

globus gram client Reference Manual

12.4

Generated by Doxygen 1.4.7

Thu Nov 21 15:25:02 2013

Contents

1	Resource Management Client API	1
2	globus gram client Module Index	1
3	globus gram client Data Structure Index	1
4	globus gram client Page Index	2
5	globus gram client Module Documentation	2
6	globus gram client Data Structure Documentation	28
7	globus gram client Page Documentation	29

1 Resource Management Client API

The resource management API provides function for requesting that a job be started or terminated, as well as for requesting information about the status of a job.

2 globus gram client Module Index

2.1 globus gram client Modules

Here is a list of all modules:

Job state callbacks	2
GRAM Job Functions	6
Other GRAM Client Functions	22
GRAM Client Attribute Functions	24

3 globus gram client Data Structure Index

3.1 globus gram client Data Structures

Here are the data structures with brief descriptions:

globus_gram_client_job_info_s (Extensible job information structure)	28
--	-----------

4 globus gram client Page Index

4.1 globus gram client Related Pages

Here is a list of all related documentation pages:

Resource Manager Contact

29

5 globus gram client Module Documentation

5.1 Job state callbacks

Data Structures

- struct **globus_gram_client_job_info_s**
Extensible job information structure.

Typedefs

- typedef void(*) **globus_gram_client_callback_func_t** (void *user_callback_arg, char *job_contact, int state, int errorcode)
- typedef **globus_gram_client_job_info_s** **globus_gram_client_job_info_t**
- typedef void(*) **globus_gram_client_info_callback_func_t** (void *user_callback_arg, const char *job_contact, **globus_gram_client_job_info_t** *job_info)
- typedef void(*) **globus_gram_client_nonblocking_func_t** (void *user_callback_arg, **globus_gram_protocol_error_t** operation_failure_code, const char *job_contact, **globus_gram_protocol_job_state_t** job_state, **globus_gram_protocol_error_t** job_failure_code)

Functions

- int **globus_gram_client_callback_allow** (**globus_gram_client_callback_func_t** callback_func, void *user_callback_arg, char **callback_contact)
- int **globus_gram_client_info_callback_allow** (**globus_gram_client_info_callback_func_t** callback_func, void *user_callback_arg, char **callback_contact)
- int **globus_gram_client_callback_disallow** (char *callback_contact)

5.1.1 Typedef Documentation

5.1.1.1 typedef void(*) globus_gram_client_callback_func_t(void *user_callback_arg, char *job_contact, int state, int errorcode)

Signature for GRAM state notification callback functions.

The **globus_gram_client_callback_func_t** type describes the function signature for job state callbacks. A pointer to a function of this type is passed to the **globus_gram_client_callback_allow()** (p. 4) function to create a callback contact. The contact string can be passed to **globus_gram_client_job_request()** (p. 10) or **globus_gram_client_job_callback_register()** (p. 18) to let the job management service know to where to send information on GRAM job state changes.

Parameters:

user_callback_arg A pointer to application-specific data.

job_contact A string containing the job contact. This string indicates which job this callback is referring to. It should in most cases match the return value *job_contact* from a call to **globus_gram_client_job_request()** (p. 10) or in the *job_contact* parameter to the **globus_gram_client_nonblocking_func_t** used with **globus_gram_client_register_job_request()** (p. 9). However, in some cases, the port number in the job contact URL may change if the job manager is restarted.

state The new state (one of the **globus_gram_protocol_job_state_t** (v ??)lues) of the job.

errorcode The error code if the *state* parameter is equal to **GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED**.

5.1.1.2 typedef struct globus_gram_client_job_info_s globus_gram_client_job_info_t

Extensible job information structure.

The **globus_gram_client_job_info_t** (p. 3) data type is used to pass protocol extensions along with the standard job status information included in the GRAM2 protocol. This structure contains the information returned in job state callbacks plus a hash table of extension entries that contain **globus_gram_protocol_extension_t** (n ??)me-value pairs.

5.1.1.3 typedef void(*) globus_gram_client_info_callback_func_t(void *user_callback_arg, const char *job_contact, globus_gram_client_job_info_t *job_info)

Signature for GRAM state notification callback functions with extension support.

The **globus_gram_client_info_callback_func_t** (p. 3) type describes the function signature for job state callbacks that carry any GRAM protocol extensions beyond the set used in GRAM2. A pointer to a function of this type is passed to the **globus_gram_client_info_callback_allow()** (p. 5) function to create a callback contact that can handle extensions. The contact string can be passed to **globus_gram_client_job_request()** (p. 10) or **globus_gram_client_job_callback_register()** (p. 18) to let the job management service know to where to send information on GRAM job state changes.

Parameters:

user_callback_arg Application-specific callback information.

job_contact Job this information is related to

job_info Job state and extensions

See also:

globus_gram_client_info_callback_allow() (p. 5)

5.1.1.4 typedef void(*) globus_gram_client_nonblocking_func_t(void *user_callback_arg, globus_gram_protocol_error_t operation_failure_code, const char *job_contact, globus_gram_protocol_job_state_t job_state, globus_gram_protocol_error_t job_failure_code)

Signature for callbacks signalling completion of non-blocking GRAM requests.

The **globus_gram_client_info_callback_func_t** (p. 3) type describes the function signature for callbacks which indicate that a GRAM operation has completed. A pointer to a function of this type is passed to the following functions:

- **globus_gram_client_register_job_request()** (p. 9)
- **globus_gram_client_register_job_cancel()** (p. 13)
- **globus_gram_client_register_job_status()** (p. 16)

- **globus_gram_client_register_job_refresh_credentials()** (p. 20)
- **globus_gram_client_register_job_signal()** (p. 14)
- **globus_gram_client_register_job_callback_registration()** (p. 21)
- **globus_gram_client_register_job_callback_unregistration()** (p. 21)
- **globus_gram_client_register_ping()** (p. 7)

Parameters:

user_callback_arg Application-specific callback information.

operation_failure_code The result of the nonblocking operation, indicating whether the operation was processed by the job manager successfully or not.

job_contact A string containing the job contact associated with this non-blocking operation.

job_state The state (one of the **globus_gram_protocol_job_state_t** (v ??)lues) of the job related to this non-blocking operation.

job_failure_code The error code of the job request if the job_state parameter is **GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED**. Otherwise, its value is undefined.

5.1.2 Function Documentation

5.1.2.1 **int globus_gram_client_callback_allow (globus_gram_client_callback_func_t callback_func, void * user_callback_arg, char ** callback_contact)**

Begin listening for job state change callbacks.

The **globus_gram_client_callback_allow()** (p. 4) function initializes a GRAM protocol service in the current process which will process job state updates from GRAM Job Managers. The URL to contact this service is returned and may be used with the **globus_gram_client_job_request()** (p. 10) or **globus_gram_client_callback_register()** family of functions.

Parameters:

callback_func A pointer to a function to call when a new job state update is received.

user_callback_arg A pointer to application-specific data which is passed to the function pointed to by **callback_func** as its **user_callback_arg** parameter.

callback_contact An output parameter that points to a string that will be allocated and set to the URL that the GRAM callback listener is waiting on.

Returns:

Upon success, **globus_gram_client_callback_allow()** (p. 4) returns **GLOBUS_SUCCESS** opens a TCP port to accept job state updates and modifies the value pointed to by the **callback_contact** parameter as described above. If an error occurs, **globus_gram_client_callback_allow()** (p. 4) returns an integer error code.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER Null parameter

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.1.2.2 `int globus_gram_client_info_callback_allow(globus_gram_client_info_callback_func_t callback_func, void * user_callback_arg, char ** callback_contact)`

Begin listening for job state change callbacks.

The `globus_gram_client_info_callback_allow()` (p. 5) function initializes a GRAM protocol service in the current process which will process job state updates from GRAM Job Managers. The URL to contact this service is returned and may be used with the `globus_gram_client_job_request_with_info()` (p. 12) or `globus_gram_client_register_job_status_with_info()` (p. 16) family of functions.

