GT 5.2.4 GRAM5: System Administrator's Guide
Introduction

This guide contains configuration information for system administrators working with GRAM5. It describes procedures typically performed by system administrators, including GRAM5 software installation, configuration, testing, and debugging. Readers should be familiar with the GRAM5 Key Concepts to understand the motivation for and interaction between the various deployed components.
Table of Contents

1. GRAM5 Installation ................................................................. 1
   1. Introduction ................................................................. 1
   2. Planning your GRAM5 installation ..................................... 1
   3. Installing LRM Adapter Packages ..................................... 2
2. Common Administrative Tasks ................................................. 3
   1. Managing GRAM5 Users .................................................. 3
   2. Starting and Stopping GRAM5 services ............................... 3
   3. Enabling and Disabling GRAM5 Services .............................. 3
   4. Enabling and Disabling SEG Modules ................................ 4
3. Configuring GRAM5 ................................................................ 5
   1. Gatekeeper Configuration ............................................... 5
   2. Scheduler Event Generator Configuration ............................. 6
   3. Job Manager Configuration .............................................. 7
   4. LRM Adapter Configuration ............................................. 8
   5. Auditing ........................................................................ 10
   6. RSL Attributes ................................................................ 11
4. Audit Logging ........................................................................ 12
   1. Overview ........................................................................ 12
   2. Audit and Accounting Records .......................................... 12
   3. For More Information ...................................................... 13
   4. Configuration .................................................................. 13
   5. Audit Database Interface ................................................. 13
5. Security Considerations .......................................................... 14
   1. Security Considerations ................................................... 14
6. Troubleshooting ..................................................................... 15
   1. Admin Troubleshooting .................................................... 15
7. Admin Tools .......................................................................... 18
   globus-gatekeeper ............................................................... 19
   globus-gatekeeper-admin .................................................... 22
   globus-gram-audit .............................................................. 23
   globus-job-manager ............................................................ 24
   globus-rvf-check ................................................................. 29
   globus-rvf-edit .................................................................. 30
   globus-scheduler-event-generator ........................................ 31
   globus-scheduler-event-generator-admin .............................. 32
8. Usage statistics collection by the Globus Alliance ....................... 33
   1. GRAM5-specific usage statistics ...................................... 33
Glossary .................................................................................. 36
Index ...................................................................................... 38
## List of Tables

1.1. GRAM5 LRM Adapters ................................................................. 2
3.1. Gatekeeper Configuration Path .................................................. 5
3.2. Scheduler Event Generator Configuration Path ............................... 6
3.3. Job Manager Configuration Path .................................................. 7
3.4. GRAM5 Log Levels ................................................................. 7
3.5. LRM Adapter Configuration Path ................................................. 8
3.6. GRAM Audit Configuration Path .................................................. 10
3.7. Audit Configuration Attributes .................................................... 11
List of Examples

2.1. Using globus-gatekeeper-admin to set the default service ................................................................. 4
2.2. Using globus-scheduler-event-generator-admin to disable a SEG module ........................................ 4
Chapter 1. GRAM5 Installation

1. Introduction

The Globus Toolkit provides GRAM5: a service to submit, monitor, and cancel jobs on Grid computing resources. In GRAM5, a job consists of a computation and, optionally, file transfer and management operations related to the computation. Some users, particularly interactive ones, benefit from accessing output data files as the job is running. Monitoring consists of querying for and/or subscribing to status information, such as job state changes.

GRAM5 relies on GSI C mechanisms for security, and interacts with GridFTP services to stage files to compute resources. Please see their respective Administrator's guides for information about installing, configuring, and managing those systems. In particular, you must understand the tasks in Installing GT and install the basic GRAM5 packages, and complete the tasks in Basic Security Configuration.

2. Planning your GRAM5 installation

Before installing GRAM5 on a server, you'll first need to plan what Local Resource Managers (LRMs) you want GRAM5 to interface with, what LRM you want to have as your default GRAM5 service, and whether you'll be using the globus-scheduler-event-generator to process LRM events.

GRAM5 requires a few services to be running to function: the Gatekeeper and the Scheduler Event Generator (SEG). The supported way to run these services is via the System-V style init scripts provided with the GRAM5-related packages. The gatekeeper daemon can also be configured to start via an internet superserver such as inetd or xinetd though that is beyond the scope of this document. The globus-scheduler-event-generator can not be run in that way.

2.1. Choosing an LRM Adapter

GRAM5 in GT 5.2.4 supports the following LRM adapters: Condor, PBS, GridEngine, and Fork. These LRM adapters translate GRAM5 job specifications into LRM-specific job descriptions and scripts to run them, as well as interfaces to the LRM to determine job termination status.

If you're not familiar with the supported LRMs, you might want to start with the Fork one to get familiar with how GRAM5 works. This adapter simply forks the job and runs it on the GRAM5 node. You can then install one of the other LRMs and its adapter to provide batch or high-throughput job scheduling.

2.1.1. Default GRAM5 Service

GRAM5 can be configured to support multiple LRMs on the same service machine. In that case, one LRM is typically configured as the default LRM which is used when a client uses a shortened version of a GRAM5 resource name. A common configuration is to configure a batch system interface as the default, and provide the jobmanager-fork service as well for simple jobs, such as creating directories or staging data.

2.1.2. Job Status Method

GRAM5 has two ways of determining job state transitions: polling the LRM and using the Scheduler Event Generator (SEG) service. When polling, each user's globus-job-manager will periodically execute an LRM-specific command to determine the state of each job. On systems with many users, or with users submitting a large number of jobs, this can cause significant resource use on the GRAM5 service machine. Instead, the GRAM5 service can be configured (on a per-LRM basis) to use the globus-scheduler-event-generator service to more efficiently process LRM state changes.
Note

Not all LRM adapters provide an interface to the `globus-scheduler-event-generator`, and some require LRM-specific configuration to work properly. This is described in more detail.

3. Installing LRM Adapter Packages

There are several LRM adapters included in the GT 5.2.4. For some, there is a `-setup-poll` and `-setup-seg` package which installs the adapter and configuration file needed for job status via polling or the `globus-scheduler-event-generator` program.

There are three ways to get LRM adapters: as RPM packages, as Debian packages, and from the source installer. These installation methods are described in Installing GT 5.2.4.

LRM adapter packages included in the GT 5.2.4 release are:

Table 1.1. GRAM5 LRM Adapters

<table>
<thead>
<tr>
<th>LRM Adapter</th>
<th>Poll Package</th>
<th>SEG Package</th>
<th>Installer Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>fork</td>
<td><code>globus-gram-job-manager-fork-setup-poll</code></td>
<td><code>globus-gram-job-manager-fork-setup-sega</code></td>
<td><code>globus_gram_job_manager_fork</code></td>
</tr>
<tr>
<td>pbs¹</td>
<td><code>globus-gram-job-manager-pbs-setup-poll b</code></td>
<td><code>globus-gram-job-manager-pbs-setup-seg</code></td>
<td><code>globus_gram_job_manager_pbs</code></td>
</tr>
<tr>
<td>Condor²</td>
<td>N/A</td>
<td><code>globus-gram-job-manager-condor c</code></td>
<td><code>globus_gram_job_manager_condor</code></td>
</tr>
<tr>
<td>SGE</td>
<td><code>globus-gram-job-manager-sge-setup-poll</code></td>
<td><code>globus-gram-job-manager-sge-setup-seg</code></td>
<td><code>globus_gram_job_manager_sge</code></td>
</tr>
</tbody>
</table>

¹Not recommended for production use
²This module does not work with torque 3.0.1-5 in Fedora 15 because of a bug causing `qstat` to hang. This bug is mentioned on the TORQUE user list [http://www.supercluster.org/pipermail/torqueusers/2011-May/012911.html] and is fixed in newer versions.
³This LRM uses a SEG-like mechanism included in the `globus-job-manager` program, but not the `globus-scheduler-event-generator` service.

