

globus xio gsi driver  
0.6

Generated by Doxygen 1.7.4

Sat Apr 30 2011 00:06:33

# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Module Documentation</b>	<b>2</b>
2.1	Globus XIO GSI Driver . . . . .	2
2.1.1	Detailed Description . . . . .	2
2.2	Opening/Closing . . . . .	3
2.3	Reading/Writing . . . . .	3
2.4	Server . . . . .	3
2.5	Env Variables . . . . .	4
2.6	Attributes and Cntls . . . . .	4
2.6.1	Detailed Description . . . . .	7
2.6.2	Enumeration Type Documentation . . . . .	7
2.6.3	Function Documentation . . . . .	8
2.7	Types . . . . .	15
2.7.1	Typedef Documentation . . . . .	15
2.7.2	Enumeration Type Documentation . . . . .	16
2.8	Error Types . . . . .	17
2.8.1	Detailed Description . . . . .	17
2.8.2	Enumeration Type Documentation . . . . .	17

## 1 Module Index

### 1.1 Modules

Here is a list of all modules:

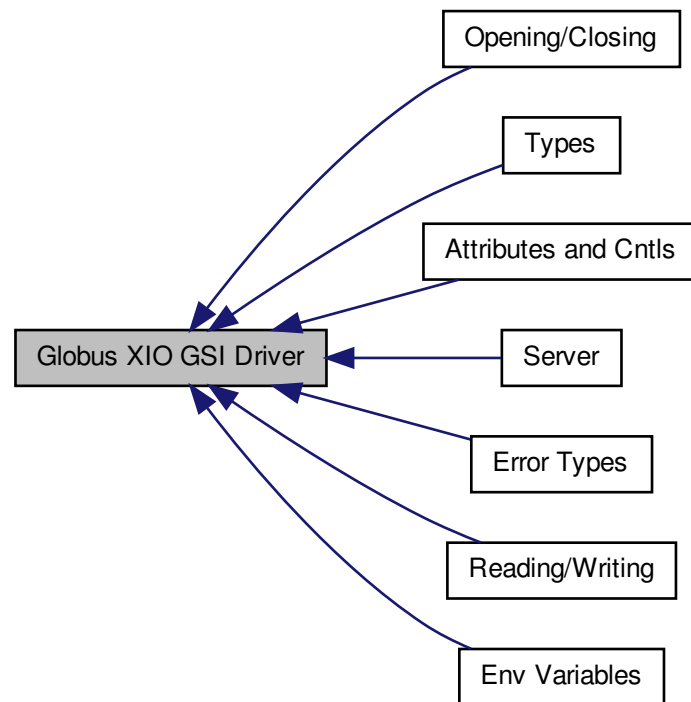
<b>Globus XIO GSI Driver</b>	<b>2</b>
<b>Opening/Closing</b>	<b>3</b>
<b>Reading/Writing</b>	<b>3</b>
<b>Server</b>	<b>3</b>
<b>Env Variables</b>	<b>4</b>
<b>Attributes and Cntls</b>	<b>4</b>
<b>Types</b>	<b>15</b>
<b>Error Types</b>	<b>17</b>

## 2 Module Documentation

### 2.1 Globus XIO GSI Driver

The GSI driver.

Collaboration diagram for Globus XIO GSI Driver:



#### Modules

- [Opening/Closing](#)
- [Reading/Writing](#)
- [Server](#)
- [Env Variables](#)
- [Attributes and Cntls](#)
- [Types](#)
- [Error Types](#)

#### 2.1.1 Detailed Description

The GSI driver.

## 2.2 Opening/Closing

An XIO handle with the gsi driver can be created with either **globus\_xio\_handle\_create()** or **globus\_xio\_server\_register\_accept()**.

Collaboration diagram for Opening/Closing:



An XIO handle with the gsi driver can be created with either **globus\_xio\_handle\_create()** or **globus\_xio\_server\_register\_accept()**. If the handle is created with **globus\_xio\_server\_register\_accept()**, the **globus\_xio\_register\_open()** call will proceed to accept a GSSAPI security context. Upon successful completion of the open, ie after the open callback has been called, the application may proceed to read or write data associated with the GSI session.

If the handle is created with **globus\_xio\_handle\_create()**, then the XIO handle will implement the client-side (init) of the GSSAPI call sequence and establish a security context with the accepting side indicated by the `contact_string` passed to **globus\_xio\_register\_open()**.

## 2.3 Reading/Writing

The GSI driver behaves similar to the underlying transport driver with respect to reads and writes, except for the try-read and try-write operations (ie.

Collaboration diagram for Reading/Writing:

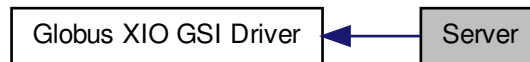


The GSI driver behaves similar to the underlying transport driver with respect to reads and writes, except for the try-read and try-write operations (ie. `waitforbytes == 0`) which always return immediately. This is due to the fact that the security layer needs to read and write tokens of a certain minimal size and thus needs to rely on the underlying transport to handle greater than 0 reads/write which is not possible in "try" mode.

## 2.4 Server

**globus\_xio\_server\_create()** causes a new transport-specific listener socket to be created to handle new GSI connections.

Collaboration diagram for Server:



**globus\_xio\_server\_create()** causes a new transport-specific listener socket to be created to handle new GSI connections. **globus\_xio\_server\_register\_accept()** will accept a new connection for processing. **globus\_xio\_server\_register\_close()** cleans up the internal resources associated with the http server and calls close on the listener.