Parameters:

callback_func A pointer to a function to call when a new job state update is received. The function signature of this parameter supports GRAM protocol extensions.

user_callback_arg A pointer to application-specific data which is passed to the function pointed to by *callback_func* as its *user_callback_arg* parameter.

callback_contact An output parameter that points to a string that will be allocated and set to the URL that the GRAM callback listener is waiting on.

Returns:

Upon success, `globus_gram_client_callback_allow()` (p. 4) returns `GLOBUS_SUCCESS` opens a TCP port to accept job state updates and modifies the value pointed to by the *callback_contact* parameter as described above. If an error occurs, `globus_gram_client_callback_allow()` (p. 4) returns an integer error code.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER Null parameter

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.1.2.3 `int globus_gram_client_callback_disallow(char * callback_contact)`

Stop listening for job state change callbacks.

The `globus_gram_client_callback_disallow()` (p. 5) function stops the GRAM protocol handler associated with a callback contact from receiving further messages. After this function returns, no further callbacks for this contact will be called. Furthermore, the network port associated with the protocol handler will be released.

This function can only be used to disable a callback contact created in the current process.

Parameters:

callback_contact A callback contact string that refers to a protocol handler in the current process.

Returns:

Upon success, `globus_gram_client_callback_disallow()` (p. 5) returns `GLOBUS_SUCCESS`, closes the network port associated with the *callback_contact* parameter and stops further callbacks from occurring. If an error occurs, `globus_gram_client_callback_disallow()` (p. 5) returns an integer error code.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_CALLBACK_NOT_FOUND Callback not found

5.2 GRAM Job Functions

Functions

- `int globus_gram_client_register_ping (const char *resource_manager_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`
- `int globus_gram_client_ping (const char *resource_manager_contact)`
- `int globus_gram_client_get_jobmanager_version (const char *resource_manager_contact, globus_hashtable_t *extensions)`
- `int globus_gram_client_register_get_jobmanager_version (const char *resource_manager_contact, globus_gram_client_attr_t attr, globus_gram_client_info_callback_func_t info_callback, void *callback_arg)`
- `int globus_gram_client_register_job_request (const char *resource_manager_contact, const char *description, int job_state_mask, const char *callback_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`
- `int globus_gram_client_job_request (const char *resource_manager_contact, const char *description, int job_state_mask, const char *callback_contact, char **job_contact)`
- `int globus_gram_client_register_job_request_with_info (const char *resource_manager_contact, const char *description, int job_state_mask, const char *callback_contact, globus_gram_client_attr_t attr, globus_gram_client_info_callback_func_t callback, void *callback_arg)`
- `int globus_gram_client_job_request_with_info (const char *resource_manager_contact, const char *description, int job_state_mask, const char *callback_contact, char **job_contact, globus_gram_client_job_info_t *info)`
- `int globus_gram_client_job_cancel (const char *job_contact)`
- `int globus_gram_client_register_job_cancel (const char *job_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`
- `int globus_gram_client_job_signal (const char *job_contact, globus_gram_protocol_job_signal_t signal, const char *signal_arg, int *job_status, int *failure_code)`
- `int globus_gram_client_register_job_signal (const char *job_contact, globus_gram_protocol_job_signal_t signal, const char *signal_arg, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`
- `int globus_gram_client_job_status (const char *job_contact, int *job_status, int *failure_code)`
- `int globus_gram_client_register_job_status (const char *job_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`
- `int globus_gram_client_register_job_status_with_info (const char *job_contact, globus_gram_client_attr_t attr, globus_gram_client_info_callback_func_t info_callback, void *callback_arg)`
- `int globus_gram_client_job_status_with_info (const char *job_contact, globus_gram_client_job_info_t *info)`
- `int globus_gram_client_job_callback_register (const char *job_contact, int job_state_mask, const char *callback_contact, int *job_status, int *failure_code)`
- `int globus_gram_client_job_callback_unregister (const char *job_contact, const char *callback_contact, int *job_status, int *failure_code)`
- `int globus_gram_client_job_refresh_credentials (char *job_contact, gss_cred_id_t creds)`
- `int globus_gram_client_register_job_refresh_credentials (char *job_contact, gss_cred_id_t creds, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`
- `int globus_gram_client_register_job_callback_registration (const char *job_contact, int job_state_mask, const char *callback_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`
- `int globus_gram_client_register_job_callback_unregister (const char *job_contact, const char *callback_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`

5.2.1 Function Documentation

5.2.1.1 `int globus_gram_client_register_ping (const char * resource_manager_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void * register_callback_arg)`

Send a ping request to a GRAM service.

The `globus_gram_client_register_ping()` (p. 7) function sends a specially-formatted GRAM protocol message which checks to see if a Gatekeeper is running on a given PORT, and whether that Gatekeeper is configured to support the desired job manager service. This is used for diagnostic purposes.

If `globus_gram_client_register_ping()` (p. 7) determines that this request could not be processed before contacting the gatekeeper (for example, a malformed *resource_manager_contact*), it will return an error, and the *register_callback* function will not be called. Otherwise, the success or failure can be determined by the *operation_failure_code* parameter to the function pointed to by the *register_callback* parameter.

Parameters:

resource_manager_contact A NULL-terminated character string containing a **GRAM contact** (p. 29) that this function will contact.

attr A set of client attributes to use to contact the gatekeeper. If no custom attributes are needed, the caller should pass the value `GLOBUS_GRAM_CLIENT_NO_ATTR`.

register_callback A pointer to a function to call when the ping request has completed or failed.

register_callback_arg A pointer to application-specific data which will be passed to the function pointed to by *register_callback* as its *user_callback_arg* parameter.

Returns:

Upon success, `globus_gram_client_register_ping()` (p. 7) returns `GLOBUS_SUCCESS` and the *register_callback* function will be called once the ping operation completes. If an error occurs, this function returns an integer error code and the function pointed to by the *register_callback* parameter will not be called.

Return values:

`GLOBUS_SUCCESS` Success

`GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED` Out of memory

`GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED` Protocol error

`GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT` Invalid job contact

`GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST` Invalid request

`GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES` No resources

5.2.1.2 `int globus_gram_client_ping (const char * resource_manager_contact)`

Send a ping request to a GRAM service.

The `globus_gram_client_ping()` (p. 7) function sends a specially-formatted GRAM protocol message which checks to see if a Gatekeeper is running on a given PORT, and whether that Gatekeeper is configured to support the desired job manager service. This is used for diagnostic purposes.

Parameters:

resource_manager_contact A NULL-terminated character string containing a **GRAM contact** (p. 29) that this function will contact.

Returns:

Upon success, **globus_gram_client_ping()** (p. 7) contacts the gatekeeper service and returns *GLOBUS_SUCCESS*. If an error occurs, this function returns an integer error code.

Return values:

GLOBUS_SUCCESS Success
GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.3 int globus_gram_client_get_jobmanager_version (const char * resource_manager_contact, globus_hashtable_t * extensions)

Get version information from a job manager.

The **globus_gram_client_get_jobmanager_version()** (p. 8) function sends a message to a GRAM service which returns information about the job manager version in the value pointed to by the *extensions* parameter. Note that job managers prior to GT5 do not support the version request and so will return a *GLOBUS_GRAM_PROTOCOL_ERROR_HTTP_UNPACK_FAILED* error. This function blocks while processing this request.

Parameters:

resource_manager_contact A NULL-terminated character string containing a **GRAM contact** (p. 29).
extensions A pointer to a hash table which will be initialized to contain the version information returned by the service. The extensions defined by GRAM5 are *toolkit-version* and *version*.

Returns:

Upon success, **globus_gram_client_get_jobmanager_version()** (p. 8) function returns *GLOBUS_SUCCESS* and modifies the *extensions* parameter as described above. If an error occurs, the integer error code will be returned and the value pointed to by the *extensions* parameter is undefined.

Return values:

GLOBUS_SUCCESS Success
GLOBUS_GRAM_PROTOCOL_ERROR_BAD_GATEKEEPER_CONTACT Bad gatekeeper contact
GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER NULL parameter
GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol failed
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.4 int globus_gram_client_register_get_jobmanager_version (const char * resource_manager_contact, globus_gram_client_attr_t attr, globus_gram_client_info_callback_func_t info_callback, void * callback_arg)

Get version information from a job manager without blocking.

