

GT 5.0.0 GRAM5: User's Guide

GT 5.0.0 GRAM5: User's Guide

Introduction

GRAM services provide secure, remote job submission to different *local resource managers* in a Grid environment. This document describes how to use the RSL language and command-line interfaces provided in GT 5.0.0 to submit jobs to Grid resources.

Table of Contents

1. Using GRAM5	1
1. Preparing to use GRAM	1
2. Java Client API Download	1
3. Delegating credentials	2
4. Submitting jobs	2
5. Using GRAM5 with Condor-G	13
I. GRAM5 Commands	14
globusrun	15
globus-job-cancel	19
globus-job-clean	20
globus-job-get-output	21
globus-job-run	23
globus-job-status	26
globus-job-submit	28
globus-personal-gatekeeper	31
globus-gram-audit	33
globus-gatekeeper	34
globus-job-manager	37
globus-job-manager-event-generator	42
globus-fork-starter	43
2. Troubleshooting	45
1. Troubleshooting tips	45
2. Errors	46
3. Known Problems in GRAM5	56
1. Known Problems	56
4. Usage statistics collection by the Globus Alliance	59
1. GRAM5-specific usage statistics	59
Glossary	62
Index	63

List of Tables

2.1. GRAM5 Errors	47
-------------------------	----

List of Examples

1.1. Generating a proxy with grid-proxy-init	1
1.2. Gatekeeper Service Contact Examples	3
1.3. Minimal job using globus-job-run	4
1.4. Multiprocess job using globus-job-run	4
1.5. Canceling an interactive job	4
1.6. Setting job environment variables with globus-job-run	4
1.7. Using custom RSL clauses with globus-job-run	5
1.8. Constructing RSL strings with globus-job-run	5
1.9. globus-job-submit	6
1.10. Checking RSL Syntax	7
1.11. GRAM Authentication test	7
1.12. GRAM version check	8
1.13. Basic Interactive Job	8
1.14. Basic Batch Job	8
1.15. Refreshing a Credential	9
1.16. Proxy Expiration Example	9
1.17. File stage in	10
1.18. File stage in shared	10
1.19. File stage out	10
1.20. Staging to scratch directory	11
1.21. Cleaning up a file	11
1.22. Two phase commit example	12
1.23. Restart example	12
1.24. Java example	13

Chapter 1. Using GRAM5

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 1.1. 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. Java Client API Download

GT 5.0.0 does not include any of the CoG JGlobus Java APIs that were included in the GT4 release series. But, the JGlobus APIs can still be used with the GT5 services. You can get them directly from the CoG JGlobus releases; see the following link:

http://dev.globus.org/wiki/CoG_jglobus

Consider the following when determining which version of CoG JGlobus to use:

- The GRAM development team used CoG JGlobus version 1.6.0 for performance testing.
- The BIRN project used CoG JGlobus version 1.6.0 (plus patches) for GridFTP testing. All patches are included in 1.8.0.
- At the time of the GT 5.0.0 release, 1.8.0 was the recommended version. In general, the latest recommended CoG JGlobus version should be used.

3. 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 5.0.0 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.

3.1. 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.

4. 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.

4.1. 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 1.2. 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`

4.2. 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 1.3. 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 1.4. 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 1.5. 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 1.6. 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:3882/16001579536700793196/5295612977485
GLOBUS_LOCATION=/opt/globus-5.0.0
GLOBUS_GASS_CACHE_DEFAULT=/home/juser/.globus/.gass_cache
TEST=1
X509_USER_PROXY=/home/juser/.globus/job/mactop.local/16001579536700793196.5295612977485997
GRID=1
```

Example 1.7. 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 1.8. 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.

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

4.3. 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 1.9. 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:38843/16001600430615223386/529561297748601358
PENDING
% globus-job-status https://grid.example.org:38843/16001600430615223386/529561297748601358
ACTIVE
% globus-job-status https://grid.example.org:38843/16001600430615223386/529561297748601358
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.

4.4. 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 `-dump_rsl` 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.

4.4.1. 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 1.10. 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

4.4.2. 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 1.11. GRAM Authentication test

```
% globusrun -a -r grid.example.org:2119/jobmanager-pbs  
GRAM Authentication test successful  
% globusrun -a -r grid.example.org:2119/jobmanager-lsf  
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.

4.4.3. Checking GRAM service version

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

Example 1.12. 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 inf
```

4.4.4. 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 1.13. 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
```

4.4.5. 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 1.14. Basic Batch Job

```
% globusrun -b -r grid.example.org:2119/jobmanager-pbs "&(executable=/bin/sleep)(arguments
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:38824/16001608125017717261/529561297748601998
PENDING
% globusrun -k https://grid.example.org:38824/16001608125017717261/5295612977486019989/
%
```

4.4.6. Refreshing a GRAM5 Credential

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

Example 1.15. Refreshing a Credential

```
% globusrun -refresh-proxy https://grid.example.org:38824/16001608125017717261/52956129774
% echo $?
0
```



Note

In GT 5.0.0, **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.

4.4.7. 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 1.16. Proxy Expiration Example

```
% 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:1997/1600170047757511"
globus_gram_client_callback_allow successful
GRAM Job submission successful
GLOBUS_GRAM_PROTOCOL_JOB_STATE_ACTIVE
GLOBUS_GRAM_PROTOCOL_JOB_STATE_DONE
```

4.4.8. 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 substitution variable to reduce the amount of redundancy in the job description.

Example 1.17. 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 1.18. 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/.ga
```

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 1.19. 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/resul" \  
%
```

Note

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

4.4.9. 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 1.20. Staging to scratch directory

```
% 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)" \  
inputfile
```

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

Example 1.21. Cleaning up a file

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

4.4.10. 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 `GLOBUS_GRAM_PROTOCOL_ERROR_WAITING_FOR_COMMIT`. The client must respond to this signal with either the `GLOBUS_GRAM_PROTOCOL_JOB_SIGNAL_COMMIT_REQUEST` or `GLOBUS_GRAM_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 `GLOBUS_GRAM_PROTOCOL_JOB_STATE_DONE` or `GLOBUS_GRAM_PROTOCOL_JOB_STATE_DONE` it will wait to clean up the job until the two phase commit occurs. The client must reply with the `GLOBUS_GRAM_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 1.22. 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.

```
% globusrun -r grid.example.org:2119/jobmanager-pbs \  
  "&(two_phase = 30) \  
    (save_state = yes) \  
    (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  
%
```

4.4.11. 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 1.23. Restart example

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

```
% 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  
%
```

4.4.12. 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 1.24. 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))) \  
  (scratch_dir = \$(HOME)) \  
  (directory = \$(SCRATCH_DIRECTORY)) \  
  (rsl_substitution = (JAVA_SERVER http://java.example.org)) \  
  (file_stage_in = \  
    (\$(JAVA_SERVER)/example.jar \$(SCRATCH_DIRECTORY)/example.jar) \  
    (\$(JAVA_SERVER)/support.jar \$(SCRATCH_DIRECTORY)/support.jar)) \  
  (executable=/bin/sh) \  
  (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 \  
%
```

5. Using GRAM5 with Condor-G

Condor-G users should upgrade their clients to condor 7.4.0 or later to achieve highest performance. That version includes the `gt5` grid type, which includes client-side optimizations to improve performance. To use an older Condor-G client, be sure to set the `GRIDMANAGER_MAX_JOBMANAGERS_PER_RESOURCE` attribute to 0 to disable the Condor-G client attempts to stop and restart the job manager service. Also, disable the Grid Monitor when using a GRAM5 resource by setting the `ENABLE_GRID_MONITOR` configuration attribute to `FALSE`.

GRAM5 Commands

Name

globusrun -- Execute and manage jobs via GRAM

