

globus authz Reference Manual

0.7

Generated by Doxygen 1.3.5

Sat Feb 6 11:41:57 2010

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_authz_decision(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 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:

GLOBUS_SUCCESS

2.2 GSI Credential Constants

Enumerations

- `enum globus_gsi_authz_error_t`
`GLOBUS_GSI_AUTHZ_ERROR_SUCCESS = 0,`
`GLOBUS_GSI_AUTHZ_ERROR_ERRNO = 1,`
`GLOBUS_GSI_AUTHZ_ERROR_BAD_PARAMETER = 2,`
`GLOBUS_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.

Enumeration values:

`GLOBUS_GSI_AUTHZ_ERROR_SUCCESS` Success.

`GLOBUS_GSI_AUTHZ_ERROR_ERRNO` Error with system call.

`GLOBUS_GSI_AUTHZ_ERROR_BAD_PARAMETER` Invalid parameter.

`GLOBUS_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_constants
 - globus_gsi_authz_error [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 Constant [3](#)