-status https://grid.example.org:38843/16001600430615223386/5295612977486013582/
DONE
% globus-job-get-output -r grid.example.org:2119/jobmanager-fork \
    https://grid.example.org:38843/16001600430615223386/5295612977486013582/
    node1.grid.example.org
% globus-job-clean -r grid.example.org:2119/jobmanager-fork \
    https://grid.example.org:38843/16001600430615223386/5295612977486013582/

WARNING: Cleaning a job means:
- Kill the job if it still running, and
- Remove the cached output on the remote resource

Are you sure you want to cleanup the job now (Y/N) ?
y
Cleanup successful.
```

3.4. Using the globusrun tool

The `globusrun` tool provides a more flexible tool for submitting, monitoring, and canceling jobs. With this tool, most of the functionality of the GRAM5 APIs are made available.

One major difference between `globusrun` and the other tools described above is that `globusrun` uses the RSL language to provide the job description, instead of multiple command-line options to describe the various aspects of the job. The section on `globus-job-run` contained a brief example RSL in the `-dumprsl` example above.

The following sections show examples of the different modes that `globusrun` can run in. Full information about `globusrun` command-line options is available in the public interface guide.

3.4.1. Checking RSL Syntax

This example shows how to check that an RSL document contains a syntactically correct job description. Note that this mode does not do semantic validation of the RSL, so an RSL document that passes this test may not work when submitted to a GRAM5 service.
Example 4.10. Checking RSL Syntax

% globusrun -p "&(executable=a.out)"
RSL Parsed Successfully...

% globusrun -p "&/executable=a.out)"
ERROR: cannot parse RSL &/executable=/adfadf/adf /adf /adf)
Syntax: globusrun [-help] [-f RSL file] [-s][-b][-d][...] [-r RM] [RSL]

Use -help to display full usage

3.4.2. Checking Service Contacts

This example shows how to check that a globus-gatekeeper is running at a particular contact and that the client and service have mutually-trusted credentials.

Example 4.11. GRAM Authentication test

% globusrun -a -r grid.example.org:2119/jobmanager-pbs
GRAM Authentication test successful
% globusrun -a -r grid.example.org:2119/jobmanager-1sf
GRAM Authentication test failure: the gatekeeper failed to find the requested service
% globusrun -a -r grid.example.org:2119/jobmanager-pbs:host@not.example.org
GRAM Authentication test failure: an authorization operation failed
globus_xio_gsi: gss_init_sec_context failed.
GSS Major Status: Unexpected Gatekeeper or Service Name
globus_gsi_gssapi: Authorization denied: The name of the remote host
(host@not.example.org), and the expected name for the remote host
(grid.example.org) do not match. This happens when the name in the host
certificate does not match the information obtained from DNS and is often a DNS
configuration problem.

Note

The DNS configuration problem was a common issue in GRAM2, but GRAM5 will not depend on DNS to
resolve names for mutual authentication.

3.4.3. Checking GRAM service version

This example shows how to determine what software version of GRAM5 is deployed at a particular service contact.
Example 4.12. GRAM version check

% globusrun -j -r grid.example.org:2119/jobmanager-pbs:host@not.example.org
Toolkit version: 4.3.0-HEAD
Job Manager version: 10.5 (1256257907-0)

Note
This example shows the version number for an unreleased development version of GRAM5. The actual numbers returned will be different.

Note
This feature is new in GRAM5. When contacting a GRAM2 service, globusrun will display the following error message:

GRAM version check failed : an incoming HTTP message did not contain the expected info

3.4.4. Basic Interactive job with globusrun

This example shows how to submit interactive job with globusrun. When the --s is used, the output of the job command is returned to the client and displayed as if the command ran locally. This is similar to the behavior of the globus-job-run program described above.

Example 4.13. Basic Interactive Job

% globusrun --s --r example.grid.org/jobmanager-pbs "& (executable=/bin/hostname) (count=5)"
node03.grid.example.org
node01.grid.example.org
node02.grid.example.org
node05.grid.example.org
node04.grid.example.org

3.4.5. Basic batch job with globusrun

This example shows how to submit, monitor, and cancel a batch job using globusrun. This method is useful for the case where the job may run for a long time, the job may be queued for a long time, or when there are network reliability issues between the client and service.

Example 4.14. Basic Batch Job

% globusrun -b --r grid.example.org:2119/jobmanager-pbs "& (executable=/bin/sleep) (arguments=500)"
globus_gram_client_callback_allow successful
GRAM Job submission successful
https://grid.example.org:38824/16001608125017717261/5295612977486019989/
GLOBUS_GRAM_PROTOCOL_JOB_STATE_PENDING
% globusrun -status https://grid.example.org:38824/16001608125017717261/5295612977486019989/
PENDING
% globusrun -k https://grid.example.org:38824/16001608125017717261/5295612977486019989/
3.4.6. Refreshing a GRAM5 Credential

The following example shows how to refresh the credential used by a job manager and a job.

Example 4.15. Refreshing a Credential

```bash
% globusrun -refresh-proxy https://grid.example.org:38824/16001608125017717261/5295612977486019989/
% echo $?
0
```

Note

In GT 5.0.2, globusrun does not print any diagnostics when given the -refresh-proxy command-line option. Therefore, check the exit code as above to ensure that the refresh is successful.

3.4.7. Dealing with credential expiration

When the Job Manager’s credential is about to expire, it sends a message to all clients registered for GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED notifications that the job manager is terminating and that the job will continue to run without the job manager.

Any client which receives such a message can (if necessary) generate a new proxy as described above and then submit a restart request to start a job manager with a new credential. This job manager will resume monitoring the jobs which were started prior to proxy expiration.

In this example, the globusrun displays an error message when the job manager's proxy is about to expire. The user creates a new proxy and resumes monitoring the job with globusrun.

Example 4.16. Proxy Expiration Example

```bash
% globusrun -r grid.example.org "&(executable=a.out)"
globus_gram_client_callback_allow successful
GRAM Job submission successful
GLOBUS_GRAM_PROTOCOL_JOB_STATE_ACTIVE
GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED
GRAM Job failed because the user proxy expired (job is still running) (error code 131)
% grid-proxy-init
Your identity: /DC=org/DC=example/OU=grid/CN=Joe User
Enter GRID pass phrase for this identity:
Creating proxy .................................................................
Your proxy is valid until: Tue Nov 10 04:25:03 2009
% globusrun -r grid.example.org "&(restart="https://grid.example.org:1997/16001700477575114131/5295612977486005428/"
gram_job_callback_allow successful
GRAM Job submission successful
GLOBUS_GRAM_PROTOCOL_JOB_STATE_ACTIVE
GLOBUS_GRAM_PROTOCOL_JOB_STATE_DONE
```

3.4.8. File staging

In addition to the standard output and error stream output done by globusrun, GRAM5 can do basic file management tasks to stage files to the GRAM5 service node before submitting a job and to stage files from the GRAM5 service node to a file service after the job completes.
GRAM5 file staging supports four URL schemes: ftp, gsiftp, http, and https. Note, that for the https scheme, GRAM expects the file server to be running with the same identity as the client.

General file staging is controlled by three RSL attributes: file_stage_in, file_stage_in_shared, and file_stage_out. In addition, the files named by the RSL attributes executable, stdin may be staged in and the files named by the RSL attributes stdout and stderr may be staged out.

The file_stage_in_shared RSL attribute instructs GRAM to store a local copy of the resource named by the URL in the GASS cache. This is useful if multiple concurrent jobs will be accessing one or more common files. The GASS cache will manage a reference count for files in the cache and remove them when all jobs that refer to them complete.

The following example shows how to stage a few files from a GridFTP server to the GRAM node. It uses the rsl_substitution mechanism to define a subsitution variable to reduce the amount of redundancy in the job description.

Example 4.17. File stage in