```
globusrun [-help] [-usage] [-version] [-versions]
globusrun { -p | -parse }
{ -f RSL_FILENAME | -file RSL_FILENAME | RSL_SPECIFICATION }
globusrun [-n] [-no-interrupt]
{ -r RESOURCE_CONTACT | -resource RESOURCE_CONTACT }
{ -a | -authenticate-only }
globusrun [-n] [-no-interrupt]
{ -r RESOURCE_CONTACT | -resource RESOURCE_CONTACT }
{ -j | -jobmanager-version }
globusrun [-n] [-no-interrupt] { -k | -kill } { JOB_ID }
globusrun [-n] [-no-interrupt] [-full-proxy] [-D] { -y | -refresh-proxy } { JOB_ID }
globusrun { -status } { JOB_ID }
globusrun [-q] [-quiet] [-o] [-output-enable] [-s] [-server] [-w] [-write-allow] [-n] [-no-interrupt] [-b] [-batch] [-F]
[-fast-batch] [-full-proxy] [-D] [-d] [-dryrun]
{ -r RESOURCE_CONTACT | -resource RESOURCE_CONTACT }
{ -f RSL_FILENAME | -file RSL_FILENAME | RSL_SPECIFICATION }
```

Description

The **globusrun** program for submits and manages jobs run on a local or remote job host. The jobs are controlled by the **globus-job-manager** program which interfaces with a local resource manager that schedules and executes the job.

The **globusrun** program can be run in a number of different modes chosen by command-line options.

When `-help`, `-usage`, `-version`, or `-versions` command-line options are used, **globusrun** will print out diagnostic information and then exit.

When the `-p` or `-parse` command-line option is present, **globusrun** will verify the syntax of the RSL specification and then terminate. If the syntax is valid, **globusrun** will print out the string "RSL Parsed Successfully..." and exit with a zero exit code; otherwise, it will print an error message and terminate with a non-zero exit code.

When the `-a` or `-authenticate-only` command-line option is present, **globusrun** will verify that the service named by *RESOURCE_CONTACT* exists and the client's credentials are granted permission to access that service. If authentication is successful, **globusrun** will display the string "GRAM Authentication test successful" and exit with a zero exit code; otherwise it will print an explanation of the problem and will with a non-zero exit code.

When the `-j` or `-jobmanager-version` command-line option is present, **globusrun** will attempt to determine the software version that the service named by *RESOURCE_CONTACT* is running. If successful, it will display both the Toolkit version and the Job Manager package version and exit with a zero exit code; otherwise, it will print an explanation of the problem and exit with a non-zero exit code.

When the `-k` or `-kill` command-line option is present, **globusrun** will attempt to terminate the job named by *JOB_ID*. If successful, **globusrun** will exit with zero; otherwise it will display an explanation of the problem and exit with a non-zero exit code.

When the `-y` or `-refresh-proxy` command-line option is present, **globusrun** will attempt to delegate a new X.509 proxy to the job manager which is managing the job named by *JOB_ID*. If successful, **globusrun** will exit with zero; otherwise it will display an explanation of the problem and exit with a non-zero exit code. This behavior can be modified by the `-full-proxy` or `-D` command-line options to enable full proxy delegation. The default is limited proxy delegation.

When the `-status` command-line option is present, **globusrun** will attempt to determine the current state of the job. If successful, the state will be printed to standard output and **globusrun** will exit with a zero exit code; otherwise, a description of the error will be displayed and it will exit with a non-zero exit code.

Otherwise, **globusrun** will submit the job to a GRAM service. By default, **globusrun** waits until the job has terminated or failed before exiting, displaying information about job state changes and at exit time, the job exit code if it is provided by the GRAM service.

The **globusrun** program can also function as a GASS file server to allow the **globus-job-manager** program to stage files to and from the machine on which **globusrun** is executed to the GRAM service node. This behavior is controlled by the `-s`, `-o`, and `-w` command-line options.

Jobs submitted by **globusrun** can be monitored interactively or detached. To have **globusrun** detach from the GRAM service after submitting the job, use the `-b` or `-F` command-line options.

Options

The full set of options to **globusrun** consist of:

<code>-help</code>	Display a help message to standard error and exit.
<code>-usage</code>	Display a one-line usage summary to standard error and exit.
<code>-version</code>	Display the software version of globusrun to standard error and exit.
<code>-versions</code>	Display the software version of all modules used by globusrun (including DiRT information) to standard error and then exit.
<code>-p, -parse</code>	Do a parse check on the job specification and print diagnostics. If a parse error occurs, globusrun exits with a non-zero exit code.
<code>-f RSL_FILENAME, -file RSL_FILENAME</code>	Read job specification from the file named by <i>RSL_FILENAME</i> .
<code>-n, -no-interrupt</code>	Disable handling of the SIGINT signal, so that the interrupt character (typically Control-C) causes globusrun to terminate without canceling the job.
<code>-r RESOURCE_CONTACT, -resource RESOURCE_CONTACT</code>	Submit the request to the resource specified by <i>RESOURCE_CONTACT</i> . A resource may be specified in the following ways: <ul style="list-style-type: none"> • <i>HOST</i> • <i>HOST:PORT</i> • <i>HOST:PORT/SERVICE</i> • <i>HOST/SERVICE</i> • <i>HOST:/SERVICE</i> • <i>HOST::SUBJECT</i> • <i>HOST:PORT:SUBJECT</i> • <i>HOST/SERVICE:SUBJECT</i>

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

If any of *PORT*, *SERVICE*, or *SUBJECT* is omitted, the defaults of 2811, jobmanager, and host@*HOST* are used respectively.

-j, -jobmanager-version	Print the software version being run by the service running at <i>RESOURCE_CONTACT</i> .
-k <i>JOB_ID</i> , -kill <i>JOB_ID</i>	Kill the job named by <i>JOB_ID</i>
-D, -full-proxy	Delegate a full impersonation proxy to the service. By default, a limited proxy is delegated when needed.
-y, -refresh-proxy	Delegate a new proxy to the service processing <i>JOB_ID</i> .
-status	Display the current status of the job named by <i>JOB_ID</i> .
-q, -quiet	Do not display job state change or exit code information.
-o, -output-enable	Start a GASS server within the globusrun application that allows access to its standard output and standard error streams only. Also, augment the <i>RSL_SPECIFICATION</i> with a definition of the GLOBUSRUN_GASS_URL RSL substitution and add <code>stdout</code> and <code>stderr</code> clauses which redirect the output and error streams of the job to the output and error streams of the interactive globusrun command. If this is specified, then globusrun acts as though the <code>-q</code> were also specified.
-s, -server	Start a GASS server within the globusrun application that allows access to its standard output and standard error streams for writing and any file local the the globusrun invocation for reading. Also, augment the <i>RSL_SPECIFICATION</i> with a definition of the GLOBUSRUN_GASS_URL RSL substitution and add <code>stdout</code> and <code>stderr</code> clauses which redirect the output and error streams of the job to the output and error streams of the interactive globusrun command. If this is specified, then globusrun acts as though the <code>-q</code> were also specified.
-w, -write-allow	Start a GASS server within the globusrun application that allows access to its standard output and standard error streams for writing and any file local the the globusrun invocation for reading or writing. Also, augment the <i>RSL_SPECIFICATION</i> with a definition of the GLOBUSRUN_GASS_URL RSL substitution and add <code>stdout</code> and <code>stderr</code> clauses which redirect the output and error streams of the job to the output and error streams of the interactive globusrun command. If this is specified, then globusrun acts as though the <code>-q</code> were also specified.
-b, -batch	Terminate after submitting the job to the GRAM service. The globusrun program will exit after the job hits any of the following states: PENDING, ACTIVE, FAILED, or DONE. The GASS-related options can be used to stage input files, but standard output, standard error, and file staging after the job completes will not be processed.
-F, -fast-batch	Terminate after submitting the job to the GRAM service. The globusrun program will exit after it receives a reply from the service. The <i>JOB_ID</i> will be displayed to standard output before terminating so that the job can be checked with the

`-status` command-line option or modified by the `-refresh-proxy` or `-kill` command-line options.

`-d`, `-dryrun`

Submit the job with the `dryrun` attribute set to true. When this is done, the job manager will prepare to start the job but start short of submitting it to the service. This can be used to detect problems with the `RSL_SPECIFICATION`.

Environment

If the following variables affect the execution of **globusrun**

`X509_USER_PROXY` Path to proxy credential.

`X509_CERT_DIR` Path to trusted certificate directory.

Bugs

The **globusrun** program assumes any failure to contact the job means the job has terminated. In fact, this may be due to the **globus-job-manager** program exiting after all jobs it is managing have reached the `DONE` or `FAILED` states. In order to reliably detect job termination, the `two_phase` RSL attribute should be used.

See Also

`globus-job-submit(1)`, `globus-job-run(1)`, `globus-job-clean(1)`, `globus-job-get-output(1)`, `globus-job-cancel(1)`

Name

`globus-job-cancel --` Cancel a GRAM batch job

```
globus-job-cancel [ -f | -force ] [ -q | -quiet ] JOBID  
globus-job-cancel [-help] [-usage] [-version] [-versions]
```

Description

The **globus-job-cancel** program cancels the job named by *JOBID*. Any cached files associated with the job will remain until **globus-job-clean** is executed for the job.

By default, **globus-job-cancel** prompts the user prior to canceling the job. This behavior can be overridden by specifying the `-f` or `-force` command-line options.

Options

The full set of options to **globus-job-cancel** are:

- `-help,` Display a help message to standard error and exit.
- `-usage`

- `-version` Display the software version of the **globus-job-cancel** program to standard output.

- `-version` Display the software version of the **globus-job-cancel** program including DiRT information to standard output.

- `-force,` Do not prompt to confirm job cancel and clean-up.
- `-f`

- `-quiet,` Do not print diagnostics for succesful cancel. Implies `-f`
- `-q`

ENVIRONMENT

If the following variables affect the execution of **globus-job-cancel**.

- `X509_USER_PROXY` Path to proxy credential.

- `X509_CERT_DIR` Path to trusted certificate directory.

Name

globus-job-clean -- Cancel and clean up a GRAM batch job

```
globus-job-clean [ -r RESOURCE | -resource RESOURCE ]  
[ -f | -force ] [ -q | -quiet ] JOBID  
globus-job-clean [-help] [-usage] [-version] [-versions]
```

Description

The **globus-job-clean** program cancels the job named by *JOBID* if it is still running, and then removes any cached files on the GRAM service node related to that job. In order to do the file clean up, it submits a job which removes the cache files. By default this cleanup job is submitted to the default GRAM resource running on the same host as the job. This behavior can be controlled by specifying a resource manager contact string as the parameter to the `-r` or `-resource` option.

By default, **globus-job-clean** prompts the user prior to canceling the job. This behavior can be overridden by specifying the `-f` or `-force` command-line options.

Options

The full set of options to **globus-job-clean** are:

<code>-help, -usage</code>	Display a help message to standard error and exit.
<code>-version</code>	Display the software version of the globus-job-clean program to standard output.
<code>-version</code>	Display the software version of the globus-job-clean program including DiRT information to standard output.
<code>-resource RESOURCE,</code> <code>-r RESOURCE</code>	Submit the clean-up job to the resource named by <i>RESOURCE</i> instead of the default GRAM service on the same host as the job contact.
<code>-force, -f</code>	Do not prompt to confirm job cancel and clean-up.
<code>-quiet, -q</code>	Do not print diagnostics for succesful clean-up. Implies <code>-f</code>

ENVIRONMENT

If the following variables affect the execution of **globus-job-clean**.

<code>X509_USER_PROXY</code>	Path to proxy credential.
<code>X509_CERT_DIR</code>	Path to trusted certificate directory.

Name

`globus-job-get-output --` Retrieve the output and error streams from a GRAM job

```
globus-job-get-output [ -r RESOURCE | -resource RESOURCE ]  
[ -out | -err ] [ -t LINES | -tail LINES ] [ -follow LINES | -f LINES ] JOBID  
globus-job-get-output [-help] [-usage] [-version] [-versions]
```

Description

The **globus-job-get-output** program retrieves the output and error streams of the job named by *JOBID*. By default, **globus-job-get-output** will retrieve all output and error data from the job and display them to its own output and error streams. Other behavior can be controlled by using command-line options. The data retrieval is implemented by submitting another job which simply displays the contents of the first job's output and error streams. By default this retrieval job is submitted to the default GRAM resource running on the same host as the job. This behavior can be controlled by specifying a particular resource manager contact string as the *RESOURCE* parameter to the `-r` or `-resource` option.

Options

The full set of options to **globus-job-get-output** are:

<code>-help, -usage</code>	Display a help message to standard error and exit.
<code>-version</code>	Display the software version of the globus-job-get-output program to standard output.
<code>-version</code>	Display the software version of the globus-job-get-output program including DiRT information to standard output.
<code>-resource <i>RESOURCE</i>,</code> <code>-r <i>RESOURCE</i></code>	Submit the retrieval job to the resource named by <i>RESOURCE</i> instead of the default GRAM service on the same host as the job contact.
<code>-out</code>	Retrieve only the standard output stream of the job. The default is to retrieve both standard output and standard error.
<code>-err</code>	Retrieve only the standard error stream of the job. The default is to retrieve both standard output and standard error.
<code>-tail <i>LINES</i>, -t</code> <code><i>LINES</i></code>	Print only the last <i>LINES</i> count lines of output from the data streams being retrieved. By default, the entire output and error file data is retrieved. This option can not be used along with the <code>-f</code> or <code>-follow</code> options.
<code>-follow <i>LINES</i>, -f</code> <code><i>LINES</i></code>	Print the last <i>LINES</i> count lines of output from the data streams being retrieved and then wait until canceled, printing any subsequent job output that occurs. By default, the entire output and error file data is retrieved. This option can not be used along with the <code>-t</code> or <code>-tail</code> options.

ENVIRONMENT

If the following variables affect the execution of **globus-job-get-output**.

`X509_USER_PROXY` Path to proxy credential.

X509_CERT_DIR Path to trusted certificate directory.

BUGS

The `-f` and `-follow` don't work in GRAM5.

Name

globus-job-run -- Execute a job using GRAM

