

globus gsi callback Reference Manual

1.10

Generated by Doxygen 1.3.5

Tue Aug 11 21:05:21 2009

Contents

1 Globus GSI Callback	1
2 globus gsi callback Module Index	1
3 globus gsi callback Page Index	1
4 globus gsi callback Module Documentation	2
5 globus gsi callback Page Documentation	14

1 Globus GSI Callback

The Globus GSI Callback library. This library contains functions that extend OpenSSL path validation.

- [Activation](#)
- [Callback Functions](#)
- [Callback Data Functions](#)

2 globus gsi callback Module Index

2.1 globus gsi callback Modules

Here is a list of all modules:

Activation	2
Callback Functions	2
Callback Data Functions	5
GSI Callback Constants	12

3 globus gsi callback Page Index

3.1 globus gsi callback Related Pages

Here is a list of all related documentation pages:

Deprecated List	14
---------------------------------	--------------------

4 globus gsi callback Module Documentation

4.1 Activation

Globus GSI Callback uses standard Globus module activation and deactivation.

Defines

- `#define GLOBUS_GSI_CALLBACK_MODULE`

4.1.1 Detailed Description

Globus GSI Callback uses standard Globus module activation and deactivation.

Before any Globus GSI Callback functions are called, the following function must be called:

```
globus_module_activate(GLOBUS_GSI_CALLBACK_MODULE)
```

This function returns `GLOBUS_SUCCESS` if Globus GSI Callback was successfully initialized, and you are therefore allowed to subsequently call Globus GSI Callback functions. Otherwise, an error code is returned, and Globus GSI Credential functions should not be subsequently called. This function may be called multiple times.

To deactivate Globus GSI Callback, the following function must be called:

```
globus_module_deactivate(GLOBUS_GSI_CALLBACK_MODULE)
```

This function should be called once for each time Globus GSI Callback was activated.

4.1.2 Define Documentation

4.1.2.1 `#define GLOBUS_GSI_CALLBACK_MODULE`

Module descriptor.

4.2 Callback Functions

Functions that plug into various plug points in the OpenSSL path validation mechanism.

Get callback data index from X509_STORE

- `globus_result_t globus_gsi_callback_get_X509_STORE_callback_data_index(index)`

Get callback data index from SSL structure

- `globus_result_t globus_gsi_callback_get_SSL_callback_data_index(index)`

Certificate verify wrapper

- `int globus_gsi_callback_X509_verify_cb(X509_STORE_CTX context, void arg)`

Independent path validation callback.

- int [globus_gsi_callback_create_proxy_callback](#)(int preverify_ok, X509_STORE_CTX x509_context)

SSL path validation callback.

- int [globus_gsi_callback_handshake_callback](#)(int preverify_ok, X509_STORE_CTX x509_context)

OpenSSL X509_check_issued() wrapper

- int [globus_gsi_callback_check_issued](#)(X509_STORE_CTX context, X509 cert, X509 issuer)

Typedefs

- typedef int([globus_gsi_extension_callback](#))(globus_gsi_callback_data_t callback_data, X509_EXTENSION extension)

4.2.1 Detailed Description

Functions that plug into various plug points in the OpenSSL path validation mechanism.

These functions add CRL checking, X509 Extension handling and proxy validation.

4.2.2 Typedef Documentation

4.2.2.1 typedef int([globus_gsi_extension_callback](#))(globus_gsi_callback_data_t callback_data, X509_EXTENSION extension)

Typedef for a callback that may be registered for dealing with unhandled X.509 extension.

4.2.3 Function Documentation

4.2.3.1 globus_result_t [globus_gsi_callback_get_X509_STORE_callback_data_index](#) (int index)

Retrieve or create the index for our callback data structure in the X509_STORE.

Parameters:

index Will contain the index upon return

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.2.3.2 globus_result_t [globus_gsi_callback_get_SSL_callback_data_index](#) (int index)

Retrieve or create the index for our callback data structure in the SSL structure.

Parameters:

index Will contain the index upon return

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.2.3.3 `int globus_gsi_callback_X509_verify_cert (X509_STORE_CTX context, void arg)`

This function wraps the OpenSSL certificate verification callback for the purpose of replacing the standard issuer check with one that deals with proxy certificates. Should be used with `SSL_CTX_set_cert_verify_callback()`

Parameters:

`context` The `X509_STORE_CTX` for which to register the callback.