```
% globusrun -s -r grid.example.org:2119/jobmanager-pbs \
  "&(rsl_substitution = (GRIDFTP_SERVER gsiftp://gridftp.example.org)) \
  (executable=/bin/ls) \
  (arguments=/tmp/staged_file) \
  (file_stage_in = ($(GRIDFTP_SERVER)/staged_file /tmp/staged_file))"
```

tmp/staged_file

The next example uses the file_stage_in_shared RSL attribute to stage a file into the cache. The file is transferred from the client using the GASS https server embedded in the globusrun program when the -s option is used.

Example 4.18. File stage in shared

```
% globusrun -s -r grid.example.org:2119/jobmanager-pbs \
  "&(executable=/bin/ls) \
  (arguments = -l /tmp/staged_file_link1 /tmp/staged_file_link1) \
  (file_stage_in_shared = \
    ($(GLOBUSRUN_GASS_URL)/staged_file1 /tmp/staged_file_link1))"
```

```
lrwxr-xr-x  1 juser   juser  120 Nov 11 20:37 /tmp/staged_file1 -> /home/juser/.globus/.gass_cache/local/md5/ff/771bded8a2c7dacc1a1c0fecafa0ce/md5/39/13ab3db7fc002ed54012083ae6ed1c/data
```

The final staging example uses the file_stage_out RSL attribute to transfer a file from the GRAM service to an FTP server using anonymous FTP.

Example 4.19. File stage out

```
% globusrun -r grid.example.org:2119/jobmanager-pbs \
  "&(executable=a.out) \
  (file_stage_out = (results.txt ftp://anonymous:nopass@ftp.example.org/incoming/results.txt))"
```

```
% Note

In all of the above cases, multiple files may be staged using any combination of the supported URL schemes.

---

19 Submitting jobs to a job scheduler
3.4.9. Temporary files and cleanup

GRAM5 supports creating a per-job scratch directory which can be used as a place to store files that will be automatically removed by GRAM when the job completes. It also supports an explicit list of files to remove when the job completes.

This example shows how to stage files into a scratch directory. It again uses the embedded GASS https server, stages to the GRAM service, then runs /bin/ls in the temporary directory. After the job completes, the contents of $(SCRATCH_DIRECTORY) and the directory itself are removed.

Example 4.20. Staging to scratch directory

```sh
% globusrun -s grid.example.org:2119/jobmanager-pbs 
   "&(scratch_dir = $(HOME)) 
    (directory = $(SCRATCH_DIRECTORY)) 
    (file_stage_in = 
      ($(GLOBUSRUN_GASS_URL)/inputfile $(SCRATCH_DIRECTORY)/inputfile)) 
    (executable = /bin/ls)"
```

This example shows how to explicitly remove a file that was created by the job.

Example 4.21. Cleaning up a file

```sh
% globusrun -s grid.example.org:2119/jobmanager-pbs 
   "&(executable = /bin/touch) 
    (arguments = temporary_file) 
    (file_clean_up = temporary_file)"
```

3.4.10. Reliable job submit

The `globusrun` command supports a two-phase commit protocol to ensure that the client knows the contact of the job which has been created so that it can be monitored or canceled in the case of a client or service error. The two-phase commit affects both job submission and termination.

The two-phase protocol is enabled by using the `two_phase` RSL attribute, as in the next example. When this is enabled, job submission will fail with the error `GLOBUSGRAM_PROTOCOL_ERROR_WAITING_FOR_COMMIT`. The client must respond to this signal with either the `GLOBUSGRAM_PROTOCOL_JOB_SIGNAL_COMMIT_REQUEST` or `GLOBUSGRAM_PROTOCOL_JOB_SIGNAL_COMMIT_EXTEND` signals to either commit the job to execution or delay the commit timeout. One of these signals must be sent prior to the two phase commit timeout, or the job will be discarded by the GRAM service.

A two phase protocol is also used at job termination if the `save_state` RSL attribute is used along with the `two_phase` attribute. When the job manager sends a callback with the job state set to `GLOBUSGRAM_PROTOCOL_JOB_STATE_DONE` or `GLOBUSGRAM_PROTOCOL_JOB_STATE_DONE` it will wait to clean up the job until the two phase commit occurs. The client must reply with the `GLOBUSGRAM_PROTOCOL_JOB_SIGNAL_COMMIT_END` signal to cause the job to be cleaned. Otherwise, the job will be unloaded from memory until a client restarts the job and sends the signal.
Example 4.22. Two phase commit example

In this example, the user submits a job with a `two_phase` timeout of 30 seconds and the `save_state` attribute. The client must send commit signals to ensure the job runs.

```bash
% globusrun -r grid.example.org:2119/jobmanager-pbs "
  (two_phase = 30) \n  (save_state = yes) \n  (executable = a.out)"
```

globus_gram_client_callback_allow successful
GRAM Job submission successful
GLOBUS_GRAM_PROTOCOL_JOB_STATE_PENDING
GLOBUS_GRAM_PROTOCOL_JOB_STATE_ACTIVE
GLOBUS_GRAM_PROTOCOL_JOB_STATE_DONE
%

3.4.11. Reconnecting to a job

If a job manager or client exits before a job has completed, the job will continue to run. The client can reconnect to a job manager and receive job state notifications and output using the `restart` RSL attribute.

Example 4.23. Restart example

This example uses `globus-job-submit` to submit a batch job and then `globusrun` to reconnect to the job.

```bash
% globus-job-submit grid.example.org:2119/jobmanager-pbs /bin/sleep 90
https://grid.example.org:38824/16001746665595486521/5295612977486005662/
% globusrun -r grid.example.org:2119/jobmanager-pbs "
  (restart = https://grid.example.org:38824/16001746665595486521/5295612977486005662/"
```

globus_gram_client_callback_allow successful
GRAM Job submission successful
GLOBUS_GRAM_PROTOCOL_JOB_STATE_DONE
%

3.4.12. Submitting a Java job

To submit a job that runs a java program, the client must ensure that the job can find the Java interpreter and its classes. This example sets the default `PATH` and `CLASSPATH` environment variables and uses the shell to locate the path to the `java` program.
Example 4.24. Java example

This example uses `globus-job-submit` to submit a java job, staging a jar file from a remote service.

```
% globusrun -r grid.example.org:2119/jobmanager-pbs \
  "& (environment = (PATH '/usr/bin:/bin') (CLASSPATH \${SCRATCH_DIRECTORY})) \n  (scratch_dir = \${HOME}) \n  (directory = \${SCRATCH_DIRECTORY}) \n  (rsl_substitution = (JAVA_SERVER http://java.example.org)) \n  (file_stage_in = \n    (\${JAVA_SERVER}/example.jar \${SCRATCH_DIRECTORY}/example.jar) \n    (\${JAVA_SERVER}/support.jar \${SCRATCH_DIRECTORY}/support.jar)) \n  (executable=/bin/sh) \n  (arguments=-c 'java -jar example.jar')"

globus_gram_client_callback_allow successful
GRAM Job submission successful
GLOBUS_GRAM_PROTOCOL_JOB_STATE_PENDING
GLOBUS_GRAM_PROTOCOL_JOB_STATE_ACTIVE
GLOBUS_GRAM_PROTOCOL_JOB_STATE_DONE
%
```
Appendix A. Globus Toolkit 5.0.2 Public Interface Guides

This page contains links to each GT 5.0.2 component's Public Interfaces Guide.

