

GT 4.2.1 Component Guide to Public Interfaces: GRAM4

GT 4.2.1 Component Guide to Public Interfaces: GRAM4

Table of Contents

1. APIs	1
1. Programming Model Overview	1
2. Component API	1
2. Services and WSDL	3
1. Protocol overview	3
2. Operations	3
3. Resource properties	5
4. WSDL and Schema Definition	10
I. GRAM4 Commands	12
globusrun-ws	13
3. Job description	20
A. Errors	21
Glossary	24

List of Tables

A.1. GRAM4 Errors	22
-------------------------	----

Chapter 1. APIs

1. Programming Model Overview

This component consists abstractly of two interfaces: the Managed Job Factory Port Type (MJFPT) and the Managed Job Port Type (MJPT).

In actuality there are three service/resource implementations, two of which implement the basic MJPT. The first one is the service which actually talks to a particular local resource manager to execute a process on the remote computer or cluster. This one is called a *Managed Executable Job Service (MEJS)* and its resource is called the Managed Executable Job Resource (MEJR). The second is a special implementation which accepts a multi-job description, breaks the description up into single-job descriptions, and then submits each of these so-called "sub-jobs" to an MEJS. This implementation is called the *Managed Multi Job Service (MMJS)*. Its resource is called the Managed Multi-Job Resource (MMJR)

Because of the fact that these two job services use the same port type, the API for accessing both the MEJR and the MMJR are identical. The *MJFS* creates the appropriate job resource depending on the factory resource used to qualify the operation call. Most of the factory resources represent local resource managers used by the MEJS (*PBS*, *LSF*, *Condor*). There is a special Multi factory resource which represents an abstract multi-job resource manager. The appropriate *job description* type is required for the two different types of managed job.

2. Component API

Java API Documentation Links (Javadoc)

- [Client API](#)¹
- [Auto-Generated Service Stubs and Persistence Data Objects API](#)²
- [Service API](#)³
- [Utilities API](#)⁴
- [Job Monitoring API](#)⁵

C API Documentation Links

- [All C APIs](#)⁶
- [GRAM4 Client Bindings](#)⁷ [[noframes](#)⁸]
- [WS-Rendezvous Client Bindings](#)⁹ [[noframes](#)¹⁰]

¹ http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_gram_client_java

² http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_gram_common_java

³ http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_gram_service_java

⁴ http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_gram_utils_java

⁵ http://www.globus.org/api/javadoc-4.0.0/globus_wsrf_gram_job_monitoring_common_java

⁶ <http://www.globus.org/api/c-globus-4.0/>

⁷ http://www.globus.org/api/c-globus-4.0/globus_c_gram_client_bindings/html/index.html

⁸ http://www.globus.org/api/c-globus-4.0/globus_c_gram_client_bindings/html/modules.html

⁹ http://www.globus.org/api/c-globus-4.0/globus_c_rendezvous_client_bindings/html/index.html

¹⁰ http://www.globus.org/api/c-globus-4.0/globus_c_rendezvous_client_bindings/html/modules.html

- [GRAM4 Scheduler Event Generator](#)¹¹ [[noframes](#)¹²]

¹¹ http://www.globus.org/api/c-globus-4.0/globus_scheduler_event_generator/html/index.html

¹² http://www.globus.org/api/c-globus-4.0/globus_scheduler_event_generator/html/modules.html

Chapter 2. Services and WSDL

1. Protocol overview

GRAM4 allows for remote execution and management of programs through the creation of a managed job. The management of the job is taken care of primarily by core toolkit functionality (WS-ResourceLifetime and WS-BaseN implementations). Please see [Java WS Core](#) on notifications and resource lifetime (destruction) for more information.

1.1. Managed Job Factory Service (MJFS)

A single MJFS is used to create all jobs for all users. For each local resource manager, a dedicated Managed Job Factory Resource (MJFR) enables the MJFS to publish information about the characteristics of the compute resource, for example:

- host information
- GridFTP URL (for file staging and streaming)
- compute cluster size and configuration, and so on...

In addition, there is a special MJFR which is used for creating MMJRs.

1.2. Managed Executable Job Service (MEJS)

A single *MEJS* is used to manage all executable jobs for all users. Each Managed Executable Job Resource (MEJR) enables the MEJS to publish information about the individual job the MEJR represents. This information can be accessed by querying the MEJS for the resource properties of a given MEJR, such as the:

- current job state
- stdout location
- stderr location
- exit code, and so on.

