

globus gsi callback Reference Manual

1.10

Generated by Doxygen 1.2.18

Fri Jun 26 15:29:21 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	13

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	13
---------------------------------	--------------------

4 globus gsi callback Module Documentation

4.1 Activation

Globus GSI Callback uses standard Globus module activation and deactivation.

De nes

```
#de ne 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 De ne Documentation

4.1.2.1 #de ne GLOBUSGSI.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_createproxy_callback(int preverify_ok, X509_STORECTX x509_context)
```

SSL path validation callback.

```
int globus_gsi_callback_handshake_callback(int preverify_ok, X509_STORECTX x509_context)
```

OpenSSL X509checkissued() wrapper

```
int globus_gsi_callback_checkissued(X509_STORECTX 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 globusgsi_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_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 globusgsi_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 globusgsi_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_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 globusgsi_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

```
globusresultt globusgsi_callbackdatainit (globusgsi_callbackdatat callbackdata)
globusresultt globusgsi_callbackdatadestroy(globusgsi_callbackdatat callbackdata)
```

Copying a callback data structure

```
globusresultt globusgsi_callbackdatacopy (globusgsi_callbackdatat source, globusgsi_callbackdata-
t dest)
```

Getting and setting the certificate chain depth

```
globusresultt globusgsi_callbackgetcertdepth(globusgsi_callbackdatat callbackdata, int certdepth)
globusresultt globusgsi_callbacksetcertdepth(globusgsi_callbackdatat callbackdata, int certdepth)
```

Getting and setting the "proxy chain" depth

```
globusresultt globusgsi_callbackgetproxy_depth (globusgsi_callbackdatat callbackdata, int proxy-
depth)
globusresultt globusgsi_callbacksetproxy_depth(globusgsi_callbackdatat callbackdata, int proxydepth)
```

Getting and setting the certificate type

```
globusresultt globusgsi_callbackgetcerttype (globusgsi_callbackdatat callbackdata, globusgsi_cert-
utils_cert_type_t certtype)
globusresultt globusgsi_callbacksetcerttype (globusgsi_callbackdatat callbackdata, globusgsi_cert-
utils_cert_type_t certtype)
```

Getting and setting the certificate chain

```
globusresultt globusgsi_callbackgetcertchain (globusgsi_callbackdatat callbackdata, STACK-
OF(X509) certchain)
globusresultt globusgsi_callbacksetcertchain (globusgsi_callbackdatat callbackdata, STACK-
OF(X509) certchain)
```

Getting and setting the limited proxy handling setting

```
globusresultt globusgsi_callbackgetmultiple_limited_proxy_ok (globusgsi_callbackdatat callbackdata,
int multiple_limited_proxy_ok)
globusresultt globusgsi_callbacksetmultiple_limited_proxy_ok (globusgsi_callbackdatat callbackdata,
int multiple_limited_proxy_ok)
```

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

```
globus_result_t globus_gsi_callback_get_extensionoids (globus_gsi_callback_data_t callbackdata, void
extensionoids)
globus_result_t globus_gsi_callback_set_extensionoids (globus_gsi_callback_data_t callbackdata, void
extensionoids)
```

Getting and setting the trusted certificate directory

```
globus_result_t globus_gsi_callback_get_cert_dir (globus_gsi_callback_data_t callbackdata, char cert_dir)
globus_result_t globus_gsi_callback_set_cert_dir (globus_gsi_callback_data_t callbackdata, char cert_dir)
```

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

```
globus_result_t globus_gsi_callback_get_extensioncb (globus_gsi_callback_data_t callbackdata, globus_gsi-
extension_callback_t extensioncb)
globus_result_t globus_gsi_callback_set_extensioncb (globus_gsi_callback_data_t callbackdata, globus_gsi-
extension_callback_t extensioncb)
```

Getting and setting the error status

```
globus_result_t globus_gsi_callback_get_error (globus_gsi_callback_data_t callbackdata, globus_result_t error)
globus_result_t globus_gsi_callback_set_error (globus_gsi_callback_data_t callbackdata, 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 call-
backdata, globus_bool_t check)
globus_result_t globus_gsi_callback_set_check_policy_for_self_signed_certs (globus_gsi_callback_data_t call-
backdata, globus_bool_t check)
```

Typedefs

```
typedef globus_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_gsi_callback_data_s globus_gsi_callback_data_t

Callback data typedef.