- Common Runtime Components
  - XIO
  - C Common Libraries
- Security
  - GSI C
  - MyProxy
  - GSI-OpenSSH
- Data Management
  - GridFTP
  - RLS
- Execution Management
  - GRAM5
Appendix B. Globus Toolkit 5.0.2 Errors

- XIO
- GSI credentials
- GSI gridmap
- MyProxy
- GSI-OpenSSH
- GridFTP
- Replica Location Service (RLS)
- GRAM5
### Table B.1. XIO Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation was canceled</td>
<td>An I/O operation has been canceled by a close or a cancel</td>
<td>In most cases this will be intentionally performed by the application developer. In unexpected cases the application developer should verify that there is not a race condition relating to closing a handle.</td>
</tr>
<tr>
<td>Operation timed out</td>
<td>Occurs when the application developer associates a timeout with a handle's I/O operations. If no I/O is performed before the timeout expires this error will be triggered.</td>
<td>The remote side of connection might be hung and busy. The network could have higher latencies than expected. The filesystem might be over worked.</td>
</tr>
<tr>
<td>An end of file occurred</td>
<td>This occurs when and EOF is detected on the file descriptor</td>
<td>When doing file I/O this like means you read to the end of the file and thus you are finished and should now close it. On network connections however it means the socket was closed on the remote end. This can happen it the remote side suddenly dies (seg-fault is common here) or if the remote side chooses to close the connection.</td>
</tr>
<tr>
<td>Contact string invalid</td>
<td>A poorly formed contact string was passed in to open</td>
<td>Verify the format of the contact string with the documentation of the drivers in use.</td>
</tr>
<tr>
<td>Memory allocation failed on XXXX</td>
<td>malloc failed. The system is likely quite overloaded</td>
<td>Free up memory in your application</td>
</tr>
<tr>
<td>System error in XXXX</td>
<td>A low level system error occurred. The errno and errstring should indicate more information.</td>
<td></td>
</tr>
<tr>
<td>Invalid stack</td>
<td>The requested stack does not meet XIO standards</td>
<td>Most likely a transport driver is not on the bottom of the stack, or 2 transport drivers are in the stack.</td>
</tr>
<tr>
<td>Operation already registered</td>
<td>With certain common drivers like TCP and FILE, only one specific operations can be registered at a time (1 read, 1 write). If another operation of the same type is posted to the handle before receiving the previous operations callback, this error can occur.</td>
<td>Restructure the application code so that it waits for the callback before registering the next IO operation.</td>
</tr>
<tr>
<td>Unexpected state</td>
<td>The internal logic of XIO came across a logical path that should not be possible. Often times this is due to application memory corruption or trying to perform an IO operation on a closed or otherwise invalid handle.</td>
<td>Use valgrind or some sort of memory management tool to verify there is no memory corruption. Try to recreate the problem in a small program. Submit the program and the memory trace at bugzilla.globus.org</td>
</tr>
<tr>
<td>Driver in handle has been unloaded</td>
<td>A driver associated with the offending operation has already been unloaded by the application code.</td>
<td>Verify that you are not unloading drivers until they are no longer in use.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Definition</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>Module not activated</td>
<td>globus_module_activate(GLOBUS_XIO_MODULE); has not been called.</td>
<td>Call this before making any other XIO API calls.</td>
</tr>
</tbody>
</table>
### Table B.2. Credential Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your proxy credential may have expired</td>
<td>Your proxy credential may have expired.</td>
<td>Use <code>grid-proxy-info</code> to check whether the proxy credential has actually expired. If it has, generate a new proxy with <code>grid-proxy-init</code>.</td>
</tr>
<tr>
<td>The system clock on either the local or remote system is wrong.</td>
<td>This may cause the server or client to conclude that a credential has expired.</td>
<td>Check the system clocks on the local and remote system.</td>
</tr>
<tr>
<td>Your end-user certificate may have expired</td>
<td>Your end-user certificate may have expired</td>
<td>Use <code>grid-cert-info</code> to check your certificate's expiration date. If it has expired, follow your CA's procedures to get a new one.</td>
</tr>
<tr>
<td>The permissions may be wrong on your proxy file</td>
<td>If the permissions on your proxy file are too lax (for example, if others can read your proxy file), Globus Toolkit clients will not use that file to authenticate.</td>
<td>You can &quot;fix&quot; this problem by changing the permissions on the file or by destroying it (with <code>grid-proxy-destroy</code>) and creating a new one (with <code>grid-proxy-init</code>). <strong>Important:</strong> However, it is still possible that someone else has made a copy of that file during the time that the permissions were wrong. In that case, they will be able to impersonate you until the proxy file expires or your permissions or end-user certificate are revoked, whichever happens first.</td>
</tr>
<tr>
<td>The permissions may be wrong on your private key file</td>
<td>If the permissions on your end user certificate private key file are too lax (for example, if others can read the file), <code>grid-proxy-init</code> will refuse to create a proxy certificate.</td>
<td>You can &quot;fix&quot; this by changing the permissions on the private key file. <strong>Important:</strong> However, you will still have a much more serious problem: it is possible that someone has made a copy of your private key file. Although this file is encrypted, it is possible that someone will be able to decrypt the private key, at which point they will be able to impersonate you as long as your end user certificate is valid. You should contact your CA to have your end-user certificate revoked and get a new one.</td>
</tr>
<tr>
<td>The remote system may not trust your CA</td>
<td>The remote system may not trust your CA</td>
<td>Verify that the remote system is configured to trust the CA that issued your end-entity certificate. See Installing GT 5.0.2 for details.</td>
</tr>
<tr>
<td>You may not trust the remote system's CA</td>
<td>You may not trust the remote system's CA</td>
<td>Verify that your system is configured to trust the remote CA (or that your environment is set up to trust the remote CA). See Installing GT 5.0.2 for details.</td>
</tr>
<tr>
<td>There may be something wrong with the remote service's credentials</td>
<td>There may be something wrong with the remote service's credentials</td>
<td>It is sometimes difficult to distinguish between errors reported by the remote service regarding your credentials and errors reported by the client interface regarding the remote service's credentials. If you cannot find anything wrong with your credentials, check for the same conditions on the remote system (or ask a remote administrator to do so).</td>
</tr>
</tbody>
</table>
Table B.3. Gridmap Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The content of the grid map file does not conform to the expected format</td>
<td>The content of the grid map file does not conform to the expected format</td>
<td>Run <code>grid-mapfile-check-consistency</code> to make sure that your gridmap file conforms to the expected format.</td>
</tr>
<tr>
<td>The grid map file does not contain a entry for your DN</td>
<td>The grid map file does not contain a entry for your DN</td>
<td>Use <code>grid-mapfile-add-entry</code> to add the relevant entry.</td>
</tr>
</tbody>
</table>
## Table B.4. MyProxy Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MyProxy server name does not match expected name</td>
<td>This error appears as a mutual authentication failure or a server authentication failure, and the error message should list two names: the expected name of the MyProxy server and the actual authenticated name. By default, the MyProxy clients expect the MyProxy server to be running with a host certificate that matches the target hostname. This error can occur when running the MyProxy server under a non-host certificate or if the server is running on a machine with multiple hostnames. The MyProxy clients authenticate the identity of the MyProxy server to avoid sending passphrases and credentials to rogue servers. If the expected name contains an IP address, your system is unable to do a reverse lookup on that address to get the canonical hostname of the server, indicating either a problem with that machine's DNS record or a problem with the resolver on your system.</td>
<td>If the server name shown in the error message is acceptable, set the MYPROXY_SERVER_DN environment variable to that name to resolve the problem.</td>
</tr>
<tr>
<td>Error in bind(): Address already in use</td>
<td>This error indicates that the myproxy-server port (default: 7512) is in use by another process, probably another myproxy-server instance. You cannot run multiple instances of the myproxy-server on the same network port.</td>
<td>If you want to run multiple instances of the myproxy-server on a machine, you can specify different ports with the -p option, and then give the same -p option to the MyProxy commands to tell them to use the myproxy-server on that port.</td>
</tr>
<tr>
<td>grid-proxy-init failed</td>
<td>This error indicates that the grid-proxy-init command failed when myproxy-init attempted to run it, which implies a problem with the underlying Globus installation.</td>
<td>Run grid-proxy-init -debug -verify for more information.</td>
</tr>
<tr>
<td>User not authorized</td>
<td>An error from the myproxy-server saying you are &quot;not authorized&quot; to complete an operation typically indicates that the myproxy-server.config file settings are restricting your access to the myproxy-server. It is possible that the myproxy-server is running with the default myproxy-server.config file, which does not authorize any operations.</td>
<td>See Configuring MyProxy for more information.</td>
</tr>
</tbody>
</table>
### Error Code

