

globus gram client
7.2

Generated by Doxygen 1.6.1

Mon Oct 19 15:07:50 2009

Contents

1	Resource Management Client API	1
2	Resource Manager Contact	1
3	Module Index	2
3.1	Modules	2
4	Module Documentation	2
4.1	Job state callbacks	2
4.1.1	Typedef Documentation	2
4.1.2	Function Documentation	3
4.2	GRAM Job Functions	4
4.2.1	Function Documentation	5
4.3	Other GRAM Client Functions	12
4.3.1	Function Documentation	13
4.4	GRAM Client Attribute Functions	14
4.4.1	Function Documentation	14

1 Resource Management Client API

The resource manager 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 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://".

3 Module Index

3.1 Modules

Here is a list of all modules:

Job state callbacks	2
GRAM Job Functions	4
Other GRAM Client Functions	12
GRAM Client Attribute Functions	14

4 Module Documentation

4.1 Job state callbacks

Typedefs

- typedef void(* [globus_gram_client_callback_func_t](#))(void *user_callback_arg, char *job_contact, int state, int errorcode)
- typedef void * [globus_gram_client_attr_t](#)
- 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_callback_disallow](#) (char *callback_contact)

4.1.1 Typedef Documentation

- 4.1.1.1 **typedef void(* [globus_gram_client_callback_func_t](#))(void *user_callback_arg, char *job_contact, int state, int errorcode)**

GRAM state callback type. Type of a GRAM Client state callback function. A pointer to a function of this type is passed to the [globus_gram_client_callback_allow\(\)](#) function to create a callback contact. This contact can be passed to [globus_gram_client_job_request\(\)](#) or [globus_gram_client_job_callback_register\(\)](#) to let the job manager know to send information on GRAM job state changes to the user's function.

Parameters:

- user_callback_arg* A pointer to arbitrary user data.
- job_contact* A string containing the job contact. This string will contain the same value as the return job_contact parameter from [globus_gram_client_job_request\(\)](#).
- state* The new state (one of the **globus_gram_protocol_job_state_t** values) of the job.
- errorcode* The error code if the *state* parameter is equal to GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED.

4.1.1.2 `typedef void* globus_gram_client_attr_t`

GRAM operation attribute.

4.1.1.3 `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)`

GRAM nonblocking operation callback function. Type of a callback indicating completion of a nonblocking GRAM call.

Parameters:

- user_callback_arg* The register_callback_arg value passed to the nonblocking function.
- operation_failure_code* The result of nonblocking call, indicating whether the call was processed by the job manager successfully or not.
- job_contact* A string containing the job contact.
- job_state* The new state (one of the **globus_gram_protocol_job_state_t** values) of the job.
- job_failure_code* The error code of the job request if the job_state parameter is GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED.

4.1.2 Function Documentation

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

Create a callback contact. Creates a small GRAM server which can handle GRAM state updates from job managers. The contact information for this server is returned and may be used with the [globus_gram_client_job_request\(\)](#) or [globus_gram_client_callback_register\(\)](#) functions.

Parameters:

- callback_func* A pointer to the user's callback function to be called when GRAM state changes are received from a Job Manager.

user_callback_arg A pointer to arbitrary data which is passed as the first parameter to the *callback_func* function when it is called.

callback_contact A pointer to a char *. This pointer will be initialized with a newly allocated string containing the information needed by the Job Manager to contact this GRAM callback server. This string should be freed by the user when it is no longer used.

Returns:

This function returns GLOBUS_SUCCESS if the callback contact create was successful. Otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR_* values will be returned, indicating why the client could not create the callback contact.

4.1.2.2 int globus_gram_client_callback_disallow (char * *callback_contact*)

Disable a callback handler. Disables the GRAM server created by calling [globus_gram_client_callback_allow\(\)](#). This function blocks until all pending job state updates being handled by this server are dispatched.

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

Parameters:

callback_contact The callback contact string returned by calling [globus_gram_client_callback_allow](#).

Returns:

This function returns GLOBUS_SUCCESS if the callback contact was disabled successful. Otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR_* values will be returned, indicating why the client could not disable the callback contact.