The **globus_gram_client_register_get_jobmanager_version()** (p. 8) function sends a message to a GRAM service which returns information about the job manager version to the function pointed to by the *info_callback* function. Note that job managers prior to GT5 do not support the version request and so will return a **GLOBUS_GRAM_PROTOCOL_ERROR_HTTP_UNPACK_FAILED** error. This function blocks while processing this request.

Parameters:

- resource_manager_contact* A NULL-terminated character string containing a **GRAM contact** (p. 29).
- attr* A set of client attributes to use to contact the gatekeeper. If no custom attributes are needed, the caller should pass the value *GLOBUS_GRAM_CLIENT_NO_ATTR*.
- info_callback* A pointer to a function to call when the version request has completed or failed.
- callback_arg* A pointer to application-specific data which will be passed to the function pointed to by *info_callback* as its *user_callback_arg* parameter.

Returns:

Upon success, **globus_gram_client_register_get_jobmanager_version()** (p. 8) function returns **GLOBUS_SUCCESS** and begins processing the version request to contact *resource_manager_contact*; when complete, the *info_callback* function will be called. If an error occurs, the integer error code will be returned and the value pointed to by the *extensions* parameter is undefined.

Return values:

- GLOBUS_SUCCESS* Success
- GLOBUS_GRAM_PROTOCOL_ERROR_BAD_GATEKEEPER_CONTACT* Bad gatekeeper contact
- GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER* NULL parameter
- GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED* Out of memory
- GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED* Protocol failed
- GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT* Invalid job contact
- GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST* Invalid request
- GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES* No resources

5.2.1.5 int globus_gram_client_register_job_request (const char * *resource_manager_contact*, const char * *description*, int *job_state_mask*, const char * *callback_contact*, globus_gram_client_attr_t *attr*, globus_gram_client_nonblocking_func_t *register_callback*, void * *register_callback_arg*)

Send a job request to a GRAM service.

The **globus_gram_client_register_job_request()** (p. 9) function sends a GRAM protocol message to a service to request that it start a job on behalf of the client.

If **globus_gram_client_register_job_request()** (p. 9) determines that this request could not be processed before contacting the gatekeeper (for example, a malformed *resource_manager_contact*), it will return an error, and the *register_callback* function will not be called. Otherwise, the success or failure can be determined by the *operation_failure_code* parameter to the function pointed to by the *register_callback* parameter.

Parameters:

- resource_manager_contact* A NULL-terminated character string containing a **GRAM contact** (p. 29) that this function will contact.
- description* A pointer to a string containing the job request information formatted in RSL syntax.
- job_state_mask* A bitwise-or of the **GLOBUS_GRAM_PROTOCOL_JOB_STATE_*** states that the job manager will send job state notification messages for to the contact named by *callback_contact*.

callback_contact A GRAM listener contact that the job manager will send job state notification messages to.

attr A set of client attributes to use to contact the gatekeeper. If no custom attributes are needed, the caller should pass the value *GLOBUS_GRAM_CLIENT_NO_ATTR*.

register_callback A pointer to a function to call when the job_request request has completed or failed.

register_callback_arg A pointer to application-specific data which will be passed to the function pointed to by *register_callback* as its *user_callback_arg* parameter.

Returns:

Upon success, **globus_gram_client_register_job_request()** (p. 9) returns *GLOBUS_SUCCESS* and the *register_callback* function will be called once the job request operation completes. If an error occurs, this function returns an integer error code and the function pointed to by the *register_callback* parameter will not be called.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.6 **int globus_gram_client_job_request (const char * resource_manager_contact, const char * description, int job_state_mask, const char * callback_contact, char ** job_contact)**

Send a job request to a GRAM service.

The **globus_gram_client_job_request()** (p. 10) function sends a GRAM protocol message to a service to request that it start a job on behalf of the client. Unlike, **globus_gram_client_register_job_request()** (p. 9), **globus_gram_client_job_request()** (p. 10) blocks until the job request has been processed by the service.

Parameters:

resource_manager_contact A NULL-terminated character string containing a **GRAM contact** (p. 29) that this function will contact.

description A pointer to a string containing the job request information formatted in RSL syntax.

job_state_mask A bitwise-or of the *GLOBUS_GRAM_PROTOCOL_JOB_STATE_** states that the job manager will send job state notification messages for to the contact named by *callback_contact*.

callback_contact A GRAM listener contact that the job manager will send job state notification messages to.

job_contact An output parameter pointing to a string that will be set to the job contact for this job. This value will only be set if the job request is successful or the two-phase commit protocol is being used and the return code is *GLOBUS_GRAM_PROTOCOL_ERROR_WAITING_FOR_COMMIT*.

Returns:

Upon success, **globus_gram_client_job_request()** (p. 10) returns *GLOBUS_SUCCESS* and modifies the value pointed to by *job_contact* as described above. If an error occurs, this function returns an integer error code and the value pointed to by *job_contact*. In addition to the error codes described below, any **globus_gram_protocol_error_t** (v ??)lue may be returned as a cause for the job to fail.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.7 `int globus_gram_client_register_job_request_with_info (const char * resource_manager_contact, const char * description, int job_state_mask, const char * callback_contact, globus_gram_client_attr_t attr, globus_gram_client_info_callback_func_t callback, void * callback_arg)`

Send a job request to a GRAM service with extensions-aware callback.

The `globus_gram_client_register_job_request_with_info()` (p. 11) function sends a GRAM protocol message to a service to request that it start a job on behalf of the client.

If `globus_gram_client_register_job_request_with_info()` (p. 11) determines that this request could not be processed before contacting the gatekeeper (for example, a malformed *resource_manager_contact*), it will return an error, and the *callback* function will not be called. Otherwise, the success or failure can be determined by the *operation_failure_code* parameter to the function pointed to by the *callback* parameter. The difference between this function and `globus_gram_client_register_job_request()` (p. 9) is the function signature of the callback function.

Parameters:

resource_manager_contact A NULL-terminated character string containing a **GRAM contact** (p. 29) that this function will contact.

description A pointer to a string containing the job request information formatted in RSL syntax.

job_state_mask A bitwise-or of the `GLOBUS_GRAM_PROTOCOL_JOB_STATE_*` states that the job manager will send job state notification messages for to the contact named by *callback_contact*.

callback_contact A GRAM listener contact that the job manager will send job state notification messages to.

attr A set of client attributes to use to contact the gatekeeper. If no custom attributes are needed, the caller should pass the value `GLOBUS_GRAM_CLIENT_NO_ATTR`.

callback A pointer to a function to call when the job_request request has completed or failed.

callback_arg A pointer to application-specific data which will be passed to the function pointed to by *callback* as its *user_callback_arg* parameter.

Returns:

Upon success, `globus_gram_client_register_job_request_with_info()` (p. 11) returns `GLOBUS_SUCCESS` and the *callback* function will be called once the job request operation completes. If an error occurs, this function returns an integer error code and the function pointed to by the *callback* parameter will not be called.

Return values:

GLOBUS_SUCCESS Success
GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.8 `int globus_gram_client_job_request_with_info (const char * resource_manager_contact, const char * description, int job_state_mask, const char * callback_contact, char ** job_contact, globus_gram_client_job_info_t * info)`

Send a job request to a GRAM service and parse extensions in the response.

The `globus_gram_client_job_request_with_info()` (p. 12) function sends a GRAM protocol message to a service to request that it start a job on behalf of the client. Unlike, `globus_gram_client_register_job_request_with_info()` (p. 11), `globus_gram_client_job_request_with_info()` (p. 12) blocks until the job request has been processed by the service.

Parameters:

resource_manager_contact A NULL-terminated character string containing a **GRAM contact** (p. 29) that this function will contact.

description A pointer to a string containing the job request information formatted in RSL syntax.

job_state_mask A bitwise-or of the `GLOBUS_GRAM_PROTOCOL_JOB_STATE_*` states that the job manager will send job state notification messages for to the contact named by *callback_contact*.

callback_contact A GRAM listener contact that the job manager will send job state notification messages to.

job_contact An output parameter pointing to a string that will be set to the job contact for this job. This value will only be set if the job request is successful or the two-phase commit protocol is being used and the return code is `GLOBUS_GRAM_PROTOCOL_ERROR_WAITING_FOR_COMMIT`.

info An output parameter pointing to a structure to hold the extensions in the GRAM response. The caller is responsible for destroying this by calling the `globus_gram_client_job_info_destroy()` (p. 24) function.