4.3.3 Function Documentation

4.3.3.1 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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_extensionoids (globus_gsi_callback_data_t callback_data, void extensionoids)`

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.
`extensionoids` 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_extensionoids (globus_gsi_callback_data_t callback_data, void extensionoids)`

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.
`extensionoids` 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_extensioncb (globus_gsi_callback_data_t callback_data, globus_gsi_extension_callback_t extensioncb)`

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

`extensioncb` 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_extensioncb (globus_gsi_callback_data_t callback_data, globus_gsi_extension_callback_t extensioncb)`

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

`extensioncb` 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" 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_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" 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_callbackerror_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_ERRORCANT_GET_LOCAL_CA_CERT = 3, GLOBUS_GSI_CALLBACK_ERRORCERT_HAS_EXPIRED = 4, GLOBUS_GSI_CALLBACK_ERRORINVALID_PROXY = 5, GLOBUS_GSI_CALLBACK_ERRORLIMITED_PROXY = 6, GLOBUS_GSI_CALLBACK_ERRORINVALID_CRL = 7, GLOBUS_GSI_CALLBACK_ERRORREVOKED_CERT = 8, GLOBUS_GSI_CALLBACK_ERRORMIXING_DIFFERENT_PROXY_TYPES = 9, GLOBUS_GSI_CALLBACK_ERRORWITH_SIGNING_POLICY = 10, GLOBUS_GSI_CALLBACK_ERROROLD_GAA = 11, GLOBUS_GSI_CALLBACK_ERRORCALLBACK_DATA = 12, GLOBUS_GSI_CALLBACK_ERRORERRNO = 13, GLOBUS_GSI_CALLBACK_ERRORCERT_CHAIN = 14, GLOBUS_GSI_CALLBACK_ERRORWITH_CALLBACK_DATA_INDEX = 15, GLOBUS_GSI_CALLBACK_ERRORPROXY_PATH_LENGTH_EXCEEDED = 16, GLOBUS_GSI_CALLBACK_ERRORLAST = 18g

4.4.1 Enumeration Type Documentation

4.4.1.1 enum globusgsi.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 certi cate is not yet valid.
 GLOBUS_GSI_CALLBACK_ERROR_CANT_GET_LOCAL_CA_CERT Unable to discover a local trusted CA for a given cert cate.
 GLOBUS_GSI_CALLBACK_ERROR_CERT_HAS_EXPIRED The certi cate 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 certi cate 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 certi cate 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 speci ed 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 [globusgsi.callback.getmultiple_limited_proxy_ok](#)(globusgsi.callback_data_t callback_data, int 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](#)(globus_gsi_callback_data_t callback_data, int multiple_limited_pro
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](#)

globusgsi.callback

globusgsi.callbackcheckissued, [4](#)

globusgsi.callbackcreateproxy.callback, [4](#)

globusgsi.callbackget.SSL.callbackdata-index, [3](#)

globusgsi.callbackget.X509.STORE-callbackdataindex, [3](#)

globusgsi.callbackhandshake.callback, [4](#)

globusgsi.callbackX509.verify.cert, [3](#)

globusgsi.extensioncallbackt, [3](#)

globusgsi.callbackactivation

GLOBAL_GSI_CALLBACK_MODULE, [2](#)

globusgsi.callbackcheckissued

globusgsi.callback, [4](#)

globusgsi.callbackconstants

GLOBAL_GSI_CALLBACK_ERROR-CALLBACK_DATA, [13](#)

