

globus scheduler event generator Reference Manual

2.1

Generated by Doxygen 1.3.5

Sat Feb 6 11:54:30 2010

Contents

1	Globus Scheduler Event Generator	1
2	globus scheduler event generator Module Index	1
3	globus scheduler event generator Page Index	2
4	globus scheduler event generator Module Documentation	2
5	globus scheduler event generator Page Documentation	5

1 Globus Scheduler Event Generator

The Scheduler Event Generator (SEG) is a program which uses scheduler-specific monitoring modules to generate job state change events. At the SEG level, the state change events correspond to changes in any jobs which are managed by the scheduler, even if they do not correspond to jobs initiated by the Managed Job Service. These state change events are propagated to the Job State Monitor.

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.

When executed, the SEG is able to start issuing events from some time in the past. The SEG will, in general, not require any persistent state between invocations. 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. It is invoked with the following command line:

```
globus-scheduler-event-generator -s SCHEDULER_NAME [-t TIMESTAMP]
```

It produces events in the format described in [SEG Protocol](#) section of this document on the standard output of the process.

When begun, it loads the scheduler module for the scheduler named on the command line and then defers to it for most functionality. When it detects an error writing to stdout or reading stdin, it terminates. The scheduler specific code uses the SEG API to emit events to the JSM.

Scheduler implementations use [SEG API](#) to send messages to the JSM.

2 globus scheduler event generator Module Index

2.1 globus scheduler event generator Modules

Here is a list of all modules:

Scheduler Implementation API	2
------------------------------	---

3 globus scheduler event generator Page Index

3.1 globus scheduler event generator Related Pages

Here is a list of all related documentation pages:

SEG Protocol

5

4 globus scheduler event generator Module Documentation

4.1 Scheduler Implementation API

Scheduler-specific SEG module implementations use this API to issue events to the Job State Monitor.

Enumerations

- enum `globus_scheduler_event_generator_error_t`
 - `GLOBUS_SEG_ERROR_TYPE_NULL` 1024,
 - `GLOBUS_SEG_ERROR_TYPE_ALREADY_SET`
 - `GLOBUS_SEG_ERROR_TYPE_INVALID_MODULE`
 - `GLOBUS_SEG_ERROR_TYPE_INVALID_FORMAT`
 - `GLOBUS_SEG_ERROR_TYPE_OUT_OF_MEMORY`
 - `GLOBUS_SEG_ERROR_TYPE_LOADING_MODULE`

Functions

- `globus_result_t globus_scheduler_event_activate(time_t timestamp, const char *jobid)`
- `globus_result_t globus_scheduler_event_pending(time_t timestamp, const char *jobid)`
- `globus_result_t globus_scheduler_event_active(time_t timestamp, const char *jobid)`
- `globus_result_t globus_scheduler_event_failed(time_t timestamp, const char *jobid, int failure_code)`
- `globus_result_t globus_scheduler_event_done(time_t timestamp, const char *jobid, int exit_code)`
- `globus_result_t globus_scheduler_event_generator_get_times(time_t timestamp)`

4.1.1 Detailed Description

Scheduler-specific SEG module implementations use this API to issue events to the Job State Monitor.

Events occur whenever a job is placed in the scheduler's queue (PENDING), begins execution (ACTIVE), terminates successfully (DONE), or ends abnormally (FAILED).

A SEG module should register an event with the Globus event driver (most likely using either the Globus Callback or Globus XIO interfaces) in its activation function and then return. All events should be triggered from callbacks. When the SEG detects that it should terminate, it will deactivate the SEG module it started. The SEG module should wait for any outstanding callbacks to subside and before returning from its deactivation function to ensure that all events will be properly dispatched. After deactivation is complete, the SEG will unload the shared module and terminate.

4.1.2 Enumeration Type Documentation

4.1.2.1 `enum globus_scheduler_event_generator_error_t`

Error types used by the SEG.

Enumeration values:

`GLOBUS_SEG_ERROR_TYPE_NULL` NULL Parameter.

`GLOBUS_SEG_ERROR_TYPE_ALREADY_SET` Already called a one-time function.

`GLOBUS_SEG_ERROR_TYPE_INVALID_MODULE` Shared module missing descriptor.

`GLOBUS_SEG_ERROR_TYPE_INVALID_FORMAT` Invalid printf format for SEG protocol message.

`GLOBUS_SEG_ERROR_TYPE_OUT_OF_MEMORY` Out of memory.

`GLOBUS_SEG_ERROR_TYPE_LOADING_MODULE` Unable to load scheduler module.

4.1.3 Function Documentation

4.1.3.1 `globus_result_t globus_scheduler_event (const char *format, ...)`

Send an arbitrary SEG notification.

Parameters:

`format` Printf-style format of the SEG notification message

... Varargs which will be interpreted as per format.

Return values:

`GLOBUS_SUCCESS` Scheduler message sent or queued.

`GLOBUS_SEG_ERROR_NULL` Null format.

`GLOBUS_SEG_ERROR_INVALID_FORMAT` Unable to determine length of formatted string.

4.1.3.2 `globus_result_t globus_scheduler_event_pending (time_t timestamp, const char *jobid)`

Send a job pending event to the JobSchedulerMonitor implementation.

Parameters:

`timestamp` Timestamp to use for the event. If set to 0, the time which this function was called is used.

`jobid` String indicating the scheduler-specific name of the job.