```
globus-job-run [-dumprsl] [-dryrun] [-verify]
[-file ARGUMENT_FILE]
SERVICE_CONTACT
[ -np PROCESSES | -count PROCESSES ]
[ -m MAX_TIME | -maxtime MAX_TIME ]
[ -p PROJECT | -project PROJECT ]
[ -q QUEUE | -queue QUEUE ]
[ -d DIRECTORY | -directory DIRECTORY ] [-env NAME=VALUE]...
[-stdin [ -l | -s ] STDIN_FILE ] [-stdout [ -l | -s ] STDOUT_FILE ] [-stderr [ -l | -s ] STDERR_FILE ]
[-x RSL_CLAUSE]
[ -l | -s ] EXECUTABLE [ARGUMENT...]
globus-job-run [-help] [-usage] [-version] [-versions]
```

Description

The **globus-job-run** program constructs a job description from its command-line options and then submits the job to the GRAM service running at *SERVICE_CONTACT*. The executable and arguments to the executable are provided on the command-line after all other options. Note that the `-dumprsl`, `-dryrun`, `-verify`, and `-file` command-line options must occur before the first non-option argument, the *SERVICE_CONTACT*.

The **globus-job-run** provides similar functionality to **globusrun** in that it allows interactive start-up of GRAM jobs. However, unlike **globusrun**, it uses command-line parameters to define the job instead of RSL expressions.

Options

The full set of options to **globus-job-run** are:

<code>-help, -usage</code>	Display a help message to standard error and exit.
<code>-version</code>	Display the software version of the globus-job-run program to standard output.
<code>-version</code>	Display the software version of the globus-job-run program including DiRT information to standard output.
<code>-dumprsl</code>	Translate the command-line options to globus-job-run into an RSL expression that can be used with tools such as globusrun .
<code>-dryrun</code>	Submit the job request to the GRAM service with the <code>dryrun</code> option enabled. When this option is used, the GRAM service prepares to execute the job but stops before submitting the job to the LRM. This can be used to diagnose some problems such as missing files.
<code>-verify</code>	Submit the job request to the GRAM service with the <code>dryrun</code> option enabled and then without it enabled if the <code>dryrun</code> is successful.
<code>-file ARGUMENT_FILE</code>	Read additional command-line options from <i>ARGUMENT_FILE</i> .
<code>-np PROCESSES, -count PRO- CESSES</code>	Start <i>PROCESSES</i> instances of the executable as a single job.

<p><code>-m MAX_TIME, -maxtime MAX_TIME</code></p>	<p>Schedule the job to run for a maximum of <i>MAX_TIME</i> minutes.</p>
<p><code>-p PROJECT, -project PRO- JECT</code></p>	<p>Request that the job use the allocation <i>PROJECT</i> when submitting the job to the LRM.</p>
<p><code>-q QUEUE, -queue QUEUE</code></p>	<p>Request that the job be submitted to the LRM using the named <i>QUEUE</i>.</p>
<p><code>-d DIRECTORY, -directory DIRECTORY</code></p>	<p>Run the job in the directory named by <i>DIRECTORY</i>. Input and output files will be interpreted relative to this directory. This directory must exist on the file system on the LRM-managed resource. If not specified, the job will run in the home directory of the user the job is running as.</p>
<p><code>-env NAME=VALUE</code></p>	<p>Define an environment variable named by <i>NAME</i> with the value <i>VALUE</i> in the job environment. This option may be specified multiple times to define multiple environment variables.</p>
<p><code>-stdin [-l -s] STDIN_FILE</code></p>	<p>Use the file named by <i>STDIN_FILE</i> as the standard input of the job. If the <code>-l</code> option is specified, then this file is interpreted to be on a file system local to the LRM. If the <code>-s</code> option is specified, then this file is interpreted to be on the file system where globus-job-run is being executed, and the file will be staged via GASS. If neither is specified, the local behavior is assumed.</p>
<p><code>-stdout [-l -s] STDOUT_FILE</code></p>	<p>Use the file named by <i>STDOUT_FILE</i> as the destination for the standard output of the job. If the <code>-l</code> option is specified, then this file is interpreted to be on a file system local to the LRM. If the <code>-s</code> option is specified, then this file is interpreted to be on the file system where globus-job-run is being executed, and the file will be staged via GASS. If neither is specified, the local behavior is assumed.</p>
<p><code>-stderr [-l -s] STDERR_FILE</code></p>	<p>Use the file named by <i>STDERR_FILE</i> as the destination for the standard error of the job. If the <code>-l</code> option is specified, then this file is interpreted to be on a file system local to the LRM. If the <code>-s</code> option is specified, then this file is interpreted to be on the file system where globus-job-run is being executed, and the file will be staged via GASS. If neither is specified, the local behavior is assumed.</p>
<p><code>-x RSL_CLAUSE</code></p>	<p>Add a set of custom RSL attributes described by <i>RSL_CLAUSE</i> to the job description. The clause must be an RSL conjunction and may contain one or more attributes. This can be used to include attributes which can not be defined by other command-line options of globus-job-run.</p>
<p><code>-l</code></p>	<p>When included outside the context of <code>-stdin</code>, <code>-stdout</code>, or <code>-stderr</code> command-line options, <code>-l</code> option alters the interpretation of the executable path. If the <code>-l</code> option is specified, then the executable is interpreted to be on a file system local to the LRM.</p>
<p><code>-s</code></p>	<p>When included outside the context of <code>-stdin</code>, <code>-stdout</code>, or <code>-stderr</code> command-line options, <code>-s</code> option alters the interpretation of the executable path. If the <code>-s</code> option is specified, then the executable is interpreted to be on the file system where globus-job-run is being executed, and the file will be staged via GASS. If neither is specified, the local behavior is assumed.</p>

ENVIRONMENT

If the following variables affect the execution of **globus-job-run**.

X509_USER_PROXY Path to proxy credential.

X509_CERT_DIR Path to trusted certificate directory.

See Also

globusrun(1), globus-job-submit(1), globus-job-clean(1), globus-job-get-output(1), globus-job-cancel(1)

Name

`globus-job-status` -- Check the status of a GRAM5 job

`globus-job-status` *JOBID*

`globus-job-status` [-help] [-usage] [-version] [-versions]

Description

The **globus-job-status** program checks the status of a GRAM job by sending a status request to the job manager contact for that job specified by the *JOBID* parameter. If successful, it will print the job status to standard output. The states supported by **globus-job-status** are:

PENDING	The job has been submitted to the LRM but has not yet begun execution.
ACTIVE	The job has begun execution.
FAILED	The job has failed.
SUSPENDED	The job is currently suspended by the LRM.
DONE	The job has completed.
UNSUBMITTED	The job has been accepted by GRAM, but not yet submitted to the LRM.
STAGE_IN	The job has been accepted by GRAM and is currently staging files prior to being submitted to the LRM.
STAGE_OUT	The job has completed execution and is currently staging files from the service node to other http, GASS, or GridFTP servers.

Options

The full set of options to **globus-job-status** are:

`-help, -usage` Display a help message to standard error and exit.

`-version` Display the software version of the **globus-job-status** program to standard output.

`-versions` Display the software version of the **globus-job-status** program including DiRT information to standard output.

ENVIRONMENT

If the following variables affect the execution of **globus-job-status**.

`X509_USER_PROXY` Path to proxy credential.

`X509_CERT_DIR` Path to trusted certificate directory.

Bugs

The **globus-job-status** program can not distinguish between the case of the job manager terminating for any reason and the job being in the DONE state.

See Also

globusrun(1)

Name

globus-job-submit -- Submit a batch job using GRAM

```
globus-job-submit [-dumprsl] [-dryrun] [-verify]
[-file ARGUMENT_FILE]
SERVICE_CONTACT
[ -np PROCESSES | -count PROCESSES ]
[ -m MAX_TIME | -maxtime MAX_TIME ]
[ -p PROJECT | -project PROJECT ]
[ -q QUEUE | -queue QUEUE ]
[ -d DIRECTORY | -directory DIRECTORY ] [-env NAME=VALUE]...
[-stdin [ -l | -s ] STDIN_FILE ] [-stdout [ -l | -s ] STDOUT_FILE ] [-stderr [ -l | -s ] STDERR_FILE ]
[-x RSL_CLAUSE]
[ -l | -s ] EXECUTABLE [ARGUMENT...]
globus-job-submit [-help] [-usage] [-version] [-versions]
```

Description

The **globus-job-submit** program constructs a job description from its command-line options and then submits the job to the GRAM service running at *SERVICE_CONTACT*. The executable and arguments to the executable are provided on the command-line after all other options. Note that the `-dumprsl`, `-dryrun`, `-verify`, and `-file` command-line options must occur before the first non-option argument, the *SERVICE_CONTACT*.

The **globus-job-submit** provides similar functionality to **globusrun** in that it allows batch submission of GRAM jobs. However, unlike **globusrun**, it uses command-line parameters to define the job instead of RSL expressions.

To retrieve the output and error streams of the job, use the program **globus-job-get-output**. To reclaim resources used by the job by deleting cached files and job state, use the program **globus-job-clean**. To cancel a batch job submitted by **globus-job-submit**, use the program **globus-job-cancel**.

Options

The full set of options to **globus-job-submit** are:

<code>-help, -usage</code>	Display a help message to standard error and exit.
<code>-version</code>	Display the software version of the globus-job-submit program to standard output.
<code>-versions</code>	Display the software version of the globus-job-submit program including DiRT information to standard output.
<code>-dumprsl</code>	Translate the command-line options to globus-job-submit into an RSL expression that can be used with tools such as globusrun .
<code>-dryrun</code>	Submit the job request to the GRAM service with the <code>dryrun</code> option enabled. When this option is used, the GRAM service prepares to execute the job but stops before submitting the job to the LRM. This can be used to diagnose some problems such as missing files.
<code>-verify</code>	Submit the job request to the GRAM service with the <code>dryrun</code> option enabled and then without it enabled if the <code>dryrun</code> is successful.

<code>-file ARGUMENT_FILE</code>	Read additional command-line options from <i>ARGUMENT_FILE</i> .
<code>-np PROCESSES, -count PROCESSES</code>	Start <i>PROCESSES</i> instances of the executable as a single job.
<code>-m MAX_TIME, -maxtime MAX_TIME</code>	Schedule the job to run for a maximum of <i>MAX_TIME</i> minutes.
<code>-p PROJECT, -project PROJECT</code>	Request that the job use the allocation <i>PROJECT</i> when submitting the job to the LRM.
<code>-q QUEUE, -queue QUEUE</code>	Request that the job be submitted to the LRM using the named <i>QUEUE</i> .
<code>-d DIRECTORY, -directory DIRECTORY</code>	Run the job in the directory named by <i>DIRECTORY</i> . Input and output files will be interpreted relative to this directory. This directory must exist on the file system on the LRM-managed resource. If not specified, the job will run in the home directory of the user the job is running as.
<code>-env NAME=VALUE</code>	Define an environment variable named by <i>NAME</i> with the value <i>VALUE</i> in the job environment. This option may be specified multiple times to define multiple environment variables.
<code>-stdin [-l -s] STDIN_FILE</code>	Use the file named by <i>STDIN_FILE</i> as the standard input of the job. If the <code>-l</code> option is specified, then this file is interpreted to be on a file system local to the LRM. If the <code>-s</code> option is specified, then this file is interpreted to be on the file system where globus-job-submit is being executed, and the file will be staged via GASS. If neither is specified, the local behavior is assumed.
<code>-stdout [-l -s] STDOUT_FILE</code>	Use the file named by <i>STDOUT_FILE</i> as the destination for the standard output of the job. If the <code>-l</code> option is specified, then this file is interpreted to be on a file system local to the LRM. If the <code>-s</code> option is specified, then this file is interpreted to be on the file system where globus-job-submit is being executed, and the file will be staged via GASS. If neither is specified, the local behavior is assumed.
<code>-stderr [-l -s] STDERR_FILE</code>	Use the file named by <i>STDERR_FILE</i> as the destination for the standard error of the job. If the <code>-l</code> option is specified, then this file is interpreted to be on a file system local to the LRM. If the <code>-s</code> option is specified, then this file is interpreted to be on the file system where globus-job-submit is being executed, and the file will be staged via GASS. If neither is specified, the local behavior is assumed.
<code>-x RSL_CLAUSE</code>	Add a set of custom RSL attributes described by <i>RSL_CLAUSE</i> to the job description. The clause must be an RSL conjunction and may contain one or more attributes. This can be used to include attributes which can not be defined by other command-line options of globus-job-submit .
<code>-l</code>	When included outside the context of <code>-stdin</code> , <code>-stdout</code> , or <code>-stderr</code> command-line options, <code>-l</code> option alters the interpretation of the executable path. If the <code>-l</code> option is specified, then the executable is interpreted to be on a file system local to the LRM.
<code>-s</code>	When included outside the context of <code>-stdin</code> , <code>-stdout</code> , or <code>-stderr</code> command-line options, <code>-s</code> option alters the interpretation of the executable path. If the <code>-s</code> option is specified, then the executable is interpreted to be on the file

system where **globus-job-run** is being executed, and the file will be staged via GASS. If neither is specified, the local behavior is assumed.

ENVIRONMENT

If the following variables affect the execution of **globus-job-submit**.

X509_USER_PROXY Path to proxy credential.

X509_CERT_DIR Path to trusted certificate directory.

See Also

globusrun(1), globus-job-run(1), globus-job-clean(1), globus-job-get-output(1), globus-job-cancel(1)

Name

`globus-personal-gatekeeper --` Manage a user's personal gatekeeper daemon