Returns:

Upon success, `globus_gram_client_job_request_with_info()` (p. 12) returns `GLOBUS_SUCCESS` and modifies the values pointed to by *job_contact* and *info* as described above. If an error occurs, this function returns an integer error code and the value pointed to by *job_contact*. In addition to the error codes described below, any `globus_gram_protocol_error_t` (v ??)lue may be returned as a cause for the job to fail.

Return values:

`GLOBUS_SUCCESS` Success

`GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED` Out of memory

`GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED` Protocol error

`GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT` Invalid job contact

`GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST` Invalid request

`GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES` No resources

5.2.1.9 `int globus_gram_client_job_cancel (const char * job_contact)`

Cancel a GRAM job.

The `globus_gram_client_job_cancel()` (p. 12) function cancels a GRAM job. Depending on the job's current state, this cancellation may be immediate or require a delay for interacting with the LRM servicing the job. Notification when the job has been successfully canceled will be sent to all client contacts registered for notifications after the cancellation has been completely processed. Unlike, `globus_gram_client_register_job_cancel()` (p. 13), `globus_gram_client_job_cancel()` (p. 12) blocks until the job cancel request has been processed by the service.

Parameters:

job_contact A NULL-terminated character string containing a GRAM job contact that this function will contact to cancel the job.

Returns:

Upon succes, **globus_gram_client_job_cancel()** (p. 12) returns *GLOBUS_SUCCESS* if the cancellation was successful posted to the service. If an error occurs, **globus_gram_client_job_cancel()** (p. 12) returns one of the **globus_gram_protocol_error_t** (v ??)lues values indicating why the client could not cancel the job.

See also:

globus_gram_client_register_job_cancel() (p. 13)

5.2.1.10 **int globus_gram_client_register_job_cancel** (const char **job_contact*, globus_gram_client_attr_t *attr*, globus_gram_client_nonblocking_func_t *register_callback*, void **register_callback_arg*)

Cancel a GRAM job.

The **globus_gram_client_register_job_cancel()** (p. 13) function sends a message to a GRAM service to cancel a GRAM job. Depending on the job's current state, this cancellation may be immediate or require a delay for interacting with the LRM servicing the job. In either case, this function returns as soon as it is able to start sending the message. Notification when the job has been successfully canceled will be sent to all client contacts registered for notifications after the cancellation has been completely processed.

Parameters:

job_contact A NULL-terminated character string containing a GRAM job contact that this function will contact to cancel the job.

attr A set of client attributes to use to contact the job. If no custom attributes are needed, the caller should pass the value *GLOBUS_GRAM_CLIENT_NO_ATTR*.

register_callback A pointer to a function to call when the job_request request has completed or failed.

register_callback_arg A pointer to application-specific data which will be passed to the function pointed to by *register_callback* as its *user_callback_arg* parameter.

Returns:

Upon succes, **globus_gram_client_register_job_cancel()** (p. 13) returns *GLOBUS_SUCCESS* if the cancellation was successful posted to the service. If an error occurs, **globus_gram_client_register_job_cancel()** (p. 13) returns one an integer error code indicating why it could not cancel the job.

Return values:

GLOBUS_GRAM_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

See also:

globus_gram_client_job_cancel() (p. 12)

5.2.1.11 **int globus_gram_client_job_signal** (const char **job_contact*, globus_gram_protocol_job_signal_t *signal*, const char **signal_arg*, int **job_status*, int **failure_code*)

Send a signal a GRAM job.

The **globus_gram_client_job_signal()** (p. 13) function sends a signal message to a job managed by the GRAM service. Signals consist of a signal number and an optional string argument. The meanings of the signals supported by the GRAM job manager are defined in the GRAM Protocol documentation. Unlike **globus_gram_client_register_job_signal()** (p. 14), this function blocks until the signal has been delivered and acknowledged by the GRAM service.

Parameters:

job_contact The job contact string of the job manager to contact. This is the same value returned from **globus_gram_client_job_request()** (p. 10).

signal The signal code to send to the job manager.

signal_arg Parameters for the signal, as described in the documentation for the **globus_gram_protocol_job_signal_t** (enumeration).

job_status An output parameter pointing to an integer to set to the status of the job after the signal has been processed.

failure_code An output parameter pointing to an integer to set to the reason why the job has failed if the value pointed to by *job_status* is set to **GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED** by this function.

Returns:

Upon success, **globus_gram_client_job_signal()** (p. 13) returns **GLOBUS_SUCCESS** after sending the signal and receiving a response and modifies the *job_status* and *failure_code* parameters as described above. If an error occurs, this function returns an integer error code indicating why the client could not signal the job.

Return values:

GLOBUS_GRAM_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

See also:

globus_gram_client_register_job_signal() (p. 14)

5.2.1.12 int globus_gram_client_register_job_signal (const char * *job_contact*, globus_gram_protocol_job_signal_t *signal*, const char * *signal_arg*, globus_gram_client_attr_t *attr*, globus_gram_client_nonblocking_func_t *register_callback*, void * *register_callback_arg*)

Send a signal a GRAM job.

The **globus_gram_client_register_job_signal()** (p. 14) function sends a signal message to a job managed by the GRAM service. Signals consist of a signal number and an optional string argument. The meanings of the signals supported by the GRAM job manager are defined in the GRAM Protocol documentation. This function returns as soon as it has determined that its parameters are valid and it has begun to send the message to the GRAM service.

Parameters:

job_contact The job contact string of the job manager to contact. This is the same value returned from **globus_gram_client_job_request()** (p. 10).

signal The signal code to send to the job manager.

signal_arg Parameters for the signal, as described in the documentation for the **globus_gram_protocol_job_signal_t** enumeration.

attr A set of client attributes to use to contact the job. If no custom attributes are needed, the caller should pass the value *GLOBUS_GRAM_CLIENT_NO_ATTR*.

register_callback A pointer to a function to call when the signal request has completed or failed.

register_callback_arg A pointer to application-specific data which will be passed to the function pointed to by *register_callback* as its *user_callback_arg* parameter.

Returns:

Upon success, *globus_gram_client_job_register_signal()* returns *GLOBUS_SUCCESS* after beginning to send the signal to the GRAM job and registers the *register_callback* function to be called once that has completed. If an error occurs, this function returns an integer error code indicating why the client could not signal the job.

Return values:

GLOBUS_GRAM_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

See also:

globus_gram_client_register_job_signal() (p. 14)

5.2.1.13 *int globus_gram_client_job_status (const char * job_contact, int * job_status, int * failure_code)*

Send a status query to a GRAM job.

The *globus_gram_client_status()* function queries the current status of the job associated with the job contact, returning its current status, as well as the job's failure reason if it has failed in this function's return parameters. This function blocks until the service has responded to the status query.

Parameters:

job_contact The job contact string of the job to query. This is the same value returned from *globus_gram_client_job_request()* (p. 10).

job_status An output parameter that points to an integer to be set to the current status of the job named by the *job_contact* parameter.

failure_code An output parameter that points to an integer to be set to the reason why the job failed if its current status is *GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED*. If the job has not failed, the value will be set to 0.

Returns:

Upon success, the *globus_gram_client_job_status()* (p. 15) function returns *GLOBUS_SUCCESS*, sends a job state query to the job named by *job_contact* and parses the service response, modifying the values pointed to by *job_status* and *failure_code* as described above. If an error occurs, *globus_gram_client_job_status()* (p. 15) returns an integer error code.

Return values:

GLOBUS_GRAM_SUCCESS Success
GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.14 `int globus_gram_client_register_job_status (const char * job_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void * register_callback_arg)`

Send a status query to a GRAM job.