4.2 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_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_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_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)

4.2.1 Function Documentation

4.2.1.1 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*)

Verify that a gatekeeper is running (nonblocking). Sends a specially-formatted GRAM protocol message which checks to see if a Globus Gatekeeper is running on a given PORT, and whether that Gatekeeper is configured to support the desired job manager service. This is primarily used for diagnostic purposes.

If this function determines that the ping 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.

Parameters:

resource_manager_contact A NULL-terminated character string containing a [GRAM contact](#).

attr Client attributes to be used. Should be set to GLOBUS_GRAM_CLIENT_NO_ATTR if no attributes are to be used.

register_callback The callback function to call when the ping request has completed.

register_callback_arg A pointer to user data which will be passed to the callback as it's *user_callback_arg*.

Returns:

This function returns GLOBUS_SUCCESS if The gatekeeper contact is valid, the client was able to authenticate with the Gatekeeper, and the Gatekeeper was able to locate the requested service. Otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR values is returned.

4.2.1.2 int [globus_gram_client_ping](#) (const char * *resource_manager_contact*)

Verify that a gatekeeper is running. Sends a specially-formatted GRAM protocol message which checks to see if a Globus Gatekeeper is running on a given PORT, and whether that Gatekeeper is configured to support the desired job manager service. This is primarily used for diagnostic purposes.

This function blocks while processing the ping request.

Parameters:

resource_manager_contact A NULL-terminated character string containing a [GRAM contact](#).

Returns:

This function returns GLOBUS_SUCCESS if The gatekeeper contact is valid, the client was able to authenticate with the Gatekeeper, and the Gatekeeper was able to locate the requested service. Otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR values is returned.

4.2.1.3 `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)`

Request a job be started (nonblocking). Request access to interactive resources at the current time. A job request is atomic: either all of the requested processes are created, or none are created. This is the nonblocking version of [globus_gram_client_job_request\(\)](#). Instead of waiting for the job manager to acknowledge that the job has been submitted or started, this function immediately returns after beginning the job submission. The *register_callback* function will be called to let the caller know whether the job request has been submitted successfully or not.

If this function determines that the job request could not be processed before contacting the job manager (for example, a malformed *resource_manager_contact*) it will return an error, and the *register_callback* function will not be called.

Parameters:

resource_manager_contact A NULL-terminated character string containing a [GRAM contact](#).

description An RSL description of the requested job. A GRAM RSL consists of a conjunction of [RSL parameters](#).

job_state_mask 0, a bitwise OR of the GLOBUS_GRAM_PROTOCOL_JOB_STATE_* states, or GLOBUS_GRAM_PROTOCOL_JOB_STATE_ALL.

callback_contact The URL which will receive all messages about the job.

attr Client attributes to be used. Should be set to GLOBUS_GRAM_CLIENT_NO_ATTR if no attributes are to be used.

register_callback The callback function to call when the job request submission has completed. This function will be passed a copy of the job_contact which the user must free, and an error code (the job status value is undefined).

register_callback_arg A pointer to user data which will be passed to the callback as it's *user_callback_arg*.

Returns:

This function returns GLOBUS_SUCCESS if successful, otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR values is returned.

See also:

[Resource Manager Contact](#)
[globus_gram_client_job_request\(\)](#)

4.2.1.4 `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)`

Request a job be started. Request access to interactive resources at the current time. A job request is atomic: either all of the requested processes are created, or none are created.

Parameters:

resource_manager_contact A NULL-terminated character string containing a [GRAM contact](#).

description An RSL description of the requested job. A GRAM RSL consists of a conjunction of [RSL parameters](#).

job_state_mask 0, a bitwise OR of the GLOBUS_GRAM_PROTOCOL_JOB_STATE_* states, or GLOBUS_GRAM_PROTOCOL_JOB_STATE_ALL.

callback_contact The URL which will receive all messages about the job.

job_contact In a successful case, this is set to a unique identifier for each job.

Returns:

This function returns GLOBUS_SUCCESS if successful, otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR values is returned.

See also:

[Resource Manager Contact](#)

4.2.1.5 int globus_gram_client_job_cancel (const char **job_contact*)

Cancel a GRAM-managed job. Removes a *PENDING* job request, or kills all processes associated with an *ACTIVE* job, releasing any associated resources

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from [globus_gram_client_job_request\(\)](#) or [globus_gram_client_register_job_request\(\)](#).