`arg` Arguments to the callback. Currently ignored.

Returns:

1 on success 0 on failure

4.2.3.4 `int globus_gsi_callback_create_proxy_callback (int preverify_ok, X509_STORE_CTX x509_context)`

This function provides a path validation callback for validation outside of a SSL session. It should be used in `X509_STORE_set_verify_cb_func()`.

Parameters:

`preverify_ok` Communicates the result of default validation steps performed by OpenSSL

`x509_context` The validation state object

Returns:

1 on success 0 on failure

4.2.3.5 `int globus_gsi_callback_handshake_callback (int preverify_ok, X509_STORE_CTX x509_context)`

This function provides a path validation callback for the validation part of establishing a SSL session. It handles proxy certificates, X509 Extensions and CRL checking. It should be used in `SSL_CTX_set_verify()`.

Parameters:

`preverify_ok` Communicates the result of default validation steps performed by OpenSSL

`x509_context` The validation state object.

Returns:

1 on success 0 on failure

4.2.3.6 `int globus_gsi_callback_check_issued (X509_STORE_CTX context, X509 cert, X509 issuer)`

This function wraps the OpenSSL `X509_check_issued()` call and catches the error caused by the fact that a proxy certificate issuer may not have to have the correct `KeyUsage` fields set.

Parameters:

`context` The validation state object.

`cert` The certificate to check

`issuer` The issuer certificate to check

Returns:

1 on success 0 on failure

4.3 Callback Data Functions

Functions that deal with the data structure that contains state associated with the path validation callback.

Initializing and destroying a callback data structure

- `globus_result_t globus_gsi_callback_data_init(globus_gsi_callback_data_t callback_data)`
- `globus_result_t globus_gsi_callback_data_destroy(globus_gsi_callback_data_t callback_data)`

Copying a callback data structure

- `globus_result_t globus_gsi_callback_data_copy(globus_gsi_callback_data_t source, globus_gsi_callback_data_t dest)`

Getting and setting the certificate chain depth

- `globus_result_t globus_gsi_callback_get_cert_depth(globus_gsi_callback_data_t callback_data, int cert_depth)`
- `globus_result_t globus_gsi_callback_set_cert_depth(globus_gsi_callback_data_t callback_data, int cert_depth)`

Getting and setting the "proxy chain" depth

- `globus_result_t globus_gsi_callback_get_proxy_depth(globus_gsi_callback_data_t callback_data, int proxy_depth)`
- `globus_result_t globus_gsi_callback_set_proxy_depth(globus_gsi_callback_data_t callback_data, int proxy_depth)`

Getting and setting the certificate type

- `globus_result_t globus_gsi_callback_get_cert_type(globus_gsi_callback_data_t callback_data, globus_gsi_cert_utils_cert_type_t cert_type)`
- `globus_result_t globus_gsi_callback_set_cert_type(globus_gsi_callback_data_t callback_data, globus_gsi_cert_utils_cert_type_t cert_type)`

Getting and setting the certificate chain

- `globus_result_t globus_gsi_callback_get_cert_chain(globus_gsi_callback_data_t callback_data, STACK_OF(X509) cert_chain)`
- `globus_result_t globus_gsi_callback_set_cert_chain(globus_gsi_callback_data_t callback_data, STACK_OF(X509) cert_chain)`

Getting and setting the limited proxy handling setting

- `globus_result_t globus_gsi_callback_get_multiple_limited_proxy_ok(globus_gsi_callback_data_t callback_data, int multiple_limited_proxy_ok)`
- `globus_result_t globus_gsi_callback_set_multiple_limited_proxy_ok(globus_gsi_callback_data_t callback_data, int multiple_limited_proxy_ok)`

Getting and setting a set of X.509 extension OIDs.