Chapter 2. Common Administrative Tasks

There are several tools provided with GT 5.2.4 to manage GRAM5, as well as OS-specific tools to start and stop some of the services. There are tools to manage user authorization, which services are enabled, which scheduler event generator modules are enabled, and to test the globus-gatekeeper service.

1. Managing GRAM5 Users

Before a user may interact with the GRAM5 service to submit jobs, he or she must be authorized to use the service. In order to be authorized, a GRAM5 administrator must add the user's credential name and local account mapping to the /etc/grid-mapfile. This can be done using the grid-mapfile-add-entry and grid-mapfile-delete-entry tools. For more information, see the GSI C manual.

2. Starting and Stopping GRAM5 services

In order to run the service, the globus-gatekeeper, and, if applicable to your configuration, the globus-scheduler-event-generator services must be running on your system. The packages for these services include init scripts and configuration files which can be used to configure, start, and stop the service.

The globus-gatekeeper and globus-scheduler-event-generator init scripts handle the following actions: start, stop, status, restart, condrestart, try-restart, reload, and force-reload. The globus-scheduler-event-generator script also accepts another optional parameter to start or stop a particular globus-scheduler-event-generator module. If the second parameter is not present, then all services will be acted on.

2.1. Debian Specifics

If you installed using Debian packaging tools, then the services will automatically be started upon installation. To start or stop the service, use the command invoke-rc.d with the service name and action.

2.2. RPM Specifics

If you installed using the RPM packaging tools, then the services will be installed but not enabled by default. To enable the services to start at boot time, use the commands:

```
# chkconfig globus-gatekeeper on
# chkconfig globus-scheduler-event-generator on
```

To start or stop the services, use the service command to run the init scripts with the service name and action and optional globus-scheduler-event-generator module.

3. Enabling and Disabling GRAM5 Services

The GRAM5 packages described in Section 3, “Installing LRM Adapter Packages” will automatically register themselves with the globus-gatekeeper and globus-scheduler-event-generator services. The first LRM adapter installed will be configured as the default Job Manager service. To list the installed services, change the default, or disable a service, use the (8) tool.
Example 2.1. Using globus-gatekeeper-admin to set the default service

This example shows how to use the globus-gatekeeper-admin tool to list the available services and then choose one as the default:

```
# globus-gatekeeper-admin -l
jobmanager-condor [ENABLED]
jobmanager-fork-poll [ENABLED]
jobmanager-fork [ALIAS to jobmanager-fork-poll]
# globus-gatekeeper-admin -e jobmanager-condor -n jobmanager
# globus-gatekeeper-admin -l
jobmanager-condor [ENABLED]
jobmanager-fork-poll [ENABLED]
jobmanager [ALIAS to jobmanager-condor]
jobmanager-fork [ALIAS to jobmanager-fork-poll]
```

4. Enabling and Disabling SEG Modules

The -setup-seg packages described in Section 3, “Installing LRM Adapter Packages” will automatically register themselves with the globus-scheduler-event-generator service. To disable a module from running when the globus-scheduler-event-generator service is started, use the (8) tool.

Example 2.2. Using globus-scheduler-event-generator-admin to disable a SEG module

This example shows how to stop the pbs globus-scheduler-event-generator module and disable it so it will not restart when the system is rebooted:

```
# /etc/init.d/globus-scheduler-event-generator stop pbs
Stopped globus-scheduler-event-generator [ OK ]
# globus-scheduler-event-generator-admin -d pbs
# globus-scheduler-event-generator-admin -l
pbs [DISABLED]
```
Chapter 3. Configuring GRAM5

GRAM5 is designed to be usable by default without any manual configuration. However, there are many ways to customize a GRAM5 installation to better interact with site policies, filesystem layouts, LRM interactions, logging, and auditing. In addition to GRAM5-specific configuration, see Configuring GSI for information about configuring GSI security.

1. Gatekeeper Configuration

The globus-gatekeeper has many configuration options related to network configuration, security, logging, service path, and nice level. This configuration is located in:

**Table 3.1. Gatekeeper Configuration Path**

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>Configuration Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
<td>/etc/sysconfig/globus-gatekeeper</td>
</tr>
<tr>
<td>Debian Package</td>
<td>/etc/default/globus-gatekeeper</td>
</tr>
<tr>
<td>Source Installer</td>
<td>$PREFIX/etc/globus-gatekeeper.conf</td>
</tr>
</tbody>
</table>

The following configuration variables are available in the globus-gatekeeper configuration file:

- **GLOBUS_GATEKEEPER_PORT**: Gatekeeper Service Port. If not set, the globus-gatekeeper uses the default of 2119.
- **GLOBUS_LOCATION**: Globus Installation Path. If not set, the globus-gatekeeper uses the paths defined at package compilation time.
- **GLOBUS_GATEKEEPER_LOG**: Gatekeeper Log Filename. If not set, the globus-gatekeeper logs to syslog using the GRAM-gatekeeper log identification prefix. The default configuration value is /var/log/globus-gatekeeper.log
- **GLOBUS_GATEKEEPER_GRID SERVICES**: Path to grid service definitions. If not set, the globus-gatekeeper uses the default of /etc/grid-services.
- **GLOBUS_GATEKEEPER_GRIDMAP**: Path to grid-mapfile for authorization. If not set, the globus-gatekeeper uses the default of /etc/grid-security/grid-mapfile.
- **GLOBUS_GATEKEEPER_CERT_DIR**: Path to a trusted certificate root directory. If not set, the globus-gatekeeper uses the default of /etc/grid-security/certificates.
- **GLOBUS_GATEKEEPER_CERT_FILE**: Path to the gatekeeper's certificate. If not set, the globus-gatekeeper uses the default of /etc/grid-security/hostcert.pem.
- **GLOBUS_GATEKEEPER_KEY_FILE**: Path to the gatekeeper's private key. If not set, the globus-gatekeeper uses the default of /etc/grid-security/hostkey.pem.
Configuring GRAM5

GLOBUS_GATEKEEPER_KERBEROS_ENABLED
Flag indicating whether or not the globus-gatekeeper will use a kerberos GSSAPI implementation instead of the GSI GSSAPI implementation (untested).

GLOBUS_GATEKEEPER_KMAP
Path to the KMAP authentication module. (untested).

GLOBUS_GATEKEEPER_PIDFILE
Path to a file where the globus-gatekeeper's process ID is written. If not set, globus-gatekeeper uses /var/run/globus-gatekeeper.pid

GLOBUS_GATEKEEPER_NICE_LEVEL
Process nice level for globus-gatekeeper and globus-job-manager processes. If not set, the default system process nice level is used.

After modifying the configuration file, restart the globus-gatekeeper using the methods described in Section 2, “Starting and Stopping GRAM5 services”.

2. Scheduler Event Generator Configuration

The globus-scheduler-event-generator has several configuration options related to filesystem paths. This configuration is located in:

Table 3.2. Scheduler Event Generator Configuration Path

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>Configuration Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
<td>/etc/sysconfig/globus-scheduler-event-generator</td>
</tr>
<tr>
<td>Debian Package</td>
<td>/etc/default/globus-scheduler-event-generator</td>
</tr>
<tr>
<td>Source Installer</td>
<td>PREFIX/etc/globus-scheduler-event-generator.conf</td>
</tr>
</tbody>
</table>

The following configuration variables are available in the globus-scheduler-event-generator configuration file:

GLOBUS_SEG_PIDFMT
Scheduler Event Generator PID file path format. Modify this to be the location where the globus-scheduler-event-generator writes its process IDs (one per configured LRM). The format is a printf format string with one %s to be replaced by the LRM name. By default, globus-scheduler-event-generator uses /var/run/globus-scheduler-event-generator-%s.pid.