Returns:

This function returns GLOBUS_SUCCESS if the cancellation was successful. Otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR_* values will be returned, indicating why the client could not cancel the job.

See also:

[globus_gram_client_register_job_cancel\(\)](#)

4.2.1.6 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*)

Nonblocking cancel of a GRAM-managed job. Removes a *PENDING* job request, or kills all processes associated with an *ACTIVE* job, releasing any associated resources

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from [globus_gram_client_job_request\(\)](#) or [globus_gram_client_register_job_request\(\)](#).

attr Client attributes to be used. Should be set to GLOBUS_GRAM_CLIENT_NO_ATTR if no attributes are to be used.

register_callback The callback function to call when the job request cancel has completed.

register_callback_arg A pointer to user data which will be passed to the callback as it's *user_callback_arg*.

Returns:

This function returns GLOBUS_SUCCESS if successful, otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR values is returned.

See also:

[globus_gram_client_job_cancel\(\)](#)

4.2.1.7 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*)

Signal a job manager. Send a signal to a GRAM job manager to modify the way it handles a job request. 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](#)

Parameters:

job_contact The job contact string of the job manager to signal.

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.

job_status A pointer to an integer which will return the new job status, if the signal causes the job's state to change (for example, the GLOBUS_GRAM_PROTOCOL_JOB_CANCEL signal will cause the job to enter the GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED).

failure_code An error code indicating why the job manager could not process the signal.

Returns:

This function returns GLOBUS_SUCCESS if the signal was successful. Otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR_* values will be returned, indicating why the client could not signal the job.

See also:

[globus_gram_client_register_job_signal\(\)](#)

4.2.1.8 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*)

Nonblocking signal a job manager. Send a signal to a GRAM job manager to modify the way it handles a job request. 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](#)

Parameters:

job_contact The job contact string of the job manager to signal.

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`.

attr Client attributes to be used. Should be set to `GLOBUS_GRAM_CLIENT_NO_ATTR` if no attributes are to be used.

register_callback The callback function to call when the job signal has completed.

register_callback_arg A pointer to user data which will be passed to the callback as it's `user_callback_arg`.

Returns:

This function returns `GLOBUS_SUCCESS` if successful, otherwise one of the `GLOBUS_GRAM_PROTOCOL_ERROR` values is returned.

4.2.1.9 `int globus_gram_client_job_status (const char *job_contact, int *job_status, int *failure_code)`

Query a job's status. This function queries the status of the job associated with the job contact, returning it's current job status and job failure reason if it has failed.

Parameters:

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

job_status A pointer to an integer which will be populated with the current status of the job. This will be one of the `GLOBUS_GRAM_PROTOCOL_JOB_STATE_*` values if this function is successful.

failure_code The reason why the job failed if the `job_status` is set to `GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED`.

Returns:

This function returns `GLOBUS_SUCCESS` if the job state query was successfully. Otherwise one of the `GLOBUS_GRAM_PROTOCOL_ERROR_*` values will be returned, indicating why the client could not query the job state.

4.2.1.10 `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)`

Nonblocking query of a job's status. This function queries the status of the job associated with the job contact, returning it's current job status and job failure reason if it has failed.

Parameters:

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

attr Client attributes to be used. Should be set to `GLOBUS_GRAM_CLIENT_NO_ATTR` if no attributes are to be used.

register_callback Callback function to be called when the job status query has been processed.

register_callback_arg A pointer to user data which will be passed to the callback as it's `user_callback_arg`.

Returns:

This function returns `GLOBUS_SUCCESS` if the job state query was successfully. Otherwise one of the `GLOBUS_GRAM_PROTOCOL_ERROR_*` values will be returned, indicating why the client could not query the job state.

4.2.1.11 `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 callback contact for job state changes.

Parameters:

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

job_state_mask A mask indicating which job state changes should be sent to the *callback_contact*. This may be 0 (no job state changes), a bitwise-or of the `GLOBUS_GRAM_PROTOCOL_JOB_STATE_*` states, or `GLOBUS_GRAM_PROTOCOL_JOB_STATE_ALL` to register for all job states.

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

job_status A pointer to an integer which will be populated with the current status of the job. This will be one of the `GLOBUS_GRAM_PROTOCOL_JOB_STATE_*` values if this function is successful.

failure_code Set to an error code when the job manager is unable to process this registration.