All accepted handles inherit all gsi specific attributes set in the attr to **globus\_xio\_server\_create()**, but can be overridden with the attr to **globus\_xio\_register\_open()**. Furthermore, accepted handles will use the GSSAPI accept security context call unless explicitly overridden during the **globus\_xio\_register\_open()** call ([GLOBUS\\_XIO\\_GSI\\_FORCE\\_SERVER\\_MODE](#)).

## 2.5 Env Variables

The gsi driver uses the following environment variables.

Collaboration diagram for Env Variables:



The gsi driver uses the following environment variables.

- X509\_USER\_PROXY
- X509\_USER\_CERT
- X509\_USER\_KEY
- X509\_CERT\_DIR

For details see [Globus: GSI Environment Variables](#)

## 2.6 Attributes and Cntls

GSI driver specific attrs and cntls.

Collaboration diagram for Attributes and Cntls:



## Enumerations

- `enum globus_xio_gsi_cmd_t {`
  - `GLOBUS_XIO_GSI_SET_CREDENTIAL,`
  - `GLOBUS_XIO_GSI_GET_CREDENTIAL,`
  - `GLOBUS_XIO_GSI_SET_GSSAPI_REQ_FLAGS,`
  - `GLOBUS_XIO_GSI_GET_GSSAPI_REQ_FLAGS,`
  - `GLOBUS_XIO_GSI_SET_PROXY_MODE,`
  - `GLOBUS_XIO_GSI_GET_PROXY_MODE,`
  - `GLOBUS_XIO_GSI_SET_AUTHORIZATION_MODE,`
  - `GLOBUS_XIO_GSI_GET_AUTHORIZATION_MODE,`
  - `GLOBUS_XIO_GSI_SET_DELEGATION_MODE,`
  - `GLOBUS_XIO_GSI_GET_DELEGATION_MODE,`
  - `GLOBUS_XIO_GSI_SET_SSL_COMPATIBLE,`
  - `GLOBUS_XIO_GSI_SET_ANON,`
  - `GLOBUS_XIO_GSI_SET_WRAP_MODE,`
  - `GLOBUS_XIO_GSI_GET_WRAP_MODE,`
  - `GLOBUS_XIO_GSI_SET_BUFFER_SIZE,`
  - `GLOBUS_XIO_GSI_GET_BUFFER_SIZE,`
  - `GLOBUS_XIO_GSI_SET_PROTECTION_LEVEL,`
  - `GLOBUS_XIO_GSI_GET_PROTECTION_LEVEL,`
  - `GLOBUS_XIO_GSI_GET_TARGET_NAME,`
  - `GLOBUS_XIO_GSI_SET_TARGET_NAME,`
  - `GLOBUS_XIO_GSI_GET_CONTEXT,`
  - `GLOBUS_XIO_GSI_GET_DELEGATED_CRED,`
  - `GLOBUS_XIO_GSI_GET_PEER_NAME,`
  - `GLOBUS_XIO_GSI_GET_LOCAL_NAME,`
  - `GLOBUS_XIO_GSI_INIT_DELEGATION,`
  - `GLOBUS_XIO_GSI_REGISTER_INIT_DELEGATION,`
  - `GLOBUS_XIO_GSI_ACCEPT_DELEGATION,`
  - `GLOBUS_XIO_GSI_REGISTER_ACCEPT_DELEGATION,`
  - `GLOBUS_XIO_GSI_FORCE_SERVER_MODE }`

## Functions

- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_CREDENTIAL, gss\_cred\_id\_t credential)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_SET\_CREDENTIAL, gss\_cred\_id\_t credential)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_CREDENTIAL, gss\_cred\_id\_t \*credential)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_GET\_CREDENTIAL, gss\_cred\_id\_t \*credential)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_GSSAPI\_REQ\_FLAGS, OM\_uint32 req\_flags)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_GSSAPI\_REQ\_FLAGS, OM\_uint32 \*req\_flags)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_PROXY\_MODE, [globus\\_xio\\_gsi\\_proxy\\_mode\\_t](#) proxy\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_PROXY\_MODE, [globus\\_xio\\_gsi\\_proxy\\_mode\\_t](#) \*proxy\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_AUTHORIZATION\_MODE, [globus\\_xio\\_gsi\\_authorization\\_mode\\_t](#) authz\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_AUTHORIZATION\_MODE, [globus\\_xio\\_gsi\\_authorization\\_mode\\_t](#) \*authz\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_DELEGATION\_MODE, [globus\\_xio\\_gsi\\_delegation\\_mode\\_t](#) delegation\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_DELEGATION\_MODE, [globus\\_xio\\_gsi\\_delegation\\_mode\\_t](#) \*delegation\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_SSL\_COMPATIBLE, globus\_bool\_t ssl\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_ANON, globus\_bool\_t anon\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_WRAP\_MODE, globus\_boolean\_t wrap\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_WRAP\_MODE, globus\_boolean\_t \*wrap\_mode)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_BUFFER\_SIZE, globus\_size\_t buffer\_size)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_BUFFER\_SIZE, globus\_size\_t \*buffer\_size)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_PROTECTION\_LEVEL, [globus\\_xio\\_gsi\\_protection\\_level\\_t](#) protection\_level)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_PROTECTION\_LEVEL, [globus\\_xio\\_gsi\\_protection\\_level\\_t](#) \*protection\_level)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_GET\_TARGET\_NAME, gss\_name\_t \*target\_name)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_SET\_TARGET\_NAME, gss\_name\_t target\_name)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_GET\_CONTEXT, gss\_ctx\_id\_t \*context)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_GET\_DELEGATED\_CRED, gss\_cred\_id\_t \*credential)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_GET\_PEER\_NAME, gss\_name\_t \*peer\_name)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_GET\_LOCAL\_NAME, gss\_name\_t \*local\_name)

- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_INIT\_DELEGATION, gss\_cred\_id\_t credential, gss\_OID\_set restriction\_oids, gss\_buffer\_set\_t restriction\_buffers, OM\_uint32 time\_req)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_REGISTER\_INIT\_DELEGATION, gss\_cred\_id\_t credential, gss\_OID\_set restriction\_oids, gss\_buffer\_set\_t restriction\_buffers, OM\_uint32 time\_req, [globus\\_xio\\_gsi\\_delegation\\_init\\_callback\\_t](#) callback, void \*callback\_arg)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_ACCEPT\_DELEGATION, gss\_cred\_id\_t \*credential, gss\_OID\_set restriction\_oids, gss\_buffer\_set\_t restriction\_buffers, OM\_uint32 time\_req)
- globus\_result\_t [globus\\_xio\\_handle\\_cntl](#) (handle, driver, GLOBUS\_XIO\_GSI\_REGISTER\_ACCEPT\_DELEGATION, gss\_OID\_set restriction\_oids, gss\_buffer\_set\_t restriction\_buffers, OM\_uint32 time\_req, [globus\\_xio\\_gsi\\_delegation\\_accept\\_callback\\_t](#) callback, void \*callback\_arg)
- globus\_result\_t [globus\\_xio\\_attr\\_cntl](#) (attr, driver, GLOBUS\_XIO\_GSI\_FORCE\_SERVER\_MODE, globus\_bool\_t server\_mode)

## 2.6.1 Detailed Description

GSI driver specific attrs and cntls.

See also

[globus\\_xio\\_attr\\_cntl\(\)](#)  
[globus\\_xio\\_handle\\_cntl\(\)](#)

## 2.6.2 Enumeration Type Documentation

### 2.6.2.1 enum globus\_xio\_gsi\_cmd\_t

doxygen varargs filter stuff

GSI driver specific cntls

Enumerator:

**GLOBUS\_XIO\_GSI\_SET\_CREDENTIAL** See usage for: [globus\\_xio\\_attr\\_cntl](#) , [globus\\_xio\\_handle\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_GET\_CREDENTIAL** See usage for: [globus\\_xio\\_attr\\_cntl](#) , [globus\\_xio\\_handle\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_GSSAPI\_REQ\_FLAGS** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_GET\_GSSAPI\_REQ\_FLAGS** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_PROXY\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_GET\_PROXY\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_AUTHORIZATION\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_GET\_AUTHORIZATION\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_DELEGATION\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_GET\_DELEGATION\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_SSL\_COMPATIBLE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_ANON** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_WRAP\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_GET\_WRAP\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_BUFFER\_SIZE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_GET\_BUFFER\_SIZE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBUS\_XIO\_GSI\_SET\_PROTECTION\_LEVEL** See usage for: [globus\\_xio\\_attr\\_cntl](#) .



**GLOBAL\_XIO\_GSI\_GET\_PROTECTION\_LEVEL** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_GET\_TARGET\_NAME** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_SET\_TARGET\_NAME** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_GET\_CONTEXT** See usage for: [globus\\_xio\\_handle\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_GET\_DELEGATED\_CRED** See usage for: [globus\\_xio\\_handle\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_GET\_PEER\_NAME** See usage for: [globus\\_xio\\_handle\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_GET\_LOCAL\_NAME** See usage for: [globus\\_xio\\_handle\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_INIT\_DELEGATION** See usage for: [globus\\_xio\\_handle\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_REGISTER\_INIT\_DELEGATION** See usage for: [globus\\_xio\\_handle\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_ACCEPT\_DELEGATION** See usage for: [globus\\_xio\\_handle\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_REGISTER\_ACCEPT\_DELEGATION** See usage for: [globus\\_xio\\_handle\\_cntl](#) .

**GLOBAL\_XIO\_GSI\_FORCE\_SERVER\_MODE** See usage for: [globus\\_xio\\_attr\\_cntl](#) .

## 2.6.3 Function Documentation

### 2.6.3.1 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBAL_XIO_GSI_SET_CREDENTIAL , gss_cred_id_t credential )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the credential to be used.

#### Parameters

<i>credential</i>	The credential to set. The credential structure needs to remain valid for the lifetime of any xio datastructure it is used by.
-------------------	--

#### Note

If this is called with the `handle_cntl`, there must be no outstanding operations on the handle.

### 2.6.3.2 `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBAL_XIO_GSI_SET_CREDENTIAL , gss_cred_id_t credential )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the credential to be used.

#### Parameters

<i>credential</i>	The credential to set. The credential structure needs to remain valid for the lifetime of any xio datastructure it is used by.
-------------------	--

#### Note

If this is called with the `handle_cntl`, there must be no outstanding operations on the handle.

### 2.6.3.3 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBAL_XIO_GSI_GET_CREDENTIAL , gss_cred_id_t * credential )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the credential to be used.

#### Parameters

<i>credential</i>	The credential that is currently set. This will only return a credential if a credential was explicitly set prior to this call. It will not return any credential automatically acquired during context initialization.
-------------------	---

2.6.3.4 `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI.GET_CREDENTIAL , gss_cred_id_t * credential )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the credential to be used.