1.3. Managed Multi Job Service (MMJS)

A single MMJS is used to manage all multi-jobs for all users. Each Managed Multi-Job Resource (MMJR) enables the MMJS to publish information about the individual multi-job the MMJR represents. This information can be accessed by querying the MMJS for the resource properties of a given MMJR, such as the:

- current overall job state
- list of sub-job EPRs

2. Operations

There are just two operations defined in the GRAM port types (not counting the Rendezvous port type which is used for MPI job synchronization): "createManagedJob" in the Managed Job Factory port type, and "release" in the Managed

Job port type. All other operations (such as canceling/killing the job and querying for resource properties) are provided by the underlying WSRF implementation of the toolkit.

2.1. ManagedJobFactoryPortType

- `createManagedJob`: This operation creates either a MEJR or MMJR, subscribes the client for notifications if requested, and replies with one or two endpoint references (EPRs). The input of this operation consists of a *job description*, an optional initial termination time for the job resource, and an optional state notification subscription request.

The first EPR:

- is qualified with the identifier to the newly created MEJR or MMJR
- points to either the MEJS or MMJS.

The second EPR:

- is only present if a notification subscription was requested
- is qualified with the identifier to the newly created subscription resource
- points to the subscription manager service.

Using the optional subscription request provides an efficient means of subscribing to the newly created MEJR or MMJR without additional round-trip messages. Clients who subscribe afterwards must check the current status of the job, since the inherent race-condition means some state-changes may have occurred prior to the separate subscription request. In any event, there is a slight risk of lost notifications due to the lack of reliability guarantees in the notification delivery mechanism from WS-BaseNotification.

The `ManagedJobFactoryPortType` also has all the operations and publishes all the resource properties (via the MJFR) defined in the following `WS-ResourceProperties`¹ port types:

- `GetResourceProperty`
- `GetMultipleResourceProperties`
- `QueryResourceProperties`

2.2. ManagedJobPortType

This port type does not define any new operations itself, but has all the operations and publishes all the resource properties defined in the following port types:

ReleaseManagedJob port type:

- `release`: This operation takes no parameters and returns nothing. Its purpose is to release a hold placed on a state through the use of the "holdState" field in the job description. See the [do-main-specific GRAM4 component documentation](#)² for more information on the "holdState" field.

TerminateManagedJob port type:

¹ <http://www.ibm.com/developerworks/library/ws-resource/ws-resourceproperties.pdf>

² [../schemas/gram_job_description.html](#)

- `terminate`: This operation terminates a job. Depending on arguments and the state of the job this may result in immediate destruction of the job resource or in starting of clean up steps and resource destruction after the clean up is done.

*WS-ResourceProperties*³ port types:

- `GetResourceProperty`
- `GetMultipleResourceProperties`
- `QueryResourceProperties`

*WS-ResourceLifetime*⁴ port types:

- `ScheduledResourceTermination`

*WS-BaseNotification*⁵ port type:

- `NotificationProducer`

2.3. Managed Executable Job Port Type

This port type does not define any new operations. See "Resources Properties" under [Services and WSDL](#).

2.4. Managed Multi-Job Port Type

This port type does not define any new operations. See "Resources properties" below.

3. Resource properties

3.1. Managed Job Factory Port Type

- `{http://www.globus.org/namespaces/2008/03/gram/job}condorArchitecture`
Condor architecture label.
- `{http://www.globus.org/namespaces/2008/03/gram/job}condorOS`
Condor OS label.
- `{http://www.globus.org/namespaces/2008/03/gram/job}delegationFactoryEndpoint`
The endpoint reference to the delegation factory used to delegated credentials to the job.
- `{http://mds.globus.org/glue/ce/1.1}GLUECE`
GLUE data
- `{http://mds.globus.org/glue/ce/1.1}GLUECESummary`
GLUE data summary

³ <http://www.ibm.com/developerworks/library/ws-resource/ws-resourceproperties.pdf>

⁴ <http://www.ibm.com/developerworks/library/ws-resource/ws-resourcelifetime.pdf>

⁵ <ftp://www6.software.ibm.com/software/developer/library/ws-notification/WS-BaseN.pdf>