GLOBUS_SEG_LOGFMT
Scheduler Event Generator Log path format. Modify this to be the location where globus-scheduler-event-generator writes its event logs. The format is a printf format string with one %s to be replaced by the LRM name. By default, globus-scheduler-event-generator uses /var/lib/globus/globus-seg-%s. If you modify this value, you'll need to also update the LRM configuration file to look for the log file in the new location.

GLOBUS_SEG_NICE_LEVEL
Process nice level for globus-scheduler-event-generator processes. If not set, the default system process nice level is used.

After modifying the configuration file, restart the globus-scheduler-event-generator using the methods described in Section 2, “Starting and Stopping GRAM5 services”.

6
3. Job Manager Configuration

The `globus-job-manager` process is started by the `globus-gatekeeper` and uses the configuration defined in the service entry for the resource name. By default, these service entries use a common configuration file for most job manager features. This configuration is located in:

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>Configuration Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
<td><code>/etc/globus/globus-gram-job-manager.conf</code></td>
</tr>
<tr>
<td>Debian Package</td>
<td><code>/etc/globus/globus-gram-job-manager.conf</code></td>
</tr>
<tr>
<td>Source Installer</td>
<td><code>PREFIX/etc/globus-gram-job-manager.conf</code></td>
</tr>
</tbody>
</table>

This configuration file is used to construct the command-line options for the `globus-job-manager` program. Thus, all of the options described in (8) may be used.

3.1. Job Manager Logging

From an administrator's perspective, the most important job manager configuration options are likely the ones related to logging and auditing. The default GRAM5 configuration puts logs in `/var/log/globus/gram_USERNAME.log`, with logging enabled at the FATAL and ERROR levels. To enable more fine-grained logging, add the option `-log-levels LEVELS` to `/etc/globus/globus-gram-job-manager.conf`. The value for `LEVELS` is a set of log levels joined by the `|` character. The available log levels are:

<table>
<thead>
<tr>
<th>Level</th>
<th>Meaning</th>
<th>Default Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>FATAL</td>
<td>Problems which cause the job manager to terminate prematurely.</td>
<td>Enabled</td>
</tr>
<tr>
<td>ERROR</td>
<td>Problems which cause a job or operation to fail.</td>
<td>Enabled</td>
</tr>
<tr>
<td>WARN</td>
<td>Problems which cause minor problems with job execution or monitoring.</td>
<td>Disabled</td>
</tr>
<tr>
<td>INFO</td>
<td>Major events in the lifetime of the job manager and its jobs.</td>
<td>Disabled</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Minor events in the lifetime of jobs.</td>
<td>Disabled</td>
</tr>
<tr>
<td>TRACE</td>
<td>Job processing details.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

In RPM or Debian package installs, these logs will be configured to be rotated via `logrotate`. See `/etc/logrotate.d/globus-job-manager` for details on the default log rotation configuration.

3.2. Firewall Configuration

There are also a few configuration options related to the TCP ports the Job Manager users. This port configuration is useful when dealing with firewalls that restrict incoming or outgoing ports. To restrict incoming ports (those...
that the Job Manager listens on), add the command-line option `-globus-tcp-port-range` to the Job Manager configuration file like this:

```
-globus-tcp-port-range MIN-PORT,MAX-PORT
```

Where `MIN-PORT` is the minimum TCP port number the Job Manager will listen on and `MAX-PORT` is the maximum TCP port number the Job Manager will listen on.

Similarly, to restrict the outgoing port numbers that the job manager connects from, use the command-line option `-globus-tcp-source-range`, like this:

```
-globus-tcp-source-range MIN-PORT,MAX-PORT
```

Where `MIN-PORT` is the minimum outgoing TCP port number the Job Manager will use and `MAX-PORT` is the maximum TCP outgoing port number the Job Manager will use.

For more information about Globus and firewalls, see Section 4, “Firewall configuration”.

# 4. LRM Adapter Configuration

Each LRM adapter has its own configuration file which can help customize the adapter to the site configuration. Some LRMs use non-standard programs to launch parallel or MPI jobs, and some might want to provide queue or project validation to make it easier to translate job failures into problems that can be described by GRAM5. All of the LRM adapter configuration files consist of simple `variable=value` pairs, with a leading `#` starting a comment until end-of-line.

Generally, the GRAM5 LRM configuration files are located in the globus configuration directory, with each configuration file named by the LRM name (fork, condor, pbs, sge). The following table contains the paths to these configurations:

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>Configuration Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
<td>/etc/globus/globus-LRM.conf</td>
</tr>
<tr>
<td>Debian Package</td>
<td>/etc/globus/globus-LRM.conf</td>
</tr>
<tr>
<td>Source Installer</td>
<td>PREFIX/etc/globus/globus-LRM.conf</td>
</tr>
</tbody>
</table>

## 4.1. Fork

The `globus-fork.conf` configuration file can define the following configuration parameters:

- `log_path`: Path to the `globus-fork.log` file used by the `globus-fork-starter` and fork SEG module.
- `mpixec`, `mpirun`: Path to `mpixec` and `mpirun` for parallel jobs which use MPI. By default, these are not configured. The LRM adapter will use `mpixec` over `mpirun` if both are defined.
- `softenv_dir`: Path to an installation of `softenv`\(^1\), which is used on some systems to manage application environment variables.

## 4.2. Condor

The `globus-condor.conf` configuration file can define the following configuration parameters:

Configuring GRAM5

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>condor_os</td>
<td>Custom value for the OpSys requirement for condor jobs. If not specified, the system-wide default will be used.</td>
</tr>
<tr>
<td>condor_arch</td>
<td>Custom value for the OpSys requirement for condor jobs. If not specified, the system-wide default will be used.</td>
</tr>
<tr>
<td>condor_submit, condor_rm</td>
<td>Path to the condor commands that the LRM adapter uses. These are usually determined when the LRM adapter is compiled if the commands are in the PATH.</td>
</tr>
<tr>
<td>condor_config</td>
<td>Value of the CONDOR_CONFIG environment variable, which might be needed to use condor in some cases.</td>
</tr>
<tr>
<td>check_vanilla_files</td>
<td>Enable checking if executable, standard input, and directory are valid paths for vanilla universe jobs. This can detect some types of errors before submitting jobs to condor, but only if the filesystems between the condor submit host and condor execution hosts are equivalent. In other cases, this may cause unnecessary job failures.</td>
</tr>
<tr>
<td>condor_mpi_script</td>
<td>Path to a script to launch MPI jobs on condor.</td>
</tr>
</tbody>
</table>