#### Parameters

<i>credential</i>	The credential that is currently set. This will only return a credential if a credential was explicitly set prior to this call. It will not return any credential automatically acquired during context initialization.
-------------------	---

2.6.3.5 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI.SET_GSSAPI_REQ_FLAGS , OM_uint32 req_flags )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the GSSAPI req\_flags to be used.

#### Parameters

<i>req_flags</i>	The req_flags to set
------------------	----------------------

2.6.3.6 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI.GET_GSSAPI_REQ_FLAGS , OM_uint32 * req_flags )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the GSSAPI req\_flags to be used.

#### Parameters

<i>req_flags</i>	The req flags currently in effect
------------------	-----------------------------------

2.6.3.7 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI.SET_PROXY_MODE , globus_xio_gsi_proxy_mode_t proxy_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the proxy mode.

#### Parameters

<i>proxy_mode</i>	The proxy mode to set
-------------------	-----------------------

#### Note

Changing the proxy mode changes the req\_flags

2.6.3.8 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI.GET_PROXY_MODE , globus_xio_gsi_proxy_mode_t * proxy_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the proxy mode.

#### Parameters

<i>proxy_mode</i>	The proxy mode that is currently in effect
-------------------	--

## Note

Changing the proxy mode changes the req\_flags

2.6.3.9 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_SET_AUTHORIZATION_MODE , globus_xio_gsi_authorization_mode_t authz_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the authorization mode.

## Parameters

<i>authz_mode</i>	The authorization mode to set
-------------------	-------------------------------

2.6.3.10 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_GET_AUTHORIZATION_MODE , globus_xio_gsi_authorization_mode_t * authz_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the authorization mode.

## Parameters

<i>authz_mode</i>	The authorization mode that is currently in effect
-------------------	--

2.6.3.11 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_SET_DELEGATION_MODE , globus_xio_gsi_delegation_mode_t delegation_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the delegation mode.

## Parameters

<i>delegation_mode</i>	The delegation mode to use
------------------------	----------------------------

## Note

Changing the delegation mode changes the req\_flags

2.6.3.12 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_GET_DELEGATION_MODE , globus_xio_gsi_delegation_mode_t * delegation_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the delegation mode.

## Parameters

<i>delegation_mode</i>	The delegation mode currently in effect
------------------------	---

2.6.3.13 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_SET_SSL_COMPATIBLE , globus_bool_t ssl_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Make the on the wire protocol SSL compatible.

This implies no wrapping of security tokens and no delegation

## Parameters

<i>ssl_mode</i>	The ssl compatibility mode to use
-----------------	-----------------------------------

## Note

Changing the ssl compatibility mode changes the req\_flags

**2.6.3.14** `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_SET_ANON , globus_bool_t anon_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Do anonymous authentication.

## Parameters

<i>anon_mode</i>	The ssl compatibility mode to use
------------------	-----------------------------------

## Note

Changing the ssl compatibility mode changes the req\_flags and the wrapping mode

**2.6.3.15** `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_SET_WRAP_MODE , globus_boolean_t wrap_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the wrapping mode

This mode determines whether tokens will be wrapped with a Globus IO style header or not.

## Parameters

<i>wrap_mode</i>	The wrapping mode to use
------------------	--------------------------

**2.6.3.16** `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_GET_WRAP_MODE , globus_boolean_t * wrap_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the wrapping mode

This mode determines whether tokens will be wrapped with a Globus IO style header or not.

## Parameters

<i>wrap_mode</i>	The wrapping mode currently in use.
------------------	-------------------------------------

**2.6.3.17** `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_SET_BUFFER_SIZE , globus_size_t buffer_size )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the read buffer size

The read buffer is used for buffering wrapped data, is initialized with a default size of 128K and scaled dynamically to always be able to fit whole tokens.

## Parameters

<i>buffer_size</i>	The size of the read buffer
--------------------	-----------------------------

2.6.3.18 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_GET_BUFFER_SIZE , globus_size_t * buffer_size )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the read buffer size

The read buffer is used for buffering wrapped data, is initialized with a default size of 128K and scaled dynamically to always be able to fit whole tokens.

#### Parameters

<i>buffer_size</i>	The size of the read buffer
--------------------	-----------------------------

2.6.3.19 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_SET_PROTECTION_LEVEL , globus_xio_gsi_protection_level_t protection_level )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the protection level.

#### Parameters

<i>protection_level</i>	The protection level to set
-------------------------	-----------------------------

#### Note

Changing the proxy mode changes the req\_flags

2.6.3.20 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_GET_PROTECTION_LEVEL , globus_xio_gsi_protection_level_t * protection_level )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the protection level.

#### Parameters

<i>protection_level</i>	The current protection level
-------------------------	------------------------------

2.6.3.21 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_GET_TARGET_NAME , gss_name_t * target_name )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Set the expected peer name.

#### Parameters

<i>target_name</i>	The expected peer name
--------------------	------------------------

2.6.3.22 `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_SET_TARGET_NAME , gss_name_t target_name )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the expected peer name.

#### Parameters

<i>target_name</i>	The expected peer name
--------------------	------------------------

**2.6.3.23** `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI_GET_CONTEXT , gss_ctx_id_t * context )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the GSS context.

#### Parameters

<i>context</i>	The GSS context
----------------	-----------------

**2.6.3.24** `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI_GET_DELEGATED_CRED , gss_cred_id_t * credential )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the delegated credential.

#### Parameters

<i>credential</i>	The delegated credential
-------------------	--------------------------

**2.6.3.25** `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI_GET_PEER_NAME , gss_name_t * peer_name )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the name of the peer.