The `globus_gram_client_register_job_status()` (p. 16) function initiates a query of the current status of the job associated with the *job_contact* parameter. The job's status and failure code are passed to the function pointed to by the *register_callback* parameter. This function returns as soon as it has begun requesting the job status.

Parameters:

job_contact The job contact string of the job to query. This is the same value returned from `globus_gram_client_job_request()` (p. 10).
attr A set of client attributes to use to contact the job. If no custom attributes are needed, the caller should pass the value `GLOBUS_GRAM_CLIENT_NO_ATTR`.
register_callback A pointer to a function to call when the status request has completed or failed.
register_callback_arg A pointer to application-specific data which will be passed to the function pointed to by *register_callback* as its *user_callback_arg* parameter.

Returns:

Upon success, the `globus_gram_client_register_job_status()` (p. 16) function returns `GLOBUS_SUCCESS` and begins to send a job state query to the job named by *job_contact* and registers the function pointed to by the *register_callback* parameter to be called once the status query terminates or fails. If an error occurs, `globus_gram_client_register_job_status()` (p. 16) returns an integer error code.

Return values:

GLOBUS_GRAM_SUCCESS Success
GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources
GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER Null parameter

5.2.1.15 `int globus_gram_client_register_job_status_with_info (const char * job_contact, globus_gram_client_attr_t attr, globus_gram_client_info_callback_func_t info_callback, void * callback_arg)`

Send a status query to a GRAM job.

The `globus_gram_client_register_job_status_with_info()` (p. 16) function initiates a query of the current status of the job associated with the *job_contact* parameter. Job information is returned via the *job_info* parameter passed to the function pointed to by the *info_callback* parameter. This function returns as soon as it has begun requesting the job status.

Parameters:

job_contact The job contact string of the job to query. This is the same value returned from **globus_gram_client_job_request()** (p. 10).

attr A set of client attributes to use to contact the job. If no custom attributes are needed, the caller should pass the value *GLOBUS_GRAM_CLIENT_NO_ATTR*.

info_callback A pointer to a function to call when the status request has completed or failed.

callback_arg A pointer to application-specific data which will be passed to the function pointed to by *info_callback* as its *user_callback_arg* parameter.

Returns:

Upon success, the **globus_gram_client_register_job_status_with_info()** (p. 16) function returns *GLOBUS_SUCCESS* and begins to send a job state query to the job named by *job_contact* and registers the function pointed to by the *info_callback* parameter to be called once the status query terminates or fails. If an error occurs, **globus_gram_client_register_job_status_with_info()** (p. 16) returns an integer error code.

Return values:

GLOBUS_GRAM_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER Null parameter

5.2.1.16 **int globus_gram_client_job_status_with_info (const char * *job_contact*, globus_gram_client_job_info_t * *info*)**

Send a status query to a GRAM job.

The **globus_gram_client_status_with_info()** function queries the current status of the job associated with the job contact, returning its current status, as well as the job's failure reason if it has failed in this function's return parameters. This function blocks until the service has responded to the status query.

Parameters:

job_contact The job contact string of the job to query. This is the same value returned from **globus_gram_client_job_request()** (p. 10).

info An output parameter that points to a **globus_gram_client_job_info_t** structure which will be populated with the state information associated with the job named by the *job_contact* parameter. The caller is responsible for calling **globus_gram_client_job_info_destroy()** (p. 24) to free the state pointed to by this parameter if this function returns *GLOBUS_SUCCESS*.

Returns:

Upon success, the **globus_gram_client_job_status_with_info()** (p. 17) function returns *GLOBUS_SUCCESS*, sends a job state query to the job named by *job_contact* and parses the service response, modifying the structure pointed to by *info* as described above. If an error occurs, **globus_gram_client_job_status_with_info()** (p. 17) returns an integer error code.

Return values:

GLOBUS_GRAM_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.17 `int globus_gram_client_job_callback_register (const char *job_contact, int job_state_mask, const char *callback_contact, int *job_status, int *failure_code)`

Register a new callback contact to be notified for job state changes.

The **`globus_gram_client_job_callback_register()`** (p. 18) function contacts a GRAM service managing a job and instructs it to send subsequent job state callbacks to the client listening for callbacks at the contact url named by the *callback_contact* parameter. This function blocks until the registration operation either completes or exits.

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from **`globus_gram_client_job_request()`** (p. 10).
job_state_mask A bitwise-or of the ***GLOBUS_GRAM_PROTOCOL_JOB_STATE_**** states that the job manager will send job state notification messages for to the contact named by *callback_contact*.
callback_contact A URL string containing a GRAM client callback. This string is normally be generated by a process calling **`globus_gram_client_callback_allow()`** (p. 4).
job_status An output parameter pointing to an integer to set to the status of the job after the registration message has been processed.
failure_code An output parameter that points to an integer to be set to the reason why the job failed if its current status is ***GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED***. If the job has not failed, the value will be set to 0.

Returns:

Upon success, the **`globus_gram_client_job_callback_register()`** (p.18) function returns ***GLOBUS_SUCCESS***, sends a registration request the job named by *job_contact* and parses the service response, modifying the values pointed to by the *job_status* and *failure_code* parameters as described above. If an error occurs, **`globus_gram_client_job_callback_register()`** (p. 18) returns an integer error code indicating why it can't register the callback contact. The return code may be any value defined by the *globus_gram_protocol_error_t* enumeration in addition to those listed below.

Return values:

GLOBUS_GRAM_SUCCESS Success
GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.18 `int globus_gram_client_job_callback_unregister (const char *job_contact, const char *callback_contact, int *job_status, int *failure_code)`

Unregister a callback contact to stop job state change notifications.

The **globus_gram_client_job_callback_unregister()** (p. 18) function contacts a GRAM service managing a job and instructs it to stop sending job state callbacks to the client listening for callbacks at the contact url named by the *callback_contact* parameter. This function blocks until the unregistration operation either completes or exits. It is possible that callbacks related to the job arrive at the contact after this function returns depending on network delays.

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from **globus_gram_client_job_request()** (p. 10).

callback_contact A URL string containing a GRAM client callback. This string is normally be generated by a process calling **globus_gram_client_callback_allow()** (p. 4).

job_status An output parameter pointing to an integer to set to the status of the job after the registration message has been processed.

failure_code An output parameter that points to an integer to be set to the reason why the job failed if its current status is *GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED*. If the job has not failed, the value will be set to 0.

Returns:

Upon success, the **globus_gram_client_job_callback_unregister()** (p. 18) function returns *GLOBUS_SUCCESS*, sends an unregister request the job named by *job_contact* and parses the service response, modifying the values pointed to by the *job_status* and *failure_code* parameters as described above. If an error occurs, **globus_gram_client_job_callback_unregister()** (p. 18) returns an integer error code indicating why it can't unregister the callback contact. The return code may be any value defined by the *globus_gram_protocol_error_t* enumeration in addition to those listed below.

Return values:

GLOBUS_GRAM_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.19 int globus_gram_client_job_refresh_credentials (char **job_contact*, gss_cred_id_t *creds*)

Delegate a new credential to a job.

The **globus_gram_client_job_refresh_credentials()** (p. 19) function sends a "renew" signal to a GRAM service and then initiates the delegation of a new credential to the job manager and job. This prevents errors that can occur when a credential expires. This function blocks until the delegation has completed or failed.

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from **globus_gram_client_job_request()** (p. 10).

creds A GSSAPI credential handle which will be used to authenticate with the job manager and sign the delegated credential. This parameter's value may be set to *GSS_C_NO_CREDENTIAL* to indicate the desire to use this process's default credential.

Returns:

Upon success, the **globus_gram_client_job_refresh_credentials()** (p. 19) function returns *GLOBUS_SUCCESS*, sends an proxy renew request the job named by *job_contact*, parses the service response and

performs a GSSAPI delegation to send a new credential to the job service. If an error occurs, **globus_gram_client_job_refresh_credentials()** (p. 19) returns an integer error code indicating why it can't refresh the job service's credential. The return code may be any value defined by the *globus_gram_protocol_error_t* enumeration in addition to those listed below.