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unable to verify remote side's credentials</td>
<td>An error saying &quot;Unable to verify remote side's credentials,&quot; &quot;Couldn't verify the remote certificate,&quot; or &quot;alert bad certificate&quot; often indicates that the client or server's certificate is signed by an untrusted Certification Authority (CA). The client must have a CA certificate and signing policy file installed in <code>/etc/grid-security/certificates</code> for the CA that signed the server's certificate. Likewise, the server must have a CA certificate and signing policy file installed in <code>/etc/grid-security/certificates</code> for the CA that signed the client's certificate.</td>
<td>See <a href="#">Configuring Certificates</a> for more information.</td>
</tr>
</tbody>
</table>
Table B.5. GSI-OpenSSH Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSS-API error Failure acquiring GSSAPI credentials: GSS_S_CREDENTIALS_EXPIRED</td>
<td>This means that your proxy certificate has expired.</td>
<td>Run <code>grid-proxy-init</code> to acquire a new proxy certificate, then run <code>gsissh</code> again.</td>
</tr>
<tr>
<td>...no proxy credentials...</td>
<td>Failing to run grid-proxy-init to create a user proxy with which to connect will result in the client notifying you that no local credentials exist. Any attempt to authenticate using GSI will fail in this case.</td>
<td>Verify that your GSI proxy has been properly initialized via <code>grid-proxy-info</code>. If you need to initialize the proxy, use the command <code>grid-proxy-init</code>.</td>
</tr>
<tr>
<td>...bad file system permissions on private key; key must only be readable by the user...</td>
<td>The host key that the SSH server is using for GSI authentication must only be readable by the user which owns it. Any other permissions will cause this error.</td>
<td>Make sure that the host key's UNIX permissions are mode 400 (that is, it should only have mode readable for the user that owns the file, and no other mode bits should be set).</td>
</tr>
<tr>
<td>...gssapi received empty username; failed to set username from gssapi context; Failed external-keyx for &lt;user&gt; from &lt;host&gt; &lt;port&gt;...</td>
<td>If the server was passed an &quot;implicit username&quot; (i.e. requested to map the incoming connection to a username based on some contextual clues such as the certificate's subject), and no entry exists in the grid-mapfile for the incoming connection's certificate subject, the server should output a clue that states it is unable to set the username against which to authenticate.</td>
<td>Add an entry for the user to the Section 1.2, &quot;Gridmap file&quot;.</td>
</tr>
<tr>
<td>...INTERNAL ERROR: authenticated invalid user xxx...</td>
<td>If the subject name given in the system's grid-mapfile points to a non-existent user, the server will give an internal error which is best caught when it is running in debugging mode.</td>
<td>Add a new account to the system matching the username pointed at by the user's subject in the grid-mapfile.</td>
</tr>
<tr>
<td>...gssapi received empty username; no suitable client data; failed to set username from gssapi context; Failed external-keyx for &lt;user&gt; from &lt;host&gt; &lt;port&gt;...</td>
<td>Should the user attempt to connect without first creating a proxy certificate, or if the user is connecting via a SSH client that does not support GSI authentication, the server will note that no GSSAPI data was sent to it. Verify that the client is able to connect through another GSI service (such as the gatekeeper) to make sure that the user's proxy has been created correctly.</td>
<td>Verify that you are using a GSI-enabled SSH client and that your GSI proxy has been properly initialized via <code>grid-proxy-info</code>. If you need to initialize this proxy, use the command <code>grid-proxy-init</code>.</td>
</tr>
</tbody>
</table>
**Table B.6. GridFTP Errors**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>globus_ftp_client</code>: the server responded with an error 530 <code>530-globus_xio: Authentication Error 530-OpenSSL Error: s3_srvc.c:2525: in library: SSL routines, function SSL3_GET_CLIENT_CERTIFICATE: no certificate returned</code></td>
<td>This error message indicates that the GridFTP server doesn't trust the certificate authority (CA) that issued your certificate.</td>
<td>You need to ask the GridFTP server administrator to install your CA certificate chain in the GridFTP server's trusted certificates directory.</td>
</tr>
<tr>
<td><code>globus_ftp_control: gss_init_sec_context failed OpenSSL Error: s3_clnt.c:951: in library: SSL routines, function SSL3_GET_SERVER_CERTIFICATE: certificate verify failed</code></td>
<td>This error message indicates that your local system doesn't trust the certificate authority (CA) that issued the certificate on the resource you are connecting to.</td>
<td>You need to ask the resource administrator which CA issued their certificate and install the CA certificate in the local trusted certificates directory.</td>
</tr>
</tbody>
</table>