#### Parameters

<i>peer_name</i>	The GSS name of the peer.
------------------	---------------------------

**2.6.3.26** `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI_GET_LOCAL_NAME , gss_name_t * local_name )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Get the GSS name associated with the local credentials.

#### Parameters

<i>local_name</i>	The GSS name of the local credentials
-------------------	---------------------------------------

**2.6.3.27** `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI_INIT_DELEGATION , gss_cred_id_t credential, gss_OID_set restriction_oids, gss_buffer_set_t restriction_buffers, OM_uint32 time_req )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Initialize delegation-at-any-time process.

#### Parameters

<i>credential</i>	The GSS credential to delegate
<i>restriction_oids</i>	The OIDS for X.509 extensions to embed in the delegated credential
<i>restriction_buffers</i>	The corresponding bodies for the X.509 extensions
<i>time_req</i>	The lifetime of the delegated credential

**2.6.3.28** `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI_REGISTER_INIT_DELEGATION , gss_cred_id_t credential, gss_OID_set restriction_oids, gss_buffer_set_t restriction_buffers, OM_uint32 time_req, globus_xio_gsi_delegation_init_callback_t callback, void * callback_arg )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Initialize non-blocking delegation-at-any-time process.

#### Parameters

<i>credential</i>	The GSS credential to delegate
<i>restriction_oids</i>	The OIDS for X.509 extensions to embed in the delegated credential
<i>restriction_buffers</i>	The corresponding bodies for the X.509 extensions
<i>time_req</i>	The lifetime of the delegated credential
<i>callback</i>	The callback to call when the operation completes
<i>callback_arg</i>	The arguments to pass to the callback

**2.6.3.29** `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI_ACCEPT_DELEGATION , gss_cred_id_t * credential, gss_OID_set restriction_oids, gss_buffer_set_t restriction_buffers, OM_uint32 time_req )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Accept delegation-at-any-time process.

#### Parameters

<i>credential</i>	The delegated GSS credential
<i>restriction_oids</i>	The OIDS for X.509 extensions to embed in the delegated credential
<i>restriction_buffers</i>	The corresponding bodies for the X.509 extensions
<i>time_req</i>	The requested lifetime of the delegated credential

**2.6.3.30** `globus_result_t globus_xio_handle_cntl ( handle , driver , GLOBUS_XIO_GSI_REGISTER_ACCEPT_DELEGATION , gss_OID_set restriction_oids, gss_buffer_set_t restriction_buffers, OM_uint32 time_req, globus_xio_gsi_delegation_accept_callback_t callback, void * callback_arg )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Accept non-blocking delegation-at-any-time process.

#### Parameters

<i>restriction_oids</i>	The OIDS for X.509 extensions to embed in the delegated credential
<i>restriction_buffers</i>	The corresponding bodies for the X.509 extensions
<i>time_req</i>	The lifetime of the delegated credential
<i>callback</i>	The callback to call when the operation completes
<i>callback_arg</i>	The arguments to pass to the callback

**2.6.3.31** `globus_result_t globus_xio_attr_cntl ( attr , driver , GLOBUS_XIO_GSI_FORCE_SERVER_MODE , globus_bool_t server_mode )`

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Force the server mode setting.

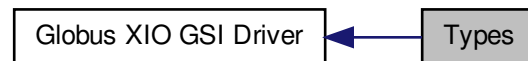
This explicitly sets the directionality of context establishment and delegation.

## Parameters

<code>server_mode</code>	The server mode.
--------------------------	------------------

## 2.7 Types

Collaboration diagram for Types:



### Typedefs

- typedef void(\* [globus\\_xio\\_gsi\\_delegation\\_init\\_callback\\_t](#))(globus\_result\_t result, void \*user\_arg)
- typedef void(\* [globus\\_xio\\_gsi\\_delegation\\_accept\\_callback\\_t](#))(globus\_result\_t result, gss\_cred\_id\_t delegated\_cred, OM\_uint32 time\_rec, void \*user\_arg)

### Enumerations

- enum [globus\\_xio\\_gsi\\_protection\\_level\\_t](#) {  
    [GLOBUS\\_XIO\\_GSI\\_PROTECTION\\_LEVEL\\_NONE](#),  
    [GLOBUS\\_XIO\\_GSI\\_PROTECTION\\_LEVEL\\_INTEGRITY](#),  
    [GLOBUS\\_XIO\\_GSI\\_PROTECTION\\_LEVEL\\_PRIVACY](#) }
- enum [globus\\_xio\\_gsi\\_delegation\\_mode\\_t](#) {  
    [GLOBUS\\_XIO\\_GSI\\_DELEGATION\\_MODE\\_NONE](#),  
    [GLOBUS\\_XIO\\_GSI\\_DELEGATION\\_MODE\\_LIMITED](#),  
    [GLOBUS\\_XIO\\_GSI\\_DELEGATION\\_MODE\\_FULL](#) }
- enum [globus\\_xio\\_gsi\\_proxy\\_mode\\_t](#) {  
    [GLOBUS\\_XIO\\_GSI\\_PROXY\\_MODE\\_FULL](#),  
    [GLOBUS\\_XIO\\_GSI\\_PROXY\\_MODE\\_LIMITED](#),  
    [GLOBUS\\_XIO\\_GSI\\_PROXY\\_MODE\\_MANY](#) }
- enum [globus\\_xio\\_gsi\\_authorization\\_mode\\_t](#) {  
    [GLOBUS\\_XIO\\_GSI\\_NO\\_AUTHORIZATION](#),  
    [GLOBUS\\_XIO\\_GSI\\_SELF\\_AUTHORIZATION](#),  
    [GLOBUS\\_XIO\\_GSI\\_IDENTITY\\_AUTHORIZATION](#),  
    [GLOBUS\\_XIO\\_GSI\\_HOST\\_AUTHORIZATION](#) }

### 2.7.1 Typedef Documentation

#### 2.7.1.1 typedef void(\* [globus\\_xio\\_gsi\\_delegation\\_init\\_callback\\_t](#))(globus\_result\_t result, void \*user\_arg)

Globus XIO GSI init delegation callback.