- `{http://www.globus.org/namespaces/2008/03/gram/job}globusLocation`
The location of the Globus Toolkit installation that these services are running under.
- `{http://www.globus.org/namespaces/2008/03/gram/job}hostCPUType`
The job host CPU architecture (i686, x86_64, etc...)
- `{http://www.globus.org/namespaces/2008/03/gram/job}hostManufacturer`
The host manufacturer name. May be "unknown".
- `{http://www.globus.org/namespaces/2008/03/gram/job}hostOSName`
The host OS name (Linux, Solaris, etc...)
- `{http://www.globus.org/namespaces/2008/03/gram/job}hostOSVersion`
The host OS version.
- `{http://www.globus.org/namespaces/2008/03/gram/job}localResourceManager`
The local resource manager type (i.e. *Condor*, Fork, *LSF*, Multi, *PBS*, etc...)
- `{http://www.globus.org/namespaces/2008/03/gram/job}availableLocalResourceManager`
All local resource managers that are configured in this GRAM4 instance
- `{http://www.globus.org/namespaces/2008/03/gram/job}jobTTLAfterProcessing`
Time in seconds a job resource will stay alive after a job finished processing in GRAM4 (including fileStageOut, fileCleanUp). When this time elapsed the job resource is destroyed and no longer be available for a client. A negative values means that the job resource will never be destroyed.
- `{http://www.globus.org/namespaces/2008/03/gram/job}maxJobLifetime`
Max time in seconds a user can set as initial lifetime in job submission or in subsequent setTerminationTime calls. A negative value means that there is no limit.
- `{http://mds.globus.org/metadata/2005/02}ServiceMetaDataInfo`
service start time, Globus Toolkit(R) version, service type name
- `{http://www.globus.org/namespaces/2008/03/gram/job}scratchBaseDirectory`
The directory recommended by the system administrator to be used for temporary job data.
- `{http://www.globus.org/namespaces/2008/03/gram/job}stagingDelegationFactoryEndpoint`
The endpoint reference to the delegation factory used to delegated credentials to the staging service (RFT).

3.2. Managed Job Port Type

- `{http://www.globus.org/namespaces/2008/04/rendezvous}Capacity`

Used for Rendezvous.

- {http://docs.oasis-open.org/wsrf/r1-2}CurrentTime
Time of creation.
- {http://docs.oasis-open.org/wsrf/rp-2}QueryExpressionDialect
From the QueryResourceProperties port type.
- {http://www.globus.org/namespaces/2008/03/gram/job/faults}fault
Faults (if generated) that happen along job processing and that cause a job to fail.
- {http://www.globus.org/namespaces/2008/03/gram/job/types}holding
Indicates whether a hold has been placed on this job.
- {http://www.globus.org/namespaces/2008/03/gram/job/types}localUserId
The job owner's local user account name.
- {http://www.globus.org/namespaces/2008/04/rendezvous}RegistrantData
Used for Rendezvous.
- {http://www.globus.org/namespaces/2008/04/rendezvous}RendezvousCompleted
Used for Rendezvous.
- {http://www.globus.org/namespaces/2008/03/gram/job/description}service-
LevelAgreement
A wrapper around fields containing the single-job and multi-job descriptions or *RSLs*. Only one of these sub-fields shall have a non-null value.
- {http://www.globus.org/namespaces/2008/03/gram/job/types}state
The current state of the job.
- {http://docs.oasis-open.org/wsrf/r1-2}TerminationTime
Time when the resource expires.
- {http://www.globus.org/namespaces/2008/03/gram/job/types}userSubject
The GSI certificate DN of the job owner.

3.3. Managed Executable Job Port Type

- {http://docs.oasis-open.org/wsrf/r1-2}CurrentTime
Time of creation.
- {http://docs.oasis-open.org/wsrf/r1-2}TerminationTime
Time when the resource expires.

- `{http://www.globus.org/namespaces/2008/03/gram/job/exec}credentialPath`
The path (relative to the job process) to the file containing the user proxy used by the job to authenticate out to other services.
- `{http://www.globus.org/namespaces/2008/03/gram/job/types}exitCode`
The exit code generated by the job process.
- `{http://www.globus.org/namespaces/2008/03/gram/job/faults}fault`
The fault (if generated) indicating the reason for failure of the job to complete.
- `{http://www.globus.org/namespaces/2008/03/gram/job/types}holding`
Indicates whether a hold has been placed on this job.
- `{http://www.globus.org/namespaces/2008/03/gram/job/types}localUserId`
The job owner's local user account name.
- `{http://www.globus.org/namespaces/2008/03/gram/job/exec}localJobId`
The job id(s) of the job in the local resource manager. Note that for Fork jobs these id's are prefixed with the uuid of the job.
- `{http://www.globus.org/namespaces/2008/03/gram/job/description}service-LevelAgreement`
A wrapper around fields containing the single-job and multi-job descriptions or *RSLs*. Only one of these sub-fields shall have a non-null value.
- `{http://www.globus.org/namespaces/2008/03/gram/job/types}state`
The current state of the job.
- `{http://www.globus.org/namespaces/2008/03/gram/job/exec}stderrURL`
A GridFTP URL to the file generated by the job which contains the stderr.
- `{http://www.globus.org/namespaces/2008/03/gram/job/exec}stdoutURL`
A GridFTP URL to the file generated by the job which contains the stdout.
- `{http://www.globus.org/namespaces/2008/03/gram/job/types}userSubject`
The GSI certificate DN of the job owner.
- `{http://www.globus.org/namespaces/2008/04/rendezvous}Capacity`
Used for Rendezvous.
- `{http://www.globus.org/namespaces/2008/04/rendezvous}RegistrantData`
Used for Rendezvous.
- `{http://www.globus.org/namespaces/2008/04/rendezvous}RendezvousCompleted`