Globus Toolkit 5.0.2 Errors
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>530-globus_xio: Authentication Error</td>
<td>This error message indicates one of the following: Certificate Revocation List (CRL) for the source or destination server CA at the client has expired or CRL for client CA has expired at source or destination server or CRL for source (destination) server CA has expired at destination (source) server. CRL is a file <code>{CA_hash}.r0</code> in <code>/etc/grid-security/certificates</code> or <code>${USER_HOME}/.globus/certificates</code> or <code>${X509_CERT_DIR}</code>.</td>
<td>The tool available at <a href="http://dist.eu-gridpma.info/distribution/util/fetch-crl/">http://dist.eu-gridpma.info/distribution/util/fetch-crl/</a> can be run in a crontab to keep the CRLs up to date.</td>
</tr>
</tbody>
</table>
Table B.7. Replica Locator Service (RLS) Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Definition</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error with credential: The proxy credential: <code>&lt;credential&gt;</code> with subject: <code>&lt;subject&gt;</code> expired <code>&lt;minutes&gt;</code> minutes ago</td>
<td>Expired proxy credential</td>
<td>Create a new proxy with <code>grid-proxy-init</code>.</td>
</tr>
</tbody>
</table>
| Unable to connect to local-host:xxxx | Unable to connect to the local host. This can be due to a variety of reasons, including a wrong address or port number in the RLS connection URL or an issue with a firewall configuration. | • Double-check the address and port number in the RLS connection URL. parameters are correct.  
  • If a firewall configuration is preventing connections to the target host for a particular port, you may need to consult the system administrator. |
| "connection timeout" | At times, a client may experience a connection timeout when interacting with the RLS server due to a variety of reasons:  
  • One reason could simply be due to wide-area network latency or congestion.  
  • Another situation that users eventually encounter is due to scaling of the system. As the RLS server’s database of replica location mappings grows in size, some query operations, such as bulk queries involving large quantities of mappings or wildcard queries that result in a large subset of mappings, will begin to take more time both to process the query and to return the large results set to the client over the network. | If timeouts are experienced with increasing frequency, increase the RLS server’s timeout configuration parameter found in the `$GLOBUS_LOCATION/var/globus-rls-server.conf` file. You may also use the `-t` timeout option of the `globus-rls-cli` tool. |
Table B.8. GRAM5 Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Reason</th>
<th>Possible Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>one of the RSL parameters is not supported</td>
<td>Check RSL documentation</td>
</tr>
<tr>
<td>2</td>
<td>the RSL length is greater than the maximum allowed</td>
<td>Use RSL substitutions to reduce length of RSL strings</td>
</tr>
<tr>
<td>3</td>
<td>an I/O operation failed</td>
<td>Enable trace logging and report to <a href="mailto:gram-dev@globus.org">gram-dev@globus.org</a></td>
</tr>
<tr>
<td>4</td>
<td>jobmanager unable to set default to the directory requested</td>
<td>Check that RSL directory attribute refers to a directory that exists on the target system.</td>
</tr>
<tr>
<td>5</td>
<td>the executable does not exist</td>
<td>Check that the RSL executable attribute refers to an executable that exists on the target system.</td>
</tr>
<tr>
<td>6</td>
<td>of an unused INSUFFICIENT_FUNDS</td>
<td>Unimplemented feature.</td>
</tr>
<tr>
<td>7</td>
<td>authentication with the remote server failed</td>
<td>Check that the contact string contains the proper X.509 DN.</td>
</tr>
<tr>
<td>8</td>
<td>the user cancelled the job</td>
<td>Don’t cancel jobs you want to complete.</td>
</tr>
<tr>
<td>9</td>
<td>the system cancelled the job</td>
<td>Check RSL requirements such as maximum time and memory are valid for the job.</td>
</tr>
<tr>
<td>10</td>
<td>data transfer to the server failed</td>
<td>Check gatekeeper and/or job manager logs to see why the process failed.</td>
</tr>
<tr>
<td>11</td>
<td>the stdin file does not exist</td>
<td>Check that the RSL stdin attribute refers to a file that exists on the target system or has a valid ftp, gsiftp, http, or https URL.</td>
</tr>
<tr>
<td>12</td>
<td>the connection to the server failed (check host and port)</td>
<td>Check that the service is running on the expected TCP/IP port. Check that no firewall prevents contacting that TCP/IP port. Check $GLOBUS_LOCATION/var/globus-gatekeeper.log for runtime configuration errors.</td>
</tr>
<tr>
<td>13</td>
<td>the provided RSL 'maxtime' value is not an integer</td>
<td>Check that the RSL maxtime value evaluates to an integer.</td>
</tr>
<tr>
<td>14</td>
<td>the provided RSL 'count' value is not an integer</td>
<td>Check that the RSL count value evaluates to an integer.</td>
</tr>
<tr>
<td>15</td>
<td>the job manager received an invalid RSL</td>
<td>Check that the RSL string can be parsed by using <code>globusrun -p RSL</code>.</td>
</tr>
<tr>
<td>16</td>
<td>the job manager failed in allowing others to make contact</td>
<td>Check job manager log.</td>
</tr>
<tr>
<td>17</td>
<td>the job failed when the job manager attempted to run it</td>
<td>Verify that the LRM is configured properly.</td>
</tr>
<tr>
<td>18</td>
<td>an invalid paradyn was specified</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>19</td>
<td>the provided RSL 'jobtype' value is invalid</td>
<td>The RSL jobtype attribute is not indicated as supported by the LRM. Valid jobtype values are single, multiple, mpi, and condor.</td>
</tr>
<tr>
<td>20</td>
<td>the provided RSL 'myjob' value is invalid</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>Error Code</td>
<td>Reason</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>21</td>
<td>the job manager failed to locate an internal script argument file</td>
<td>Check that $GLOBUS_LOCATION/libexec/globus-job-manager-script.pl exists and is executable. Check that the LRM-specific perl module is located in $GLOBUS_LOCATION/lib/perl/Globus/GRAM/JobManager/ directory and is valid. The command <code>perl -I$GLOBUS_LOCATION/lib/perl $GLOBUS_LOCATION/lib/perl/Globus/GRAM/JobManager/LRM.pm</code> can be used to check if there are any syntax errors in the script.</td>
</tr>
<tr>
<td>22</td>
<td>the job manager failed to create an internal script argument file</td>
<td>Check that your home directory is writable and not full.</td>
</tr>
<tr>
<td>23</td>
<td>the job manager detected an invalid job state</td>
<td>Check job manager logs.</td>
</tr>
<tr>
<td>24</td>
<td>the job manager detected an invalid script response</td>
<td>Check job manager logs. This is likely a bug in the LRM script.</td>
</tr>
<tr>
<td>25</td>
<td>the job manager detected an invalid script status</td>
<td>Check job manager logs. This is likely a bug in the LRM script.</td>
</tr>
<tr>
<td>26</td>
<td>the provided RSL 'jobtype' value is not supported by this job manager</td>
<td>Check that the RSL jobtype attribute is implemented by the LRM script. Note that some job types require configuration.</td>
</tr>
<tr>
<td>27</td>
<td>unused ERROR_UNIMPLEMENTED</td>
<td>LRM does not support some feature included in the job request.</td>
</tr>
<tr>
<td>28</td>
<td>the job manager failed to create an internal script submission file</td>
<td>Check that the user's home file system is not full. Check job manager log.</td>
</tr>
<tr>
<td>29</td>
<td>the job manager cannot find the user proxy</td>
<td>Check that client is delegating a proxy when authenticating with the gatekeeper. Check that the user's home filesystem and the /tmp file system are not full.</td>
</tr>
<tr>
<td>30</td>
<td>the job manager failed to open the user proxy</td>
<td>Check that the user's home filesystem and the /tmp file system are not full.</td>
</tr>
<tr>
<td>31</td>
<td>the job manager failed to cancel the job as requested</td>
<td>Check that the user's home filesystem and the /tmp file system are not full.</td>
</tr>
<tr>
<td>32</td>
<td>system memory allocation failed</td>
<td>Check job manager log for details.</td>
</tr>
<tr>
<td>33</td>
<td>the interprocess job communication initialization failed</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>34</td>
<td>the interprocess job communication setup failed</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>35</td>
<td>the provided RSL 'host count' value is invalid</td>
<td>Check that the RSL host_count attribute evaluates to an integer.</td>
</tr>
<tr>
<td>36</td>