2.7.1.2 `typedef void(* globus_xio_gsi_delegation_accept_callback_t)(globus_result_t result, gss_cred_id_t delegated_cred, OM_uint32 time_rec, void *user_arg)`

Globus XIO GSI init delegation callback.

## 2.7.2 Enumeration Type Documentation

2.7.2.1 `enum globus_xio_gsi_protection_level_t`

Globus XIO GSI protection levels.

### Enumerator:

***GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_NONE*** No security.

***GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_INTEGRITY*** Messages are signed.

***GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_PRIVACY*** Messages are signed and encrypted.

2.7.2.2 `enum globus_xio_gsi_delegation_mode_t`

Globus XIO GSI delegation modes.

### Enumerator:

***GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_NONE*** No delegation.

***GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_LIMITED*** Delegate a limited proxy.

***GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_FULL*** Delegate a full proxy.

2.7.2.3 `enum globus_xio_gsi_proxy_mode_t`

Globus XIO GSI proxy modes.

### Enumerator:

***GLOBUS\_XIO\_GSI\_PROXY\_MODE\_FULL*** Accept only full proxies.

***GLOBUS\_XIO\_GSI\_PROXY\_MODE\_LIMITED*** Accept full proxies and limited proxies if they are the only limited proxy in the cert chain.

***GLOBUS\_XIO\_GSI\_PROXY\_MODE\_MANY*** Accept both full and limited proxies unconditionally.

2.7.2.4 `enum globus_xio_gsi_authorization_mode_t`

Globus XIO GSI authorization modes.

### Enumerator:

***GLOBUS\_XIO\_GSI\_NO\_AUTHORIZATION*** Do not perform any authorization. This will cause a error when used in conjunction with delegation on the init/client side.

***GLOBUS\_XIO\_GSI\_SELF\_AUTHORIZATION*** Authorize the peer if the peer has the same identity as ourselves.

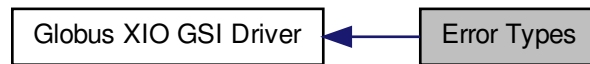
***GLOBUS\_XIO\_GSI\_IDENTITY\_AUTHORIZATION*** Authorize the peer if the peer identity matches the identity set in the target name.

***GLOBUS\_XIO\_GSI\_HOST\_AUTHORIZATION*** Authorize the peer if the identity of the peer matches the identity of the peer hostname.

## 2.8 Error Types

The GSI driver uses mostly GSSAPI calls, so it generally just wraps the underlying GSSAPI errors or uses generic xio errors.

Collaboration diagram for Error Types:



### Enumerations

- enum `globus_xio_gsi_error_t` {  
    `GLOBUS_XIO_GSI_ERROR_INVALID_PROTECTION_LEVEL`,  
    `GLOBUS_XIO_GSI_ERROR_WRAP_GSSAPI`,  
    `GLOBUS_XIO_GSI_ERROR_EMPTY_TARGET_NAME`,  
    `GLOBUS_XIO_GSI_ERROR_EMPTY_HOST_NAME`,  
    `GLOBUS_XIO_GSI_AUTHORIZATION_FAILED`,  
    `GLOBUS_XIO_GSI_ERROR_TOKEN_TOO_BIG` }

### 2.8.1 Detailed Description

The GSI driver uses mostly GSSAPI calls, so it generally just wraps the underlying GSSAPI errors or uses generic xio errors.

#### See also

`globus_xio_driver_error_match()`  
`globus_error_gssapi_match()`  
`globus_error_match_openssl_error()`

### 2.8.2 Enumeration Type Documentation

#### 2.8.2.1 enum `globus_xio_gsi_error_t`

GSI driver specific error types.

#### Enumerator:

**`GLOBUS_XIO_GSI_ERROR_INVALID_PROTECTION_LEVEL`** Indicates that the established context does not meet the required protection level.

**`GLOBUS_XIO_GSI_ERROR_WRAP_GSSAPI`** Wraps a GSSAPI error.

**`GLOBUS_XIO_GSI_ERROR_EMPTY_TARGET_NAME`** Indicates that `GLOBUS_XIO_GSI_IDENTITY_AUTHORIZATION` is set but that the target name is empty.

***GLOBUS\_XIO\_GSI\_ERROR\_EMPTY\_HOST\_NAME*** Indicates that GLOBUS\_XIO\_GSI\_HOST\_AUTHORIZATION is set but that no host name is available.