4.3. PBS

The globus-pbs.conf configuration file can define the following configuration parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>log_path</td>
<td>Path to PBS server_logs directory. The PBS SEG module parses these logs to generate LRM events.</td>
</tr>
<tr>
<td>pbs_default</td>
<td>Name of the PBS server node, if not the same as the GRAM service node.</td>
</tr>
<tr>
<td>mpiexec, mpirun</td>
<td>Path to mpiexec and mpirun for parallel jobs which use MPI. By default these are not configured. The LRM adapter will use mpiexec over mpirun if both are defined.</td>
</tr>
<tr>
<td>qsub, qstat, qdel</td>
<td>Path to the LRM-specific command to submit, check, and delete PBS jobs. These are usually determined when the LRM adapter is compiled if they are in the PATH.</td>
</tr>
<tr>
<td>cluster</td>
<td>If this value is set to yes, then the LRM adapter will attempt to use a remote shell command to launch multiple instances of the executable on different nodes, as defined by the file named by the PBS_NODEFILE environment variable.</td>
</tr>
<tr>
<td>remote_shell</td>
<td>Remote shell command to launch processes on different nodes when cluster is set to yes.</td>
</tr>
<tr>
<td>cpu_per_node</td>
<td>Number of instances of the executable to launch per allocated node.</td>
</tr>
<tr>
<td>softenv_dir</td>
<td>Path to an installation of softenv(^2) which is used on some systems to manage application environment variables.</td>
</tr>
</tbody>
</table>

4.4. SGE

The globus-sge.conf configuration file can define the following configuration parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sge_root</td>
<td>Root location of the GridEngine installation. If this is set to undefined, then the LRM adapter will try to determine it from the globus-job-manager environment, or if not there, the contents of the file named by the sge_config configuration parameter.</td>
</tr>
</tbody>
</table>

Configuring GRAM5

**sge_cell**
Name of the GridEngine cell to interact with. If this is set to `undefined`, then the LRM adapter will try to determine it from the `globus-job-manager` environment, or if not there, the contents of the file named by the `sge_config` configuration parameter.

**sge_config**
Path to a file which defines the `SGE_ROOT` and the `SGE_CELL` environment variables.

**log_path**
Path to GridEngine reporting file. This value is used by the SGE SEG module. If this is used, GridEngine must be configured to write a reporting file and not load reporting data into an ARCo database.

**qsub, qstat, qdel, qconf**
Path to the LRM-specific command to submit, check, and delete GridEngine jobs. These are usually determined when the LRM adapter is compiled if they are in the `PATH`.

**sun_mprun, mpirun**
Path to `mprun` and `mpirun` for parallel jobs which use MPI. By default these are not configured. The LRM adapter will use `mprun` over `mpirun` if both are defined.

**default_pe**
Default parallel environment to submit parallel jobs to. If this is not set, then clients must use the `parallel_environment` RSL attribute to choose one.

**validate_pes**
If this value is set to `true`, then the LRM adapter will verify that the `parallel_environment` RSL attribute value matches one of the parallel environments supported by this GridEngine service.

**available_pes**
If this value is defined, use it as a list of parallel environments supported by this GridEngine deployment for validation when `validate_pes` is set to `true`. If validation is being done but this value is not set, then the LRM adapter will query the GridEngine service to determine available parallel environments at startup.

**default_queue**
Default queue to use if the job description does not name one.

**validate_queues**
If this value is set to `true`, then the LRM adapter will verify that the `queue` RSL attribute value matches one of the queues supported by this GridEngine service.

**available_queues**
If this value is defined, use it as a list of queues supported by this GridEngine deployment for validation when `validate_queues` is set to `true`. If validation is being done but this value is not set, then the LRM adapter will query the GridEngine service to determine available queues at startup.

### 4.4.1. Enabling reporting for the GridEngine Scheduler Event Generator

In order to use the Scheduler Event Generator with GridEngine, the job reporting feature must be enabled, and ARCo database storage must not be enabled. To enable this, use the command `qconf -mconf` and modify the `reporting_params` parameter so that the options `reporting` and `joblog` are set to `true`.

### 5. Auditing

The `globus-gram-audit` configuration defines information about the database to load the GRAM5 audit records into. This configuration is located in:

**Table 3.6. GRAM Audit Configuration Path**

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>Configuration Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
<td><code>/etc/globus/gram-audit.conf</code></td>
</tr>
</tbody>
</table>
### Installation Type

<table>
<thead>
<tr>
<th>Configuration Path</th>
<th>Debian Package: /etc/globus/gram-audit.conf</th>
</tr>
</thead>
</table>

This configuration file contains the following attributes. Each attribute is defined by a `ATTRIBUTE:VALUE` pair.

#### Table 3.7. Audit Configuration Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVER</td>
<td>The name of the Perl 5 DBI driver for the database to be used. The supported drivers for this program are SQLite, Pg (for PostgreSQL), and mysql.</td>
<td>SQLite</td>
</tr>
<tr>
<td>DATABASE</td>
<td>The DBI data source specification to contact the audit database. dbname=/var/gram_audit_database/gram_audit.db</td>
<td></td>
</tr>
<tr>
<td>USERNAME</td>
<td>Username to authenticate as to the database</td>
<td></td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Password to use to authenticate with the database</td>
<td></td>
</tr>
<tr>
<td>AUDITVERSION</td>
<td>Version of the audit database table schemas to use. May be 1 or 1TG for this version of the software.</td>
<td>1</td>
</tr>
</tbody>
</table>

### 6. RSL Attributes

GRAM5 uses the RSL language to encode job descriptions. The attributes supported by gram are defined in RSL Validation Files. These definitions contain information about when the different RSL attributes are valid and what their default values might be if not present. GRAM5 will look in `/etc/globus/gram/job-manager.rvf` and `/etc/globus/gram/LRM.rvf` for site-specific changes to the RSL validation file.
Chapter 4. Audit Logging

1. Overview

GRAM5 includes mechanisms to provide access to audit and accounting information associated with jobs that GRAM5 submits to a local resource manager (LRM) such as Torque, GridEngine, or Condor.

In some scenarios, it is desirable to get general information about the usage of the underlying LRM, such as:

- What kinds of jobs were submitted via GRAM?
- How long did the processing of a job take?
- How many jobs were submitted by user X?

The following three use cases give a better overview of the meaning and purpose of auditing and accounting:

1. **Group Access**: A grid resource provider allows a remote service (e.g., a gateway or portal) to submit jobs on behalf of multiple users. The grid resource provider only obtains information about the identity of the remote submitting service and thus does not know the identity of the users for which the grid jobs are submitted. This group access is allowed under the condition that the remote service stores audit information so that, if and when needed, the grid resource provider can request and obtain information to track a specific job back to an individual user.

2. **Query Job Accounting**: A client that submits a job needs to be able to obtain, after the job has completed, information about the resources consumed by that job. In portal and gateway environments where many users submit many jobs against a single allocation, this per-job accounting information is needed soon after the job completes so that client-side accounting can be updated. Accounting information is sensitive and thus should only be released to authorized parties.

3. **Auditing**: In a distributed, multi-site environment, it can be necessary to investigate various forms of suspected intrusion and abuse. In such cases, we may need to access an audit trail of the actions performed by a service. When accessing this audit trail, it will frequently be important to be able to relate specific actions to the user.

Audit logging in GRAM5 is done when a job completes.

2. Audit and Accounting Records

While audit and accounting records may be generated and stored by different entities in different contexts, we make the following assumptions in this chapter:

<table>
<thead>
<tr>
<th></th>
<th>Audit Records</th>
<th>Accounting Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generated by:</td>
<td>GRAM service</td>
<td>LRM to which the GRAM service submits jobs</td>
</tr>
<tr>
<td>Stored in:</td>
<td>Database, indexed by GJID</td>
<td>LRM, indexed by JID</td>
</tr>
<tr>
<td>Data that is stored:</td>
<td>See list below.</td>
<td>May include all information about the duration and resource-usage of a job</td>
</tr>
</tbody>
</table>

The audit record of each job contains the following data:

- **job_grid_id**: String representation of the resource EPR
Audit Logging

- **local_job_id**: Job/process id generated by the scheduler
- **subject_name**: Distinguished name (DN) of the user
- **username**: Local username
- **idempotence_id**: Job id generated on the client-side
- **creation_time**: Date when the job resource is created
- **queued_time**: Date when the job is submitted to the scheduler
- **stage_in_grid_id**: String representation of the stageIn-EPR (RFT)
- **stage_out_grid_id**: String representation of the stageOut-EPR (RFT)
- **clean_up_grid_id**: String representation of the cleanUp-EPR (RFT)
- **globus_toolkit_version**: Version of the server-side GT
- **resource_manager_type**: Type of the resource manager (Fork, Condor, ...)
- **job_description**: Complete job description document
- **success_flag**: Flag that shows whether the job failed or finished successfully
- **finished_flag**: Flag that shows whether the job is already fully processed or still in progress
- **gateway_user**: Teragrid identity of the user which submitted the job.