- `globus_result_t globus_gsi_callback_get_extension_oids(globus_gsi_callback_data_t callback_data, void extension_oids)`
- `globus_result_t globus_gsi_callback_set_extension_oids(globus_gsi_callback_data_t callback_data, void extension_oids)`

Getting and setting the trusted certificate directory

- `globus_result_t globus_gsi_callback_get_cert_dir(globus_gsi_callback_data_t callback_data, char cert_dir)`
- `globus_result_t globus_gsi_callback_set_cert_dir(globus_gsi_callback_data_t callback_data, char cert_dir)`

Getting and setting the callback to be called for unknown X.509 extensions

- `globus_result_t globus_gsi_callback_get_extension(globus_gsi_callback_data_t callback_data, globus_gsi_extension_callback_t extension_cb)`
- `globus_result_t globus_gsi_callback_set_extension(globus_gsi_callback_data_t callback_data, globus_gsi_extension_callback_t extension_cb)`

Getting and setting the error status

- `globus_result_t globus_gsi_callback_get_error(globus_gsi_callback_data_t callback_data, globus_result_t error)`
- `globus_result_t globus_gsi_callback_set_error(globus_gsi_callback_data_t callback_data, globus_result_t error)`

Getting and setting the check self-signed policy flag

- `globus_result_t globus_gsi_callback_get_check_policy_for_self_signed_certs(globus_gsi_callback_data_t callback_data, globus_bool_t check)`
- `globus_result_t globus_gsi_callback_set_check_policy_for_self_signed_certs(globus_gsi_callback_data_t callback_data, globus_bool_t check)`

Typedefs

- `typedef globus_l_gsi_callback_data_t globus_gsi_callback_data_t`

4.3.1 Detailed Description

Functions that deal with the data structure that contains state associated with the path validation callback.

4.3.2 Typedef Documentation

4.3.2.1 typedef struct [globus_l_gsi_callback_data_t](#) [globus_gsi_callback_data_t](#)

Callback data typedef.

4.3.3 Function Documentation

4.3.3.1 `globus_result_t globus_gsi_callback_data_init(globus_gsi_callback_data_t callback_data)`

This function initializes a `globus_gsi_callback_data_t`.

Parameters:

`callback_data` Reference to the structure to be initialized

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.2 `globus_result_t globus_gsi_callback_data_destroy(globus_gsi_callback_data_t callback_data)`

This function destroys a `globus_gsi_callback_data_t`.

Parameters:

`callback_data` The structure to be destroyed

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.3 `globus_result_t globus_gsi_callback_data_copy(globus_gsi_callback_data_t source, globus_gsi_callback_data_t dest)`

This function copies a `globus_gsi_callback_data_t`.

Parameters:

`source` The structure to be copied

`dest` The destination of the copy

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.4 `globus_result_t globus_gsi_callback_get_cert_depth(globus_gsi_callback_data_t callback_data, int cert_depth)`

This function returns the certificate chain depth.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to retrieve the depth from

`cert_depth` The returned certificate chain depth

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.5 `globus_result_t globus_gsi_callback_set_cert_depth(globus_gsi_callback_data_t callback_data, int cert_depth)`

This function sets the certificate chain depth.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to retrieve the depth from

`cert_depth` The certificate chain depth

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.6 `globus_result_t globus_gsi_callback_get_proxy_depth(globus_gsi_callback_data_t callback_data, int proxy_depth)`

This function returns the number of proxies in the certificate chain.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to retrieve the depth from

`proxy_depth` The returned "proxy chain" depth

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.7 `globus_result_t globus_gsi_callback_set_proxy_depth(globus_gsi_callback_data_t callback_data, int proxy_depth)`

This function sets the number of proxies in the certificate chain.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to retrieve the depth from

`proxy_depth` The "proxy chain" depth

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.8 `globus_result_t globus_gsi_callback_get_cert_type(globus_gsi_callback_data_t callback_data, globus_gsi_cert_utils_cert_type_t cert_type)`

This function returns the certificate type of the certificate currently being processed.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to retrieve the certificate type from

`cert_type` Variable containing the certificate type on return

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.9 `globus_result_t globus_gsi_callback_set_cert_type(globus_gsi_callback_data_t callback_data, globus_gsi_cert_utils_cert_type_t cert_type)`

This function sets the certificate type of the certificate currently being processed.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to set the certificate type on
`cert_type` The certificate type

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.10 `globus_result_t globus_gsi_callback_get_cert_chain(globus_gsi_callback_data_t callback_data, STACK_OF(X509) cert_chain)`

