

# globus authz Reference Manual

## 2.2

Generated by Doxygen 1.4.7

Tue May 1 03:42:31 2012

# Contents

1	globus authz Module Index	1
2	globus authz Module Documentation	1

## 1 globus authz Module Index

### 1.1 globus authz Modules

Here is a list of all modules:

GSI Authorization API	1
GSI Credential Constants	3

## 2 globus authz Module Documentation

### 2.1 GSI Authorization API

#### Initialize Handle

- `globus_result_t globus_gsi_authz_handle_init (globus_gsi_authz_handle_t *handle, const char *service_name, const gss_ctx_id_t context, globus_gsi_authz_cb_t callback, void *callback_arg)`

#### Authorization decision made here

- `globus_result_t globus_gsi_authorize (globus_gsi_authz_handle_t handle, const void *action, const void *object, globus_gsi_authz_cb_t callback, void *callback_arg)`

#### Destroy Handle

- `globus_result_t globus_gsi_authz_handle_destroy (globus_gsi_authz_handle_t handle, globus_gsi_authz_cb_t callback, void *callback_arg)`

#### Query for authorization identity

- `globus_result_t globus_gsi_authz_get_authorization_identity (globus_gsi_authz_handle_t handle, char **identity_ptr, globus_gsi_authz_cb_t callback, void *callback_arg)`

#### 2.1.1 Function Documentation

**2.1.1.1** `globus_result_t globus_gsi_authz_handle_init (globus_gsi_authz_handle_t * handle, const char * service_name, const gss_ctx_id_t context, globus_gsi_authz_cb_t callback, void * callback_arg)`

Initializes a handle.

#### Parameters:

*handle* Pointer to the handle that is to be initialized

*service\_name* Service to authorize access to  
*context* Security context used to contact the service  
*callback* Callback function to call when authz handle init completes  
*callback\_arg* Argument to callback function

**Returns:**

GLOBUS\_SUCCESS if successful A Globus error object on failure:

**2.1.1.2 globus\_result\_t globus\_gsi\_authorize (globus\_gsi\_authz\_handle\_t handle, const void \* action, const void \* object, globus\_gsi\_authz\_cb\_t callback, void \* callback\_arg)**

Authorization decision made here.

**Parameters:**

*handle* Pointer to the handle that is to be initialized  
*action* Action to authorize  
*object* Object that the action pertains to.  
*callback* Callback function to call when authorization completes  
*callback\_arg* Argument to callback function

**Returns:**

GLOBUS\_SUCCESS if successful A Globus error object on failure:

**2.1.1.3 globus\_result\_t globus\_gsi\_authz\_handle\_destroy (globus\_gsi\_authz\_handle\_t handle, globus\_gsi\_authz\_cb\_t callback, void \* callback\_arg)**

Destroy a Globus GSI authz handle.

**Parameters:**

*handle* The handle that is to be destroyed  
*callback* Callback function to call when handle is destroyed  
*callback\_arg* Argument to callback function

**Returns:**

GLOBUS\_SUCCESS

**2.1.1.4 globus\_result\_t globus\_gsi\_authz\_get\_authorization\_identity (globus\_gsi\_authz\_handle\_t handle, char \*\* identity\_ptr, globus\_gsi\_authz\_cb\_t callback, void \* callback\_arg)**

Query for authorization identity.

**Parameters:**

*handle* The handle that is to be used for the identity check.  
*identity\_ptr* The authorization identity determined by the authorization handle. This is must be freed by the caller. If the value is NULL (and this function returned GLOBUS\_SUCCESS), the caller should use the authenticated identity.

*callback* Callback function to call when identity is determined.

*callback\_arg* Argument to callback function.

**Returns:**

GLOBAL\_SUCCESS

## 2.2 GSI Credential Constants

### Enumerations

- enum globus\_gsi\_authz\_error\_t {  
    GLOBAL\_GSI\_AUTHZ\_ERROR\_SUCCESS = 0,  
    GLOBAL\_GSI\_AUTHZ\_ERROR\_ERRNO = 1,  
    GLOBAL\_GSI\_AUTHZ\_ERROR\_BAD\_PARAMETER = 2,  
    GLOBAL\_GSI\_AUTHZ\_ERROR\_CALLOUT = 3 }

### 2.2.1 Enumeration Type Documentation

#### 2.2.1.1 enum globus\_gsi\_authz\_error\_t

GSI Authz Error codes.

**Enumerator:**

**GLOBAL\_GSI\_AUTHZ\_ERROR\_SUCCESS** Success.

**GLOBAL\_GSI\_AUTHZ\_ERROR\_ERRNO** Error with system call.

**GLOBAL\_GSI\_AUTHZ\_ERROR\_BAD\_PARAMETER** Invalid parameter.

**GLOBAL\_GSI\_AUTHZ\_ERROR\_CALLOUT** Callout returned an error.

## Index

- globus\_gsi\_authorize
  - globus\_gsi\_authz, 2
- globus\_gsi\_authz
  - globus\_gsi\_authorize, 2
  - globus\_gsi\_authz\_get\_authorization\_identity, 2
  - globus\_gsi\_authz\_handle\_destroy, 2
  - globus\_gsi\_authz\_handle\_init, 1
- globus\_gsi\_authz\_constants
  - GLOBUS\_GSI\_AUTHZ\_ERROR\_BAD\_PARAMETER, 3
  - GLOBUS\_GSI\_AUTHZ\_ERROR\_CALLOUT, 3
  - GLOBUS\_GSI\_AUTHZ\_ERROR\_ERRNO, 3
  - GLOBUS\_GSI\_AUTHZ\_ERROR\_SUCCESS, 3
  - globus\_gsi\_authz\_error\_t, 3
- GLOBUS\_GSI\_AUTHZ\_ERROR\_BAD\_PARAMETER
  - globus\_gsi\_authz\_constants, 3
- GLOBUS\_GSI\_AUTHZ\_ERROR\_CALLOUT
  - globus\_gsi\_authz\_constants, 3
- GLOBUS\_GSI\_AUTHZ\_ERROR\_ERRNO
  - globus\_gsi\_authz\_constants, 3
- GLOBUS\_GSI\_AUTHZ\_ERROR\_SUCCESS
  - globus\_gsi\_authz\_constants, 3
- globus\_gsi\_authz\_error\_t
  - globus\_gsi\_authz\_constants, 3
- globus\_gsi\_authz\_get\_authorization\_identity
  - globus\_gsi\_authz, 2
- globus\_gsi\_authz\_handle\_destroy
  - globus\_gsi\_authz, 2
- globus\_gsi\_authz\_handle\_init
  - globus\_gsi\_authz, 1
- GSI Authorization API, 1
- GSI Credential Constants, 3