3. For More Information

The rest of this chapter focuses on how to configure GRAM5 to enable Audit-Logging.

4. Configuration

Audit logging is turned off by default. To enable GRAM5 audit logging, in the job manager, add the command-line option `-audit-directory AUDIT-DIRECTORY` to the job manager configuration in one of the following locations:

- `$GLOBUS_LOCATION/etc/globus-job-manager.conf` to enable it for all job manager services
- `$GLOBUS_LOCATION/etc/grid-services/LRM_SERVICE_NAME` to enable it for a particular job manager service for a particular LRM.

5. Audit Database Interface

The `globus-gram-audit` program reads GRAM5 audit records and loads those records into a SQL database. This program is available as part of the globus_gram_job_manager_auditing package. It must be configured by installing and running the `globus_gram_job_manager_auditing_setup_scripts` setup package via `gpt-postinstall`. This setup script creates the `$GLOBUS_LOCATION/etc/globus-job-manager-audit.conf` configuration file described below and creates database tables needed by the audit system.

The `globus-gram-audit` program support three database systems: MySQL, PostgreSQL, and SQLite.
Chapter 5. Security Considerations

1. Security Considerations

1.1. Gatekeeper Security Considerations

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

1.2. Job Manager Security Considerations

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

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

1.3. Fork SEG Module Security Considerations

The Fork Scheduler Event Generator module uses a globally writable file for job state change events. This is not recommended for production use.
Chapter 6. Troubleshooting

1. Admin Troubleshooting

1.1. Security

GRAM requires a host certificate and private key in order for the `globus-gatekeeper` service to run. These are typically located in `/etc/grid-security/hostcert.pem` and `/etc/grid-security/hostkey.pem`, but the path is configurable in the `gatekeeper` configuration file. The key must be protected by file permissions allowing only the root user to read it.

GRAM also (by default) uses a `grid-mapfile` to authorize Grid users as local users. This file is typically located in `/etc/grid-security/grid-mapfile`, but is configurable in the `gatekeeper` configuration file.

Problems in either of these configurations will show up in the gatekeeper log described below. See the GSI documentation for more detailed information about obtaining and installing host certificates and maintaining a `grid-mapfile`.

1.2. Verify that Services are Running

GRAM relies on the `globus-gatekeeper` program and (in some cases) the `globus-scheduler-event-generator` programs to process jobs. If the former is not running, jobs requests will fail with a "connection refused" error. If the latter is not running, GRAM jobs will appear to "hang" in the PENDING state.

The `globus-gatekeeper` is typically started via an init script installed in `/etc/init.d/globus-gatekeeper`. The command `/etc/init.d/globus-gatekeeper status` will indicate whether the service is running. See Section 2, “Starting and Stopping GRAM5 services” for more information about starting and stopping the `globus-gatekeeper` program.

If the `globus-gatekeeper` service fails to start, the output of the command `globus-gatekeeper -test` will output information describing some types of configuration problems.

The `globus-scheduler-event-generator` is typically started via an init script installed in `/etc/init.d/globus-scheduler-event-generator`. It is only needed when the LRM-specific "setup-seg" package is installed. The command `/etc/init.d/globus-scheduler-event-generator status` will indicate whether the service is running. See Section 2, “Starting and Stopping GRAM5 services” for more information about starting and stopping the `globus-scheduler-event-generator` program.

1.3. Verify that LRM packages are installed

The `globus-gatekeeper` program starts the `globus-job-manager` service with different command-line parameters depending on the LRM being used. Use the command `globus-gatekeeper-admin -l` to list which LRMs the gatekeeper is configured to use.

The `globus-job-manager-script.pl` is the interface between the GRAM job manager process and the LRM adapter. The command `/usr/share/globus/globus-job-manager-script.pl -h` will print the list of available adapters.

```
% /usr/share/globus/globus-job-manager-script.pl -h
USAGE: /usr/share/globus/globus-job-manager-script.pl -m MANAGER -f FILE -c COMMAND
Installed managers: condor fork
```

The `globus-scheduler-event-generator` also uses an LRM-specific module to generate scheduler events for GRAM to reduce the amount of resources GRAM uses on the machine where it runs. To determine which LRMs are installed and configured, use the command `globus-scheduler-event-generator-admin -l`.
% globus-scheduler-event-generator-admin -l
fork [DISABLED]

If any of these do not show the LRM you are trying to use, install the relevant packages related to that LRM and restart the GRAM services. See the GRAM Administrator’s Guide for more information about starting and stopping the GRAM services.

1.4. Verify that the LRM packages are configured

All GRAM5 LRM adapters have a configuration file for site customizations, such as queue names, paths to executables needed to interface with the LRM, etc. Check that the values in these files are correct. These files are described in Section 4, “LRM Adapter Configuration”.

1.5. Check the Gatekeeper Log

The /var/log/globus-gatekeeper.log file contains information about service requests from clients, and will be useful when diagnosing service startup failures, authentication failures, and authorization failures.

1.5.1. Authorization failures

GRAM uses GSI to authenticate client job requests. If there is a problem with the GSI configuration for your host, or a client is trying to connect with a certificate signed by a CA your host does not trust, the job request will fail. This will show up in the log as a "GSS authentication failure". See the GSI Administrator's Guide for information about diagnosing authentication failures.

1.5.2. Gridmap failures

After authentication is complete, GRAM maps the Grid identity to a local user prior to starting the globus-job-manager process. If this fails, an error will show up in the log as "globus_gss_assist_gridmap() failed authorization". See the GSI Administrator's Guide for information about managing gridmap files.

1.6. Job Manager Logs

A per-user job manager log is typically located in /var/log/globus/gram_$$USERNAME.log. This log contains information from the job manager as it attempts to execute GRAM jobs via a local resource manager. The logs can be fairly verbose. Sometimes looking for log entries near those containing the string level=ERROR will show more information about what caused a particular failure.

Once you’ve found an error in the log, it is generally useful to find log entries related to the job which hit that error. There are two job IDs associated with each job, one a GRAM-specific ID, and one an LRM-specific ID. To determine the GRAM ID associated with a job, look for the attribute gramid in the log message. Finding that, looking for all other log messages which contain that gramid value will give a better picture of what the job manager is doing. To determine the LRM-specific ID, look for a message at TRACE level with the matching GRAM ID found above with the response value matching GRAM_SCRIPT_JOB_ID: LRM-ID. You can then find follow the state of the LRM-ID as well as the GRAM ID in the log, and correlate the LRM-ID information with local resource manager logs and administrative tools.

1.7. Email Support

If all else fails, please send information about your problem to <gram-user@globus.org>. You'll have to subscribe to a list before you can send an e-mail to it. See here for general e-mail lists and information on how to subscribe.

