globus xio gsi driver Reference Manual

0.6

Generated by Doxygen 1.4.4

Contents

1   globus xio gsi driver Module Index

1.1   globus xio gsi driver Modules

Here is a list of all modules:

2   globus xio gsi driver Module Documentation

2.1   Globus XIO GSI Driver

Collaboration diagram for Globus XIO GSI Driver:
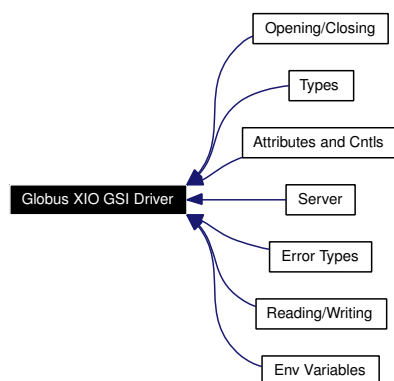
The GSI driver.

Modules

2.1.1 Detailed Description

The GSI driver.

2.2 Opening/Closing

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

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
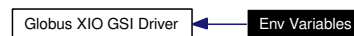
Collaboration diagram for Server:
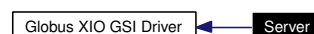
globus_xio_server_create() causes a new transport-specific listener socket to be created to handle new GSI connections. **globus_xio_server_register_accept()** will accept a new connection for processing. **globus_xio_-server_register_close()** cleans up the internal resources associated with the http 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 overriden during the **globus_xio_register_open()** call (GLOBUS_-XIO_GSI_FORCE_SERVER_MODE).

2.5 Env 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

Collaboration diagram for Attributes and Cntls:

GSI driver specific attrs 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

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 .

GLOBUS_XIO_GSI_GET_PROTECTION_LEVEL See usage for: globus_xio_attr_cntl .

GLOBUS_XIO_GSI_GET_TARGET_NAME See usage for: globus_xio_attr_cntl .

GLOBUS_XIO_GSI_SET_TARGET_NAME See usage for: globus_xio_attr_cntl .

GLOBUS_XIO_GSI_GET_CONTEXT See usage for: globus_xio_handle_cntl .

GLOBUS_XIO_GSI_GET_DELEGATED_CRED See usage for: globus_xio_handle_cntl .

GLOBUS_XIO_GSI_GET_PEER_NAME See usage for: globus_xio_handle_cntl .

GLOBUS_XIO_GSI_GET_LOCAL_NAME See usage for: globus_xio_handle_cntl .

GLOBUS_XIO_GSI_INIT_DELEGATION See usage for: globus_xio_handle_cntl .

GLOBUS_XIO_GSI_REGISTER_INIT_DELEGATION See usage for: globus_xio_handle_cntl .

GLOBUS_XIO_GSI_ACCEPT_DELEGATION See usage for: globus_xio_handle_cntl .

GLOBUS_XIO_GSI_REGISTER_ACCEPT_DELEGATION See usage for: globus_xio_handle_cntl .

GLOBUS_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, GLOBUS_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:
    credential 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, GLOBUS_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:
    credential 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, 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:
    credential 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 initizalization.

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:
    credential 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 initizalization.

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:
    req_flags 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:
>*req_flags* 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:
>*proxy_mode* 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:
>*proxy_mode* 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:
>*authz_mode* 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:
>*authz_mode* 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:
   delegation_mode 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:
   delegation_mode 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:
   ssl_mode 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:
   anon_mode 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:
   wrap_mode 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:
>   *wrap_mode*   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:
>   *buffer_size*   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:
>   *buffer_size*   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:
>   *protection_level*   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:
>   *protection_level*   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:
    target_name 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:
    target_name 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:
    context 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:
    credential 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:
    peer_name 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:
    local_name 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:

    credential The GSS credential to delegate

    restriction_oids The OIDS for X.509 extensions to embed in the delegated credential

    restriction_buffers The corresponding bodies for the X.509 extensions

    time_req 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:

    credential The GSS credential to delegate

    restriction_oids The OIDS for X.509 extensions to embed in the delegated credential

    restriction_buffers The corresponding bodies for the X.509 extensions

    time_req The lifetime of the delegated credential

    callback The callback to call when the operation completes

    callback_arg 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:

    credential The delegated GSS credential

    restriction_oids The OIDS for X.509 extensions to embed in the delegated credential

    restriction_buffers The corresponding bodies for the X.509 extensions

    time_req 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:

    restriction_oids The OIDS for X.509 extensions to embed in the delegated credential

restriction_buffers The corresponding bodies for the X.509 extensions

time_req The lifetime of the delegated credential

callback The callback to call when the operation completes

callback_arg 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:
   server_mode The server mode.

2.7 Types

Collaboration diagram for Types:
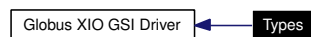
Typedefs

- 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

Collaboration diagram for Error Types:
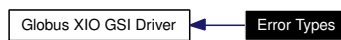
The GSI driver uses mostly GSSAPI calls, so it generally just wraps the underlying GSSAPI errors or uses generic xio errors.

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 protecetion 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