

# globus gsi callback Reference Manual

## 1.10

Generated by Doxygen 1.5.3

Fri Jun 26 20:23:19 2009

# Contents

<a href="#">1 Globus GSI Callback</a>	<a href="#">1</a>
<a href="#">2 globus gsi callback Module Index</a>	<a href="#">1</a>
<a href="#">3 globus gsi callback Page Index</a>	<a href="#">1</a>
<a href="#">4 globus gsi callback Module Documentation</a>	<a href="#">1</a>
<a href="#">5 globus gsi callback Page Documentation</a>	<a href="#">14</a>

## 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:

<b>Activation</b>	<a href="#">1</a>
<b>Callback Functions</b>	<a href="#">2</a>
<b>Callback Data Functions</b>	<a href="#">5</a>
<b>GSI Callback Constants</b>	<a href="#">13</a>

## 3 globus gsi callback Page Index

### 3.1 globus gsi callback Related Pages

Here is a list of all related documentation pages:

<b>Deprecated List</b>	<a href="#">14</a>
------------------------	--------------------

## 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](#) (`int *index`)

### Get callback data index from SSL structure

- `globus_result_t` [globus\\_gsi\\_callback\\_get\\_SSL\\_callback\\_data\\_index](#) (`int *index`)

### Certificate verify wrapper

- `int` [globus\\_gsi\\_callback\\_X509\\_verify\\_cert](#) (`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_t)(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_t)(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\\_cb](#) (globus\_gsi\_callback\_data\_t callback\_data, [globus\\_gsi\\_extension\\_callback\\_t](#) \*extension\_cb)
- globus\_result\_t [globus\\_gsi\\_callback\\_set\\_extension\\_cb](#) (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 struct globus\_l\_gsi\_callback\_data\_s \* [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\_s\* 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\_certs (globus\_gsi\_callback\_data\_t *callback\_data*, globus\_bool\_t \* *check*)**

This function gets the value of the "check policy for self-signed certificates flag" 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 flag 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\_certs (globus\_gsi\_callback\_data\_t *callback\_data*, globus\_bool\_t *check*)**

This function sets the value of the "check policy for self-signed certificates flag" 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 flag

**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.

**Enumerator:**

***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 cert 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*** OldGAA 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\\_ok\*\*\*](#) This function always returns true now. It will be removed in the next release.

Global [\*\*\*globus\\_gsi\\_callback\\_set\\_multiple\\_limited\\_proxy\\_ok\*\*\*](#) This function has been turned into a no-op. It will be removed in the next release.

# Index

Activation, [1](#)

Callback Data Functions, [4](#)

Callback Functions, [2](#)

globus\_gsi\_callback

    globus\_gsi\_callback\_check\_issued, [4](#)

    globus\_gsi\_callback\_create\_proxy\_callback, [3](#)

    globus\_gsi\_callback\_get\_SSL\_callback\_data\_index, [3](#)

    globus\_gsi\_callback\_get\_X509\_STORE\_callback\_data\_index, [3](#)

    globus\_gsi\_callback\_handshake\_callback, [4](#)

    globus\_gsi\_callback\_X509\_verify\_cert, [3](#)

    globus\_gsi\_extension\_callback\_t, [3](#)

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, [14](#)

    GLOBUS\_GSI\_CALLBACK\_ERROR\_CERT\_HAS\_EXPIRED, [13](#)

    GLOBUS\_GSI\_CALLBACK\_ERROR\_CERT\_NOT\_YET\_VALID, [13](#)

    GLOBUS\_GSI\_CALLBACK\_ERROR\_ERRNO, [14](#)

    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, [14](#)

    GLOBUS\_GSI\_CALLBACK\_ERROR\_WITH\_SIGNING\_POLICY, [13](#)

globus\_gsi\_callback\_constants

    globus\_gsi\_callback\_error\_t, [13](#)

globus\_gsi\_callback\_create\_proxy\_callback

    globus\_gsi\_callback, [3](#)

globus\_gsi\_callback\_data

    globus\_gsi\_callback\_data\_copy, [7](#)

    globus\_gsi\_callback\_data\_destroy, [6](#)

    globus\_gsi\_callback\_data\_init, [6](#)

    globus\_gsi\_callback\_data\_t, [6](#)

    globus\_gsi\_callback\_get\_cert\_chain, [8](#)

    globus\_gsi\_callback\_get\_cert\_depth, [7](#)

    globus\_gsi\_callback\_get\_cert\_dir, [10](#)

    globus\_gsi\_callback\_get\_cert\_type, [8](#)

    globus\_gsi\_callback\_get\_check\_policy\_for\_self\_signed\_certs, [12](#)

    globus\_gsi\_callback\_get\_error, [11](#)

    globus\_gsi\_callback\_get\_extension\_cb, [11](#)

    globus\_gsi\_callback\_get\_extension\_oids, [10](#)

    globus\_gsi\_callback\_get\_multiple\_limited\_proxy\_ok, [9](#)

    globus\_gsi\_callback\_get\_proxy\_depth, [7](#)

    globus\_gsi\_callback\_set\_cert\_chain, [9](#)

    globus\_gsi\_callback\_set\_cert\_depth, [7](#)

    globus\_gsi\_callback\_set\_cert\_dir, [10](#)

    globus\_gsi\_callback\_set\_cert\_type, [8](#)

    globus\_gsi\_callback\_set\_check\_policy\_for\_self\_signed\_certs, [12](#)

    globus\_gsi\_callback\_set\_error, [11](#)

    globus\_gsi\_callback\_set\_extension\_cb, [11](#)

    globus\_gsi\_callback\_set\_extension\_oids, [10](#)

    globus\_gsi\_callback\_set\_multiple\_limited\_proxy\_ok, [9](#)

    globus\_gsi\_callback\_set\_proxy\_depth, [8](#)

globus\_gsi\_callback\_data\_copy

    globus\_gsi\_callback\_data, [7](#)

globus\_gsi\_callback\_data\_destroy

    globus\_gsi\_callback\_data, [6](#)

globus\_gsi\_callback\_data\_init

    globus\_gsi\_callback\_data, [6](#)

globus\_gsi\_callback\_data\_t

    globus\_gsi\_callback\_data, [6](#)