1 http://dev.globus.org/wiki/Mailing_Lists
scribe to a list and here\textsuperscript{2} for GRAM-specific lists. Depending on the problem, you may be requested to file a bug report to the Globus project’s Issue Tracker\textsuperscript{3}.

\begin{itemize}
\item \textsuperscript{2} http://dev.globus.org/wiki/GRAM#Mailing_\_Lists
\item \textsuperscript{3} http://jira.globus.org
\end{itemize}
Chapter 7. Admin Tools
Name
globus-gatekeeper — Authorize and execute a grid service on behalf of a user

Synopsis

globus-gatekeeper [-help]
 [ -conf PARAMETER_FILE ]
 [ -test ] [ -d | -debug ]
 [ -inetd | -f ]
 [ -p PORT | -port PORT ]
 [ -home PATH | -L LOGFILE | -logfile LOGFILE ] [ -lf LOG_FACILITY ]
 [ -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 ]
 [ -pidfile PIDFILE ]

Description

The globus-gatekeeper program is a meta-server similar to inetd or xinetd that starts other services after authenticating a TCP connection using GSSAPI and mapping the client's credential to a local account.

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 /etc/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 PARAMETER_FILE
  Load configuration parameters from PARAMETER_FILE. 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.
Admin Tools

- **-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 2119 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 looks for service executables in `/usr/sbin`, configuration in `/etc`, and writes logs and accounting files to `/var/log`.

- **-l LOGFILE, -logfile LOGFILE** Write log entries to `LOGFILE`. If `LOGFILE` is equal to `logoff` or `LOGOFF`, then logging will be disabled, both to file and to syslog.

- **-lf LOG_FACILITY** Open syslog using the `LOG_FACILITY`. If not specified, `LOG_DAEMON` will be used as the default when using syslog.

- **-acctfile ACCTFILE** Set the path to write accounting records to `ACCTFILE`. If not set, records will be written to the log file.

- **-e LIBEXECDIR** Look for service executables in `LIBEXECDIR`. If not specified, the `sbin` subdirectory of the parameter to `-home` is used, or `/usr/sbin` if that is not set.

- **-launch_method** 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`.

- **-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.

- **-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_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 Use the **globus-k5** command to acquire Kerberos 5 credentials before starting the service.

-globuskmap *KMAP* Use *KMAP* as the path to the Grid credential to kerberos initialization mapping file.

-pidfile *PIDFILE* Write the process id of the **globus-gatekeeper** to the file named by *PIDFILE*.

**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.

`GLOBUS_LOCATION` Default path to gatekeeper service files.

**Files**

/etc/grid-services/*SERVICENAME* Service configuration for *SERVICENAME*.

/etc/grid-security/grid-mapfile Default file mapping Grid identities to POSIX identities.

/etc/globuskmap Default file mapping Grid identities to Kerberos 5 principals.

/etc/globus-nologin File to disable the **globus-gatekeeper** program.

/var/log/globus-gatekeeper.log Default gatekeeper log.

**See also**

globus-k5(8), globusrun(1), globus-job-manager(8)
Name
globus-gatekeeper-admin — Manage globus-gatekeeper services

Synopsis

globus-gatekeeper-admin [-h]
globus-gatekeeper-admin [-l] [-n NAME]
globus-gatekeeper-admin [-e SERVICE] [-n NAME]
globus-gatekeeper-admin [-E]
globus-gatekeeper-admin [-d SERVICE]

Description

The `globus-gatekeeper-admin` program manages service entries which are used by the `globus-gatekeeper` to execute services. Service entries are located in the `/etc/grid-services` directory. The `globus-gatekeeper-admin` can list, enable, or disable specific services, or set a service as the default. The `-h` command-line option shows a brief usage message.

Listing services

The `-l` command-line option to `globus-gatekeeper-admin` will cause it to list all of the services which are available to be run by the `globus-gatekeeper`. In the output, the service name will be followed by its status in brackets. Possible status strings are `ENABLED`, `DISABLED`, and `ALIAS to NAME`, where `NAME` is another service name.

If the `-n NAME` is used, then only information about the service named `NAME` is printed.

Enabling services

The `-e SERVICE` command-line option to `globus-gatekeeper-admin` will cause it to enable a service so that it may be run by the `globus-gatekeeper`.

If the `-n NAME` option is used as well, then the service will be enabled with the alias `NAME`.

Enabling a default service

The `-E` command-line option to `globus-gatekeeper-admin` will cause it to enable a service alias with the name `jobmanager`. The `globus-gatekeeper-admin` program will choose the first service it finds as the default. To enable a particular service as the default, use the `-e` parameter described above with the `-n` parameter.

Disabling services

The `-d SERVICE` command-line option to `globus-gatekeeper-admin` will cause it to disable a service so that it may not be run by the `globus-gatekeeper`. All aliases to a disabled service are also disabled.

Files

`/etc/grid-services` Default location of enabled gatekeeper service descriptions.
Name
globus-gram-audit — Load GRAM4 and GRAM5 audit records into a database

Synopsis

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:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--conf CONFIG_FILE</td>
<td>Use CONFIG_FILE instead of the default from the configuration file for audit database configuration.</td>
</tr>
<tr>
<td>--check</td>
<td>Check whether the insertion of a record was successful by querying the database after inserting the records. This is used in tests.</td>
</tr>
<tr>
<td>--delete</td>
<td>Delete audit records from the database right after inserting them. This is used in tests to avoid filling the database with test records.</td>
</tr>
<tr>
<td>--audit-directory DIR</td>
<td>Look for audit records in DIR, 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.</td>
</tr>
<tr>
<td>--query SQL</td>
<td>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.</td>
</tr>
<tr>
<td>--quiet</td>
<td>Reduce the amount of output for common operations.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>etc/globus-gram-job-manager.conf</td>
<td>GRAM5 job manager configuration. It includes the default path to the audit directory</td>
</tr>
<tr>
<td>etc/globus-gram-audit.conf</td>
<td>Audit configuration. It includes the information needed to contact the audit database.</td>
</tr>
</tbody>
</table>
**Name**
globus-job-manager — Execute and monitor jobs