```
globus-personal-gatekeeper [-help] [-usage] [-version] [-versions] [-list] [-directory CONTACT]
globus-personal-gatekeeper [-debug] {-start} [-jmtime LRM] [-auditdir AUDIT_DIRECTORY] [-port PORT]
[-log [=DIRECTORY]] [-seg] [-acctfile ACCOUNTING_FILE]
globus-personal-gatekeeper [-killall] [-kill]
```

Description

The **globus-personal-gatekeeper** command is a utility which manages a gatekeeper and job manager service for a single user. Depending on the command-line arguments it will operate in one of several modes. In the first set of arguments indicated in the synopsis, the program provides information about the **globus-personal-gatekeeper** command or about instances of the **globus-personal-gatekeeper** that are running currently. The second set of arguments indicated in the synopsis provide control over starting a new **globus-personal-gatekeeper** instance. The final set of arguments provide control for terminating one or more **globus-personal-gatekeeper** instances.

The `-start` mode will create a new subdirectory of `$HOME/.globus` and write the configuration files needed to start a **globus-gatekeeper** daemon which will invoke the **globus-job-manager** service when new authenticated connections are made to its service port. The **globus-personal-gatekeeper** then exits, printing the contact string for the new gatekeeper prefixed by `GRAM contact:` to standard output. In addition to the arguments described above, any arguments described in **globus-job-manager(8)** can be appended to the command-line and will be added to the job manager configuration for the service started by the **globus-gatekeeper**.

The new **globus-gatekeeper** will continue to run in the background until killed by invoking **globus-personal-gatekeeper** with the `-kill` or `-killall` argument. When killed, it will kill the **globus-gatekeeper** and **globus-job-manager** processes, remove state files and configuration data, and then exit. Jobs which are running when the personal gatekeeper is killed will continue to run, but their job directory will be destroyed so they may fail in the LRM.

The full set of command-line options to **globus-personal-gatekeeper** consists of:

<code>-help, -usage</code>	Print command-line option summary and exit
<code>-version</code>	Print software version
<code>-versions</code>	Print software version including DiRT information
<code>-list</code>	Print a list of all currently running personal gatekeepers. These entries will be printed one per line.
<code>-directory CONTACT</code>	Print the configuration directory for the personal gatekeeper with the contact string <i>CONTACT</i> .
<code>-debug</code>	Print additional debugging information when starting a personal gatekeeper. This option is ignored in other modes.
<code>-start</code>	Start a new personal gatekeeper process.
<code>-jmtime LRM</code>	Use <i>LRM</i> as the local resource manager interface. If not provided when starting a personal gatekeeper, the job manager will use the default <code>fork</code> LRM.
<code>-auditdir AUDIT_DIRECTORY</code>	Write audit report files to <i>AUDIT_DIRECTORY</i> . If not provided, the job manager will not write any audit files.

- `-port PORT` Listen for gatekeeper TCP/IP connections on the port *PORT*. If not provided, the gatekeeper will let the operating system choose.
- `-log[=DIRECTORY]` Write job manager log files to *DIRECTORY*. If *DIRECTORY* is omitted, the default of `$HOME` will be used. If this option is not present, the job manager will not write any log files.
- `-seg` Try to use the SEG mechanism to receive job state change information, instead of polling for these. These require either the system administrator or the user to run an instance of the **globus-job-manager-event-generator** program for the LRM specified by the `-jmtime` option.
- `-acctfile ACCOUNTING_FILE` Write gatekeeper accounting entries to *ACCOUNTING_FILE*. If not provided, no accounting records are written.

Examples

This example shows the output when starting a new personal gatekeeper which will schedule jobs via the lsf LRM, with debugging enabled.

```
% globus-personal-gatekeeper -start -jmtime lsf
```

```
verifying setup...
```

```
done.
```

```
GRAM contact: personal-grid.example.org:57846:/DC=org/DC=example/CN=Joe User
```

This example shows the output when listing the current active personal gatekeepers.

```
% globus-personal-gatekeeper -list
```

```
personal-grid.example.org:57846:/DC=org/DC=example/CN=Joe User
```

This example shows the output when querying the configuration directory for the above personal gatekeeper. gatekeepers.

```
% globus-personal-gatekeeper -directory "personal-grid.example.org:57846:/DC=org/DC=example/CN=Joe User"
```

```
/home/juser/.globus/.personal-gatekeeper.personal-grid.example.org.1337
```

```
% globus-personal-gatekeeper -kill "personal-grid.example.org:57846:/DC=org/DC=example/CN=Joe User"
```

```
killing gatekeeper: "personal-grid.example.org:57846:/DC=org/DC=example/CN=Joe User"
```

See Also

globusrun(1), globus-job-manager(8), globus-gatekeeper(8)

Name

globus-gram-audit -- Load GRAM4 and GRAM5 audit records into a database

globus-gram-audit [--conf *CONFIG_FILE*] [--check] [--delete] [--audit-directory *AUDITDIR*]

Description

The **globus-gram-audit** program loads audit records to an SQL-based database. It reads `$GLOBUS_LOCATION/etc/globus-job-manager.conf` by default to determine the audit directory and then uploads all files in that directory that contain valid audit records to the database configured by the **globus_gram_job_manager_auditing_setup_scripts** package. If the upload completes successfully, the audit files will be removed.

The full set of command-line options to **globus-gram-audit** consist of:

<code>--conf <i>CONFIG_FILE</i></code>	Use <i>CONFIG_FILE</i> instead of the default from the configuration file for audit database configuration.
<code>--check</code>	Check whether the insertion of a record was successful by querying the database after inserting the records. This is used in tests.
<code>--delete</code>	Delete audit records from the database right after inserting them. This is used in tests to avoid filling the database with test records.
<code>--audit-directory <i>DIR</i></code>	Look for audit records in <i>DIR</i> , instead of looking in the directory specified in the job manager configuration. This is used in tests to control which records are loaded to the database and then deleted.
<code>--query <i>SQL</i></code>	Perform the given SQL query on the audit database. This uses the database information from the configuration file to determine how to contact the database.

FILES

The **globus-gram-audit** uses the following files (paths relative to `$GLOBUS_LOCATION`).

<code>etc/globus-gram-job-manager.conf</code>	GRAM5 job manager configuration. It includes the default path to the audit directory
<code>etc/globus-gram-audit.conf</code>	Audit configuration. It includes the information needed to contact the audit database.

Name

globus-gatekeeper -- Authorize and execute a grid service on behalf of a user

```
globus-gatekeeper [-help]
[-conf PARAMETER_FILE]
[-test] [ -d | -debug ]
{ -inetd | -f }
[ -p PORT | -port PORT ]
[-home PATH] [ -l LOGFILE | -logfile LOGFILE ]
[-acctfile ACCTFILE]
[-e LIBEXECDIR]
[-launch_method { fork_and_exit | fork_and_wait | dont_fork } ]
[-grid_services SERVICEDIR]
[-globusid GLOBUSID]
[-gridmap GRIDMAP]
[-x509_cert_dir TRUSTED_CERT_DIR]
[-x509_cert_file TRUSTED_CERT_FILE]
[-x509_user_cert CERT_PATH]
[-x509_user_key KEY_PATH]
[-x509_user_proxy PROXY_PATH]
[-k]
[-globuskmap KMAP]
```

Description

The **globus-gatekeeper** program is a meta-server similar to **inetd** or **xinetd** that starts other services after authenticating the TCP connection using GSSAPI.

The most common use for the **globus-gatekeeper** program is to start instances of the globus-job-manager(8) service. A single **globus-gatekeeper** deployment can handle multiple different service configurations by having entries in the grid-services directory.

Typically, users interact with the **globus-gatekeeper** program via client applications such as globusrun(1), **globus-job-submit**, or tools such as CoG jglobus or Condor-G.

The full set of command-line options to **globus-gatekeeper** consists of:

-help	Display a help message to standard error and exit
-conf <i>PARAMETER_FILE</i>	Load configuration parameters from <i>PARAMETER_FILE</i> . The parameters in that file are treated as additional command-line options.
-test	Parse the configuration file and print out the POSIX user id of the globus-gatekeeper process, service home directory, service execution directory, and X.509 subject name and then exits.
-d, -debug	Run the globus-gatekeeper process in the foreground.
-inetd	Flag to indicate that the globus-gatekeeper process was started via inetd or a similar super-server. If this flag is set and the globus-gatekeeper was not started via inetd, a warning will be printed in the gatekeeper log.

- f Flag to indicate that the **globus-gatekeeper** process should run in the foreground. This flag has no effect when the **globus-gatekeeper** is started via inetd.
- p *PORT*, -port *PORT* Listen for connections on the TCP/IP port *PORT*. This option has no effect if the **globus-gatekeeper** is started via inetd or a similar service. If not specified and the gatekeeper is running as root, the default of 754 is used. Otherwise, the gatekeeper defaults to an ephemeral port.
- home *PATH* Sets the gatekeeper deployment directory to *PATH*. This is used to interpret relative paths for accounting files, libexecdir, certificate paths, and also to set the GLOBUS_LOCATION environment variable in the service environment. If not specified, the gatekeeper uses its working directory.
- l *LOGFILE*, -logfile *LOGFILE* Write status log entries to *LOGFILE*
- acctfile *ACCTFILE* Set the path to write accounting records to *ACCTFILE*. If not set, no accounting records will be written.
- e *LIBEXECDIR* Look for service executables in *LIBEXECDIR*. If not specified, the default of *HOME/libexec* is used.
- launch_method *fork_and_exit*, *fork_and_wait*, *fork*, *exec*, *execv*, *execvp*, *execve* Determine how to launch services. The method may be either *fork_and_exit* (the service runs completely independently of the gatekeeper, which exits after creating the new service process), *fork_and_wait* (the service is run in a separate process from the gatekeeper but the gatekeeper does not exit until the service terminates), or *dont_fork*, where the gatekeeper process becomes the service process via the *exec()* system call.
- grid_services *SERVICEDIR* Look for service descriptions in *SERVICEDIR*. If this is a relative path, it is interpreted relative to the *HOME* value. If this is not specified, the default of *HOME/etc/grid-services* is used.
- globusid *GLOBUSID* Sets the GLOBUSID environment variable to *GLOBUSID*. This variable is used to construct the gatekeeper contact string if it can not be parsed from the service credential.
- gridmap *GRIDMAP* Use the file at *GRIDMAP* to map GSSAPI names to POSIX user names. If not specified, the default of *HOME/etc/grid-mapfile* is used.
- x509_cert_dir *TRUSTED_CERT_DIR* Use the directory *TRUSTED_CERT_DIR* to locate trusted CA X.509 certificates. The gatekeeper sets the environment variable *X509_CERT_DIR* to this value.
- x509_cert_file *TRUSTED_CERT_FILE* OBSOLETE GSI OPTION
- x509_user_cert *CERT_PATH* Read the service X.509 certificate from *CERT_PATH*. The gatekeeper sets the *X509_USER_CERT* environment variable to this value.
- x509_user_key *KEY_PATH* Read the private key for the service from *KEY_PATH*. The gatekeeper sets the *X509_USER_KEY* environment variable to this value.
- x509_user_proxy *PROXY_PATH* Read the X.509 proxy certificate from *PROXY_PATH*. The gatekeeper sets the *X509_USER_PROXY* environment variable to this value.
- k Assume authentication with Kerberos 5 GSSAPI instead of X.509 GSSAPI.

`-globusmap KMAP` Assume authentication with Kerberos 5 GSSAPI instead of X.509 GSSAPI and use *KMAP* as the path to the kerberos principal to POSIX user mapping file.

ENVIRONMENT

If the following variables affect the execution of **globus-gatekeeper**

`X509_CERT_DIR` Directory containing X.509 trust anchors and signing policy files.

`X509_USER_PROXY` Path to file containing an X.509 proxy.

`X509_USER_CERT` Path to file containing an X.509 user certificate.

`X509_USER_KEY` Path to file containing an X.509 user key.

Files

`$GLOBUS_LOCATION/etc/globus-gatekeeper.conf` Default path to gatekeeper configuration file.

`$GLOBUS_LOCATION/etc/grid-services/SERVICENAME` Service configuration for *SERVICENAME*.

See also

`globusrun(1)`, `globus-job-manager(8)`

Name

globus-job-manager -- Execute and monitor jobs