Used for Rendezvous.

- `{http://docs.oasis-open.org/wsrf/rp-2}QueryExpressionDialect`

From the QueryResourceProperties port type.

3.4. Managed Multi-Job Port Type

- `{http://docs.oasis-open.org/wsrf/r1-2}CurrentTime`

Time of creation.

- `{http://docs.oasis-open.org/wsrf/r1-2}TerminationTime`

Time when the resource expires.

- `{http://www.globus.org/namespaces/2008/03/gram/job/faults}fault`

The fault (if generated) indicating the reason for failure of the job to complete.

- `{http://www.globus.org/namespaces/2008/03/gram/job/types}holding`

Indicates whether a hold has been placed on this job.

- `{http://www.globus.org/namespaces/2008/03/gram/job/types}localUserId`

The job owner's local user account name.

- `{http://www.globus.org/namespaces/2008/03/gram/job/description}service-LevelAgreement`

A wrapper around fields containing the single-job and multi-job descriptions or *RSLs*. Only one of these sub-fields shall have a non-null value.

- `{http://www.globus.org/namespaces/2008/03/gram/job/types}state`

The current state of the job.

- `{http://www.globus.org/namespaces/2008/03/gram/job/multi}subJobEndpoint`

A set of endpoint references to the sub-jobs created by this multi-job.

- `{http://www.globus.org/namespaces/2008/03/gram/job/types}userSubject`

The GSI certificate DN of the job owner.

- `{http://www.globus.org/namespaces/2008/04/rendezvous}Capacity`

Used for Rendezvous.

- `{http://www.globus.org/namespaces/2008/04/rendezvous}RegistrantData`

Used for Rendezvous.

- `{http://www.globus.org/namespaces/2008/04/rendezvous}RendezvousCompleted`

Used for Rendezvous.

- `{http://docs.oasis-open.org/wsrf/rp-2}QueryExpressionDialect`

From the QueryResourceProperties port type.

4. WSDL and Schema Definition

WSDL links:

- [ManagedJobFactoryPortType](#)⁶
- [ManagedJobPortType](#)⁷
- [ReleaseManagedJobPortType](#)⁸
- [TerminateManagedJobPortType](#)⁹
- [ManagedExecutableJobPortType](#)¹⁰
- [ManagedMultiJobPortType](#)¹¹

Schema links:

- [CAS Types](#)¹²
- [File System Mapping Config Schema](#)¹³
- GLUE Schema
 - [Batch Providers](#)¹⁴
 - [Compute Element](#)¹⁵
 - [Metadata](#)¹⁶
- [Job Description Schema](#)¹⁷
- [Managed Job Faults Schema](#)¹⁸
- [Managed Job Types Schema](#)¹⁹
- [RFT Types Schema](#)²⁰

⁶ http://viewcvs.globus.org/viewcvs.cgi/ws-gram/common/schema/gram/jdd/managed_job_factory_port_type.wsdl?revision=1.6

⁷ http://viewcvs.globus.org/viewcvs.cgi/ws-gram/common/schema/gram/jdd/managed_job_port_type.wsdl?revision=1.9

⁸ http://viewcvs.globus.org/viewcvs.cgi/ws-gram/common/schema/gram/jdd/release_managed_job_provider_port_type.wsdl?revision=1.2

⁹ http://viewcvs.globus.org/viewcvs.cgi/ws-gram/common/schema/gram/jdd/terminate_managed_job_provider_port_type.wsdl?revision=1.3

¹⁰ http://viewcvs.globus.org/viewcvs.cgi/ws-gram/common/schema/gram/jdd/managed_executable_job_port_type.wsdl?revision=1.9

¹¹ http://viewcvs.globus.org/viewcvs.cgi/ws-gram/common/schema/gram/jdd/managed_multi_job_port_type.wsdl?revision=1.9

¹² [../schemas/cas_types.html](#)

¹³ [../schemas/gram_fs_map.html](#)