**Synopsis**


**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 LRM` Execute jobs using the local resource manager named LRM.
- `-conf CONFIG_PATH` Read additional command-line arguments from the file **CONFIG_PATH**. If present, this must be the first command-line argument to the **globus-job-manager** program.
- `-globus-host-manufacturer MANUFACTURER` Indicate the manufacturer of the system which the jobs will execute on. This parameter sets the value of the `$GLOBUS_HOST_MANUFACTURER` RSL substitution to MANUFACTURER
- `-globus-host-cputype CPUTYPE` Indicate the CPU type of the system which the jobs will execute on. This parameter sets the value of the `$GLOBUS_HOST_CPUTYPE` RSL substitution to CPUTYPE
- `-globus-host-OSNAME OSNAME` Indicate the operating system type of the system which the jobs will execute on. This parameter sets the value of the `$GLOBUS_HOST_OSNAME` RSL substitution to OSNAME
- `-globus-host-osversion OSVERSION` Indicate the operating system version of the system which the jobs will execute on. This parameter sets the value of the `$GLOBUS_HOST_OSVERSION` RSL substitution to OSVERSION
- `-globus-gatekeeper-host HOST` 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 HOST
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-globus-gatekeeper-port PORT</td>
<td>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 PORT.</td>
</tr>
<tr>
<td>-globus-gatekeeper-subject SUBJECT</td>
<td>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 SUBJECT.</td>
</tr>
<tr>
<td>-home GLOBUS_LOCATION</td>
<td>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.</td>
</tr>
<tr>
<td>-target-globus-location TARGET_GLOBUS_LOCATION</td>
<td>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 TARGET_GLOBUS_LOCATION.</td>
</tr>
<tr>
<td>-history HISTORY_DIRECTORY</td>
<td>Configure the job manager to write job history files to HISTORY_DIRECTORY. These files are described in the FILES section below.</td>
</tr>
<tr>
<td>-scratch-dir-base SCRATCH_DIRECTORY</td>
<td>Configure the job manager to use SCRATCH_DIRECTORY as the default scratch directory root if a relative path is specified in the job RSL’s scratch_dir attribute.</td>
</tr>
<tr>
<td>-enable-syslog</td>
<td>Configure the job manager to write log messages via syslog. Logging is further controlled by the argument to the -log-levels parameter described below.</td>
</tr>
<tr>
<td>-log-pattern PATTERN</td>
<td>Configure the job manager to write log messages to files named by the string PATTERN. The PATTERN string may contain job-independent RSL substitutions such as $(HOME), $(LOGNAME), etc, as well as the special RSL substitution $(DATE) which will be resolved at log time to the date in YYYYM-MDD form.</td>
</tr>
<tr>
<td>-stdio-log LOG_DIRECTORY</td>
<td>Configure the job manager to write log messages to files in the LOG_DIRECTORY directory. This is a backward-compatible parameter, equivalent to -log-pattern LOG_DIRECTORY/gram_$(DATE).log.</td>
</tr>
<tr>
<td>-log-levels LEVELS</td>
<td>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</td>
</tr>
<tr>
<td>-state-file-dir STATE_DIRECTORY</td>
<td>Configure the job manager to write state files to STATE_DIRECTORY. 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.</td>
</tr>
<tr>
<td>-globus-tcp-port-range PORT_RANGE</td>
<td>Configure the job manager to restrict its TCP/IP communication to use ports in the range described by PORT_RANGE. This value is also made available in the job environment via the GLOBUS_TCP_PORT_RANGE environment variable.</td>
</tr>
<tr>
<td>-globus-tcp-source-range SOURCE_RANGE</td>
<td>Configure the job manager to restrict its TCP/IP communication to use source ports in the range described by SOURCE_RANGE. This value is also made</td>
</tr>
</tbody>
</table>
available in the job environment via the GLOBUS_TCP_SOURCE_RANGE environment variable.

-\texttt{x509-cert-dir} \texttt{TRUSTED_CERTIFICATE_DIRECTORY} Configure the job manager to search TRUSTED_CERTIFICATE_DIRECTORY for its list of trusted CA certificates and their signing policies. This value is also made available in the job environment via the X509_CERT_DIR environment variable.

-\texttt{cache-location} \texttt{GASS_CACHE_DIRECTORY} Configure the job manager to use the path GASS_CACHE_DIRECTORY for its temporary GASS-cache files. This value is also made available in the job environment via the GLOBUS_GASS_CACHE_DEFAULT environment variable.

-\texttt{k} 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.

-\texttt{extra-envvars} \texttt{VAR=VAL,...} 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 VAR=VAL pairs, where VAR is the variable name and VAL is the variable's value. If the value is not specified, then the value of the variable in the job manager's environment is used. This option may be present multiple times on the command-line or the job manager configuration file to append multiple environment settings.

-\texttt{seg-module} \texttt{SEG_MODULE} Configure the job manager to use the schedule event generator module named by SEG_MODULE 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.

-\texttt{audit-directory} \texttt{AUDIT_DIRECTORY} Configure the job manager to write audit records to the directory named by AUDIT_DIRECTORY. This records can be loaded into a database using the globus-gram-audit program.

-\texttt{globus-toolkit-version} \texttt{TOOLKIT_VERSION} Configure the job manager to use TOOLKIT_VERSION as the version for audit and usage stats records.

-\texttt{service-tag} \texttt{SERVICE_TAG} Configure the job manager to use SERVICE_TAG 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 untagged will be used.

-\texttt{disable-streaming} Configure the job manager to disable file streaming. This is propagated to the LRM script interface but has no effect in GRAM5.

-\texttt{disable-usagestats} Disable sending of any usage stats data, even if -usagestats-targets is present in the configuration.

-\texttt{usagestats-targets} \texttt{TAR\texttt{GET}} Send usage packets to a data collection service for analysis. The TARGET string consists of a comma-separated list of HOST:PORT combinations, each containing an optional list of data to send. See Usage Stats Packets\textsuperscript{1} for more information about the tags. Special tag strings of all (which enables all

\textsuperscript{1} http://confluence.globus.org/display/~bester/GRAM5+Usage+Stats+Packets
tags) and default may be used, or a sequence of characters for the various tags. If this option is not present in the configuration, then the default of usage-stats.globus.org:4810 is used.

- **condor-arch ARCH**

Set the architecture specification for condor jobs to be ARCH in job classified ads generated by the GRAM5 condor LRM script. This is required for the condor LRM but ignored for all others.

- **condor-os OS**

Set the operating system specification for condor jobs to be OS in job classified ads generated by the GRAM5 condor LRM script. This is required for the condor LRM but ignored for all others.

## 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_HTTP_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.

- **GLOBUS_USAGE_TARGETS**

  Default list of usagestats services to send usage packets to.

- **GLOBUS_TCP_PORT_RANGE**

  Default range of allowed TCP ports to listen on. The `-globus-tcp-port-range` command-line option overrides this.

- **GLOBUS_TCP_SOURCE_RANGE**

  Default range of allowed TCP ports to bind to. The `-globus-tcp-source-range` command-line option overrides this.

## Files

- **$HOME/.globus/job/HOST-NAME/LRM.TAG.red**

  Job manager delegated user credential.
Admin Tools

$HOME/.globus/job/HOST-NAME/LRM.TAG.lock
Job manager state lock file.

$HOME/.globus/job/HOST-NAME/LRM.TAG.pid
Job manager pid file.

$HOME/.globus/job/HOST-NAME/LRM.TAG.sock
Job manager socket for inter-job manager communications.

$HOME/.globus/job/HOST-NAME/JOB_ID/
Job-specific state directory.

$HOME/.globus/job/HOST-NAME/JOB_ID/stdin
Standard input which has been staged from a remote URL.

$HOME/.globus/job/HOST-NAME/JOB_ID/stdout
Standard output which will be staged from a remote URL.

$HOME/.globus/job/HOST-NAME/JOB_ID/stderr
Standard error which will be staged from a remote URL.

$HOME/.globus/job/HOSTNAME/JOB_ID/x509_user_proxy
Job-specific delegated credential.

$GLOBUS_LOCATION/tmp/gram_job_state/job.HOSTNAME.JOB_ID
Job state file.

$GLOBUS_LOCATION/tmp/gram_job_state/job.HOSTNAME.JOB_ID.lock
Job state lock file. In most cases this will be a symlink to the job manager lock file.

$GLOBUS_LOCATION/etc/globus-job-manager.conf
Default location of the global job manager configuration file.

$GLOBUS_LOCATION/etc/grid-services/jobmanager-LRM
Default location of the LRM-specific gatekeeper configuration file.

$GLOBUS_LOCATION/etc/globus/gram/jobmanager.rvf
Default location of the site-specific job manager RSL validation file.

$GLOBUS_LOCATION/etc/globus/gram/lrm.rvf
Default location of the site-specific job manager RSL validation file for the named lrm.

See Also
globusrun(1), globus-gatekeeper(8), globus-personal-gatekeeper(1), globus-gram-audit(8)
Name
globus-rvf-check — Edit a GRAM5 RSL validation file

Synopsis
globus-rvf-check [-h] [-help]
globus-rvf-check [-d] {FILENAME...}

Description
The globus-rvf-check command is a utility which checks the syntax of a RSL validation file, and prints out parse errors when encountered. It can also parse the RVF file contents and then dump file's contents to stdout, after canonicalizing values and quoting. The exit code of globus-rvf-check is 0 if all files specified on the command line exist and have no parse errors.