```
globus-job-manager {-type LRM} [-conf CONFIG_PATH] [-help] [-globus-host-manufacturer MANUFACTURER]
[-globus-host-cputype CPUTYPE] [-globus-host-osname OSNAME] [-globus-host-osversion OSVERSION] [-globus-
gatekeeper-host HOST] [-globus-gatekeeper-port PORT] [-globus-gatekeeper-subject SUBJECT] [-home GLOBUS_LOC-
ATION] [-target-globus-location TARGET_GLOBUS_LOCATION] [-condor-arch ARCH] [-condor-os OS] [-history
HISTORY_DIRECTORY] [-scratch-dir-base SCRATCH_DIRECTORY] [-enable-syslog] [-stdio-log LOG_DIRECTORY]
[-log-levels LEVELS] [-state-file-dir STATE_DIRECTORY] [-globus-tcp-port-range PORT_RANGE] [-x509-cert-dir
TRUSTED_CERTIFICATE_DIRECTORY] [-cache-location GASS_CACHE_DIRECTORY] [-k] [-extra-envvars
VAR=VAL, . . .] [-seg-module SEG_MODULE] [-audit-directory AUDIT_DIRECTORY] [-globus-toolkit-version
TOOLKIT_VERSION] [-disable-streaming] [-disable-usagestats] [-usagestats-target TARGET] [-service-tag SER-
VICE_TAG]
```

Description

The **globus-job-manager** program is a service which starts and controls GRAM jobs which are executed by a local resource management system, such as LSF or Condor. The **globus-job-manager** program is typically started by the **globus-gatekeeper** program and not directly by a user. It runs until all jobs it is managing have terminated or its delegated credentials have expired.

Typically, users interact with the **globus-job-manager** program via client applications such as **globusrun**, **globus-job-submit**, or tools such as CoG jglobus or Condor-G.

The full set of command-line options to **globus-job-manager** consists of:

-help	Display a help message to standard error and exit
-type <i>LRM</i>	Execute jobs using the local resource manager named <i>LRM</i> .
-conf <i>CONFIG_PATH</i>	Read additional command-line arguments from the file <i>CONFIG_PATH</i> . If present, this must be the first command-line argument to the globus-job-manager program.
-globus-host-manufacturer <i>MANUFACTURER</i>	Indicate the manufacturer of the system which the jobs will execute on. This parameter sets the value of the \$(GLOBUS_HOST_MANUFACTURER) RSL substitution to <i>MANUFACTURER</i>
-globus-host-cputype <i>CPU-TYPE</i>	Indicate the CPU type of the system which the jobs will execute on. This parameter sets the value of the \$(GLOBUS_HOST_CPUTYPE) RSL substitution to <i>CPUTYPE</i>
-globus-host-osname <i>OS-NAME</i>	Indicate the operating system type of the system which the jobs will execute on. This parameter sets the value of the \$(GLOBUS_HOST_OSNAME) RSL substitution to <i>OSNAME</i>
-globus-host-osversion <i>OSVERSION</i>	Indicate the operating system version of the system which the jobs will execute on. This parameter sets the value of the \$(GLOBUS_HOST_OSVERSION) RSL substitution to <i>OSVERSION</i>
-globus-gatekeeper-host <i>HOST</i>	Indicate the host name of the machine which the job was submitted to. This parameter sets the value of the \$(GLOBUS_GATEKEEPER_HOST) RSL substitution to <i>HOST</i>

-globus-gatekeeper-port <i>PORT</i>	Indicate the TCP port number of gatekeeper to which jobs are submitted to. This parameter sets the value of the $\$(GLOBUS_GATEKEEPER_PORT)$ RSL substitution to <i>PORT</i>
-globus-gatekeeper-subject <i>SUBJECT</i>	Indicate the X.509 identity of the gatekeeper to which jobs are submitted to. This parameter sets the value of the $\$(GLOBUS_GATEKEEPER_SUBJECT)$ RSL substitution to <i>SUBJECT</i>
-home <i>GLOBUS_LOCATION</i>	Indicate the path where the Globus Toolkit(r) is installed on the service node. This is used by the job manager to locate its support and configuration files.
-target-globus-location <i>TARGET_GLOBUS_LOCATION</i>	Indicate the path where the Globus Toolkit(r) is installed on the execution host. If this is omitted, the value specified as a parameter to -home is used. This parameter sets the value of the $\$(GLOBUS_LOCATION)$ RSL substitution to <i>TARGET_GLOBUS_LOCATION</i>
-history <i>HISTORY_DIRECTORY</i>	Configure the job manager to write job history files to <i>HISTORY_DIRECTORY</i> . These files are described in the FILES section below.
-scratch-dir-base <i>SCRATCH_DIRECTORY</i>	Configure the job manager to use <i>SCRATCH_DIRECTORY</i> as the default scratch directory root if a relative path is specified in the job RSL's <i>scratch_dir</i> attribute.
-enable-syslog	Configure the job manager to write log messages via syslog. Logging is further controlled by the argument to the -log-levels parameter described below.
-stdio-log <i>LOG_DIRECTORY</i>	Configure the job manager to write log messages to files in the <i>LOG_DIRECTORY</i> directory. Files will be named <i>LOG_DIRECTORY/gram_YYYYMM-DD.log</i> . Logging is further controlled by the argument to the -log-levels parameter described below. The <i>LOG_DIRECTORY</i> value can include variables derived from the job manager environment using the same syntax as RSL substitutions. For example, -stdio-log $\$(HOME)$ would cause each user's logs to be stored in their individual home directories.
-log-levels <i>LEVELS</i>	Configure the job manager to write log messages of certain levels to syslog and/or log files. The available log levels are FATAL, ERROR, WARN, INFO, DEBUG, and TRACE. Multiple values can be combined with the character. The default value of logging when enabled is FATAL ERROR.
-state-file-dir <i>STATE_DIRECTORY</i>	Configure the job manager to write state files to <i>STATE_DIRECTORY</i> . If not specified, the job manager uses the default of $\$GLOBUS_LOCATION/tmp/gram_job_state/$. This directory must be writable by all users and be on a file system which supports POSIX advisory file locks.
-globus-tcp-port-range <i>PORT_RANGE</i>	Configure the job manager to restrict its TCP/IP communication to use ports in the range described by <i>PORT_RANGE</i> . This value is also made available in the job environment via the <i>GLOBUS_TCP_PORT_RANGE</i> environment variable.
-x509-cert-dir <i>TRUSTED_CERTIFICATE_DIRECTORY</i>	Configure the job manager to search <i>TRUSTED_CERTIFICATE_DIRECTORY</i> for its list of trusted CA certificates and their signing policies. This value is also made available in the job environment via the <i>X509_CERT_DIR</i> environment variable.

<code>-cache-location</code> <code>GASS_CACHE_DIRECTORY</code>	Configure the job manager to use the path <code>GASS_CACHE_DIRECTORY</code> for its temporary GASS-cache files. This value is also made available in the job environment via the <code>GLOBUS_GASS_CACHE_DEFAULT</code> environment variable.
<code>-k</code>	Configure the job manager to assume it is using Kerberos for authentication instead of X.509 certificates. This disables some certificate-specific processing in the job manager.
<code>-extra-envvars</code> <code>VAR=VAL,...</code>	Configure the job manager to define a set of environment variables in the job environment beyond those defined in the base job environment. The format of the parameter to this argument is a comma-separated sequence of <code>VAR=VAL</code> pairs, where <code>VAR</code> is the variable name and <code>VAL</code> is the variables value.
<code>-seg-module</code> <code>SEG_MODULE</code>	Configure the job manager to use the schedule event generator module named by <code>SEG_MODULE</code> to detect job state changes events from the local resource manager, in place of the less efficient polling operations used in GT2. To use this, one instance of the globus-job-manager-event-generator must be running to process events for the LRM into a generic format that the job manager can parse.
<code>-audit-directory</code> <code>AUDIT_DIRECTORY</code>	Configure the job manager to write audit records to the directory named by <code>AUDIT_DIRECTORY</code> . This records can be loaded into a database using the globus-gram-audit program.
<code>-globus-toolkit-version</code> <code>TOOLKIT_VERSION</code>	Configure the job manager to use <code>TOOLKIT_VERSION</code> as the version for audit and usage stats records.
<code>-service-tag</code> <code>SERVICE_TAG</code>	Configure the job manager to use <code>SERVICE_TAG</code> as a unique identifier to allow multiple GRAM instances to use the same job state directories without interfering with each other's jobs. If not set, the value <code>untagged</code> will be used.
<code>-disable-streaming</code>	Configure the job manager to disable file streaming. This is propagated to the LRM script interface but has no effect in GRAM5.
<code>-disable-usagestats</code>	Disable sending of any usage stats data, even if <code>-usagestats-target</code> is present in the configuration.
<code>-usagestats-target</code> <code>TARGET</code>	Send usage packets to a data collection service for analysis. The <code>TARGET</code> string consists of a comma-separated list of <code>HOST:PORT</code> combinations, each containing an optional list of data to send. See Usage Stats Packets ¹ for more information about the tags. Special tag strings of <code>all</code> (which enables all tags) and <code>default</code> may be used, or a sequence of characters for the various tags.
<code>-condor-arch</code> <code>ARCH</code>	Set the architecture specification for condor jobs to be <code>ARCH</code> in job classified ads generated by the GRAM5 condor LRM script. This is required for the condor LRM but ignored for all others.
<code>-condor-os</code> <code>OS</code>	Set the operating system specification for condor jobs to be <code>OS</code> in job classified ads generated by the GRAM5 condor LRM script. This is required for the condor LRM but ignored for all others.

¹ <http://confluence.globus.org/display/~bester/GRAM5+Usage+Stats+Packets>

Environment

If the following variables affect the execution of **globus-job-manager**

HOME	User's home directory.
LOGNAME	User's name.
JOBMANAGER_SYSLOG_ID	String to prepend to syslog audit messages.
JOBMANAGER_SYSLOG_FAC	Facility to log syslog audit messages as.
JOBMANAGER_SYSLOG_LVL	Priority level to use for syslog audit messages.
GATEKEEPER_JM_ID	Job manager ID to be used in syslog audit records.
GATEKEEPER_PEER	Peer information to be used in syslog audit records
GLOBUS_ID	Credential information to be used in syslog audit records
GLOBUS_JOB_MANAGER_SLEEP	Time (in seconds) to sleep when the job manager is started. [For debugging purposes only]
GRID_SECURITY_HOST- TP_BODY_FD	File descriptor of an open file which contains the initial job request and to which the initial job reply should be sent. This file descriptor is inherited from the globus-gatekeeper .
X509_USER_PROXY	Path to the X.509 user proxy which was delegated by the client to the globus-gatekeeper program to be used by the job manager.
GRID_SECURITY_CONTEXT_FD	File descriptor containing an exported security context that the job manager should use to reply to the client which submitted the job.

Files

<code>\$HOME/.globus/job/HOST- NAME/LRM.TAG.red</code>	Job manager delegated user credential.
<code>\$HOME/.globus/job/HOST- NAME/LRM.TAG.lock</code>	Job manager state lock file.
<code>\$HOME/.globus/job/HOST- NAME/LRM.TAG.pid</code>	Job manager pid file.
<code>\$HOME/.globus/job/HOST- NAME/LRM.TAG.sock</code>	Job manager socket for inter-job manager communications.
<code>\$HOME/.globus/job/HOST- NAME/JOB_ID/</code>	Job-specific state directory.
<code>\$HOME/.globus/job/HOST- NAME/JOB_ID/stdin</code>	Standard input which has been staged from a remote URL.
<code>\$HOME/.globus/job/HOST- NAME/JOB_ID/stdout</code>	Standard output which will be staged from a remote URL.

<code>\$HOME/.globus/job/HOST-NAME/JOB_ID/stderr</code>	Standard error which will be staged from a remote URL.
<code>\$HOME/.globus/job/HOST-NAME/JOB_ID/x509_user_proxy</code>	Job-specific delegated credential.
<code>\$GLOBUS_LOCATION/tmp/gram_job_state/job.HOST-NAME.JOB_ID</code>	Job state file.
<code>\$GLOBUS_LOCATION/tmp/gram_job_state/job.HOST-NAME.JOB_ID.lock</code>	Job state lock file. In most cases this will be a symlink to the job manager lock file.