Return values:

GLOBUS_GRAM_SUCCESS Success
GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.20 `int globus_gram_client_register_job_refresh_credentials(char *job_contact, gss_cred_id_t creds, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void *register_callback_arg)`

Delegate a new credential to a job.

The **globus_gram_client_register_job_refresh_credentials()** (p. 20) function sends a "renew" signal to a GRAM service and then initiates the delegation of a new credential to the job manager and job. This prevents errors that can occur when a credential expires. This function returns as soon as it has determined that its parameters are valid and it has begun to send the message to the GRAM service.

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from **globus_gram_client_job_request()** (p. 10).
creds A GSSAPI credential handle which will be used to authenticate with the job manager and sign the delegated credential. This parameter's value may be set to *GSS_C_NO_CREDENTIAL* to indicate the desire to use this process's default credential.
attr A set of client attributes to use to contact the job. If no custom attributes are needed, the caller should pass the value *GLOBUS_GRAM_CLIENT_NO_ATTR*.
register_callback A pointer to a function to call when the status request has completed or failed.
register_callback_arg A pointer to application-specific data which will be passed to the function pointed to by *register_callback* as its *user_callback_arg* parameter.

Returns:

Upon success, the **globus_gram_client_job_refresh_credentials()** (p. 19) function returns *GLOBUS_SUCCESS* and begins sending the "renew" request to the GRAM service. If an error occurs, **globus_gram_client_job_refresh_credentials()** (p. 19) returns an integer error code indicating why it can't refresh the job service's credential. The return code may be any value defined by the *globus_gram_protocol_error_t* enumeration in addition to those listed below.

Return values:

GLOBUS_GRAM_SUCCESS Success
GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory
GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact
GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request
GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

5.2.1.21 `int globus_gram_client_register_job_callback_registration (const char * job_contact, int job_state_mask, const char * callback_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void * register_callback_arg)`

Register a new callback contact to be notified for job state changes.

The `globus_gram_client_register_job_callback_registration()` (p. 21) function initiates the protocol to contact a GRAM service and request that it send subsequent job state callbacks to the client listening for callbacks at the contact url named by the *callback_contact* parameter. This function returns as soon as it has validated its parameters and begun sending the GRAM message. When the registration is complete, the function pointed to by *register_callback* is called.

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from `globus_gram_client_job_request()` (p. 10).

job_state_mask A bitwise-or of the `GLOBUS_GRAM_PROTOCOL_JOB_STATE_*` states that the job manager will send job state notification messages for to the contact named by *callback_contact*.

callback_contact A URL string containing a GRAM client callback. This string is normally be generated by a process calling `globus_gram_client_callback_allow()` (p. 4).

attr A set of client attributes to use to contact the job. If no custom attributes are needed, the caller should pass the value `GLOBUS_GRAM_CLIENT_NO_ATTR`.

register_callback A pointer to a function to call when the registration request has completed or failed.

register_callback_arg A pointer to application-specific data which will be passed to the function pointed to by *register_callback* as its *user_callback_arg* parameter.

Returns:

Upon success, the `globus_gram_client_register_job_callback_registration()` (p. 21) function returns `GLOBUS_SUCCESS`, begins to send a registration request to the job named by *job_contact*, and schedules the *register_callback* to be called once the registration completes or fails. If an error occurs, this function returns an integer error code indicating why it can't process the request.

Return values:

`GLOBUS_GRAM_SUCCESS` Success

`GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED` Out of memory

`GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED` Protocol error

`GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT` Invalid job contact

`GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST` Invalid request

`GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES` No resources

See also:

`globus_gram_client_job_callback_register()` (p. 18)

5.2.1.22 `int globus_gram_client_register_job_callback_unregistration (const char * job_contact, const char * callback_contact, globus_gram_client_attr_t attr, globus_gram_client_nonblocking_func_t register_callback, void * register_callback_arg)`

Unregister a callback contact to stop job state change notifications.

The `globus_gram_client_register_job_callback_unregistration()` (p. 21) function initiates the protocol to contact a GRAM service and request that it stop sending job state callbacks to the client listening at the contact url named by the *callback_contact* parameter. This function returns as soon as it has validated its parameters and begun sending the GRAM message. When the unregistration is complete, the function pointed to by *register_callback* is called.

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from **globus_gram_client_job_request()** (p. 10).

callback_contact A URL string containing a GRAM client callback. This string is normally generated by a process calling **globus_gram_client_callback_allow()** (p. 4).

attr A set of client attributes to use to contact the job. If no custom attributes are needed, the caller should pass the value **GLOBUS_GRAM_CLIENT_NO_ATTR**.

register_callback A pointer to a function to call when the registration request has completed or failed.

register_callback_arg A pointer to application-specific data which will be passed to the function pointed to by **register_callback** as its **user_callback_arg** parameter.

Returns:

Upon success, the **globus_gram_client_register_job_callback_unregister()** (p. 21) function returns **GLOBUS_SUCCESS**, begins sending an unregister request to the job named by **job_contact** and schedules the function pointed to by the **register_callback** parameter to be called. If an error occurs, **globus_gram_client_register_job_callback_unregister()** (p. 21) returns an integer error code indicating why it can't process the unregister request.

Return values:

GLOBUS_GRAM_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

GLOBUS_GRAM_PROTOCOL_ERROR_PROTOCOL_FAILED Protocol error

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_JOB_CONTACT Invalid job contact

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_REQUEST Invalid request

GLOBUS_GRAM_PROTOCOL_ERROR_NO_RESOURCES No resources

See also:

globus_gram_client_job_callback_unregister() (p. 18)

5.3 Other GRAM Client Functions

Functions

- void **globus_gram_client_debug** (void)
- int **globus_gram_client_version** (void)
- int **globus_gram_client_set_credentials** (gss_cred_id_t new_credentials)
- const char * **globus_gram_client_error_string** (int error_code)
- int **globus_gram_client_job_contact_free** (char *job_contact)
- void **globus_gram_client_job_info_destroy** (globus_gram_client_job_info_t *info)

5.3.1 Function Documentation

5.3.1.1 void **globus_gram_client_debug** (void)

Enable GRAM debugging.

The **globus_gram_client_debug()** (p. 22) function enables the displaying of internal GRAM debug messages to standard output. Most of the information printed by this debugging system is related to errors that occur during GRAM Client API functions. The messages printed to standard output are not structured in any way.

Returns:

void

5.3.1.2 `int globus_gram_client_version (void)`

Return GRAM protocol version.

The `globus_gram_client_version()` (p. 23) function returns the version of the GRAM protocol understood by this implementation.

Returns:

The integer protocol revision.

5.3.1.3 `int globus_gram_client_set_credentials (gss_cred_id_t new_credentials)`

Set the default GRAM credential.

The `globus_gram_client_set_credentials()` (p. 23) function causes subsequent GRAM operations to use the GSS-API credential *new_credentials*. These operations include job requests, job signals, callback registration, and job state callbacks. After this function returns, the caller must not use the credential, as it may be freed by GRAM when it is no longer needed.

Parameters:

new_credentials New GSSAPI credential to use.

Returns:

Upon success, `globus_gram_client_set_credentials()` (p. 23) returns `GLOBUS_SUCCESS`. There are no error values returned by this function.

Return values:

`GLOBUS_SUCCESS` Success

5.3.1.4 `const char* globus_gram_client_error_string (int error_code)`

Get a description of a GRAM error code.

The `globus_gram_client_error_string()` (p. 23) function takes a GRAM error code value and returns the associated error code string. The string is statically allocated by the Globus GRAM Client library and should not be modified or freed. The string is intended to complete a sentence of the form "[operation] failed because ..."

Parameters:

error_code The error code to translate into a string.

Returns:

The `globus_gram_client_error_string()` (p. 23) function returns a static string containing an explanation of the error.

5.3.1.5 `int globus_gram_client_job_contact_free (char *job_contact)`

Free a job contact string.

The `globus_gram_client_job_contact_free()` (p. 23) function frees a job contact string that was allocated by a call to one of the functions in the `globus_gram_client_job_request()` (p. 10) family. The `free()` function can be used in place of this function. After this function returns, the string pointed to by the *job_contact* parameter has an undefined value.