The full set of command-line options to globus-rvf-check consists of:

- **-h, --help**: Print command-line option summary and exit
- **-d**: Dump the RVF contents to stdout. In the output, Each file which is parsed will be prefixed by an RVF comment which contains the input filename. If not specified, globus-rvf-check just prints a diagnostic message to standard output indicating whether the file could be parsed.
Name
globus-rvf-edit — Edit a GRAM5 RSL validation file

Synopsis
globus-rvf-edit [-h]
globus-rvf-edit [[-s] [-l LRM] [-f PATH]]

Description
The `globus-rvf-edit` command is a utility which opens the default editor on a specified RSL validation file, and then, when editing completes, runs the `globus-rvf-check` command to verify that the RVF file syntax is correct. If a parse error occurs, the user will be given an option to rerun the editor or discard the modifications.

The full set of command-line options to `globus-rvf-edit` consists of:

- `-h` Print command-line option summary and exit
- `-s` Edit of the site-specific RVF file, which provides override values applicable to all LRMs installed on the system.
- `-l LRM` Edit the site-specific LRM overrides for the LRM named by the `LRM` parameter to the option.
- `-f PATH` Edit the RVF file located at `PATH`
Name
globus-scheduler-event-generator — Process LRM events into a common format for use with GRAM

Synopsis
globus-scheduler-event-generator -s LRM
[-t TIMESTAMP] [-d DIRECTORY]
[-b] [-p PIDFILE]

Description
The `globus-scheduler-event-generator` program processes information from a local resource manager to generate LRM-independent events which GRAM can use to track job state changes. Typically, the `globus-scheduler-event-generator` is started at system boot time for all LRM adapters which have been installed. The only required parameter to `globus-scheduler-event-generator` is `-s LRM`, which indicates what LRM-specific module to load. A list of available modules can be found by using the `globus-scheduler-event-generator-admin -l` command.

Other options control how the `globus-scheduler-event-generator` program runs and where its output goes. These options are:

- `-t TIMESTAMP` Start processing events which start at `TIMESTAMP` in seconds since the UNIX epoch. If not present, the `globus-scheduler-event-generator` will process events from the time it was started, and not look for historical events.

- `-d DIRECTORY` Write the event log to files in `DIRECTORY`, instead of printing them to standard output. Within `DIRECTORY`, logs will be named by the time when they were created in `YYYYMMDD` format.

- `-b` Run the `globus-scheduler-event-generator` program in the background.

- `-p PIDFILE` Write the process-id of `globus-scheduler-event-generator` to `PIDFILE`.

Files

/var/lib/globus/globus-seg-LRM/YYYYMMDD LRM-independent event log generated by `globus-scheduler-event-generator`

See Also
globus-scheduler-event-generator-admin(8), globus-job-manager(8)
Name
globus-scheduler-event-generator-admin — Manage SEG modules

Synopsis
globus-scheduler-event-generator-admin [-h]
globus-scheduler-event-generator-admin [-l]
globus-scheduler-event-generator-admin [-e MODULE]
globus-scheduler-event-generator-admin [-d MODULE]

Description
The globus-scheduler-event-generator-admin program manages SEG modules which are used by the globus-scheduler-event-generator to monitor a local resource manager or batch system for events. The globus-scheduler-event-generator-admin can list, enable, or disable specific SEG modules. The -h command-line option shows a brief usage message.

Listing SEG Modules
The -l command-line option to globus-scheduler-event-generator-admin will cause it to list all of the SEG modules which are available to be run by the globus-scheduler-event-generator. In the output, the service name will be followed by its status in brackets. Possible status strings are ENABLED and DISABLED.

Enabling SEG Modules
The -e MODULE command-line option to globus-scheduler-event-generator-admin will cause it to enable the module so that the init script for the globus-scheduler-event-generator will run it.

Disabling SEG Modules
The -d MODULE command-line option to globus-scheduler-event-generator-admin will cause it to disable the module so that it will not be started by the globus-scheduler-event-generator init script.

Files
/etc/globus/scheduler-event-generator Default location of enabled SEG modules.

See Also
globus-scheduler-event-generator(8)
Chapter 8. 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
Usage statistics collection by the Globus Alliance

- 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.

\(^1\)/toolkit/docs/latest-stable/Usage_Stats.html
Glossary

C

certificate
A public key plus information about the certificate owner bound together by the
digital signature of a CA. In the case of a CA certificate, the certificate is self
signed, i.e. it was signed using its own private key.

Condor
A Local Resource Manager mechanism supported by GRAM. See the Condor
Project Website\(^1\) for more information.
See Also Local Resource Manager.

F

fork
A POSIX-specific way of creating new processes. GRAM implements a basic
fork LRM Adapter which runs jobs on the GRAM head node.

G

Gatekeeper
A part of GRAM that runs as root and authenticates clients prior to starting the
Job Manager.

grid map file
A file containing entries mapping certificate subjects to local user names. This
file can also serve as a access control list for GSI enabled services and is typi-
cally found in /etc/grid-security/grid-mapfile. For more informa-
tion see the Gridmap section here.

J

Job Manager
A part of GRAM that runs as a local user and interfaces with a Local Resource
Manager for that user.

L

Local Resource Manager (LRM)
A system which controls access to a compute resource, such as a compute clus-
ter or parallel computer. Such systems provide batch execution interfaces, which
GRAM uses to execute jobs. Condor, Portable Batch System, GridEngine are
examples of local resource managers.
See Also Condor, Portable Batch System, Oracle GridEngine.

LRM Adapter
The interface code between a Local Resource Manager and GRAM.
In most cases, this consists of a Perl module that implements the
Globus::GRAM::JobManager class and a Scheduler Event Generator
module.
See Also Local Resource Manager.

\(^1\) http://www.cs.wisc.edu/condor/
## Glossary

### O

**Oracle GridEngine**  
See Also Local Resource Manager.

### P

**Portable Batch System (PBS)**  
A *Local Resource Manager* mechanism supported by GRAM. Multiple implementations of PBS exist: GRAM currently supports TORQUE. See also **TORQUE**.  
See Also Local Resource Manager, TORQUE.

**proxy certificate**  
A short lived certificate issued using a EEC. A proxy certificate typically has the same effective subject as the EEC that issued it and can thus be used in its place. GSI uses proxy certificates for single sign on and delegation of rights to other entities.  
For more information about types of proxy certificates and their compatibility in different versions of GT, see [http://dev.globus.org/wiki/Security/ProxyCert-Types](http://dev.globus.org/wiki/Security/ProxyCert-Types).

### S

**Scheduler Event Generator (SEG)**  
The Scheduler Event Generator (SEG) is a program which uses scheduler-specific monitoring modules to generate job state change events. Depending on scheduler-specific requirements, the SEG may need to run with privileges to enable it to obtain scheduler event notifications. As such, one SEG runs per scheduler resource. For example, on a host which provides access to both PBS and fork jobs, two SEGs, running at (potentially) different privilege levels will be running. One SEG instance exists for any particular scheduled resource instance (one for all homogeneous PBS queues, one for all fork jobs, etc). The SEG is implemented in an executable called the globus-scheduler-event-generator, located in the Globus Toolkit's libexec directory.

**Sun GridEngine (SGE)**  
The old name for *Oracle GridEngine*.  
See Also Oracle GridEngine.

### T

**TORQUE**  
See Also Portable Batch System.

---

Index

A
audit logging, 12

U
usage statistics, 33