<td>one of the provided RSL parameters is unsupported</td>
<td>Check job manager log for details about invalid parameter.</td>
</tr>
<tr>
<td>37</td>
<td>the provided RSL 'queue' parameter is invalid</td>
<td>Check that the RSL queue attribute evaluates to a string that corresponds to an LRM-specific queue name.</td>
</tr>
<tr>
<td>38</td>
<td>the provided RSL 'project' parameter is invalid</td>
<td>Check that the RSL project attribute evaluates to a string that corresponds to an LRM-specific project name.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Reason</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>39</td>
<td>the provided RSL string includes variables that could not be identified</td>
<td>Check that all RSL substitutions are defined before being used in the job description.</td>
</tr>
<tr>
<td>40</td>
<td>the provided RSL 'environment' parameter is invalid</td>
<td>Check that the RSL environment attribute contains a sequence of VARIABLE VALUE pairs.</td>
</tr>
<tr>
<td>41</td>
<td>the provided RSL 'dryrun' parameter is invalid</td>
<td>Remove the RSL dryrun attribute from the job description.</td>
</tr>
<tr>
<td>42</td>
<td>the provided RSL is invalid (an empty string)</td>
<td>Include a non-empty RSL string in your job submission request.</td>
</tr>
<tr>
<td>43</td>
<td>the job manager failed to stage the executable</td>
<td>Check that the file service hosting the executable is reachable from the GRAM5 service node. Check that the executable exists on the file service node. Check that there is sufficient disk space in the user's home directory on the service node to store the executable.</td>
</tr>
<tr>
<td>44</td>
<td>the job manager failed to stage the stdin file</td>
<td>Check that the file service hosting the standard input file is reachable from the GRAM5 service node. Check that the standard input file exists on the file service node. Check that there is sufficient disk space in the user's home directory on the service node to store the standard input file.</td>
</tr>
<tr>
<td>45</td>
<td>the requested job manager type is invalid</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>46</td>
<td>the provided RSL 'arguments' parameter is invalid</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>47</td>
<td>the gatekeeper failed to run the job manager</td>
<td>Check the gatekeeper or job manager logs for more information.</td>
</tr>
<tr>
<td>48</td>
<td>the provided RSL could not be properly parsed</td>
<td>Check that the RSL string can be parsed by using globusrun -p RSL.</td>
</tr>
<tr>
<td>49</td>
<td>there is a version mismatch between GRAM components</td>
<td>Ask system administrator to upgrade GRAM service to GRAM2 or GRAM5</td>
</tr>
<tr>
<td>50</td>
<td>the provided RSL 'arguments' parameter is invalid</td>
<td>Check that the RSL arguments attribute evaluates to a sequence of strings.</td>
</tr>
<tr>
<td>51</td>
<td>the provided RSL 'count' parameter is invalid</td>
<td>Check that the RSL count attribute evaluates to a positive integer value.</td>
</tr>
<tr>
<td>52</td>
<td>the provided RSL 'directory' parameter is invalid</td>
<td>Check that the RSL directory attribute evaluates to a string.</td>
</tr>
<tr>
<td>53</td>
<td>the provided RSL 'dryrun' parameter is invalid</td>
<td>Check that the RSL dryrun attribute evaluates to either yes or no.</td>
</tr>
<tr>
<td>54</td>
<td>the provided RSL 'environment' parameter is invalid</td>
<td>Check that the RSL environment attribute evaluates to a sequence of VARIABLE VALUE pairs.</td>
</tr>
<tr>
<td>55</td>
<td>the provided RSL 'executable' parameter is invalid</td>
<td>Check that the RSL executable attribute evaluates to a string value.</td>
</tr>
<tr>
<td>56</td>
<td>the provided RSL 'host_count' parameter is invalid</td>
<td>Check that the RSL host_count attribute evaluates to a positive integer value.</td>
</tr>
<tr>
<td>57</td>
<td>the provided RSL 'jobtype' parameter is invalid</td>
<td>Check that the RSL jobtype attribute evaluates to one of single, multiple, mpi, or condor</td>
</tr>
<tr>
<td>Error Code</td>
<td>Reason</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>58</td>
<td>the provided RSL 'maxtime' parameter is invalid</td>
<td>Check that the RSL maxtime attribute evaluates to a positive integer value.</td>
</tr>
<tr>
<td>59</td>
<td>the provided RSL 'myjob' parameter is invalid</td>
<td>OBSOLETE IN GRAM5.</td>
</tr>
<tr>
<td>60</td>
<td>the provided RSL 'paradyn' parameter is invalid</td>
<td>OBSOLETE IN GRAM2.</td>
</tr>
<tr>
<td>61</td>
<td>the provided RSL 'project' parameter is invalid</td>
<td>Check that the RSL project attribute evaluates to a string value.</td>
</tr>
<tr>
<td>62</td>
<td>the provided RSL 'queue' parameter is invalid</td>
<td>Check that the RSL queue attribute evaluates to a string value.</td>
</tr>
<tr>
<td>63</td>
<td>the provided RSL 'stderr' parameter is invalid</td>
<td>Check that the RSL stderr attribute evaluates to a string value or a sequence of DESTINATION URLs with optional CACHE_TAG string parameters.</td>
</tr>
<tr>
<td>64</td>
<td>the provided RSL 'stdin' parameter is invalid</td>
<td>Check that the RSL stdin attribute evaluates to a string value.</td>
</tr>
<tr>
<td>65</td>
<td>the provided RSL 'stdout' parameter is invalid</td>
<td>Check that the RSL stdout attribute evaluates to a string value or a sequence of DESTINATION URLs with optional CACHE_TAG string parameters.</td>
</tr>
<tr>
<td>66</td>
<td>the job manager failed to locate an internal script</td>
<td>Check job manager log for more details.</td>
</tr>
<tr>
<td>67</td>
<td>the job manager failed on the system call pipe()</td>
<td>OBSOLETE IN GRAM5.</td>
</tr>
<tr>
<td>68</td>
<td>the job manager failed on the system call fcntl()</td>
<td>OBSOLETE IN GRAM2.</td>
</tr>
<tr>
<td>69</td>
<td>the job manager failed to create the temporary stdout filename</td>
<td>OBSOLETE IN GRAM5.</td>
</tr>
<tr>
<td>70</td>
<td>the job manager failed to create the temporary stderr filename</td>
<td>OBSOLETE IN GRAM5.</td>
</tr>
<tr>
<td>71</td>
<td>the job manager failed on the system call fork()</td>
<td>OBSOLETE IN GRAM2.</td>
</tr>
<tr>
<td>72</td>
<td>the executable file permissions do not allow execution</td>
<td>Check that the RSL executable attribute refers to an executable program or script.</td>
</tr>
<tr>
<td>73</td>
<td>the job manager failed to open stdout</td>
<td>Check that the RSL stdout attribute refers to one or more valid destination files or URLs.</td>
</tr>
<tr>
<td>74</td>
<td>the job manager failed to open stderr</td>
<td>Check that the RSL stderr attribute refers to one or more valid destination files or URLs.</td>
</tr>
<tr>
<td>75</td>
<td>the cache file could not be opened in order to relocate the user proxy</td>
<td>Check that the user's home directory is writable and not full on the GRAM5 service node.</td>
</tr>
<tr>
<td>76</td>
<td>cannot access cache files in ~/.globus/gass_cache, check permissions, quota, and disk space</td>
<td>Check that the user's home directory is writable and not full on the GRAM5 service node.</td>
</tr>
<tr>
<td>77</td>
<td>the job manager failed to insert the contact in the client contact list</td>
<td>Check job manager log.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Reason</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>78</td>
<td>the contact was not found in the job-manager's client contact list</td>
<td>Don’t attempt to unregister callback contacts that are not registered</td>
</tr>
<tr>
<td>79</td>
<td>connecting to the job manager failed. Possible reasons: job terminated, invalid job contact, network problems, ...</td>
<td>Check that the job manager process is running. Check that the job manager credential has not expired. Check that the job manager contact refers to the correct TCP/IP host and port. Check that the job manager contact is not blocked by a firewall.</td>
</tr>
<tr>
<td>80</td>
<td>the syntax of the job contact is invalid</td>
<td>Check the syntax of job contact string.</td>
</tr>
<tr>
<td>81</td>
<td>the executable parameter in the RSL is undefined</td>
<td>Include the RSL executable in all job requests.</td>
</tr>
<tr>
<td>82</td>
<td>the job manager service is misconfigured. condor arch undefined</td>
<td>Add the <code>-condor-arch</code> to the command-line or configuration file for a job manager configured to use the condor LRM.</td>
</tr>
<tr>
<td>83</td>
<td>the job manager service is misconfigured. condor os undefined</td>
<td>Add the <code>-condor-os</code> to the command-line or configuration file for a job manager configured to use the condor LRM.</td>
</tr>
<tr>
<td>84</td>
<td>the provided RSL 'min_memory' parameter is invalid</td>
