

globus gssapi error

4.1

Generated by Doxygen 1.7.5

Thu Jan 26 2012 15:05:11

## Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Module Documentation</b>	<b>1</b>
2.1	Globus GSSAPI Error API . . . . .	2
2.1.1	Detailed Description . . . . .	2
2.2	Error Construction . . . . .	3
2.2.1	Detailed Description . . . . .	3
2.2.2	Define Documentation . . . . .	3
2.2.3	Function Documentation . . . . .	3
2.3	Error Data Accessors and Modifiers . . . . .	5
2.3.1	Detailed Description . . . . .	5
2.3.2	Function Documentation . . . . .	5
2.4	Error Handling Helpers . . . . .	7
2.4.1	Detailed Description . . . . .	7
2.4.2	Function Documentation . . . . .	7

## 1 Module Index

### 1.1 Modules

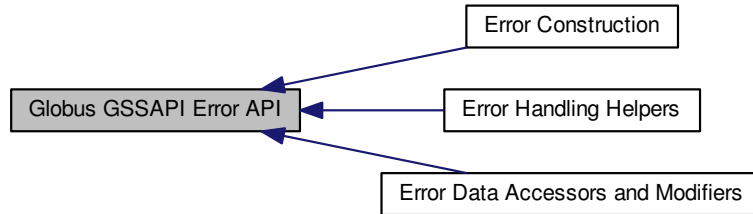
Here is a list of all modules:

<b>Globus GSSAPI Error API</b>	<b>2</b>
<b>Error Construction</b>	<b>3</b>
<b>Error Data Accessors and Modifiers</b>	<b>5</b>
<b>Error Handling Helpers</b>	<b>7</b>

## 2 Module Documentation

## 2.1 Globus GSSAPI Error API

Collaboration diagram for Globus GSSAPI Error API:



### Modules

- **Error Construction**
- **Error Data Accessors and Modifiers**
- **Error Handling Helpers**

#### 2.1.1 Detailed Description

These `globus_error` functions are motivated by the desire to provide a easier way of generating new error types, while at the same time preserving all features (e.g. memory management, chaining) of the current error handling framework. The functions in this API are auxiliary to the function in the Globus Generic Error API in the sense that they provide a wrapper for representing GSSAPI errors in terms of a `globus_error_t`.

Any program that uses Globus GSSAPI Error functions must include "`globus_error_gssapi.h`".

## 2.2 Error Construction

Collaboration diagram for Error Construction:



### Defines

- `#define GLOBUS_ERROR_TYPE_GSSAPI`

### Construct Error

- `globus_object_t * globus_error_construct_gssapi_error (globus_module_descriptor_t *base_source, globus_object_t *base_cause, const OM_uint32 major_status, const OM_uint32 minor_status)`

### Initialize Error

- `globus_object_t * globus_error_initialize_gssapi_error (globus_object_t *error, globus_module_descriptor_t *base_source, globus_object_t *base_cause, const OM_uint32 major_status, const OM_uint32 minor_status)`

### 2.2.1 Detailed Description

Create and initialize a Globus GSSAPI Error object. This section defines operations to create and initialize Globus GSSAPI Error objects.

### 2.2.2 Define Documentation

#### 2.2.2.1 `#define GLOBUS_ERROR_TYPE_GSSAPI`

Error type definition.

### 2.2.3 Function Documentation

#### 2.2.3.1 `globus_object_t* globus_error_construct_gssapi_error ( globus_module_descriptor_t * base_source, globus_object_t * base_cause, const OM_uint32 major_status, const OM_uint32 minor_status )`

Allocate and initialize an error of type `GLOBUS_ERROR_TYPE_GSSAPI`.

### Parameters

<i>base_source</i>	Pointer to the originating module.
<i>base_cause</i>	The error object causing the error. If this is the original error, this parameter may be NULL.
<i>major_status</i>	The GSSAPI major status
<i>minor_status</i>	The GSSAPI minor status

## Returns

The resulting error object. It is the user's responsibility to eventually free this object using `globus_object_free()`. A `globus_result_t` may be obtained by calling `globus_error_put()` on this object.

**2.2.3.2** `globus_object_t* globus_error_initialize_gssapi_error ( globus_object_t * error, globus_module_descriptor_t * base_source, globus_object_t * base_cause, const OM_uint32 major_status, const OM_uint32 minor_status )`

Initialize a previously allocated error of type `GLOBUS_ERROR_TYPE_GSSAPI`.

## Parameters

<i>error</i>	The previously allocated error object.
<i>base_source</i>	Pointer to the originating module.
<i>base_cause</i>	The error object causing the error. If this is the original error this parameter may be NULL.
<i>major_status</i>	The GSSAPI major status
<i>minor_status</i>	The GSSAPI minor status

## Returns

The resulting error object. You may have to call `globus_error_put()` on this object before passing it on.