GLOBAL_GSI_CALLBACK_ERRORCANT-GET_LOCAL_CA_CERT, [13](#)

GLOBAL_GSI_CALLBACK_ERRORCERT-CHAIN, [13](#)

GLOBAL_GSI_CALLBACK_ERRORCERT-HAS_EXPIRED, [13](#)

GLOBAL_GSI_CALLBACK_ERRORCERT-NOT_YET_VALID, [13](#)

GLOBAL_GSI_CALLBACK_ERRORERRNO, [13](#)

GLOBAL_GSI_CALLBACK_ERROR-INVALID_CRL, [13](#)

GLOBAL_GSI_CALLBACK_ERROR-INVALID_PROXY, [13](#)

GLOBAL_GSI_CALLBACK_ERRORLAST, [13](#)

GLOBAL_GSI_CALLBACK_ERROR-LIMITED_PROXY, [13](#)

GLOBAL_GSI_CALLBACK_ERROR-MIXING_DIFFERENT_PROXY_TYPES, [13](#)

GLOBAL_GSI_CALLBACK_ERROROLD-GAA, [13](#)

GLOBAL_GSI_CALLBACK_ERRORPROXY-PATH_LENGTH_EXCEEDED, [13](#)

GLOBAL_GSI_CALLBACK_ERROR-REVOKED_CERT, [13](#)

GLOBAL_GSI_CALLBACK_ERROR-SUCCESS, [13](#)

GLOBAL_GSI_CALLBACK_ERROR-VERIFY_CRED, [13](#)

GLOBAL_GSI_CALLBACK_ERRORWITH-CALLBACK_DATA_INDEX, [13](#)

GLOBAL_GSI_CALLBACK_ERRORWITH-SIGNING_POLICY, [13](#)

globusgsi.callbackconstants

globusgsi.callbackerror.t, [13](#)

globusgsi.callbackcreateproxy.callback

globusgsi.callback, [4](#)

globusgsi.callbackdata

globusgsi.callbackdatacopy, [7](#)

globusgsi.callbackdatadestroy, [7](#)

globusgsi.callbackdatainit, [7](#)

globusgsi.callbackdatat, [6](#)

globusgsi.callbackget.cert.chain, [9](#)

globusgsi.callbackget.cert.depth, [7](#)

globusgsi.callbackget.cert.dir, [10](#)

globusgsi.callbackget.cert.type, [8](#)

globusgsi.callbackget.checkpolicy_for_self-signedcerts, [12](#)

globusgsi.callbackget.error, [11](#)

globusgsi.callbackget.extensioncb, [11](#)

globusgsi.callbackget.extensionoids, [10](#)

globusgsi.callbackget.multiple.limited-proxy_ok, [9](#)

globusgsi.callbackget.proxy_depth, [8](#)

globusgsi.callbackset.cert.chain, [9](#)

globusgsi.callbackset.cert.depth, [7](#)

globusgsi.callbackset.cert.dir, [10](#)

globusgsi.callbackset.cert.type, [8](#)

globusgsi.callbackset.checkpolicy_for_self-signedcerts, [12](#)

globusgsi.callbackset.error, [11](#)

globusgsi.callbackset.extensioncb, [11](#)

globusgsi.callbackset.extensionoids, [10](#)

globusgsi.callbackset.multiple.limited.proxy-ok, [9](#)

globusgsi.callbackset.proxy_depth, [8](#)

globusgsi.callbackdatacopy

globusgsi.callbackdata, [7](#)

globusgsi.callbackdatadestroy

globusgsi.callbackdata, [7](#)

globusgsi.callbackdatainit

globusgsi.callbackdata, [7](#)

globusgsi.callbackdatat

globusgsi.callbackdata, [6](#)

GLOBAL_GSI_CALLBACK_ERRORCALLBACK-DATA

globusgsi.callbackconstants, [13](#)