Parameters:

job_contact Pointer to a job contact string returned by a GRAM client API function.

Returns:

This function always returns GLOBUS_SUCCESS.

Return values:

GLOBUS_SUCCESS Success

5.3.1.6 void globus_gram_client_job_info_destroy (globus_gram_client_job_info_t * info)

Free memory associated with a globus_gram_client_job_info_t structure.

The **globus_gram_client_job_info_destroy()** (p. 24) function frees data pointed to by the *extensions* and *job_contact* fields of the *globus_gram_client_job_info_t* structure pointed to by the *info* parameter.

Parameters:

info A structure containing data to free.

5.4 GRAM Client Attribute Functions**Defines**

- #define GLOBUS_GRAM_CLIENT_NO_ATTR

Typedefs

- typedef void * globus_gram_client_attr_t

Functions

- int globus_gram_client_attr_init (globus_gram_client_attr_t *attr)
- int globus_gram_client_attr_destroy (globus_gram_client_attr_t *attr)
- int globus_gram_client_attr_set_credential (globus_gram_client_attr_t attr, gss_cred_id_t credential)
- int globus_gram_client_attr_get_credential (globus_gram_client_attr_t attr, gss_cred_id_t *credential)
- int globus_gram_client_attr_set_delegation_mode (globus_gram_client_attr_t attr, globus_io_secure_delegation_mode_t mode)
- int globus_gram_client_attr_get_delegation_mode (globus_gram_client_attr_t attr, globus_io_secure_delegation_mode_t *mode)

5.4.1 Define Documentation**5.4.1.1 #define GLOBUS_GRAM_CLIENT_NO_ATTR**

Default GRAM client operation attribute.

The GLOBUS_GRAM_CLIENT_NO_ATTR macro defines a constant for use when a user of the GRAM client API does not want to specify any non-default GRAM attributes.

5.4.2 Typedef Documentation

5.4.2.1 typedef void* globus_gram_client_attr_t

GRAM client operation attribute.

The **globus_gram_client_attr_t** (p. 25) type is an opaque type describing GRAM attributes. It can be accessed or modified by functions in the **GRAM Client Attribute Functions** (p. 24) documentation.

5.4.3 Function Documentation

5.4.3.1 int globus_gram_client_attr_init (globus_gram_client_attr_t * attr)

Initialize a GRAM client attribute.

The **globus_gram_client_attr_init()** (p. 25) function creates a new opaque structure that can be used to specify custom attributes for performing GRAM client operations.

Parameters:

attr An output parameter which will be set to the newly initialized attribute.

Returns:

Upon success, **globus_gram_client_attr_init()** (p. 25) modifies the *attr* parameter to point to a new GRAM client attribute and returns *GLOBUS_SUCCESS*. If an error occurs, **globus_gram_client_attr_init()** (p. 25) returns an integer error code and value of *attr* is undefined.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR Invalid attribute

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Out of memory

See also:

globus_gram_client_attr_destroy() (p. 25)

5.4.3.2 int globus_gram_client_attr_destroy (globus_gram_client_attr_t * attr)

Destroy a GRAM client attribute.

The **globus_gram_client_attr_destroy()** (p. 25) function destroys and frees a GRAM client attribute. After this function returns, the value pointed to by *attr* is no longer valid and must not be used.

Parameters:

attr A pointer to the attribute to destroy. All data associated with the attribute will be freed and it will be an invalid attribute.

Returns:

Upon success, **globus_gram_client_attr_destroy()** (p. 25) destroys the attribute pointed to by the *attr* parameter and sets it to an invalid state. If an error occurs, **globus_gram_client_attr_destroy()** (p. 25) returns an integer error code and value of *attr* is unchanged.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR Invalid attribute

See also:

globus_gram_client_attr_init() (p. 25)

5.4.3.3 **int globus_gram_client_attr_set_credential** (globus_gram_client_attr_t *attr*, gss_cred_id_t *credential*)

Set a GRAM client attribute's security credential.

The **globus_gram_client_attr_set_credential()** (p. 26) function sets the value of the **credential** in an attribute to the GSSAPI credential named by the *credential* parameter. This is done as a shallow copy, so the value of *credential* must not be freed until the attribute will no longer be used.

Parameters:

attr The attribute set to modify to use the credential named by the *credential* parameter.

credential The GSSAPI credential to use with the attribute named by the *attr* parameter. This may be *GSS_C_NO_CREDENTIAL* to set the attribute to use the default security credential.

Returns:

Upon success, **globus_gram_client_attr_set_credential()** (p. 26) modifies the attribute pointed to by the *attr* parameter to use the credential specified by the *credential* parameter and returns *GLOBUS_SUCCESS*. If an error occurs, **globus_gram_client_attr_set_credential()** (p. 26) returns an integer error code and the attribute named by *attr* is unchanged.

Return values:

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR An invalid attribute set was passed to this function.

See also:

globus_gram_client_attr_get_credential() (p. 26)

5.4.3.4 **int globus_gram_client_attr_get_credential** (globus_gram_client_attr_t *attr*, gss_cred_id_t * *credential*)

Get a GRAM client attribute's security credential.

The **globus_gram_client_attr_get_credential()** (p. 26) function gets the value of the **credential** in an attribute and modifies the *credential* parameter to point to it. This is a shallow copy.

Parameters:

attr The attribute set to query for its *credential*.

credential An output parameter that will be initialized to point to the GSSAPI credential which the *attr* is currently using.

Returns:

Upon success, **globus_gram_client_attr_get_credential()** (p. 26) modifies the value pointed to by the *credential* parameter to be the same credential as that being used by the attribute named by the *attr* parameter and returns *GLOBUS_SUCCESS*. If an error occurs, **globus_gram_client_attr_get_credential()** (p. 26) returns an integer error code and the value pointed to by the *credential* parameter is undefined.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR Invalid attribute

GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER Null parameter

See also:

globus_gram_client_attr_set_credential() (p. 26)

5.4.3.5 int globus_gram_client_attr_set_delegation_mode (globus_gram_client_attr_t *attr*, globus_io_secure_delegation_mode_t *mode*)

Set a GRAM client attribute's delegation mode.

The **globus_gram_client_attr_set_delegation_mode()** (p. 27) function sets the value of the **delegation_mode** in an attribute to the delegation mode in the *mode* parameter.

The GRAM client supports the following delegation modes:

- **GLOBUS_IO_SECURE_DELEGATION_MODE_LIMITED_PROXY**
- **GLOBUS_IO_SECURE_DELEGATION_MODE_FULL_PROXY**

Parameters:

attr The attribute set to modify to use the delegation mode in the *mode* parameter.

mode The new value of the delegation mode.

Returns:

Upon success, **globus_gram_client_attr_set_delegation_mode()** (p. 27) modifies the attribute named by the *attr* parameter to use the delegation mode in the *mode* parameter and returns **GLOBUS_SUCCESS**. If an error occurs, **globus_gram_client_attr_set_delegation_mode()** (p. 27) returns an integer error code and the *delegation_mode* attribute value is unchanged.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR Invalid attribute

See also:

globus_gram_client_attr_get_delegation_mode() (p. 27)

5.4.3.6 int globus_gram_client_attr_get_delegation_mode (globus_gram_client_attr_t *attr*, globus_io_secure_delegation_mode_t * *mode*)

Get a GRAM client attribute's security credential.

The **globus_gram_client_attr_get_delegation_mode()** (p. 27) function gets the value of the **delegation_mode** in an attribute and modifies the *mode* parameter to point to its value.

Parameters:

attr The attribute set to query for its *delegation_mode*.

mode An output parameter that will be set to point to the delegation mode which the *attr* is currently using.

Returns:

Upon success, **globus_gram_client_attr_get_delegation_mode()** (p. 27) modifies the the value pointed to by the *mode* parameter as described above and returns *GLOBUS_SUCCESS*. If an error occurs, **globus_gram_client_attr_get_delegation_mode()** (p. 27) returns an integer error code and the value pointed to by the *mode* parameter is undefined.