This function returns the certificate chain associated with the callback data.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to retrieve the certificate chain from.
`cert_chain` Contains the certificate chain upon successful return

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.11 `globus_result_t globus_gsi_callback_set_cert_chain(globus_gsi_callback_data_t callback_data, STACK_OF(X509) cert_chain)`

This function sets the certificate chain associated with the callback data.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to set the certificate chain on
`cert_chain` The certificate chain

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.12 `globus_result_t globus_gsi_callback_get_multiple_limited_proxy_ok(globus_gsi_callback_data_t callback_data, int multiple_limited_proxy_ok)`

This function gets the value of the limited proxy handling setting. This setting determines whether path validation will accept limited proxies that have been further delegated, ie certificate chains with a limited proxy followed by further proxies.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to get the limited proxy setting from
`multiple_limited_proxy_ok` Contains the value of the setting upon successful return.

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

Deprecated

This function always returns true now. It will be removed in the next release.

4.3.3.13 `globus_result_t globus_gsi_callback_set_multiple_limited_proxy_ok(globus_gsi_callback_data_t callback_data, int multiple_limited_proxy_ok)`

This function sets the value of the limited proxy handling setting. This setting determines whether path validation will accept limited proxies that have been further delegated, ie certificate chains with a limited proxy followed by further proxies.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to set the limited proxy setting on
`multiple_limited_proxy_ok` The value of the setting

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

Deprecated

This function has been turned into a no-op. It will be removed in the next release.

4.3.3.14 `globus_result_t globus_gsi_callback_get_extension_oids(globus_gsi_callback_data_t callback_data, void * extension_oids)`

This function gets a list of X.509 extension OIDs that may be used by the extensions callback to allow or disallow certain extensions.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to get the array of extension OIDs from.
`extension_oids` Contains the list of extension OIDs upon successful return.

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.15 `globus_result_t globus_gsi_callback_set_extension_oids(globus_gsi_callback_data_t callback_data, void * extension_oids)`

This function sets a list of X.509 extension OIDs that may be used by the extensions callback to allow or disallow certain extensions.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to get the array of extension OIDs from.
`extension_oids` The list of extension OIDs

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.16 `globus_result_t globus_gsi_callback_get_cert_dir(globus_gsi_callback_data_t callback_data, char * cert_dir)`

This function gets the trusted certificate directory from the callback data.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to get the trusted certificates directory from.
`cert_dir` Contains the path to the trusted certificate directory upon successful return.

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.17 `globus_result_t globus_gsi_callback_set_cert_dir(globus_gsi_callback_data_t callback_data, char cert_dir)`

This function sets the trusted certificate directory on the callback data.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to set the trusted certificates directory on.

`cert_dir` The path to the trusted certificate directory

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.18 `globus_result_t globus_gsi_callback_get_extension_cb(globus_gsi_callback_data_t callback_data, globus_gsi_extension_callback_t extension_cb)`

This function gets the callback that is called for unknown X.509 extensions.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to get the callback information from

`extension_cb` Contains the extension callback upon successful return.

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.19 `globus_result_t globus_gsi_callback_set_extension_cb(globus_gsi_callback_data_t callback_data, globus_gsi_extension_callback_t extension_cb)`

This function sets the callback that is called for unknown X.509 extensions.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to set the callback information on

`extension_cb` The extension callback

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.20 `globus_result_t globus_gsi_callback_get_error(globus_gsi_callback_data_t callback_data, globus_result_t error)`

This function gets the error status stored in the callback data.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to get the error from

`error` Contains the error upon successful return.

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.21 `globus_result_t globus_gsi_callback_set_error(globus_gsi_callback_data_t callback_data, globus_result_t error)`

This function sets the error status stored in the callback data.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to set the error on
`error` The error

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

4.3.3.22 `globus_result_t globus_gsi_callback_get_check_policy_for_self_signed_cert(globus_gsi_callback_data_t callback_data, globus_bool_t check)`

This function gets the value of the "check policy for self-signed certificates" in the callback data. If this is set then the CA signing policy for a self-signed certificate must include a policy line that allows it to sign itself.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to get the error from
`check` Contains the value of the `ag` upon successful return.