GLOBUS.GSI_CALLBACK_ERRORCANT_GET_-
 LOCAL_CA_CERT
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORCERT_CHAIN
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORCERT_HAS_-
 EXPIRED
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORCERT_NOT_-
 YET_VALID
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORERRNO
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORINVALID_-
 CRL
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORINVALID_-
 PROXY
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORLAST
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORLIMITED_-
 PROXY
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORMIXING_-
 DIFFERENT_PROXY_TYPES
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERROROLD_GAA
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORPROXY_-
 PATH_LENGTH_EXCEEDED
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORREVOKED_-
 CERT
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORSUCCESS
 [globusgsi.callbackconstants](#)[13](#)
[globusgsi.callbackerror.t](#)
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORVERIFY_-
 CRED
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORWITH_-
 CALLBACK_DATA_INDEX
 [globusgsi.callbackconstants](#)[13](#)
 GLOBUS.GSI_CALLBACK_ERRORWITH_-
 SIGNING_POLICY
 [globusgsi.callbackconstants](#)[13](#)
[globusgsi.callbackget.cert.chain](#)
 [globusgsi.callbackdata](#)[9](#)
[globusgsi.callbackget.cert.depth](#)
 [globusgsi.callbackdata](#)[7](#)
[globusgsi.callbackget.cert.dir](#)
 [globusgsi.callbackdata](#)[10](#)
[globusgsi.callbackget.cert.type](#)
 [globusgsi.callbackdata](#)[8](#)
[globusgsi.callbackget.checkpolicy_for_self_-
 signedcerts](#)
 [globusgsi.callbackdata](#)[12](#)
[globusgsi.callbackget.error](#)
 [globusgsi.callbackdata](#)[11](#)
[globusgsi.callbackget.extensioncb](#)
 [globusgsi.callbackdata](#)[11](#)
[globusgsi.callbackget.extensionoids](#)
 [globusgsi.callbackdata](#)[10](#)
[globusgsi.callbackget.multiple_limited_proxy_ok](#)
 [globusgsi.callbackdata](#)[9](#)
[globusgsi.callbackget.proxy.depth](#)
 [globusgsi.callbackdata](#)[8](#)
[globusgsi.callbackget.SSL.callbackdataindex](#)
 [globusgsi.callback](#)[3](#)
[globusgsi.callbackget.X509_STORE.callback-
 dataindex](#)
 [globusgsi.callback](#)[3](#)
[globusgsi.callbackhandshake.callback](#)
 [globusgsi.callback](#)[4](#)
 GLOBUS.GSI_CALLBACK_MODULE
 [globusgsi.callbackactivation](#)[2](#)
[globusgsi.callbackset.cert.chain](#)
 [globusgsi.callbackdata](#)[9](#)
[globusgsi.callbackset.cert.depth](#)
 [globusgsi.callbackdata](#)[7](#)
[globusgsi.callbackset.cert.dir](#)
 [globusgsi.callbackdata](#)[10](#)
[globusgsi.callbackset.cert.type](#)
 [globusgsi.callbackdata](#)[8](#)
[globusgsi.callbackset.checkpolicy_for_self_-
 signedcerts](#)
 [globusgsi.callbackdata](#)[12](#)
[globusgsi.callbackset.error](#)
 [globusgsi.callbackdata](#)[11](#)
[globusgsi.callbackset.extensioncb](#)
 [globusgsi.callbackdata](#)[11](#)
[globusgsi.callbackset.extensionoids](#)
 [globusgsi.callbackdata](#)[10](#)
[globusgsi.callbackset.multiple_limited_proxy_ok](#)
 [globusgsi.callbackdata](#)[9](#)
[globusgsi.callbackset.proxy.depth](#)
 [globusgsi.callbackdata](#)[8](#)
[globusgsi.callbackX509.verify_cert](#)
 [globusgsi.callback](#)[3](#)
[globusgsi.extensioncallbackt](#)
 [globusgsi.callback](#)[3](#)
 GSI Callback Constants[13](#)[2](#)