***GLOBUS\_XIO\_GSI\_AUTHORIZATION\_FAILED*** Indicates that the peer is not authorized.

***GLOBUS\_XIO\_GSI\_ERROR\_TOKEN\_TOO\_BIG*** Indicates the the token being read is too big. Usually happens when someone tries to establish a non secure session with a endpoint that expects security

## Index

### Attributes and Cntls, 4

- GLOBUS\_XIO\_GSI\_ACCEPT\_DELEGATION, 8
- GLOBUS\_XIO\_GSI\_FORCE\_SERVER\_MODE, 8
- GLOBUS\_XIO\_GSI\_GET\_AUTHORIZATION\_MODE, 7
- GLOBUS\_XIO\_GSI\_GET\_BUFFER\_SIZE, 7
- GLOBUS\_XIO\_GSI\_GET\_CONTEXT, 8
- GLOBUS\_XIO\_GSI\_GET\_CREDENTIAL, 7
- GLOBUS\_XIO\_GSI\_GET\_DELEGATED\_CRED, 8
- GLOBUS\_XIO\_GSI\_GET\_DELEGATION\_MODE, 7
- GLOBUS\_XIO\_GSI\_GET\_GSSAPI\_REQ\_FLAGS, 7
- GLOBUS\_XIO\_GSI\_GET\_LOCAL\_NAME, 8
- GLOBUS\_XIO\_GSI\_GET\_PEER\_NAME, 8
- GLOBUS\_XIO\_GSI\_GET\_PROTECTION\_LEVEL, 7
- GLOBUS\_XIO\_GSI\_GET\_PROXY\_MODE, 7
- GLOBUS\_XIO\_GSI\_GET\_TARGET\_NAME, 8
- GLOBUS\_XIO\_GSI\_GET\_WRAP\_MODE, 7
- GLOBUS\_XIO\_GSI\_INIT\_DELEGATION, 8
- GLOBUS\_XIO\_GSI\_REGISTER\_ACCEPT\_DELEGATION, 8
- GLOBUS\_XIO\_GSI\_REGISTER\_INIT\_DELEGATION, 8
- GLOBUS\_XIO\_GSI\_SET\_ANON, 7
- GLOBUS\_XIO\_GSI\_SET\_AUTHORIZATION\_MODE, 7
- GLOBUS\_XIO\_GSI\_SET\_BUFFER\_SIZE, 7
- GLOBUS\_XIO\_GSI\_SET\_CREDENTIAL, 7
- GLOBUS\_XIO\_GSI\_SET\_DELEGATION\_MODE, 7
- GLOBUS\_XIO\_GSI\_SET\_GSSAPI\_REQ\_FLAGS, 7
- GLOBUS\_XIO\_GSI\_SET\_PROTECTION\_LEVEL, 7
- GLOBUS\_XIO\_GSI\_SET\_PROXY\_MODE, 7
- GLOBUS\_XIO\_GSI\_SET\_SSL\_COMPATIBLE, 7
- GLOBUS\_XIO\_GSI\_SET\_TARGET\_NAME, 8
- GLOBUS\_XIO\_GSI\_SET\_WRAP\_MODE, 7
- globus\_xio\_attr\_cntl, 8–12, 14
- globus\_xio\_gsi\_cmd\_t, 7
- globus\_xio\_handle\_cntl, 8, 9, 12–14
- GLOBUS\_XIO\_GSI\_ACCEPT\_DELEGATION  
Attributes and Cntls, 8
- GLOBUS\_XIO\_GSI\_AUTHORIZATION\_FAILED  
Error Types, 18
- GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_FULL  
Types, 16
- GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_LIMITED  
Types, 16
- GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_NONE  
Types, 16
- GLOBUS\_XIO\_GSI\_ERROR\_EMPTY\_HOST\_NAME  
Error Types, 17
- GLOBUS\_XIO\_GSI\_ERROR\_EMPTY\_TARGET\_NAME  
Error Types, 17
- GLOBUS\_XIO\_GSI\_ERROR\_INVALID\_PROTECTION\_LEVEL  
Error Types, 17
- GLOBUS\_XIO\_GSI\_ERROR\_TOKEN\_TOO\_BIG  
Error Types, 18
- GLOBUS\_XIO\_GSI\_ERROR\_WRAP\_GSSAPI  
Error Types, 17
- GLOBUS\_XIO\_GSI\_FORCE\_SERVER\_MODE  
Attributes and Cntls, 8
- GLOBUS\_XIO\_GSI\_GET\_AUTHORIZATION\_MODE  
Attributes and Cntls, 7
- GLOBUS\_XIO\_GSI\_GET\_BUFFER\_SIZE  
Attributes and Cntls, 7
- GLOBUS\_XIO\_GSI\_GET\_CONTEXT  
Attributes and Cntls, 8
- GLOBUS\_XIO\_GSI\_GET\_CREDENTIAL  
Attributes and Cntls, 7
- GLOBUS\_XIO\_GSI\_GET\_DELEGATED\_CRED  
Attributes and Cntls, 8
- GLOBUS\_XIO\_GSI\_GET\_DELEGATION\_MODE  
Attributes and Cntls, 7
- GLOBUS\_XIO\_GSI\_GET\_GSSAPI\_REQ\_FLAGS  
Attributes and Cntls, 7
- GLOBUS\_XIO\_GSI\_GET\_LOCAL\_NAME  
Attributes and Cntls, 8
- GLOBUS\_XIO\_GSI\_GET\_PEER\_NAME  
Attributes and Cntls, 8
- GLOBUS\_XIO\_GSI\_GET\_PROTECTION\_LEVEL  
Attributes and Cntls, 7
- GLOBUS\_XIO\_GSI\_GET\_PROXY\_MODE  
Attributes and Cntls, 7
- GLOBUS\_XIO\_GSI\_GET\_TARGET\_NAME  
Attributes and Cntls, 8
- GLOBUS\_XIO\_GSI\_GET\_WRAP\_MODE  
Attributes and Cntls, 7
- GLOBUS\_XIO\_GSI\_HOST\_AUTHORIZATION  
Types, 16
- GLOBUS\_XIO\_GSI\_IDENTITY\_AUTHORIZATION