¹⁴ [../schemas/batchproviders.html](#)

¹⁵ [../schemas/glue_ce.html](#)

¹⁶ [../schemas/metadata.html](#)

¹⁷ [../schemas/gram_job_description.html](#)

¹⁸ [../schemas/mj_faults.html](#)

¹⁹ [../schemas/mj_types.html](#)

²⁰ [../schemas/rft_types.html](#)

- [WS Addressing Schema](#)²¹
- [WS Base Faults Schema](#)²²

²¹ ../schemas/ws_addressing.html
²² ../schemas/ws_base_faults.html

GRAM4 Commands

Name

globusrun-ws -- Official job submission client for GRAM4

```
globusrun-ws -submit [-batch] [-quiet] [-no-cleanup] [-streaming] [-streaming-out filename] [-streaming-err filename] [-host-authz] [-self-authz] [-subject-authz subject name] [-private] [-http-timeout milliseconds] [-debug] [-allow-ipv6] [-passive] [-nodcau] [[-factory-epr-file filename] [[-factory contact] | [-factory-type type]]] [[-submission-id uuid] | [-submission-id-file filename]] [-submission-id-output-file filename] [-job-epr-output-file filename] [-job-delegate] [-staging-delegate] [-job-credential-file filename] [-staging-credential-file filename] [-transfer-credential-file filename] [-termination [+HH:MMmm/dd/yyyy HH:MM] ] [[-job-description-file filename] | [-job-command [--] program arg ...]]
globusrun-ws -validate -job-description-file filename
globusrun-ws -monitor -job-epr-file filename [-quiet] [-no-cleanup] [-streaming] [-streaming-out filename] [-streaming-err filename] [-host-authz] [-self-authz] [-subject-authz subject name] [-private] [-http-timeout milliseconds] [-debug] [-allow-ipv6] [-passive] [-nodcau]
globusrun-ws -status -job-epr-file filename [-host-authz] [-self-authz] [-subject-authz subject name] [-private] [-http-timeout milliseconds] [-debug]
globusrun-ws -kill -job-epr-file filename [-host-authz] [-self-authz] [-subject-authz subject name] [-private] [-http-timeout milliseconds] [-debug]
globusrun-ws -help
globusrun-ws -usage [-submit] [-validate] [-monitor] [-status] [-kill]
globusrun-ws -version(s)
```

Description

globusrun-ws (GRAM4 client) is a program for submitting and managing jobs to a local or remote job host. GRAM4 provides secure job submission to many types of *job scheduler* for users who have the right to access a job hosting resource in a Grid environment. All GRAM4 submission options are supported transparently through the embedded request document input. globusrun-ws offers additional features to fetch job output files incrementally during the run as well as to automatically delegate credentials needed for certain optional GRAM4 features. Online and batch submission modes are supported with reattachment (recovery) for jobs whether they were started with this client or another GRAM4 client application.

Command options

Quiet mode

A variety of protocol status messages, warning messages, and output data may be printed to standard output and error under multiple command modes. The *quiet mode* suppresses all but fatal standard error messages in order to have clean outputs for use in scripting or with the *streaming output mode* where application output is retrieved and output.

-q, -quiet If supplied, all non-fatal status and protocol-related messages are suppressed.

Debug mode

-dbg, -debug If supplied, all soap messages and ftp control messages will be displayed on stderr.

Protocol Options

Service authorization

Usually, secure communication includes mutual authentication. In addition to the service authorizing the client for the requested operation(s), an authorization decision is made by the client to determine whether the remote service is the one intended.

- host, -host-authz The GSI "host authorization" rule is used to verify that the service is using a host credential appropriate for the underlying service address information. This is the default.

- self, -self-authz The GSI "self authorization" rule is used to verify that the service is using a (proxy) credential derived from the same identity as the client's.

- subject, -subject-authz subject name The service must be using a credential with the exact subject name provided by this option.

Security Protocol

The client uses secure transport for all https endpoints and secure message for http. Secure conversation is currently unsupported.

- p, -private If supplied, privacy-protection is enabled between globusrun-ws and GRAM4 or GridFTP services. It is a fatal error to select privacy protection if it is not available due to build options or other security settings. **Note:** Currently only supported with https endpoints.

Timeouts

- T, -http-timeout milli-seconds Set timeout for HTTP socket, in milliseconds, for all Web services interactions. The default value is 120000 (2 minutes).

Signal handling

- n, -no-cleanup If supplied, the default behavior of trapping interrupts (SIG_INTR) and cancelling the job is disabled. Instead, the interrupt simply causes the tool to exit without affecting the ManagedJob resource.