Returns:

This function returns `GLOBUS_SUCCESS` if the callback registration was successful. Otherwise one of the `GLOBUS_GRAM_PROTOCOL_ERROR_*` values will be returned, indicating why the client could not register the callback contact.

4.2.1.12 `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 from future job state changes.

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()`.

callback_contact A URL string containing a GRAM client callback. This string should normally be generated by a process calling `globus_gram_client_callback_allow()`. If this function returns successfully, the process managing the *callback_contact* should not receive future job state changes.

job_status A pointer to an integer which will be populated with the current status of the job. This will be one of the `GLOBUS_GRAM_PROTOCOL_JOB_STATE_*` values if this function is successful.

failure_code Set to an error code when the job manager is unable to process this registration.

Returns:

This function returns `GLOBUS_SUCCESS` if the callback unregistration was successful. Otherwise one of the `GLOBUS_GRAM_PROTOCOL_ERROR_*` values will be returned, indicating why the client could not unregister the callback contact.

4.2.1.13 `int globus_gram_client_job_refresh_credentials (char *job_contact, gss_cred_id_t creds)`

Delegate new credentials to a job manager. This function performs a new delegation handshake with the job manager, updating it with a new user proxy. This will allow the job manager to continue to send job state callbacks after the original proxy would have expired.

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\(\)](#).

creds A credential which should be used to contact the job manager. This may be GSS_C_NO_CREDENTIAL to use the process's default credential.

Returns:

This function returns GLOBUS_SUCCESS if the delegation was successful. Otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR_* values will be returned, indicating why the client could not un-register the callback contact.

4.2.1.14 `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 new credentials to a job manager (nonblocking). This function performs the same operation as [globus_gram_client_job_refresh_credentials\(\)](#), but without blocking the calling thread. Once the delegation has completed, it's final status will be reported in the *register_callback*.

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\(\)](#).

creds A credential which should be used to contact the job manager. This may be GSS_C_NO_CREDENTIAL to use the process's default credential.

attr Client attributes to be used. Should be set to GLOBUS_GRAM_CLIENT_NO_ATTR if no attributes are to be used.

register_callback Callback function to be called when the job refresh has been processed.

register_callback_arg A pointer to user data which will be passed to the callback as it's *user_callback_arg*.

4.2.1.15 `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)`

Nonblocking register a callback contact for job state changes.

Parameters:

job_contact The job contact string of the job to contact. This is the same value returned from [globus_gram_client_job_request\(\)](#).

job_state_mask A mask indicating which job state changes should be sent to the *callback_contact*. This may be 0 (no job state changes), a bitwise-or of the GLOBUS_GRAM_PROTOCOL_JOB_STATE_* states, or GLOBUS_GRAM_PROTOCOL_JOB_STATE_ALL to register for all job states.

callback_contact A URL string containing a GRAM client callback. This string should normally be generated by a process calling [globus_gram_client_callback_allow\(\)](#).

attr Client attributes to be used. Should be set to GLOBUS_GRAM_CLIENT_NO_ATTR if no attributes are to be used.

register_callback The callback function to call when the job signal has completed.

register_callback_arg A pointer to user data which will be passed to the callback as it's *user_callback_arg*.

Returns:

This function returns GLOBUS_SUCCESS if the successful, otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR_* values will be returned, indicating why the operation failed.

See also:

[globus_gram_client_job_callback_register\(\)](#)

4.2.1.16 `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)`

Nonblocking unregistration of a callback contact.

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\(\)](#).

callback_contact A URL string containing a GRAM client callback. This string should normally be generated by a process calling [globus_gram_client_callback_allow\(\)](#). If this function returns successfully, the process managing the *callback_contact* should not receive future job state changes.

attr Client attributes to be used. Should be set to GLOBUS_GRAM_CLIENT_NO_ATTR if no attributes are to be used.

register_callback The callback function to call when the job signal has completed.

register_callback_arg A pointer to user data which will be passed to the callback as it's *user_callback_arg*.

Returns:

This function returns GLOBUS_SUCCESS if the successful, otherwise one of the GLOBUS_GRAM_PROTOCOL_ERROR_* values will be returned, indicating why the operation failed.