### Env Variables, 4

### Error Types, 17

- GLOBUS\_XIO\_GSI\_AUTHORIZATION\_FAILED, 18
- GLOBUS\_XIO\_GSI\_ERROR\_EMPTY\_HOST\_NAME, 17
- GLOBUS\_XIO\_GSI\_ERROR\_EMPTY\_TARGET\_NAME, 17
- GLOBUS\_XIO\_GSI\_ERROR\_INVALID\_PROTECTION\_LEVEL, 17
- GLOBUS\_XIO\_GSI\_ERROR\_TOKEN\_TOO\_BIG, 18
- GLOBUS\_XIO\_GSI\_ERROR\_WRAP\_GSSAPI, 17
- globus\_xio\_gsi\_error\_t, 17

### Globus XIO GSI Driver, 1

Types, [16](#)  
 GLOBUS\_XIO\_GSI\_INIT\_DELEGATION  
     Attributes and Cntls, [8](#)  
 GLOBUS\_XIO\_GSI\_NO\_AUTHORIZATION  
     Types, [16](#)  
 GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_INTEGRITY  
     Types, [16](#)  
 GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_NONE  
     Types, [16](#)  
 GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_PRIVACY  
     Types, [16](#)  
 GLOBUS\_XIO\_GSI\_PROXY\_MODE\_FULL  
     Types, [16](#)  
 GLOBUS\_XIO\_GSI\_PROXY\_MODE\_LIMITED  
     Types, [16](#)  
 GLOBUS\_XIO\_GSI\_PROXY\_MODE\_MANY  
     Types, [16](#)  
 GLOBUS\_XIO\_GSI\_REGISTER\_ACCEPT\_DELEGATION  
     Attributes and Cntls, [8](#)  
 GLOBUS\_XIO\_GSI\_REGISTER\_INIT\_DELEGATION  
     Attributes and Cntls, [8](#)  
 GLOBUS\_XIO\_GSI\_SELF\_AUTHORIZATION  
     Types, [16](#)  
 GLOBUS\_XIO\_GSI\_SET\_ANON  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_AUTHORIZATION\_MODE  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_BUFFER\_SIZE  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_CREDENTIAL  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_DELEGATION\_MODE  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_GSSAPI\_REQ\_FLAGS  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_PROTECTION\_LEVEL  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_PROXY\_MODE  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_SSL\_COMPATIBLE  
     Attributes and Cntls, [7](#)  
 GLOBUS\_XIO\_GSI\_SET\_TARGET\_NAME  
     Attributes and Cntls, [8](#)  
 GLOBUS\_XIO\_GSI\_SET\_WRAP\_MODE  
     Attributes and Cntls, [7](#)  
 globus\_xio\_attr\_cntl  
     Attributes and Cntls, [8–12](#), [14](#)  
 globus\_xio\_gsi\_authorization\_mode\_t  
     Types, [16](#)  
 globus\_xio\_gsi\_cmd\_t  
     Attributes and Cntls, [7](#)  
 globus\_xio\_gsi\_delegation\_accept\_callback\_t  
     Types, [15](#)  
 globus\_xio\_gsi\_delegation\_init\_callback\_t  
     Types, [15](#)  
 globus\_xio\_gsi\_delegation\_mode\_t  
     Types, [16](#)  
 globus\_xio\_gsi\_error\_t  
     Error Types, [17](#)  
 globus\_xio\_gsi\_protection\_level\_t  
     Types, [16](#)  
 globus\_xio\_gsi\_proxy\_mode\_t  
     Types, [16](#)  
 globus\_xio\_handle\_cntl  
     Attributes and Cntls, [8](#), [9](#), [12–14](#)  
 Opening/Closing, [2](#)  
 Reading/Writing, [3](#)  
 Server, [3](#)  
 Types, [15](#)  
 GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_FULL,  
     [16](#)  
 GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_LIMITED,  
     [16](#)  
 GLOBUS\_XIO\_GSI\_DELEGATION\_MODE\_NONE,  
     [16](#)  
 GLOBUS\_XIO\_GSI\_HOST\_AUTHORIZATION, [16](#)  
 GLOBUS\_XIO\_GSI\_IDENTITY\_AUTHORIZATION,  
     [16](#)  
 GLOBUS\_XIO\_GSI\_NO\_AUTHORIZATION, [16](#)  
 GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_INTEGRITY,  
     [16](#)  
 GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_NONE,  
     [16](#)  
 GLOBUS\_XIO\_GSI\_PROTECTION\_LEVEL\_PRIVACY,  
     [16](#)  
 GLOBUS\_XIO\_GSI\_PROXY\_MODE\_FULL, [16](#)  
 GLOBUS\_XIO\_GSI\_PROXY\_MODE\_LIMITED, [16](#)  
 GLOBUS\_XIO\_GSI\_PROXY\_MODE\_MANY, [16](#)  
 GLOBUS\_XIO\_GSI\_SELF\_AUTHORIZATION, [16](#)  
 globus\_xio\_gsi\_authorization\_mode\_t, [16](#)  
 globus\_xio\_gsi\_delegation\_accept\_callback\_t, [15](#)  
 globus\_xio\_gsi\_delegation\_init\_callback\_t, [15](#)  
 globus\_xio\_gsi\_delegation\_mode\_t, [16](#)  
 globus\_xio\_gsi\_protection\_level\_t, [16](#)  
 globus\_xio\_gsi\_proxy\_mode\_t, [16](#)