Submit options

- submit The -submit command submits (or *resubmits*) a job to a job host using an XML-based job description document. The -submit command can submit jobs in one of three output modes: batch, interactive, or interactive-streaming.

Output Mode

The user can select several tool behaviors following submission. In *batch mode*, the tool prints the resulting ManagedJob EPR as the sole standard output (unless in *quiet mode*) and exits. In *interactive mode*, the tool keeps running in order to monitor job status. Interactive mode is qualitatively equivalent to a batch-mode submission immediately followed a second invocation of globusrun-ws using the -monitor command. In interactive mode, an optional *streaming mode* where job output files are fetched and output from globusrun-ws.

- b, -batch If supplied, the batch mode is enabled. The default is interactive mode. The tool prints the resulting ManagedJob EPR as the sole standard output (unless in quiet mode) and exits.
- s, -streaming The standard output and standard error files of the job are monitored and data is written to the corresponding output of globusrun-ws. The standard output will contain ONLY job output data, while the standard error may be a mixture of job error output as well as globusrun-ws messages, unless the *quiet mode* is also enabled.
- Streaming output depends on the ability to access job outputs via GridFTP. If -streaming mode is selected and the *job description* does not already specify output file redirection for the job host, then globusrun-ws adds unique output file name redirections and automatic cleanup directives to the job description.
- If you are using -batch mode, but intend to use -streaming with -monitor, you may want to still include -streaming. -streaming always introduces a 'CleanUp Hold' state which ensures that all the data is streamed before the files are destroyed. If you do use -streaming with -batch, you **must** come back with -monitor so the hold can be released.
- This option implies -staging-delegate if the stdout and stderr entries are not specified in the job description.
- so, -stdout-file filename append stdout out stream to the specified file instead of to stdout.
- se, -stderr-file filename append stderr out stream to the specified file instead of to stderr.

Streaming Options

Streaming makes use of GridFTP client calls to retrieve user data. The following options apply to such transfers.

- ipv6, -allow-ipv6 Allow streaming transfers to use IPV6.
- passive Force streaming transfers to use MODE S to allow for passive mode transfers. (Useful if you're behind a firewall, but expensive because there is no connection caching).
- nodcau Disable data channel authentication on streaming transfers

Factory information

Addressing information for the ManagedJobFactory target of this submission must be provided. If neither option is specified, and no EPR is supplied in the job description, then "-factory localhost -factory-type fork" is assumed.

- Ff, -factory-epr-file filename If supplied, this option causes the EPR for the ManagedJobFactory to be read from the given file. This EPR is used as the service endpoint for submission of the job.
- F, -factory contact If supplied, this option causes an EPR to be constructed using ad-hoc methods that depend on GT implementation details. For interoperability to other implementations of GRAM4_, the -factory-epr-file option should be used instead.

[protocol://][{hostname|hostaddr}][:port][/service]

Default values form the following contact information if not overridden:

https://localhost:8443/wsrf/services/ManagedJobFactoryService

-Ft, -factory-type type In the absence of `-factory-epr-file`, this option refines the behavior of the `-factory` option to select a specific type of scheduler. The default is "Fork" for single jobs and "Multi" for *multijobs*.

Job description

A description of the job to be submitted must be provided with the `-submit` command, either using the GRAM4 XML description syntax or a simpler Unix command and argument list.

-f, -job-description-file filename If supplied, this option causes the job description to be read from the given file. This description is modified according to the other options and passed in the GRAM4 submission messages. The root element of this file must be 'job' for a single job or 'multiJob' for a multijob.

-c, -job-command [--] prog [arg ...] If supplied, this option take all remaining globusrun-ws arguments as its arguments; therefore it must appear last among globusrun-ws options. This option causes globusrun-ws to generate a simple job description with the named program and arguments.

Submission ID

A submission ID may be used in the GRAM4 protocol for robust reliability in the face of message faults or other transient errors to ensure that at most one instance of a job is executed, i.e. to prevent accidental duplication of jobs under rare circumstances with client retry on failure. The globusrun-ws tool always uses this feature, requiring either a submission ID to be passed in as input or a new unique ID to be created by the tool itself. If a new ID is created, it should be captured by the user who wishes to exploit this reliability interface. The ID in use, whether created or passed as input, will be written to the optional output file when provided, as well as to the standard error output unless the *quiet mode* is in effect.

If a user is unsure whether a job was submitted successfully, he should resubmit using the same ID as was used for the previous attempt.

-I, -submission-id ID If supplied, this option causes the job to be submitted using the given ID in the reliability protocol.