See also:

[globus_gram_client_job_callback_unregister\(\)](#)

4.3 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)

4.3.1 Function Documentation

4.3.1.1 void globus_gram_client_debug (void)

Enable debugging messages. Enables the displaying of internal debugging information.

Returns:

void

4.3.1.2 int globus_gram_client_version (void)

Version checking

Return the version of the GRAM protocol which this implementation of the GRAM client library understands.

Returns:

An integer representing the protocol revision.

4.3.1.3 int globus_gram_client_set_credentials (gss_cred_id_t *new_credentials*)

Set the default GRAM credential

Set the credential to use by default for all future GRAM network operations. These include job requests, job signals, callback registration, and job state callbacks. The credential must no longer be used by the caller, and will be freed by GRAM when no longer needed.

Parameters:

new_credentials New GSSAPI credential to use.

Returns:

This function returns GLOBUS_SUCCESS if the default credential has been set.

4.3.1.4 const char* globus_gram_client_error_string (int *error_code*)

Error code translation. This function takes the 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.

Parameters:

error_code The error code to look up.

Returns:

An error string containing the reason for the error. The error string is written to be used in the context "[operation] failed because [error_string]".

4.3.1.5 `int globus_gram_client_job_contact_free (char *job_contact)`

Releases the resources storing a job contact string.

Parameters:

job_contact A job contact string returned in a successful call to [globus_gram_client_job_request\(\)](#)

4.4 GRAM Client Attribute Functions

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)`

4.4.1 Function Documentation

4.4.1.1 `int globus_gram_client_attr_init (globus_gram_client_attr_t *attr)`

Initialize a GRAM client attribute.

Parameters:

attr Pointer to attribute to initialize.

Return values:

GLOBUS_SUCCESS Attribute was initialized successfully.

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR Null value was passed in for attribute.

GLOBUS_GRAM_PROTOCOL_ERROR_MALLOC_FAILED Unable to allocate memory to initialize attribute.

See also:

[globus_gram_client_attr_destroy\(\)](#)

4.4.1.2 `int globus_gram_client_attr_destroy (globus_gram_client_attr_t *attr)`

Destroy a GRAM client attribute.

Parameters:

attr Pointer to attribute to destroy.

Return values:

GLOBUS_SUCCESS Attribute was initialized successfully.

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR Null value was passed in for attribute.

See also:

[globus_gram_client_attr_init\(\)](#)

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

Associate a credential with an attribute set.

Parameters:

attr The attribute set to associate the credential with.

credential The credential to use.

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\(\)](#)

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

Get the credential associated with an attribute set.

Parameters:

attr The attribute set associated with the credential.

credential A pointer to a GSSAPI credential handle which will be set to the value of the credential currently associated with the attribute set.

Return values:

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR An invalid attribute set was passed to this function.

GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER A null credential pointer was passed to this function.

See also:

[globus_gram_client_attr_set_credential\(\)](#)

4.4.1.5 `int globus_gram_client_attr_set_delegation_mode (globus_gram_client_attr_t attr, globus_io_secure_delegation_mode_t mode)`

Set the delegation mode associated with an attribute set.

Parameters:

attr The attribute set to update.

mode The new value of the delegation mode.

Return values:

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR An invalid attribute set was passed to this function.

See also:

[globus_gram_client_attr_get_delegation_mode\(\)](#)

4.4.1.6 `int globus_gram_client_attr_get_delegation_mode (globus_gram_client_attr_t attr, globus_io_secure_delegation_mode_t * mode)`

Get the delegation mode associated with an attribute set.

Parameters:

attr The attribute set to query.

mode Pointer to a location to set to the value of the delegation mode.

Return values:

GLOBUS_GRAM_PROTOCOL_ERROR_INVALID_ATTR An invalid attribute set was passed to this function.

GLOBUS_GRAM_PROTOCOL_ERROR_NULL_PARAMETER An null *mode* pointer was passed to this function.