<code>\$GLOBUS_LOCATION/etc/globus-job-manager.conf</code>	Default location of the global job manager configuration file.
<code>\$GLOBUS_LOCATION/etc/grid-services/jobmanager-LRM</code>	Default location of the LRM-specific gatekeeper configuration file.

See Also

`globusrun(1)`, `globus-gatekeeper(8)`, `globus-personal-gatekeeper(1)`, `globus-gram-audit(8)`

Name

globus-job-manager-event-generator -- Create LRM-independent SEG files for the job manager to use

globus-job-manager-event-generator [-help] {-scheduler *LRM*} [-background] [-pidfile *PIDPATH*]

Description

The **globus-job-manager-event-generator** program is a utility which uses LRM-specific SEG parsers to generate a LRM-independent log file that a job manager instance can use to process job status change events. This program runs independently of all **globus-job-manager** instances so that only one process needs to deal with the LRM interface. The **globus-job-manager-event-generator** program can be run as a privileged user if required to interface with the LRM.

In order for **globus-job-manager-event-generator** to handle events for a particular LRM, the `globus_scheduler_event_generator_job_manager_setup` setup package must be configured after the LRM-specific setup package has been run. This can be forced by **gpt-postinstall -force** or running the command **cd \$GLOBUS_LOCATION/setup/globus; ./setup-seg-job-manager.pl**.

The full set of command-line options to **globus-job-manager-event-generator** consists of:

- help Print command-line option summary and exit.
- scheduler *LRM* Process events for the local resource manager named by *LRM*.
- background Run **globus-job-manager-event-generator** as a background process. It will fork a new process, print out its process ID and then the original process will terminate.
- pidfile *PIDPATH* Write the process ID of an instance of **globus-job-manager-event-generator** to the file named by *PIDPATH*. This file can be used to kill or monitor the **globus-job-manager-event-generator** process.

Files

globus-job-manager-seg.conf Configuration file for **globus-job-manager-event-generator**. Each line consists of a string of the form `LRM_log_path=PATH`, which indicates the directory containing LRM-independent format SEG log files for the LRM. This file is created by the running the `globus_scheduler_event_generator_job_manager_setup` setup package.

See Also

globus-scheduler-event-generator(8), globus-job-manager(8)

Name

globus-fork-starter -- Start and monitor a fork job

globus-fork-starter

Description

The **globus-fork-starter** program is executes jobs specified on its standard input stream, recording the job state changes to a file defined in the `$GLOBUS_LOCATION/etc/globus-fork.conf` configuration file. It runs until its standard input stream is closed and all jobs it is managing have terminated. The log generated by this program can be used by the SEG to provide job state changes and exit codes to the GRAM service. The **globus-fork-starter** program is typically started by the fork GRAM module.

The **globus-fork-starter** program expects its input to be a series of task definitions, separated by the newline character, each representing a separate job. Each task definition contains a number of fields, separated by the colon character. The first field is always the literal string `100` indicating the message format, the second field is a unique job tag that will be distinguish the reply from this program when multiple jobs are submitted. The rest of fields contain attribute bindings. The supported attributes are:

<code>directory</code>	Working directory of the job
<code>environment</code>	Comma-separated list of strings defining environment variables. The form of these strings is <code>var=value</code>
<code>count</code>	Number of processes to start
<code>executable</code>	Full path to the executable to run
<code>arguments</code>	Comma-separated list of command-line arguments for the job
<code>stdin</code>	Full path to a file containing the input of the job
<code>stdout</code>	Full path to a file to write the output of the job to
<code>stderr</code>	Full path to a file to write the error stream of the job

Within each field, the following characters may be escaped by preceding them with the backslash character:

- backslash (\)
- semicolon (;)
- comma (,)
- equal (=)

Additionally, newline can be represented within a field by using the escape sequence `\n`.

For each job the **globus-fork-starter** processes, it replies by writing a single line to standard output. The replies again consist of a number of fields separated by the semicolon character.

For a successful job start, the first field of the reply is the literal `101`, the second field is the tag from the input, and the third field is a comma-separated list of SEG job identifiers which consist the concatenation of a UUID and a process id. The **globus-fork-starter** program will write state changes to the SEG log using these job identifiers.

For a failure, the first field of the reply is the literal 102, the second field is the tag from the input, the third field is the integer representation of a GRAM error code, and the fourth field is a string explaining the error.

ENVIRONMENT

If the following variables affect the execution of **globus-fork-starter**

GLOBUS_LOCATION Path to Globus Toolkit installation. This is used to locate the `globus-fork.conf` configuration file.

Files

`$GLOBUS_LOCATION/etc/globus-fork.conf` Path to fork SEG configuration file.

Chapter 2. Troubleshooting

For a list of error codes generated by GRAM5, see [Section 2, “Errors”](#).

For information about sys admin logging, see [Admin Debugging](#) in the GRAM5 Admin Guide.

1. Troubleshooting tips

In case you run into problems you can do the following

- Check the GRAM5 documentation. Maybe you'll find hints here to solve your problem.
- Check the GRAM5 log for errors.

In case you don't find anything suspicious you can increase the log-level of GRAM5 or other relevant components. Maybe the additional logging-information will tell you what's going wrong.

- Send e-mails to <gram-user@globus.org>. You'll have to subscribe to a list before you can send an e-mail to it. See [here](#)¹ for general e-mail lists and information on how to subscribe to a list and [here](#)² for GRAM specific lists.

¹ http://dev.globus.org/wiki/Mailing_Lists

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

2. Errors

Table 2.1. GRAM5 Errors

Error Code	Reason	Possible Solutions
1	one of the RSL parameters is not supported	Check RSL documentation
2	the RSL length is greater than the maximum allowed	Use RSL substitutions to reduce length of RSL strings
3	an I/O operation failed	Enable trace logging and report to gram-dev@globus.org
4	jobmanager unable to set default to the directory requested	Check that RSL <code>directory</code> attribute refers to a directory that exists on the target system.
5	the executable does not exist	Check that the RSL <code>executable</code> attribute refers to an executable that exists on the target system.
6	of an unused INSUFFICIENT_FUNDS	Unimplemented feature.
7	authentication with the remote server failed	Check that the contact string contains the proper X.509 DN.
8	the user cancelled the job	Don't cancel jobs you want to complete.
9	the system cancelled the job	Check RSL requirements such as maximum time and memory are valid for the job.
10	data transfer to the server failed	Check gatekeeper and/or job manager logs to see why the process failed.
11	the stdin file does not exist	Check that the RSL <code>stdin</code> attribute refers to a file that exists on the target system or has a valid ftp, gsiftp, http, or https URL.
12	the connection to the server failed (check host and port)	Check that the service is running on the expected TCP/IP port. Check that no firewall prevents contacting that TCP/IP port. Check <code>\$GLOBUS_LOCATION/var/globus-gatekeeper.log</code> for runtime configuration errors.
13	the provided RSL 'maxtime' value is not an integer	Check that the RSL <code>maxtime</code> value evaluates to an integer.
14	the provided RSL 'count' value is not an integer	Check that the RSL <code>count</code> value evaluates to an integer.
15	the job manager received an invalid RSL	Check that the RSL string can be parsed by using <code>globusrun -p RSL</code> .
16	the job manager failed in allowing others to make contact	Check job manager log.
17	the job failed when the job manager attempted to run it	Verify that the LRM is configured properly.
18	an invalid paradyn was specified	OBSOLETE IN GRAM2
19	the provided RSL 'jobtype' value is invalid	The RSL <code>jobtype</code> attribute is not indicated as supported by the LRM. Valid <code>jobtype</code> values are <code>single</code> , <code>multiple</code> , <code>mpi</code> , and <code>condor</code> .
20	the provided RSL 'myjob' value is invalid	OBSOLETE IN GRAM5

Error Code	Reason	Possible Solutions
21	the job manager failed to locate an internal script argument file	Check that <code>\$GLOBUS_LOCATION/libexec/globus-job-manager-script.pl</code> exists and is executable. Check that the LRM-specific perl module is located in <code>\$GLOBUS_LOCATION/lib/perl/Globus/GRAM/JobManager/</code> directory and is valid. The command perl -I\$GLOBUS_LOCATION/lib/perl \$GLOBUS_LOCATION/lib/perl/Globus/GRAM/JobManager/LRM.pm can be used to check if there are any syntax errors in the script.
22	the job manager failed to create an internal script argument file	Check that your home directory is writable and not full.
23	the job manager detected an invalid job state	Check job manager logs.
24	the job manager detected an invalid script response	Check job manager logs. This is likely a bug in the LRM script.
25	the job manager detected an invalid script status	Check job manager logs. This is likely a bug in the LRM script.
26	the provided RSL 'jobtype' value is not supported by this job manager	Check that the RSL <code>jobtype</code> attribute is implemented by the LRM script. Note that some job types require configuration
27	unused ERROR_UNIMPLEMENTED	LRM does not support some feature included in the job request.
28	the job manager failed to create an internal script submission file	Check that the user's home file system is not full. Check job manager log
29	the job manager cannot find the user proxy	Check that client is delegating a proxy when authenticating with the gatekeeper. Check that the user's home filesystem and the <code>/tmp</code> file system are not full.
30	the job manager failed to open the user proxy	Check that the user's home filesystem and the <code>/tmp</code> file system are not full.
31	the job manager failed to cancel the job as requested	Check that the user's home filesystem and the <code>/tmp</code> file system are not full.
32	system memory allocation failed	Check job manager log for details.
33	the interprocess job communication initialization failed	OBSOLETE IN GRAM5
34	the interprocess job communication setup failed	OBSOLETE IN GRAM5
35	the provided RSL 'host count' value is invalid	Check that the RSL <code>host_count</code> attribute evaluates to an integer.
36	one of the provided RSL parameters is unsupported	Check job manager log for details about invalid parameter.
37	the provided RSL 'queue' parameter is invalid	Check that the RSL <code>queue</code> attribute evaluates to a string that corresponds to an LRM-specific queue name.
38	the provided RSL 'project' parameter is invalid	Check that the RSL <code>project</code> attribute evaluates to a string that corresponds to an LRM-specific project name.

Error Code	Reason	Possible Solutions
39	the provided RSL string includes variables that could not be identified	Check that all RSL substitutions are defined before being used in the job description.
40	the provided RSL 'environment' parameter is invalid	Check that the RSL <code>environment</code> attribute contains a sequence of <code>VARIABLE VALUE</code> pairs.