<td>Check that the RSL <code>min_memory</code> attribute evaluates to a positive integer value.</td>
</tr>
<tr>
<td>85</td>
<td>the provided RSL 'max_memory' parameter is invalid</td>
<td>Check that the RSL <code>max_memory</code> attribute evaluates to a positive integer value.</td>
</tr>
<tr>
<td>86</td>
<td>the RSL 'min_memory' value is not zero or greater</td>
<td>Check that the RSL <code>min_memory</code> attribute evaluates to a positive integer value.</td>
</tr>
<tr>
<td>87</td>
<td>the RSL 'max_memory' value is not zero or greater</td>
<td>Check that the RSL <code>max_memory</code> attribute evaluates to a positive integer value.</td>
</tr>
<tr>
<td>88</td>
<td>the creation of a HTTP message failed</td>
<td>Check job manager log.</td>
</tr>
<tr>
<td>89</td>
<td>parsing incoming HTTP message failed</td>
<td>Check job manager log.</td>
</tr>
<tr>
<td>90</td>
<td>the packing of information into a HTTP message failed</td>
<td>Check job manager log.</td>
</tr>
<tr>
<td>91</td>
<td>an incoming HTTP message did not contain the expected information</td>
<td>Check job manager log.</td>
</tr>
<tr>
<td>92</td>
<td>the job manager does not support the service that the client requested</td>
<td>Check that the client is talking to the correct service</td>
</tr>
<tr>
<td>93</td>
<td>the gatekeeper failed to find the requested service</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>94</td>
<td>the job manager does not accept any new requests (shutting down)</td>
<td>Execute queries before the job has been cleaned up.</td>
</tr>
<tr>
<td>95</td>
<td>the client failed to close the listener associated with the callback URL</td>
<td>Call <code>globus_gram_client_callback_disallow()</code> with a valid the callback contact.</td>
</tr>
<tr>
<td>96</td>
<td>the gatekeeper contact cannot be parsed</td>
<td>Check the syntax of the gatekeeper contact string you are attempting to contact.</td>
</tr>
<tr>
<td>97</td>
<td>the job manager could not find the 'poe' command</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>98</td>
<td>the job manager could not find the 'mpirun' command</td>
<td>Configure the LRM script with <code>mpirun</code> in your path.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Reason</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>99</td>
<td>the provided RSL 'start_time' parameter is invalid</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>100</td>
<td>the provided RSL 'reservation_handle' parameter is invalid</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>101</td>
<td>the provided RSL 'max_wall_time' parameter is invalid</td>
<td>Check that the RSL max_wall_time attribute evaluates to a positive integer.</td>
</tr>
<tr>
<td>102</td>
<td>the RSL 'max_wall_time' value is not zero or greater</td>
<td>Check that the RSL max_wall_time attribute evaluates to a positive integer.</td>
</tr>
<tr>
<td>103</td>
<td>the provided RSL 'max_cpu_time' parameter is invalid</td>
<td>Check that the RSL max_cpu_time attribute evaluates to a positive integer.</td>
</tr>
<tr>
<td>104</td>
<td>the RSL 'max_cpu_time' value is not zero or greater</td>
<td>Check that the RSL max_cpu_time attribute evaluates to a positive integer.</td>
</tr>
<tr>
<td>105</td>
<td>the job manager is misconfigured, a scheduler script is missing</td>
<td>Check that the administrator has configured the LRM by running its setup script.</td>
</tr>
<tr>
<td>106</td>
<td>the job manager is misconfigured, a scheduler script has invalid permissions</td>
<td>Check that the administrator has installed the GLLOBUS_LOCATION/libexec/globus-job-manager-script.pl script. Check that the file system containing that script allows file execution.</td>
</tr>
<tr>
<td>107</td>
<td>the job manager failed to signal the job</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>108</td>
<td>the job manager did not recognize/support the signal type</td>
<td>Check that your signal operation is using the correct signal constant.</td>
</tr>
<tr>
<td>109</td>
<td>the job manager failed to get the job id from the local scheduler</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>110</td>
<td>the job manager is waiting for a commit signal</td>
<td>Send a two-phase commit signal to the job manager to acknowledge receiving the job contact from the job manager.</td>
</tr>
<tr>
<td>111</td>
<td>the job manager timed out while waiting for a commit signal</td>
<td>Send a two-phase commit signal to the job manager to acknowledge receiving the job contact from the job manager. Increase the two-phase commit time out for your job. Check that the job manager contact TCP/IP port is reachable from your client.</td>
</tr>
<tr>
<td>112</td>
<td>the provided RSL 'save_state' parameter is invalid</td>
<td>Check that the RSL save_state attribute is set to yes or no.</td>
</tr>
<tr>
<td>113</td>
<td>the provided RSL 'restart' parameter is invalid</td>
<td>Check that the RSL restart attribute evaluates to a string containing a job contact string.</td>
</tr>
<tr>
<td>114</td>
<td>the provided RSL 'two_phase' parameter is invalid</td>
<td>Check that the RSL two_phase attribute evaluates to a positive integer.</td>
</tr>
<tr>
<td>115</td>
<td>the RSL 'two_phase' value is not zero or greater</td>
<td>Check that the RSL two_phase attribute evaluates to a positive integer.</td>
</tr>
<tr>
<td>116</td>
<td>the provided RSL 'stdout_position' parameter is invalid</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>117</td>
<td>the RSL 'stdout_position' value is not zero or greater</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>Error Code</td>
<td>Reason</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>118</td>
<td>the provided RSL 'stderr_position' parameter is invalid</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>119</td>
<td>the RSL 'stderr_position' value is not zero or greater</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>120</td>
<td>the job manager restart attempt failed</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>121</td>
<td>the job state file doesn't exist</td>
<td>Check that the job contact you are trying to restart matches one that the job manager returned to you.</td>
</tr>
<tr>
<td>122</td>
<td>could not read the job state file</td>
<td>Check that the state file directory is not full.</td>
</tr>
<tr>
<td>123</td>
<td>could not write the job state file</td>
<td>Check that the state file directory is not full.</td>
</tr>
<tr>
<td>124</td>
<td>old job manager is still alive</td>
<td>Contact the returned job manager contact to manage the job you are trying to restart.</td>
</tr>
<tr>
<td>125</td>
<td>job manager state file TTL expired</td>
<td>OBSOLETE in GRAM2</td>
</tr>
<tr>
<td>126</td>
<td>it is unknown if the job was submitted</td>
<td>Check job manager log.</td>
</tr>
<tr>
<td>127</td>
<td>the provided RSL 'remote_io_url' parameter is invalid</td>
<td>Check that the RSL remote_io_url attribute evaluates to a string value.</td>
</tr>
<tr>
<td>128</td>
<td>could not write the remote io url file</td>
<td>Check that the user's home file system on the job manager service node is writable and not full.</td>
</tr>
<tr>
<td>129</td>
<td>the standard output/error size is different</td>
<td>Send a stdio update signal to redirect the job manager output to a new URL.</td>
</tr>
<tr>
<td>130</td>
<td>the job manager was sent a stop signal (job is still running)</td>
<td>Submit a restart request to monitor the job.</td>
</tr>
<tr>
<td>131</td>
<td>the user proxy expired (job is still running)</td>
<td>Generate a new proxy and then submit a restart request to monitor the job.</td>
</tr>
<tr>
<td>132</td>
<td>the job was not submitted by original job manager</td>
<td>OBSOLETE IN GRAM2</td>
</tr>
<tr>
<td>133</td>
<td>the job manager is not waiting for that commit signal</td>
<td>Do not send a commit signal to a job that is not waiting for a commit signal.</td>
</tr>
<tr>
<td>134</td>
<td>the provided RSL scheduler specific parameter is invalid</td>
<td>Check the LRM-specific documentation to determine what values are legal for the RSL extensions implemented by the LRM.</td>
</tr>
<tr>
<td>135</td>
<td>the job manager could not stage in a file</td>
<td>Check that the file service hosting the file to stage is reachable from the GRAM5 service node. Check that the file to stage exists on the file service node. Check that there is sufficient disk space in the user's home directory on the service node to store the file to stage.</td>
</tr>
<tr>
<td>136</td>
<td>the scratch directory could not be created</td>
<td>Check that the directory named by the RSL scratch_dir attribute exists and is writable. Check that the directory named by the RSL scratch_dir attribute is not full.</td>