See also:

[globus_gram_client_attr_get_delegation_mode\(\)](#)

Index

- globus_gram_client
 - globus_gram_client_debug, 12
 - globus_gram_client_error_string, 13
 - globus_gram_client_job_contact_free, 13
 - globus_gram_client_set_credentials, 13
 - globus_gram_client_version, 12
- globus_gram_client_attr
 - globus_gram_client_attr_destroy, 14
 - globus_gram_client_attr_get_credential, 15
 - globus_gram_client_attr_get_delegation_mode, 16
 - globus_gram_client_attr_init, 14
 - globus_gram_client_attr_set_credential, 14
 - globus_gram_client_attr_set_delegation_mode, 15
- globus_gram_client_attr_destroy
 - globus_gram_client_attr, 14
- globus_gram_client_attr_get_credential
 - globus_gram_client_attr, 15
- globus_gram_client_attr_get_delegation_mode
 - globus_gram_client_attr, 16
- globus_gram_client_attr_init
 - globus_gram_client_attr, 14
- globus_gram_client_attr_set_credential
 - globus_gram_client_attr, 14
- globus_gram_client_attr_set_delegation_mode
 - globus_gram_client_attr, 15
- globus_gram_client_attr_t
 - globus_gram_client_callback, 2
- globus_gram_client_callback
 - globus_gram_client_attr_t, 2
 - globus_gram_client_callback_allow, 3
 - globus_gram_client_callback_disallow, 3
 - globus_gram_client_callback_func_t, 2
 - globus_gram_client_nonblocking_func_t, 2
- globus_gram_client_callback_allow
 - globus_gram_client_callback, 3
- globus_gram_client_callback_disallow
 - globus_gram_client_callback, 3
- globus_gram_client_callback_func_t
 - globus_gram_client_callback, 2
- globus_gram_client_debug
 - globus_gram_client, 12
- globus_gram_client_error_string
 - globus_gram_client, 13
- globus_gram_client_job_callback_register
 - globus_gram_client_job_functions, 9
- globus_gram_client_job_callback_unregister
 - globus_gram_client_job_functions, 10
- globus_gram_client_job_cancel
 - globus_gram_client_job_functions, 6
- globus_gram_client_job_contact_free
 - globus_gram_client, 13
- globus_gram_client_job_functions
 - globus_gram_client_job_callback_register, 9
 - globus_gram_client_job_callback_unregister, 10
 - globus_gram_client_job_cancel, 6
 - globus_gram_client_job_refresh_credentials, 10
 - globus_gram_client_job_request, 6
 - globus_gram_client_job_signal, 7
 - globus_gram_client_job_status, 8
 - globus_gram_client_ping, 5
 - globus_gram_client_register_job_callback_registration, 11
 - globus_gram_client_register_job_callback_unregister, 11
 - globus_gram_client_register_job_cancel, 7
 - globus_gram_client_register_job_refresh_credentials, 10
 - globus_gram_client_register_job_request, 5
 - globus_gram_client_register_job_signal, 8
 - globus_gram_client_register_job_status, 9
 - globus_gram_client_register_ping, 4
- globus_gram_client_job_refresh_credentials
 - globus_gram_client_job_functions, 10
- globus_gram_client_job_request
 - globus_gram_client_job_functions, 6
- globus_gram_client_job_signal
 - globus_gram_client_job_functions, 7
- globus_gram_client_job_status
 - globus_gram_client_job_functions, 8
- globus_gram_client_nonblocking_func_t
 - globus_gram_client_callback, 2
- globus_gram_client_ping
 - globus_gram_client_job_functions, 5
- globus_gram_client_register_job_callback_registration
 - globus_gram_client_job_functions, 11
- globus_gram_client_register_job_callback_unregister
 - globus_gram_client_job_functions, 11
- globus_gram_client_register_job_cancel
 - globus_gram_client_job_functions, 7
- globus_gram_client_register_job_refresh_credentials
 - globus_gram_client_job_functions, 10
- globus_gram_client_register_job_request
 - globus_gram_client_job_functions, 5
- globus_gram_client_register_job_signal
 - globus_gram_client_job_functions, 8
- globus_gram_client_register_job_status
 - globus_gram_client_job_functions, 9
- globus_gram_client_register_ping
 - globus_gram_client_job_functions, 4
- globus_gram_client_set_credentials
 - globus_gram_client, 13
- globus_gram_client_version
 - globus_gram_client, 12

GRAM Client Attribute Functions, [14](#)

GRAM Job Functions, [4](#)

Job state callbacks, [2](#)

Other GRAM Client Functions, [12](#)