Return values:

GLOBUS_SUCCESS Success

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR Invalid attribute

GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER Null parameter

See also:

globus_gram_client_attr_get_delegation_mode() (p. 27)

6 globus gram client Data Structure Documentation

6.1 globus_gram_client_job_info_s Struct Reference

Extensible job information structure.

Data Fields

- **globus_hashtable_t extensions**
- **char * job_contact**
- **int job_state**
- **int protocol_error_code**

6.1.1 Detailed Description

Extensible job information structure.

The **globus_gram_client_job_info_t** (p. 3) data type is used to pass protocol extensions along with the standard job status information included in the GRAM2 protocol. This structure contains the information returned in job state callbacks plus a hash table of extension entries that contain **globus_gram_protocol_extension_t** (n ??)me-value pairs.

6.1.2 Field Documentation

6.1.2.1 globus_hashtable_t globus_gram_client_job_info_s::extensions

Table of extension values.

6.1.2.2 char* globus_gram_client_job_info_s::job_contact

GRAM Job Contact String.

6.1.2.3 int globus_gram_client_job_info_s::job_state

GRAM Job State.

6.1.2.4 int globus_gram_client_job_info_s::protocol_error_code

GRAM Error Code.

7 globus gram client Page Documentation

7.1 Resource Manager Contact

The resource manager contact string is used by the GRAM Client to connect to, authenticate to, and request service from a GRAM resource manager.

The simplest contact string is just a host name, but additional fields may be present in the name to qualify the TCP port number it is listening on, the service name (for example, to choose a specific scheduler), and the subject name (security identity) of the resource manager.

For those resource manager contacts which omit the port, service or subject field the following defaults are used:

```
port = 2119 (assigned by IANA)
service = jobmanager
subject = subject based on hostname
```

The following contain all of the acceptable formats of a GRAM Resource Manager Contact:

- host
- host:port
- host:port/service
- host/service
- host:/service
- host::subject
- host:port:subject
- host/service:subject
- host:/service:subject
- host:port/service:subject

Note: Any of the above forms may be preceded by an optional "https://".

Index

extensions

- globus_gram_client_job_info_s, 28

globus_gram_client

- globus_gram_client_debug, 22
- globus_gram_client_error_string, 23
- globus_gram_client_job_contact_free, 23
- globus_gram_client_job_info_destroy, 23
- globus_gram_client_set_credentials, 22
- globus_gram_client_version, 22

globus_gram_client_attr

- globus_gram_client_attr_destroy, 25
- globus_gram_client_attr_get_credential, 26
- globus_gram_client_attr_get_delegation_mode, 27
- globus_gram_client_attr_init, 24
- globus_gram_client_attr_set_credential, 25
- globus_gram_client_attr_set_delegation_mode, 26
- globus_gram_client_attr_t, 24
- GLOBUS_GRAM_CLIENT_NO_ATTR, 24

globus_gram_client_attr_destroy

- globus_gram_client_attr, 25

globus_gram_client_attr_get_credential

- globus_gram_client_attr, 26

globus_gram_client_attr_get_delegation_mode

- globus_gram_client_attr, 27

globus_gram_client_attr_init

- globus_gram_client_attr, 24

globus_gram_client_attr_set_credential

- globus_gram_client_attr, 25

globus_gram_client_attr_set_delegation_mode

- globus_gram_client_attr, 26

globus_gram_client_attr_t

- globus_gram_client_attr, 24

globus_gram_client_callback

- globus_gram_client_callback_allow, 4
- globus_gram_client_callback_disallow, 5
- globus_gram_client_callback_func_t, 2
- globus_gram_client_info_callback_allow, 4
- globus_gram_client_info_callback_func_t, 2
- globus_gram_client_job_info_t, 2
- globus_gram_client_nonblocking_func_t, 3

globus_gram_client_callback_allow

- globus_gram_client_callback, 4

globus_gram_client_callback_disallow

- globus_gram_client_callback, 5

globus_gram_client_callback_func_t

- globus_gram_client_callback, 2

globus_gram_client_debug

- globus_gram_client, 22

globus_gram_client_error_string

- globus_gram_client, 23

globus_gram_client_get_jobmanager_version

- globus_gram_client_job_functions, 7

globus_gram_client_info_callback_allow

- globus_gram_client_callback, 4

globus_gram_client_info_callback_func_t

- globus_gram_client_callback, 2

globus_gram_client_job_callback_register

- globus_gram_client_job_functions, 17

globus_gram_client_job_callback_unregister

- globus_gram_client_job_functions, 18

globus_gram_client_job_cancel

- globus_gram_client_job_functions, 12

globus_gram_client_job_contact_free

- globus_gram_client, 23

globus_gram_client_job_functions

- globus_gram_client_get_jobmanager_version, 7
- globus_gram_client_job_callback_register, 17
- globus_gram_client_job_callback_unregister, 18
- globus_gram_client_job_cancel, 12
- globus_gram_client_job_refresh_credentials, 19
- globus_gram_client_job_request, 9
- globus_gram_client_job_request_with_info, 11
- globus_gram_client_job_signal, 13
- globus_gram_client_job_status, 14
- globus_gram_client_job_status_with_info, 16
- globus_gram_client_ping, 7
- globus_gram_client_register_get_jobmanager_version, 8
- globus_gram_client_register_job_callback_registration, 20
- globus_gram_client_register_job_callback_unregister, 21
- globus_gram_client_register_job_cancel, 12
- globus_gram_client_register_job_refresh_credentials, 19
- globus_gram_client_register_job_request, 9
- globus_gram_client_register_job_request_with_info, 10
- globus_gram_client_register_job_signal, 14
- globus_gram_client_register_job_status, 15
- globus_gram_client_register_job_status_with_info, 16
- globus_gram_client_register_ping, 6

globus_gram_client_job_info_destroy

- globus_gram_client, 23

globus_gram_client_job_info_s, 27

- extensions, 28

- job_contact, 28

- job_state, 28

- protocol_error_code, 28

globus_gram_client_job_info_t

- globus_gram_client_callback, 2

globus_gram_client_job_refresh_credentials

- globus_gram_client_job_functions, 19
- globus_gram_client_job_request
 - globus_gram_client_job_functions, 9
- globus_gram_client_job_request_with_info
 - globus_gram_client_job_functions, 11
- globus_gram_client_job_signal
 - globus_gram_client_job_functions, 13
- globus_gram_client_job_status
 - globus_gram_client_job_functions, 14
- globus_gram_client_job_status_with_info
 - globus_gram_client_job_functions, 16
- GLOBUS_GRAM_CLIENT_NO_ATTR
 - globus_gram_client_attr, 24
- globus_gram_client_nonblocking_func_t
 - globus_gram_client_callback, 3
- globus_gram_client_ping
 - globus_gram_client_job_functions, 7
- globus_gram_client_register_get_jobmanager_version
 - globus_gram_client_job_functions, 8
- globus_gram_client_register_job_callback_registration
 - globus_gram_client_job_functions, 20
- globus_gram_client_register_job_callback_-unregistration
 - globus_gram_client_job_functions, 21
- globus_gram_client_register_job_cancel
 - globus_gram_client_job_functions, 12
- globus_gram_client_register_job_refresh_credentials
 - globus_gram_client_job_functions, 19
- globus_gram_client_register_job_request
 - globus_gram_client_job_functions, 9
- globus_gram_client_register_job_request_with_info
 - globus_gram_client_job_functions, 10
- globus_gram_client_register_job_signal
 - globus_gram_client_job_functions, 14
- globus_gram_client_register_job_status
 - globus_gram_client_job_functions, 15
- globus_gram_client_register_job_status_with_info
 - globus_gram_client_job_functions, 16
- globus_gram_client_register_ping
 - globus_gram_client_job_functions, 6
- globus_gram_client_set_credentials
 - globus_gram_client, 22
- globus_gram_client_version
 - globus_gram_client, 22
- GRAM Client Attribute Functions, 24
- GRAM Job Functions, 5
- Job state callbacks, 1
- job_contact
 - globus_gram_client_job_info_s, 28
- job_state
 - globus_gram_client_job_info_s, 28
- Other GRAM Client Functions, 22
- protocol_error_code
 - globus_gram_client_job_info_s, 28