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

Since:

Globus Toolkit 4.2.1

4.3.3.23 `globus_result_t globus_gsi_callback_set_check_policy_for_self_signed_cert(globus_gsi_callback_data_t callback_data, globus_bool_t check)`

This function sets the value of the "check policy for self-signed certificates" in the callback data. If this is set then the CA signing policy for a self-signed certificate must include a policy line that allows it to sign itself.

Parameters:

`callback_data` The `globus_gsi_callback_data_t` to set the error on
`check` New value of the `ag`

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

Since:

Globus Toolkit 4.2.1

4.4 GSI Callback Constants

Enumerations

- `enum globus_gsi_callback_error{t`
`GLOBUS_GSI_CALLBACK_ERROR_SUCCESS=0,`

```

GLOBUS_GSI_CALLBACK_ERROR_VERIFY_CRED= 1,
GLOBUS_GSI_CALLBACK_ERROR_CERT_NOT_YET_VALID= 2,
GLOBUS_GSI_CALLBACK_ERROR_CANT_GET_LOCAL_CA_CERT= 3,
GLOBUS_GSI_CALLBACK_ERROR_CERT_HAS_EXPIRED= 4,
GLOBUS_GSI_CALLBACK_ERROR_INVALID_PROXY= 5,
GLOBUS_GSI_CALLBACK_ERROR_LIMITED_PROXY= 6,
GLOBUS_GSI_CALLBACK_ERROR_INVALID_CRL= 7,
GLOBUS_GSI_CALLBACK_ERROR_REVOKED_CERT= 8,
GLOBUS_GSI_CALLBACK_ERROR_MIXING_DIFFERENT_PROXY_TYPES= 9,
GLOBUS_GSI_CALLBACK_ERROR_WITH_SIGNING_POLICY= 10,
GLOBUS_GSI_CALLBACK_ERROR_OLD_GAA= 11,
GLOBUS_GSI_CALLBACK_ERROR_CALLBACK_DATA= 12,
GLOBUS_GSI_CALLBACK_ERROR_ERRNO= 13,
GLOBUS_GSI_CALLBACK_ERROR_CERT_CHAIN= 14,
GLOBUS_GSI_CALLBACK_ERROR_WITH_CALLBACK_DATA_INDEX= 15,
GLOBUS_GSI_CALLBACK_ERROR_PROXY_PATH_LENGTH_EXCEEDED= 16,
GLOBUS_GSI_CALLBACK_ERROR_LAST= 18 }

```

4.4.1 Enumeration Type Documentation

4.4.1.1 enum `globus_gsi_callback_error_t`

GSI Callback Error codes.

Enumeration values:

GLOBUS_GSI_CALLBACK_ERROR_SUCCESS	Success - never used.
GLOBUS_GSI_CALLBACK_ERROR_VERIFY_CRED	Error verifying credential.
GLOBUS_GSI_CALLBACK_ERROR_CERT_NOT_YET_VALID	The certificate is not yet valid.
GLOBUS_GSI_CALLBACK_ERROR_CANT_GET_LOCAL_CA_CERT	Unable to discover a local trusted CA for a given certificate.
GLOBUS_GSI_CALLBACK_ERROR_CERT_HAS_EXPIRED	The certificate has expired.
GLOBUS_GSI_CALLBACK_ERROR_INVALID_PROXY	The proxy format is invalid.
GLOBUS_GSI_CALLBACK_ERROR_LIMITED_PROXY	Limited proxies are not accepted.
GLOBUS_GSI_CALLBACK_ERROR_INVALID_CRL	Invalid CRL.
GLOBUS_GSI_CALLBACK_ERROR_REVOKED_CERT	The certificate has been revoked.
GLOBUS_GSI_CALLBACK_ERROR_MIXING_DIFFERENT_PROXY_TYPES	The certificate chain contains both legacy on rfc compliant proxies.
GLOBUS_GSI_CALLBACK_ERROR_WITH_SIGNING_POLICY	Could not verify certificate chain against the signing policy for the issuing CA.
GLOBUS_GSI_CALLBACK_ERROR_OLD_GAA	Old GAA error.
GLOBUS_GSI_CALLBACK_ERROR_CALLBACK_DATA	Error with the callback data structure.
GLOBUS_GSI_CALLBACK_ERROR_ERRNO	System error.
GLOBUS_GSI_CALLBACK_ERROR_CERT_CHAIN	Error setting or getting the cert chain from callback data.