Return values:

`GLOBUS_SUCCESS` Scheduler message sent or queued.

`GLOBUS_SEG_ERROR_NULL` Null jobid.

`GLOBUS_SEG_ERROR_INVALID_FORMAT` Unable to determine length of formatted string.

4.1.3.3 `globus_result_t globus_scheduler_event_active (time_t timestamp, const char *jobid)`

Send a job active event to the JobSchedulerMonitor implementation.

Parameters:

`timestamp` Timestamp to use for the event. If set to 0, the time which this function was called is used.

jobid String indicating the scheduler-specific name of the job.

Return values:

GLOBUS_SUCCESS Scheduler message sent or queued.

GLOBUS_SEG_ERROR_NULL Null jobid.

GLOBUS_SEG_ERROR_INVALID_FORMAT Unable to determine length of formatted string.

4.1.3.4 globus_result_t globus_scheduler_event_failed (time_t timestamp, const char * jobid, int failure_code)

Send a job failed event to the JobSchedulerMonitor implementation.

Parameters:

timestamp Timestamp to use for the event. If set to 0, the time which this function was called is used.

jobid String indicating the scheduler-specific name of the job.

failure_code Failure code of the process if known.

Return values:

GLOBUS_SUCCESS Scheduler message sent or queued.

GLOBUS_SEG_ERROR_NULL Null jobid.

GLOBUS_SEG_ERROR_INVALID_FORMAT Unable to determine length of formatted string.

4.1.3.5 globus_result_t globus_scheduler_event_done (time_t timestamp, const char * jobid, int exit_code)

Send a job done event to the JobSchedulerMonitor implementation.

Parameters:

timestamp Timestamp to use for the event. If set to 0, the time which this function was called is used.

jobid String indicating the scheduler-specific name of the job.

exit_code Exit code of the process if known.

Return values:

GLOBUS_SUCCESS Scheduler message sent or queued.

GLOBUS_SEG_ERROR_NULL Null jobid.

GLOBUS_SEG_ERROR_INVALID_FORMAT Unable to determine length of formatted string.

4.1.3.6 globus_result_t globus_scheduler_event_generator_get_timestamp (time_t * timestamp)

Get the timestamp for the earliest event an SEG module should send.

Parameters:

timestamp Pointer to a time_t which will be set to the timestamp passed to the SEG executable. The module should not send any events which occur prior to this timestamp.

Return values:

GLOBUS_SEG_ERROR_NULL Null timestamp.

GLOBUS_SUCCESS Timestamp value updated. If the timestamp was not set on the SEG command-line, then the value pointed to by timestamp will be set to 0.

5 globus scheduler event generator Page Documentation

5.1 SEG Protocol

The general form for the SEG protocol messages is

MESSAGE-TYPE;TIMESTAMP;message-type-specific content

- MESSAGE-TYPE is a three-digit integer. The JSM will parse the message contents based on the message type.
- TIMESTAMPIs an unsigned value indicating seconds since the UNIX epoch.

5.1.1 Message Types

5.1.1.1 001 - Job State Change

Message Format: 001;TIMESTAMP;JOBID;STATE;EXIT_CODE

Message Type Specific Content:

JOBID local scheduler-specific job id

STATE new job state (integer as per the GRAM protocol constants)

EXIT_CODE job exit code if STATE is done or failed.

Index

globus_scheduler_event
 seg_api [3](#)
globus_scheduler_event_active
 seg_api [3](#)
globus_scheduler_event_done
 seg_api [4](#)
globus_scheduler_event_failed
 seg_api [4](#)
globus_scheduler_event_generator_error_t
 seg_api [3](#)
globus_scheduler_event_generator_get_timestamp
 seg_api [4](#)
globus_scheduler_event_pending
 seg_api [3](#)
GLOBUS_SEG_ERROR_TYPE_ALREADY_SET
 seg_api [3](#)
GLOBUS_SEG_ERROR_TYPE_INVALID_-
 FORMAT
 seg_api [3](#)
GLOBUS_SEG_ERROR_TYPE_INVALID_-
 MODULE
 seg_api [3](#)
GLOBUS_SEG_ERROR_TYPE_LOADING_-
 MODULE
 seg_api [3](#)
GLOBUS_SEG_ERROR_TYPE_NULL
 seg_api [3](#)
GLOBUS_SEG_ERROR_TYPE_OUT_OF_-
 MEMORY
 seg_api [3](#)

Scheduler Implementation API [2](#),
seg_api
 GLOBUS_SEG_ERROR_TYPE_ALREADY_-
 SET, [3](#)
 GLOBUS_SEG_ERROR_TYPE_INVALID_-
 FORMAT, [3](#)
 GLOBUS_SEG_ERROR_TYPE_INVALID_-
 MODULE, [3](#)
 GLOBUS_SEG_ERROR_TYPE_LOADING_-
 MODULE, [3](#)
 GLOBUS_SEG_ERROR_TYPE_NULL, [3](#)
 GLOBUS_SEG_ERROR_TYPE_OUT_OF_-
 MEMORY, [3](#)
seg_api
 globus_scheduler_event, [3](#)
 globus_scheduler_event_active, [3](#)
 globus_scheduler_event_done, [4](#)
 globus_scheduler_event_failed, [4](#)
 globus_scheduler_event_generator_error_t, [3](#)
 globus_scheduler_event_generator_get_-
 timestamp, [4](#)
 globus_scheduler_event_pending, [3](#)