41	the provided RSL 'dryrun' parameter is invalid	Remove the RSL <code>dryrun</code> attribute from the job description.
42	the provided RSL is invalid (an empty string)	Include a non-empty RSL string in your job submission request.
43	the job manager failed to stage the executable	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.
44	the job manager failed to stage the stdin file	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.
45	the requested job manager type is invalid	OBSOLETE IN GRAM5
46	the provided RSL 'arguments' parameter is invalid	OBSOLETE IN GRAM2
47	the gatekeeper failed to run the job manager	Check the gatekeeper or job manager logs for more information.
48	the provided RSL could not be properly parsed	Check that the RSL string can be parsed by using globusrun -p RSL .
49	there is a version mismatch between GRAM components	Ask system administrator to upgrade GRAM service to GRAM2 or GRAM5
50	the provided RSL 'arguments' parameter is invalid	Check that the RSL <code>arguments</code> attribute evaluates to a sequence of strings.
51	the provided RSL 'count' parameter is invalid	Check that the RSL <code>count</code> attribute evaluates to a positive integer value.
52	the provided RSL 'directory' parameter is invalid	Check that the RSL <code>directory</code> attribute evaluates to a string.
53	the provided RSL 'dryrun' parameter is invalid	Check that the RSL <code>dryrun</code> attribute evaluates to either <code>yes</code> or <code>no</code> .
54	the provided RSL 'environment' parameter is invalid	Check that the RSL <code>environment</code> attribute evaluates to a sequence of <code>VARIABLE, VALUE</code> pairs.
55	the provided RSL 'executable' parameter is invalid	Check that the RSL <code>executable</code> attribute evaluates to a string value.
56	the provided RSL 'host_count' parameter is invalid	Check that the RSL <code>host_count</code> attribute evaluates to a positive integer value.
57	the provided RSL 'jobtype' parameter is invalid	Check that the RSL <code>jobtype</code> attribute evaluates to one of <code>single</code> , <code>multiple</code> , <code>mpi</code> , or <code>condor</code>

Error Code	Reason	Possible Solutions
58	the provided RSL 'maxtime' parameter is invalid	Check that the RSL <code>maxtime</code> attribute evaluates to a positive integer value.
59	the provided RSL 'myjob' parameter is invalid	OBSOLETE IN GRAM5.
60	the provided RSL 'paradyn' parameter is invalid	OBSOLETE IN GRAM2.
61	the provided RSL 'project' parameter is invalid	Check that the RSL <code>project</code> attribute evaluates to a string value.
62	the provided RSL 'queue' parameter is invalid	Check that the RSL <code>queue</code> attribute evaluates to a string value.
63	the provided RSL 'stderr' parameter is invalid	Check that the RSL <code>stderr</code> attribute evaluates to a string value or a sequence of <i>DESTINATION</i> URLs with optional <i>CACHE_TAG</i> string parameters.
64	the provided RSL 'stdin' parameter is invalid	Check that the RSL <code>stdin</code> attribute evaluates to a string value.
65	the provided RSL 'stdout' parameter is invalid	Check that the RSL <code>stdout</code> attribute evaluates to a string value or a sequence of <i>DESTINATION</i> URLs with optional <i>CACHE_TAG</i> string parameters.
66	the job manager failed to locate an internal script	Check job manager log for more details.
67	the job manager failed on the system call <code>pipe()</code>	OBSOLETE IN GRAM5
68	the job manager failed on the system call <code>fcntl()</code>	OBSOLETE IN GRAM2
69	the job manager failed to create the temporary stdout filename	OBSOLETE IN GRAM5
70	the job manager failed to create the temporary stderr filename	OBSOLETE IN GRAM5
71	the job manager failed on the system call <code>fork()</code>	OBSOLETE IN GRAM2
72	the executable file permissions do not allow execution	Check that the RSL <code>executable</code> attribute refers to an executable program or script.
73	the job manager failed to open stdout	Check that the RSL <code>stdout</code> attribute refers to one or more valid destination files or URLs.
74	the job manager failed to open stderr	Check that the RSL <code>stderr</code> attribute refers to one or more valid destination files or URLs.
75	the cache file could not be opened in order to relocate the user proxy	Check that the user's home directory is writable and not full on the GRAM5 service node.
76	cannot access cache files in <code>~/globus/.gass_cache</code> , check permissions, quota, and disk space	Check that the user's home directory is writable and not full on the GRAM5 service node.
77	the job manager failed to insert the contact in the client contact list	Check job manager log

Error Code	Reason	Possible Solutions
78	the contact was not found in the job manager's client contact list	Don't attempt to unregister callback contacts that are not registered
79	connecting to the job manager failed. Possible reasons: job terminated, invalid job contact, network problems, ...	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.
80	the syntax of the job contact is invalid	Check the syntax of job contact string.
81	the executable parameter in the RSL is undefined	Include the RSL <code>executable</code> in all job requests.
82	the job manager service is misconfigured. <code>condor arch</code> undefined	Add the <code>-condor-arch</code> to the command-line or configuration file for a job manager configured to use the <code>condor</code> LRM.
83	the job manager service is misconfigured. <code>condor os</code> undefined	Add the <code>-condor-os</code> to the command-line or configuration file for a job manager configured to use the <code>condor</code> LRM.
84	the provided RSL 'min_memory' parameter is invalid	Check that the RSL <code>min_memory</code> attribute evaluates to a positive integer value.
85	the provided RSL 'max_memory' parameter is invalid	Check that the RSL <code>max_memory</code> attribute evaluates to a positive integer value.
86	the RSL 'min_memory' value is not zero or greater	Check that the RSL <code>min_memory</code> attribute evaluates to a positive integer value.
87	the RSL 'max_memory' value is not zero or greater	Check that the RSL <code>max_memory</code> attribute evaluates to a positive integer value.
88	the creation of a HTTP message failed	Check job manager log.
89	parsing incoming HTTP message failed	Check job manager log.
90	the packing of information into a HTTP message failed	Check job manager log.
91	an incoming HTTP message did not contain the expected information	Check job manager log.
92	the job manager does not support the service that the client requested	Check that the client is talking to the correct service
93	the gatekeeper failed to find the requested service	OBSOLETE IN GRAM2
94	the jobmanager does not accept any new requests (shutting down)	Execute queries before the job has been cleaned up.
95	the client failed to close the listener associated with the callback URL	Call <code>globus_gram_client_callback_disallow()</code> with a valid the callback contact.
96	the gatekeeper contact cannot be parsed	Check the syntax of the gatekeeper contact string you are attempting to contact.
97	the job manager could not find the 'poe' command	OBSOLETE IN GRAM2
98	the job manager could not find the 'mpirun' command	Configure the LRM script with <code>mpirun</code> in your path.

Error Code	Reason	Possible Solutions
99	the provided RSL 'start_time' parameter is invalid	OBSOLETE IN GRAM2
100	the provided RSL 'reservation_handle' parameter is invalid	OBSOLETE IN GRAM2
101	the provided RSL 'max_wall_time' parameter is invalid	Check that the RSL <code>max_wall_time</code> attribute evaluates to a positive integer.
102	the RSL 'max_wall_time' value is not zero or greater	Check that the RSL <code>max_wall_time</code> attribute evaluates to a positive integer.
103	the provided RSL 'max_cpu_time' parameter is invalid	Check that the RSL <code>max_cpu_time</code> attribute evaluates to a positive integer.
104	the RSL 'max_cpu_time' value is not zero or greater	Check that the RSL <code>max_cpu_time</code> attribute evaluates to a positive integer.
105	the job manager is misconfigured, a scheduler script is missing	Check that the administrator has configured the LRM by running its setup script.
106	the job manager is misconfigured, a scheduler script has invalid permissions	Check that the administrator has installed the <code>GLLOBUS_LOCATION/libexec/globus-job-manager-script.pl</code> script. Check that the file system containing that script allows file execution.
107	the job manager failed to signal the job	OBSOLETE IN GRAM2
108	the job manager did not recognize/support the signal type	Check that your signal operation is using the correct signal constant.
109	the job manager failed to get the job id from the local scheduler	OBSOLETE IN GRAM2
110	the job manager is waiting for a commit signal	Send a two-phase commit signal to the job manager to acknowledge receiving the job contact from the job manager.
111	the job manager timed out while waiting for a commit signal	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.
112	the provided RSL 'save_state' parameter is invalid	Check that the RSL <code>save_state</code> attribute is set to <code>yes</code> or <code>no</code> .
113	the provided RSL 'restart' parameter is invalid	Check that the RSL <code>restart</code> attribute evaluates to a string containing a job contact string.
114	the provided RSL 'two_phase' parameter is invalid	Check that the RSL <code>two_phase</code> attribute evaluates to a positive integer.
115	the RSL 'two_phase' value is not zero or greater	Check that the RSL <code>two_phase</code> attribute evaluates to a positive integer.
116	the provided RSL 'stdout_position' parameter is invalid	OBSOLETE IN GRAM5
117	the RSL 'stdout_position' value is not zero or greater	OBSOLETE IN GRAM5

Error Code	Reason	Possible Solutions
118	the provided RSL 'stderr_position' parameter is invalid	OBSOLETE IN GRAM5
119	the RSL 'stderr_position' value is not zero or greater	OBSOLETE IN GRAM5
120	the job manager restart attempt failed	OBSOLETE IN GRAM2
121	the job state file doesn't exist	Check that the job contact you are trying to restart matches one that the job manager returned to you.
122	could not read the job state file	Check that the state file directory is not full.
123	could not write the job state file	Check that the state file directory is not full.
124	old job manager is still alive	Contact the returned job manager contact to manage the job you are trying to restart.
125	job manager state file TTL expired	OBSOLETE in GRAM2
126	it is unknown if the job was submitted	Check job manager log.
127	the provided RSL 'remote_io_url' parameter is invalid	Check that the RSL <code>remote_io_url</code> attribute evaluates to a string value.
128	could not write the remote io url file	Check that the user's home file system on the job manager service node is writable and not full.
129	the standard output/error size is different	Send a stdio update signal to redirect the job manager output to a new URL
130	the job manager was sent a stop signal (job is still running)	Submit a restart request to monitor the job.
131	the user proxy expired (job is still running)	Generate a new proxy and then submit a restart request to monitor the job.
132	the job was not submitted by original job-manager	OBSOLETE IN GRAM2
133	the job manager is not waiting for that commit signal	Do not send a commit signal to a job that is not waiting for a commit signal.
134	the provided RSL scheduler specific parameter is invalid	Check the LRM-specific documentation to determine what values are legal for the RSL extensions implemented by the LRM.
135	the job manager could not stage in a file	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.
136	the scratch directory could not be created	Check that the directory named by the RSL <code>scratch_dir</code> attribute exists and is writable. Check that the directory named by the RSL <code>scratch_dir</code> attribute is not full.