GLOBUS\_GSI\_CALLBACK\_ERROR\_CALLBACK\_DATA

    globus\_gsi\_callback\_constants, [13](#)

GLOBUS\_GSI\_CALLBACK\_ERROR\_CANT\_GET\_LOCAL\_CA\_CERT

    globus\_gsi\_callback\_constants, [13](#)



GLOBUS\_GSI\_CALLBACK\_ERROR\_CERT\_-CHAIN  
     globus\_gsi\_callback\_constants, 14  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_CERT\_HAS\_-EXPIRED  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_CERT\_NOT\_-YET\_VALID  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_ERRNO  
     globus\_gsi\_callback\_constants, 14  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_INVALID\_-CRL  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_INVALID\_-PROXY  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_LAST  
     globus\_gsi\_callback\_constants, 14  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_LIMITED\_-PROXY  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_MIXING\_-DIFFERENT\_PROXY\_TYPES  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_OLD\_GAA  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_PROXY\_-PATH\_LENGTH\_EXCEEDED  
     globus\_gsi\_callback\_constants, 14  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_REVOKED\_-CERT  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_SUCCESS  
     globus\_gsi\_callback\_constants, 13  
 globus\_gsi\_callback\_error\_t  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_VERIFY\_-CRED  
     globus\_gsi\_callback\_constants, 13  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_WITH\_-CALLBACK\_DATA\_INDEX  
     globus\_gsi\_callback\_constants, 14  
 GLOBUS\_GSI\_CALLBACK\_ERROR\_WITH\_-SIGNING\_POLICY  
     globus\_gsi\_callback\_constants, 13  
 globus\_gsi\_callback\_get\_cert\_chain  
     globus\_gsi\_callback\_data, 8  
 globus\_gsi\_callback\_get\_cert\_depth  
     globus\_gsi\_callback\_data, 7  
 globus\_gsi\_callback\_get\_cert\_dir  
     globus\_gsi\_callback\_data, 10  
 globus\_gsi\_callback\_get\_cert\_type  
     globus\_gsi\_callback\_data, 8  
 globus\_gsi\_callback\_get\_check\_policy\_for\_self\_-signed\_certs  
     globus\_gsi\_callback\_data, 12  
 globus\_gsi\_callback\_get\_error  
     globus\_gsi\_callback\_data, 11  
 globus\_gsi\_callback\_get\_extension\_cb  
     globus\_gsi\_callback\_data, 11  
 globus\_gsi\_callback\_get\_extension\_oids  
     globus\_gsi\_callback\_data, 10  
 globus\_gsi\_callback\_get\_multiple\_limited\_proxy\_ok  
     globus\_gsi\_callback\_data, 9  
 globus\_gsi\_callback\_get\_proxy\_depth  
     globus\_gsi\_callback\_data, 7  
 globus\_gsi\_callback\_get\_SSL\_callback\_data\_index  
     globus\_gsi\_callback, 3  
 globus\_gsi\_callback\_get\_X509\_STORE\_callback\_-data\_index  
     globus\_gsi\_callback, 3  
 globus\_gsi\_callback\_handshake\_callback  
     globus\_gsi\_callback, 4  
 GLOBUS\_GSI\_CALLBACK\_MODULE  
     globus\_gsi\_callback\_activation, 2  
 globus\_gsi\_callback\_set\_cert\_chain  
     globus\_gsi\_callback\_data, 9  
 globus\_gsi\_callback\_set\_cert\_depth  
     globus\_gsi\_callback\_data, 7  
 globus\_gsi\_callback\_set\_cert\_dir  
     globus\_gsi\_callback\_data, 10  
 globus\_gsi\_callback\_set\_cert\_type  
     globus\_gsi\_callback\_data, 8  
 globus\_gsi\_callback\_set\_check\_policy\_for\_self\_-signed\_certs  
     globus\_gsi\_callback\_data, 12  
 globus\_gsi\_callback\_set\_error  
     globus\_gsi\_callback\_data, 11  
 globus\_gsi\_callback\_set\_extension\_cb  
     globus\_gsi\_callback\_data, 11  
 globus\_gsi\_callback\_set\_extension\_oids  
     globus\_gsi\_callback\_data, 10  
 globus\_gsi\_callback\_set\_multiple\_limited\_proxy\_ok  
     globus\_gsi\_callback\_data, 9  
 globus\_gsi\_callback\_set\_proxy\_depth  
     globus\_gsi\_callback\_data, 8  
 globus\_gsi\_callback\_X509\_verify\_cert  
     globus\_gsi\_callback, 3  
 globus\_gsi\_extension\_callback\_t  
     globus\_gsi\_callback, 3  
 GSI Callback Constants, 13