## 2.3 Error Data Accessors and Modifiers

Collaboration diagram for Error Data Accessors and Modifiers:



### Get Major Status

- OM\_uint32 **globus\_error\_gssapi\_get\_major\_status** (globus\_object\_t \*error)

### Set Major Status

- void **globus\_error\_gssapi\_set\_major\_status** (globus\_object\_t \*error, const OM\_uint32 major\_status)

### Get Minor Status

- OM\_uint32 **globus\_error\_gssapi\_get\_minor\_status** (globus\_object\_t \*error)

#### 2.3.1 Detailed Description

Get and set data in a Globus GSSAPI Error object. This section defines operations for accessing and modifying data in a Globus GSSAPI Error object.

#### 2.3.2 Function Documentation

##### 2.3.2.1 OM\_uint32 globus\_error\_gssapi\_get\_major\_status ( globus\_object\_t \* *error* )

Retrieve the major status from a gssapi error object.

##### Parameters

<i>error</i>	The error from which to retrieve the major status
--------------	---

##### Returns

The major status stored in the object

##### 2.3.2.2 void globus\_error\_gssapi\_set\_major\_status ( globus\_object\_t \* *error*, const OM\_uint32 *major\_status* )

Set the major status in a gssapi error object.

##### Parameters

<i>error</i>	The error object for which to set the major status
<i>major_status</i>	The major status

## Returns

void

**2.3.2.3** OM\_uint32 globus\_error\_gssapi\_get\_minor\_status ( globus\_object\_t \* *error* )

Retrieve the minor status from a gssapi error object.

## Parameters

<i>error</i>	The error from which to retrieve the minor status
--------------	---

## Returns

The minor status stored in the object

## 2.4 Error Handling Helpers

Collaboration diagram for Error Handling Helpers:



### Error Match

- `globus_bool_t globus_error_gssapi_match (globus_object_t *error, globus_module_descriptor_t *module, const OM_uint32 major_status)`

### Wrap GSSAPI Error

- `globus_object_t * globus_error_wrap_gssapi_error (globus_module_descriptor_t *base_source, OM_uint32 major_status, OM_uint32 minor_status, int type, const char *source_file, const char *source_func, int source_line, const char *short_desc_format,...)`

#### 2.4.1 Detailed Description

Helper functions for dealing with Globus GSSAPI Error objects. This section defines utility functions for dealing with Globus GSSAPI Error objects.

#### 2.4.2 Function Documentation

**2.4.2.1** `globus_bool_t globus_error_gssapi_match ( globus_object_t * error, globus_module_descriptor_t * module, const OM_uint32 major_status )`

Check whether the error originated from a specific module and match a specific major status.

This function checks whether the error or any of it's causative errors originated from a specific module and contains a specific major status. If the module descriptor is left unspecified this function will check for any error of the specified major\_status and vice versa.

#### Parameters

<i>error</i>	The error object for which to perform the check
<i>module</i>	The module descriptor to check for
<i>major_status</i>	The major status to check for



## Returns

GLOBUS\_TRUE - the error matched the module and major status GLOBUS\_FALSE - the error failed to match the module and major status

2.4.2.2 `globus_object_t* globus_error_wrap_gssapi_error ( globus_module_descriptor_t * base_source, OM_uint32 major_status, OM_uint32 minor_status, int type, const char * source_file, const char * source_func, int source_line, const char * short_desc_format, ... )`

Allocate and initialize an error of type GLOBUS\_ERROR\_TYPE\_GLOBUS which contains a causal error of type GLOBUS\_ERROR\_TYPE\_GSSAPI.

## Parameters

<i>base_source</i>	Pointer to the originating module.
<i>major_status</i>	The major status to use when generating the causal error.
<i>minor_status</i>	The minor status to use when generating the causal error.
<i>type</i>	The error type. We may reserve part of this namespace for common errors. Errors not in this space are assumed to be local to the originating module.
<i>source_file</i>	Name of file. Use <code>__FILE__</code>
<i>source_func</i>	Name of function. Use <code>_globus_func_name</code> and declare your func with <code>GlobusFuncName(&lt;name&gt;)</code>
<i>source_line</i>	Line number. Use <code>__LINE__</code>
<i>short_desc_format</i>	Short format string giving a succinct description of the error. To be passed on to the user.
...	Arguments for the format string.

## Returns

The resulting error object. It is the user's responsibility to eventually free this object using `globus_object_free()`. A `globus_result_t` may be obtained by calling `globus_error_put()` on this object.