GLOBUS_GSI_CALLBACK_ERROR_WITH_CALLBACK_DATA_INDEX Error getting callback data index.

GLOBUS_GSI_CALLBACK_ERROR_PROXY_PATH_LENGTH_EXCEEDED Exceeded the path length specified in the proxy cert info extension.

GLOBUS_GSI_CALLBACK_ERROR_LAST Last marker - never used.

5 globus_gsi_callback Page Documentation

5.1 Deprecated List

Global [globus_gsi_callback_get_multiple_limited_proxy_of](#) globus_gsi_callback_data_t callback_data, int multiple_limit
This function always returns true now. It will be removed in the next release.

Global [globus_gsi_callback_set_multiple_limited_proxy_of](#) globus_gsi_callback_data_t callback_data, int multiple_limit
This function has been turned into a no-op. It will be removed in the next release.

Index

Activation, [2](#)

Callback Data Function, [5](#)

Callback Functions, [2](#)

globus_gsi_callback

globus_gsi_callback_check_issue, [4d](#),
globus_gsi_callback_create_proxy_callback, [4k](#),
globus_gsi_callback_get_SSL_callback_data_-
index, [3](#)

globus_gsi_callback_get_X509_STORE_-
callback_data_index, [8](#)

globus_gsi_callback_handshake_callback, [4k](#),
globus_gsi_callback_X509_verify_cert, [8t](#),
globus_gsi_extension_callback, [3t](#),

globus_gsi_callback_activation

 GLOBUS_GSI_CALLBACK_MODULE, [2](#)

globus_gsi_callback_check_issued

 globus_gsi_callback, [4](#)

globus_gsi_callback_constants

 GLOBUS_GSI_CALLBACK_ERROR_-
 CALLBACK_DATA, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 CANT_GET_LOCAL_CA_CERT, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_CERT_-
 CHAIN, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_CERT_-
 HAS_EXPIRED, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_CERT_-
 NOT_YET_VALID, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 ERRNO, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 INVALID_CRL, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 INVALID_PROXY, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_LAST,
 [14](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 LIMITED_PROXY, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 MIXING_DIFFERENT_PROXY_TYPES,
 [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_OLD_-
 GAA, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 PROXY_PATH_LENGTH_EXCEEDED,
 [14](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 REVOKED_CERT, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 SUCCESS, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_-
 VERIFY_CRED, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_WITH_-
 CALLBACK_DATA_INDEX, [13](#)

 GLOBUS_GSI_CALLBACK_ERROR_WITH_-
 SIGNING_POLICY, [13](#)

globus_gsi_callback_constants

 globus_gsi_callback_error, [13](#)

globus_gsi_callback_create_proxy_callback

 globus_gsi_callback, [4](#)

globus_gsi_callback_data

 globus_gsi_callback_data_copy, [7](#),

 globus_gsi_callback_data_destroy, [7](#),

 globus_gsi_callback_data_init, [7](#),

 globus_gsi_callback_data, [6](#),

 globus_gsi_callback_get_cert_chain, [9n](#),

 globus_gsi_callback_get_cert_depth, [9](#),

 globus_gsi_callback_get_cert_data, [10](#)

 globus_gsi_callback_get_cert_type, [9e](#),

 globus_gsi_callback_get_check_policy_for_-
 self_signed_certs, [2](#)

 globus_gsi_callback_get_error, [1](#)

 globus_gsi_callback_get_extension, [1b](#),

 globus_gsi_callback_get_extension_oid, [1d](#),

 globus_gsi_callback_get_multiple_limited_-
 proxy_ok, [9](#)

 globus_gsi_callback_get_proxy_depth, [8h](#),

 globus_gsi_callback_set_cert_chain, [9n](#),

 globus_gsi_callback_set_cert_depth, [9](#),

 globus_gsi_callback_set_cert_data, [10](#)

 globus_gsi_callback_set_cert_type, [9e](#),

 globus_gsi_callback_set_check_policy_for_-
 self_signed_certs, [2](#)