-If, -submission-id-file filename If supplied, this option causes the ID to be read from the given file. It is an error to use both mechanisms to provide an input ID.

-Io, -submission-id-output-file file-name If supplied, the ID in use is written to the given file, whether this ID was provided by the user or given by one of the above input options.

Job EPR output

A successful submission will create a new ManagedJob resource with its own unique EPR for messaging. The globusrun-ws tool will output this EPR to a file when requested and as the sole standard output when running in batch mode. When running in streaming output mode, it is possible that the EPR will not be output and the user's only recourse is to submit again with the same submission ID and job request in order to reattach to the existing job.

-o, -job-epr-output-file filename If supplied, the created ManagedJob EPR will be written to the given file following successful submission. The file will not be written if the submission fails.

Delegation

The job description supports the optional identification of delegated credentials for use by the GRAM4 services. These features are passed through globusrun-ws without modification. However, globusrun-ws can also perform delegation

and construct these optional request elements before submitting it to the service. The only delegation performed by default (if an endpoint does not already exist) is the multijob level jobCredential.

- J, -job-delegate If supplied AND the job description does not already provide a jobCredential element, globusrun-ws will delegate the client credential to GRAM4 and introduce the corresponding element to the submission input.
- S, -staging-delegate If supplied AND the job description does include staging or cleanup directives AND the job description does not already provide the necessary stagingCredential or transferCredential element(s), globusrun-ws will delegate the client credential to GRAM4 and RFT, and introduce the corresponding elements to the submission input.

This option is implied by -streaming
- Jf, -job-credential-file filename: If supplied AND the job description does not already provide a jobCredential element, globusrun-ws will copy the supplied epr into the job description. This should be an epr returned from the DelegationFactoryService intended for use by the job (or, in the case of a multijob, for authenticating to the subjobs).

note: for multijob descriptions, only the top level jobCredential will be copied into.
- Sf, -staging-credential-file filename: If supplied AND the job description does not already provide a stagingCredential element, globusrun-ws will copy the supplied epr into the job description. This should be an epr returned from the DelegationFactoryService intended for use with the RFT service associated with the ManagedJobService.

note: this option is ignored for multijobs.
- Tf, -transfer-credential-file filename: If supplied, globusrun-ws will copy the epr into each of the stage in, stage out, and cleanup elements that do not already contain a transferCredential element. This should be an epr returned from the DelegationFactoryService intended for use by RFT to authenticate with the target gridftp server.

note: this option is ignored for multijobs.

Lifetime

The ManagedJob resource supports lifetime management in the form of a scheduled destruction. The default lifetime requested by the client is infinite, subject to server policies.

- term, -termination mm/dd/yyyy Set an absolute termination time.
HH:MM
- term, -termination +HH:MM Set a termination time relative to the successful creation of the job. The default is +24:00

Validate options

- validate The -validate command checks the job description for syntax errors and a subset of semantic errors without making any service requests.

Job description

-f, -job-description-file filename This option causes the job description to be read from the given file. This description is checked for validity.

Monitor options

-monitor The -monitor command attaches to an existing job in interactive or interactive-streaming output modes.

Job

Addressing information for the ManagedJob target of this command must be provided.

-j, -job-epr-file filename If supplied, this option causes the EPR for the ManagedJob to be read from the given file. This EPR is used as the endpoint for service requests.

Output mode

In the default *interactive mode*, the tool keeps running in order to monitor job status. In the optional *interactive-streaming mode*, the job output files are fetched and output from globusrun-ws as well.

-s, -streaming See Output mode under Submit Options above for details on streaming.

Status options

-status The -status command reports the current state of the job and exits. See the [External States of the Managed Job Services](#) section of the developer guid for information on valid job states.

See the Job options for the -monitor command.

Kill options

-kill The -kill command requests the immediate cancellation of the job and exits.

Help options

-help Outputs an overview of the commands and features of the command.

Usage options

-usage Outputs brief usage information for the command.

Version options

-version Outputs version information for the command.