</tr>
<tr>
<td>137</td>
<td>the provided 'gass_cache' parameter is invalid</td>
<td>Check that the RSL gass_cache attribute evaluates to a string.</td>
</tr>
<tr>
<td>138</td>
<td>the RSL contains attributes which are not valid for job submission</td>
<td>Do not use restart- or signal-only RSL attributes when submitting a job.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Reason</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>-------------------</td>
</tr>
<tr>
<td>139</td>
<td>the RSL contains attributes which are not valid for stdio update</td>
<td>Do not use submit- or restart-only RSL attributes when sending a stdio update signal to a job.</td>
</tr>
<tr>
<td>140</td>
<td>the RSL contains attributes which are not valid for job restart</td>
<td>Do not use submit- or signal-only RSL attributes when restarting a job.</td>
</tr>
<tr>
<td>141</td>
<td>the provided RSL 'file_stage_in' parameter is invalid</td>
<td>Check that the RSL file_stage_in attribute evaluates to a sequence of SOURCE DESTINATION pairs.</td>
</tr>
<tr>
<td>142</td>
<td>the provided RSL 'file_stage_in_shared' parameter is invalid</td>
<td>Check that the RSL file_stage_in_shared attribute evaluates to a sequence of SOURCE DESTINATION pairs.</td>
</tr>
<tr>
<td>143</td>
<td>the provided RSL 'file_stage_out' parameter is invalid</td>
<td>Check that the RSL file_stage_out attribute evaluates to a sequence of SOURCE DESTINATION pairs.</td>
</tr>
<tr>
<td>144</td>
<td>the provided RSL 'gass_cache' parameter is invalid</td>
<td>Check that the RSL gass_cache attribute evaluates to a string.</td>
</tr>
<tr>
<td>145</td>
<td>the provided RSL 'file_cleanup' parameter is invalid</td>
<td>Check that the RSL file_clean_up attribute evaluates to a sequence of strings.</td>
</tr>
<tr>
<td>146</td>
<td>the provided RSL 'scratch_dir' parameter is invalid</td>
<td>Check that the RSL scratch_dir attribute evaluates to a string.</td>
</tr>
<tr>
<td>147</td>
<td>the provided scheduler-specific RSL parameter is invalid</td>
<td>Check the LRM-specific documentation to determine what values are legal for the RSL extensions implemented by the LRM.</td>
</tr>
<tr>
<td>148</td>
<td>a required RSL attribute was not defined in the RSL spec</td>
<td>Check that the RSL executable attribute is present in your job request RSL. Check that the RSL restart attributes is present in your restart RSL.</td>
</tr>
<tr>
<td>149</td>
<td>the gass_cache attribute points to an invalid cache directory</td>
<td>Check that the RSL gass_cache attributes evaluates to a directory that exists or can be created. Check that the user's home file system is writable and not full.</td>
</tr>
<tr>
<td>150</td>
<td>the provided RSL 'save_state' parameter has an invalid value</td>
<td>Check that the RSL save_state attribute has a value of yes or no.</td>
</tr>
<tr>
<td>151</td>
<td>the job manager could not open the RSL attribute validation file</td>
<td>Check that $GLOBUS_LOCATION/share/globus_gram_job_manager/globus-gram-job-manager.rvf is present and readable on the job manager service node. Check that $GLOBUS_LOCATION/share/globus_gram_job_manager/LRM.rvf is readable on the job manager service node if present.</td>
</tr>
<tr>
<td>152</td>
<td>the job manager could not read the RSL attribute validation file</td>
<td>Check that $GLOBUS_LOCATION/share/globus_gram_job_manager/globus-gram-job-manager.rvf is valid. Check that $GLOBUS_LOCATION/share/globus_gram_job_manager/LRM.rvf is readable on the job manager service node if present.</td>
</tr>
<tr>
<td>153</td>
<td>the provided RSL 'proxy_timeout' is invalid</td>
<td>Check that RSL proxy_timeout attribute evaluates to a positive integer.</td>
</tr>
<tr>
<td>154</td>
<td>the RSL 'proxy_timeout' value is not greater than zero</td>
<td>Check that RSL proxy_timeout attribute evaluates to a positive integer.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Reason</td>
<td>Possible Solutions</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>155</td>
<td>the job manager could not stage out a file</td>
<td>Check that the source file being staged exists on the job manager service node. Check that the directory of the destination file being staged exists on the file service node. Check that the directory of the destination file being staged is writable by the user. Check that the destination file service is reachable by the job manager service node.</td>
</tr>
<tr>
<td>156</td>
<td>the job contact string does not match any which the job manager is handling</td>
<td>Check that the job contact string matches one returned from a job request.</td>
</tr>
<tr>
<td>157</td>
<td>proxy delegation failed</td>
<td>Check that the job manager service node trusts the signer of your credential. Check that you trust the signer of the job manager service node's credential.</td>
</tr>
<tr>
<td>158</td>
<td>the job manager could not lock the state lock file</td>
<td>Check that the file system holding the job state directory supports POSIX advisory locking. Check that the job state directory is writable by the user on the service node. Check that the job state directory is not full.</td>
</tr>
<tr>
<td>159</td>
<td>an invalid globus_io_clientattr_t was used.</td>
<td>Check that you have initialized the globus_io_clientattr_t attribute prior to using it with the GRAM client API.</td>
</tr>
<tr>
<td>160</td>
<td>an null parameter was passed to the gram library</td>
<td>Check that you are passing legal values to all GRAM API calls.</td>
</tr>
<tr>
<td>161</td>
<td>the job manager is still streaming output</td>
<td>OBSOLETE IN GRAM5</td>
</tr>
<tr>
<td>162</td>
<td>the authorization system denied the request</td>
<td>Check with your GRAM system administrator to allow a particular certificate to be authorized.</td>
</tr>
<tr>
<td>163</td>
<td>the authorization system reported a failure</td>
<td>Check with your system administrator to verify that the authorization system is configured properly.</td>
</tr>
<tr>
<td>164</td>
<td>the authorization system denied the request - invalid job id</td>
<td>Check with your system administrator to verify that the authorization system is configured properly. Use a credential which is authorized to interact with a particular GRAM job.</td>
</tr>
<tr>
<td>165</td>
<td>the authorization system denied the request - not authorized to run the specified executable</td>
<td>Check with your system administrator to verify that the authorization system is configured properly. Use a credential which is authorized to interact with a particular GRAM job.</td>
</tr>
<tr>
<td>166</td>
<td>the provided RSL 'user_name' parameter is invalid.</td>
<td>Check that the RSL user_name attribute evaluates to a string.</td>
</tr>
<tr>
<td>167</td>
<td>the job is not running in the account named by the 'user_name' parameter.</td>
<td>Ask with the GRAM system administrator to add an authorization entry to allow your credential to run jobs as the specified user account.</td>
</tr>
</tbody>
</table>
Glossary

C

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.

P

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.

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.

T

third party transfers
In the simplest terms, a third party transfer moves a file between two GridFTP servers.

The following is a more detailed, programmatic description.

In a third party transfer, there are three entities involved. The client, who will only orchestrate, but not actually take place in the data transfer, and two servers one of which will be sending data to the other. This scenario is common in Grid applications where you may wish to stage data from a data store somewhere to a supercomputer you have reserved. The commands are quite similar to the client/server transfer. However, now the client must establish two control channels, one to each server. He will then choose one to listen, and send it the PASV command. When it responds with the IP/port it is listening on, the client will send that IP/port as part of the PORT command to the other server. This will cause the second server to connect to the first server, rather than the client. To initiate the actual movement of the data, the client then sends the RETR “filename” command to the server that will read from disk and write to the network (the “sending” server) and will send the STOR “filename” command to the other server which will read from the network and write to the disk (the “receiving” server).

See Also client/server transfer.
user certificate

A EEC belonging to a user. When using GSI, this certificate is typically stored in
$HOME/.globus/usercert.pem. For more information on possible user
certificate locations, see this.