137	the provided 'gass_cache' parameter is invalid	Check that the RSL <code>gass_cache</code> attribute evaluates to a string.
138	the RSL contains attributes which are not valid for job submission	Do not use restart- or signal-only RSL attributes when submitting a job.

Error Code	Reason	Possible Solutions
139	the RSL contains attributes which are not valid for stdio update	Do not use submit- or restart-only RSL attributes when sending a stdio update signal to a job.
140	the RSL contains attributes which are not valid for job restart	Do not use submit- or signal-only RSL attributes when restarting a job.
141	the provided RSL 'file_stage_in' parameter is invalid	Check that the RSL <code>file_stage_in</code> attribute evaluates to a sequence of <i>SOURCE DESTINATION</i> pairs.
142	the provided RSL 'file_stage_in_shared' parameter is invalid	Check that the RSL <code>file_stage_in_shared</code> attribute evaluates to a sequence of <i>SOURCE DESTINATION</i> pairs.
143	the provided RSL 'file_stage_out' parameter is invalid	Check that the RSL <code>file_stage_out</code> attribute evaluates to a sequence of <i>SOURCE DESTINATION</i> pairs.
144	the provided RSL 'gass_cache' parameter is invalid	Check that the RSL <code>gass_cache</code> attribute evaluates to a string.
145	the provided RSL 'file_cleanup' parameter is invalid	Check that the RSL <code>file_clean_up</code> attribute evaluates to a sequence of strings.
146	the provided RSL 'scratch_dir' parameter is invalid	Check that the RSL <code>scratch_dir</code> attribute evaluates to a string.
147	the provided scheduler-specific RSL parameter is invalid	Check the LRM-specific documentation to determine what values are legal for the RSL extensions implemented by the LRM.
148	a required RSL attribute was not defined in the RSL spec	Check that the RSL <code>executable</code> attribute is present in your job request RSL. Check that the RSL <code>restart</code> attributes is present in your restart RSL.
149	the <code>gass_cache</code> attribute points to an invalid cache directory	Check that the RSL <code>gass_cache</code> attributes evaluates to a directory that exists or can be created. Check that the user's home file system is writable and not full.
150	the provided RSL 'save_state' parameter has an invalid value	Check that the RSL <code>save_state</code> attribute has a value of <code>yes</code> or <code>no</code> .
151	the job manager could not open the RSL attribute validation file	Check that <code>\$GLOBUS_LOCATION/share/globus_gram_job_manager/globus-gram-job-manager.rvf</code> is present and readable on the job manager service node. Check that <code>\$GLOBUS_LOCATION/share/globus_gram_job_manager/LRM.rvf</code> is readable on the job manager service node if present.
152	the job manager could not read the RSL attribute validation file	Check that <code>\$GLOBUS_LOCATION/share/globus_gram_job_manager/globus-gram-job-manager.rvf</code> is valid. Check that <code>\$GLOBUS_LOCATION/share/globus_gram_job_manager/LRM.rvf</code> is valid if present.
153	the provided RSL 'proxy_timeout' is invalid	Check that RSL <code>proxy_timeout</code> attribute evaluates to a positive integer.
154	the RSL 'proxy_timeout' value is not greater than zero	Check that RSL <code>proxy_timeout</code> attribute evaluates to a positive integer.

Error Code	Reason	Possible Solutions
155	the job manager could not stage out a file	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.
156	the job contact string does not match any which the job manager is handling	Check that the job contact string matches one returned from a job request.
157	proxy delegation failed	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.
158	the job manager could not lock the state lock file	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.
159	an invalid globus_io_clientattr_t was used.	Check that you have initialized the globus_io_clientattr_t attribute prior to using it with the GRAM client API.
160	an null parameter was passed to the gram library	Check that you are passing legal values to all GRAM API calls.
161	the job manager is still streaming output	OBSOLETE IN GRAM5
162	the authorization system denied the request	Check with your GRAM system administrator to allow a particular certificate to be authorized.
163	the authorization system reported a failure	Check with your system administrator to verify that the authorization system is configured properly.
164	the authorization system denied the request - invalid job id	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.
165	the authorization system denied the request - not authorized to run the specified executable	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.
166	the provided RSL 'user_name' parameter is invalid.	Check that the RSL user_name attribute evaluates to a string.
167	the job is not running in the account named by the 'user_name' parameter.	Ask with the GRAM system administrator to add an authorization entry to allow your credential to run jobs as the specified user account.

Chapter 3. Known Problems in GRAM5

1. Known Problems

The following problems and limitations are known to exist for GRAM5 at the time of the 5.0.0 release:

1.1. Limitations

- [list limitations]

1.2. Outstanding bugs

- [Bug 108](#):¹ Fork perl zombies
- [Bug 106](#):² Fix test failures with SGE LRM adapter
- [Bug 105](#):³ Held Condor jobs should be reported as SUSPENDED
- [Bug 104](#):⁴ globus-job-manager-event-generator loads all historical events the first time run
- [Bug 103](#):⁵ Ease two phase end commit timeout
- [Bug 102](#):⁶ Fix Two Phase Commit Semantics for Failed Jobs
- [Bug 100](#):⁷ one of the RSL parameters is not supported error doesn't indicate which it is
- [Bug 99](#):⁸ Add a high-level diagram for the approach doc
- [Bug 98](#):⁹ Add Condor-G doc for using GRAM 2 and 5
- [Bug 96](#):¹⁰ GRAM-106 SGE LRM mishandles invalid environment definition
- [Bug 95](#):¹¹ GRAM-106 SGE LRM doesn't check for executable permissions
- [Bug 94](#):¹² GRAM-106 SGE LRM doesn't check for executable existence
- [Bug 93](#):¹³ GRAM-106 SGE LRM script doesn't handle environment vars with whitespace
- [Bug 92](#):¹⁴ stdout to local file doesn't work if count >1

¹ <http://jira.globus.org/browse/GRAM-108>

² <http://jira.globus.org/browse/GRAM-106>

³ <http://jira.globus.org/browse/GRAM-105>

⁴ <http://jira.globus.org/browse/GRAM-104>

⁵ <http://jira.globus.org/browse/GRAM-103>

⁶ <http://jira.globus.org/browse/GRAM-102>

⁷ <http://jira.globus.org/browse/GRAM-100>

⁸ <http://jira.globus.org/browse/GRAM-99>

⁹ <http://jira.globus.org/browse/GRAM-98>

¹⁰ <http://jira.globus.org/browse/GRAM-96>

¹¹ <http://jira.globus.org/browse/GRAM-95>

¹² <http://jira.globus.org/browse/GRAM-94>

¹³ <http://jira.globus.org/browse/GRAM-93>

¹⁴ <http://jira.globus.org/browse/GRAM-92>

- [Bug 88:](#)¹⁵ Missed two phase commit causes job to not be destroyed
- [Bug 86:](#)¹⁶ Prioritize script invocations to improve throughput
- [Bug 80:](#)¹⁷ GRAM 5 beta2 release
- [Bug 79:](#)¹⁸ Add support for OSG's "NFS Lite" concept
- [Bug 77:](#)¹⁹ GRAM zombie
- [Bug 71:](#)²⁰ GRAM protocol test package contains expired test certificate
- [Bug 70:](#)²¹ globus-job-status acts strange for completed jobs in GRAM5
- [Bug 69:](#)²² globus-job-get-output -f doesn't work in GRAM5
- [Bug 68:](#)²³ Bad error when proxy is too short-lived
- [Bug 54:](#)²⁴ make globus-job-manager-event-generator not require configuration by default
- [Bug 53:](#)²⁵ Generalize log path configuration
- [Bug 51:](#)²⁶ configurable control of number of perl scripts that can run simultaneously
- [Bug 47:](#)²⁷ simplify the throughput tester program and use improved version as doc
- [Bug 24:](#)²⁸ Debug/verbose flags for globusrun, globus-job-run
- [Bug 23:](#)²⁹ Improved error codes and error reporting for users
- [Bug 22:](#)³⁰ client connections can't be timed out
- [Bug 15:](#)³¹ transition from httpg to https
- [Bug 14:](#)³² increase availability of GRAM in linux distributions
- [Bug 12:](#)³³ Gatekeeper's syslog output cannot be controlled
- [Bug 5:](#)³⁴ Add gram-level prologue and epilogue script execution for mpi jobs

¹⁵ <http://jira.globus.org/browse/GRAM-88>

¹⁶ <http://jira.globus.org/browse/GRAM-86>

¹⁷ <http://jira.globus.org/browse/GRAM-80>

¹⁸ <http://jira.globus.org/browse/GRAM-79>

¹⁹ <http://jira.globus.org/browse/GRAM-77>

²⁰ <http://jira.globus.org/browse/GRAM-71>

²¹ <http://jira.globus.org/browse/GRAM-70>

²² <http://jira.globus.org/browse/GRAM-69>

²³ <http://jira.globus.org/browse/GRAM-68>

²⁴ <http://jira.globus.org/browse/GRAM-54>

²⁵ <http://jira.globus.org/browse/GRAM-53>

²⁶ <http://jira.globus.org/browse/GRAM-51>

²⁷ <http://jira.globus.org/browse/GRAM-47>

²⁸ <http://jira.globus.org/browse/GRAM-24>

²⁹ <http://jira.globus.org/browse/GRAM-23>

³⁰ <http://jira.globus.org/browse/GRAM-22>

³¹ <http://jira.globus.org/browse/GRAM-15>

³² <http://jira.globus.org/browse/GRAM-14>

³³ <http://jira.globus.org/browse/GRAM-12>

³⁴ <http://jira.globus.org/browse/GRAM-5>

- Bug 4:³⁵ Add support for a "managed fork" service
- Bug 2:³⁶ Investigate how to setup GRAM5 services in a HA setup

³⁵ <http://jira.globus.org/browse/GRAM-4>

³⁶ <http://jira.globus.org/browse/GRAM-2>

Chapter 4. Usage statistics collection by the Globus Alliance

1. GRAM5-specific usage statistics

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

- Job Manager Session ID
- dryrun used
- RSL Host Count
- Timestamp when job hit GLOBUS_GRAM_PROTOCOL_JOB_STATE_UNSUBMITTED
- Timestamp when job hit GLOBUS_GRAM_PROTOCOL_JOB_STATE_FILE_STAGE_IN
- Timestamp when job hit GLOBUS_GRAM_PROTOCOL_JOB_STATE_PENDING
- Timestamp when job hit GLOBUS_GRAM_PROTOCOL_JOB_STATE_ACTIVE
- Timestamp when job hit GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED
- Timestamp when job hit GLOBUS_GRAM_PROTOCOL_JOB_STATE_FILE_STAGE_OUT
- Timestamp when job hit GLOBUS_GRAM_PROTOCOL_JOB_STATE_DONE
- Job Failure Code
- Number of times status is called
- Number of times register is called
- Number of times signal is called
- Number of times refresh is called
- Number of files named in file_clean_up RSL
- Number of files being staged in (including executable, stdin) from http servers
- Number of files being staged in (including executable, stdin) from https servers
- Number of files being staged in (including executable, stdin) from ftp servers
- Number of files being staged in (including executable, stdin) from gsiftp servers
- Number of files being staged into the GASS cache from http servers
- Number of files being staged into the GASS cache from https servers
- Number of files being staged into the GASS cache from ftp servers

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

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

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

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

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

- Total number of failed jobs
- Total number of canceled jobs
- Total number of completed jobs
- Total number of dry-run jobs
- Peak number of concurrently managed jobs
- Number of jobs currently being managed
- Number of jobs currently in the UNSUBMITTED state
- Number of jobs currently in the STAGE_IN state
- Number of jobs currently in the PENDING state
- Number of jobs currently in the ACTIVE state
- Number of jobs currently in the STAGE_OUT state
- Number of jobs currently in the FAILED state
- Number of jobs currently in the DONE state

Also, please see our [policy statement](#)¹ on the collection of usage statistics.

¹ ../../Usage_Stats.html

Glossary

Index

B

bugs
 outstanding, 56

E

errors, 46

J

jobs
 preparing
 delegating credentials, 2
 generate valid proxy, 1

L

limitations, 56

T

troubleshooting, 45
 check documentation, 45
 errors, 45
 gram log, 45
 mailing lists, 45

U

usage statistics, 59