Job Handling

For every job that globusrun-ws delegates a credential, globusrun will augment the user's job description, adding annotations that will later tell globusrun-ws to destroy the credential after the job has been destroyed. Below are 2 job annotation examples. globusrun-ws only delegated the job cred...

```
<extensions>
<globusrunAnnotation>
<automaticJobDelegation>true</automaticJobDelegation>
<automaticStagingDelegation>false</automaticStagingDelegation>
<automaticStageInDelegation>false</automaticStageInDelegation>
<automaticStageOutDelegation>false</automaticStageOutDelegation>
<automaticCleanUpDelegation>false</automaticCleanUpDelegation>
</globusrunAnnotation>
</extensions>
```

globusrun-ws delegated the job, staging and stage in cred...

```
<extensions>
<globusrunAnnotation>
<automaticJobDelegation>true</automaticJobDelegation>
<automaticStagingDelegation>true</automaticStagingDelegation>
<automaticStageInDelegation>true</automaticStageInDelegation>
<automaticStageOutDelegation>false</automaticStageOutDelegation>
<automaticCleanUpDelegation>false</automaticCleanUpDelegation>
</globusrunAnnotation>
</extensions>
```

Environment

X509_USER_PROXY Overrides the default selection of user credentials when using GSI security.

Exit Codes

The client returns negative error codes for client errors, 0 for success, and positive error codes from the submitted job (where possible)

Chapter 3. Job description

Jobs are described in GRAM4's job description language. For detailed schema information check the [Job Description Schema documentation](#)¹. For more information and examples please check the [Job description overview section](#) in the [Users's guide](#).

¹ [../schemas/gram_job_description.html](#)

Appendix A. Errors

Table A.1. GRAM4 Errors

Error Code	Definition	Possible Solutions
globusrun-ws - error querying job state	<p>During job submission, an error like this occurs:</p> <pre>globusrun-ws failed: Delegating user creden- tials...Done. Sub- mitting job...Done. Job ID: xxxx Termin- ation time: xxxx Current job state: Unsubmitted globus- run-ws: Error querying job state globus_soap_mes- sage_module: Failed sending request ManagedJobPort- Type_GetMul- tipleResourceProper- ties. globus_xio: An end of file oc- curred</pre>	<p>Periodically, globusrun-ws will query the GRAM service to check on the job state. The "End of file" indicates that the GRAM server dropped a connection when globusrun-ws tried to read a response. This could be caused by temporary network issues between the client and service, or possibly caused by an overloaded service host.</p>
globusrun-ws - error querying job state	<p>During job submission, an error like this occurs:</p> <pre>globusrun-ws failed: Delegating user creden- tials...Done. Sub- mitting job...Done. Job ID: xxxx Termin- ation time: xxxx Current job state: Unsubmitted globus- run-ws: Error querying job state globus_soap_mes- sage_module: Failed sending request ManagedJobPort- Type_GetMul- tipleResourceProper- ties. globus_xio: System error in read: Connection reset by peer glo- bus_xio: A system call failed: Connec- tion reset by peer</pre>	<p>Periodically, globusrun-ws will query the GRAM service to check on the job state. The</p> <pre>System error in read: Connection reset by peer</pre> <p>indicates that the GRAM server dropped the connection while trying to write the response. This could be caused by temporary network issues between the client and service, or possibly caused by an overloaded service host.</p>

Error Code	Definition	Possible Solutions
glo- bus- run- ws - error sub- mit- ting job	During job submission, an error like this occurs: globusrun-ws -Ft PBS -F ht- tps://host.teragrid.org:8444 -submit -b -f /tmp/wsgram.rsl -o /tmp/wsgram.epr failed: Submitting job...Failed. globusrun-ws: Error submitting job globus_soap_message_module: Failed sending request ManagedJobFactoryPortType_createManagedJob. globus_xio: Operation was canceled globus_xio: Operation timed out	The Operation timed out indicates that the GRAM service was not able to accept the job request and respond in time. This could be caused by temporary network issues between the client and service, or possibly caused by an overloaded service host.

Glossary

C

Condor A job scheduler mechanism supported by GRAM. See <http://www.cs.wisc.edu/condor/> for more information.

J

job description Term used to describe a GRAM4 job for GT4.

job scheduler See the term [scheduler](#)⁴.

L

LSF A job scheduler mechanism supported by GRAM.
For more information, see <http://www.platform.com/Products/Platform.LSF.Family/Platform.LSF/>⁷.

M

Managed Executable Job Service (MEJS) [FIXME]

Managed Job Factory Service (MJFS) [FIXME]

Managed Multi Job Service (MMJS) [FIXME]

multijob A job that is itself composed of several executable jobs; these are processed by the MMJS subjob.

See also [MMJS subjob](#)¹⁰.

P

Portable Batch System (PBS) A job scheduler mechanism supported by GRAM. For more information, see <http://www.openpbs.org>.

R

Resource Specification Language (RSL) Term used to describe a GRAM job for GT2 and GT3. (Note: This is not the same as RLS - the Replica Location Service)

⁴ #scheduler

⁷ <http://www.platform.com/Products/Platform.LSF.Family/Platform.LSF/>

¹⁰ #mmjs-subjob