 globus_gsi_callback_set_error, [1](#)

 globus_gsi_callback_set_extension, [1b](#),

 globus_gsi_callback_set_extension_oid, [1d](#),

 globus_gsi_callback_set_multiple_limited_-
 proxy_ok, [9](#)

 globus_gsi_callback_set_proxy_depth, [8h](#),

globus_gsi_callback_data_copy

 globus_gsi_callback_data, [7](#),

globus_gsi_callback_data_destroy

 globus_gsi_callback_data, [7](#),

globus_gsi_callback_data_init

 globus_gsi_callback_data, [7](#),

globus_gsi_callback_data_t

 globus_gsi_callback_data, [6](#),

GLOBUS_GSI_CALLBACK_ERROR_-

CALLBACK_DATA
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_CANT_
 GET_LOCAL_CA_CERT
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_CERT_
 CHAIN
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_CERT_
 HAS_EXPIRED
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_CERT_
 NOT_YET_VALID
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_ERRNO
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_INVALID_
 CRL
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_INVALID_
 PROXY
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_LAST
 globus_gsi_callback_constan[14](#),
 GLOBUS_GSI_CALLBACK_ERROR_LIMITED_
 PROXY
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_MIXING_
 DIFFERENT_PROXY_TYPES
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_OLD_GAA
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_PROXY_
 PATH_LENGTH_EXCEEDED
 globus_gsi_callback_constan[14](#),
 GLOBUS_GSI_CALLBACK_ERROR_
 REVOKED_CERT
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_SUCCESS
 globus_gsi_callback_constan[13](#),
 globus_gsi_callback_error_t
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_VERIFY_
 CRED
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_WITH_
 CALLBACK_DATA_INDEX
 globus_gsi_callback_constan[13](#),
 GLOBUS_GSI_CALLBACK_ERROR_WITH_
 SIGNING_POLICY
 globus_gsi_callback_constan[13](#),
 globus_gsi_callback_get_cert_chain
 globus_gsi_callback_data,

globus_gsi_callback_get_cert_depth
 globus_gsi_callback_data,

globus_gsi_callback_get_cert_dir
 globus_gsi_callback_data,

globus_gsi_callback_get_cert_type
 globus_gsi_callback_data,

globus_gsi_callback_get_check_policy_for_self_
 signed_certs
 globus_gsi_callback_data,

globus_gsi_callback_get_error
 globus_gsi_callback_data,

globus_gsi_callback_get_extension_cb
 globus_gsi_callback_data,

globus_gsi_callback_get_extension_oids
 globus_gsi_callback_data,

globus_gsi_callback_get_multiple_limited_proxy_ok
 globus_gsi_callback_data,

globus_gsi_callback_get_proxy_depth
 globus_gsi_callback_data,

globus_gsi_callback_get_SSL_callback_data_index
 globus_gsi_callback,

globus_gsi_callback_get_X509_STORE_callback_
 data_index
 globus_gsi_callback,

globus_gsi_callback_handshake_callback
 globus_gsi_callback,

GLOBUS_GSI_CALLBACK_MODULE
 globus_gsi_callback_activation,

globus_gsi_callback_set_cert_chain
 globus_gsi_callback_data,

globus_gsi_callback_set_cert_depth
 globus_gsi_callback_data,

globus_gsi_callback_set_cert_dir
 globus_gsi_callback_data,

globus_gsi_callback_set_cert_type
 globus_gsi_callback_data,

globus_gsi_callback_set_check_policy_for_self_
 signed_certs
 globus_gsi_callback_data,

globus_gsi_callback_set_error
 globus_gsi_callback_data,

globus_gsi_callback_set_extension_cb
 globus_gsi_callback_data,

globus_gsi_callback_set_extension_oids
 globus_gsi_callback_data,

globus_gsi_callback_set_multiple_limited_proxy_ok
 globus_gsi_callback_data,

globus_gsi_callback_set_proxy_depth
 globus_gsi_callback_data,

globus_gsi_callback_X509_verify_cert
 globus_gsi_callback,

globus_gsi_extension_callback_t
 globus_gsi_callback,

GSI Callback Constants [42](#)