globus gsi openssl error

2.1

Generated by Doxygen 1.7.6.1

Thu Mar 15 2012 10:30:01

Contents

1 Module Index

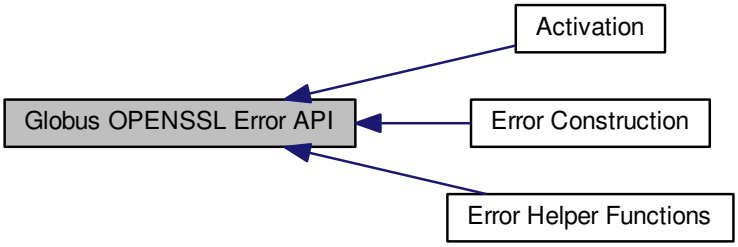
1.1 Modules

Here is a list of all modules:

2 Module Documentation

2.1 Globus OPENSSL Error API

Collaboration diagram for Globus OPENSSL Error API:



Modules

- **Activation**
- **Error Construction**
- **Error Helper Functions**

2.1.1 Detailed Description

These globus_openssl_error functions provide a wrapper to error types defined by OpenSSL

Any program that uses Globus OpenSSL Error functions must include "globus_error_openssl.h".

2.2 Activation

Collaboration diagram for Activation:

Defines

- #define **GLOBUS_GSI_OPENSSL_ERROR_MODULE**

2.2.1 Detailed Description

Globus GSI OpenSSL Error uses standard Globus module activation and deactivation. Before any Globus GSI OpenSSL Error functions are called, the following function must be called:

```
globus_module_activate(GLOBUS_GSI_OPENSSL_ERROR_MODULE)
```

This function returns GLOBUS_SUCCESS if Globus GSI OpenSSL Error was successfully initialized, and you are therefore allowed to subsequently call Globus GSI OpenSSL Error functions. Otherwise, an error code is returned, and Globus GSI OpenSSL Error functions should not be subsequently called. This function may be called multiple times.

To deactivate Globus GSI OpenSSL Error, the following function must be called:

```
globus_module_deactivate(GLOBUS_GSI_OPENSSL_ERROR_MODULE)
```

This function should be called once for each time Globus GSI OpenSSL Error was activated.

2.2.2 Define Documentation

2.2.2.1 #define GLOBUS_GSI_OPENSSL_ERROR_MODULE

Module descriptor.

2.3 Error Construction

Collaboration diagram for Error Construction:

Defines

- #define **GLOBUS_ERROR_TYPE_OPENSSL**

Get Error Code

- unsigned long **globus_openssl_error_handle_get_error_code** (globus_openssl_error_handle_t handle)

Get Error Data

- const char * **globus_openssl_error_handle_get_data** (globus_openssl_error_handle_t handle)

Get Error Data Flags

- int **globus_openssl_error_handle_get_data_flags** (globus_openssl_error_handle_t handle)

Get Filename

- const char * **globus_openssl_error_handle_get_filename** (globus_openssl_error_handle_t handle)

Get Linenumber

- int **globus_openssl_error_handle_get_linenumber** (globus_openssl_error_handle_t handle)

Get Library

- const char * **globus_openssl_error_handle_get_library** (globus_openssl_error_handle_t handle)

Get Function

- const char * **globus_openssl_error_handle_get_function** (globus_openssl_error_handle_t handle)

Get Reason

- const char * **globus_openssl_error_handle_get_reason** (globus_openssl_error_handle_t handle)

Construct Error

- globus_object_t * **globus_error_construct_openssl_error** (globus_module_descriptor_t *base_source, globus_object_t *base_cause)

Initialize Error

- globus_object_t * **globus_error_initialize_openssl_error** (globus_object_t *error, globus_module_descriptor_t *base_source, globus_object_t *base_cause, globus_openssl_error_handle_t openssl_error_handle)

2.3.1 Detailed Description

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

2.3.2 Define Documentation

2.3.2.1 #define GLOBUS_ERROR_TYPE_OPENSSL

Error type definition.

2.3.3 Function Documentation

2.3.3.1 unsigned long globus_openssl_error_handle_get_error_code (globus_openssl_error_handle_t *handle*)

Get the openssl error code which represents the openssl error from the openssl error handle.

Parameters

handle	The openssl error handle

Returns

The error code

2.3.3.2 const char* globus_openssl_error_handle_get_data (globus_openssl_error_handle_t *handle*)

Get the openssl error data which contains additional data about the error from the openssl error handle.

Parameters

handle	The openssl error handle

Returns

The error data

2.3.3.3 int globus_openssl_error_handle_get_data_flags (globus_openssl_error_handle_t *handle*)

Get the openssl error data flags from the openssl error handle.

Parameters

handle	The openssl error handle

Returns

The error data flags

2.3.3.4 const char* globus_openssl_error_handle_get_filename (globus_openssl_error_handle_t *handle*)

Get the filename where the openssl error occurred from the openssl error handle.

Parameters

handle	The openssl error handle

Returns

The filename

2.3.3.5 int globus_openssl_error_handle_get_linenumber (globus_openssl_error_handle_t *handle*)

Get the linenumber on which the openssl error occurred from the openssl error handle.

Parameters

handle	The openssl error handle

Returns

The linenumber

2.3.3.6 const char* globus_openssl_error_handle_get_library (globus_openssl_error_handle_t *handle*)

Get the library name where the openssl error occurred in from the openssl error handle.

Parameters

handle	The openssl error handle

Returns

The library name

2.3.3.7 const char* globus_openssl_error_handle_get_function (globus_openssl_error_handle_t *handle*)

Get the function name where the openssl error occurred from the openssl error handle.

Parameters

handle	The openssl error handle

Returns

The function name

2.3.3.8 const char* globus_openssl_error_handle_get_reason (globus_openssl_error_handle_t *handle*)

Get the reason string which caused the openssl error from the openssl error handle.

Parameters

handle	The openssl error handle

Returns

The reson string

2.3.3.9 globus_object_t* globus_error_construct_openssl_error (globus_module_descriptor_t * *base_source,* globus_object_t * *base_cause*)

Allocate and initialize an error of type GLOBUS_ERROR_TYPE_OPENSSL This function, combined with **globus_error_initialize_openssl_error()** (p. 7) will recursively generate globus error objects (of type globus_object_t) from the errors on openssl's static error stack.

The errors will be chained in a causal fashion to provide a path to the root cause of the actual error.

NOTE: the static stack openssl implements for its errors currently only supports at most 16 errors, so if more are added, the errors that were added first will be wiped out. If 16 errors are counted in the chain of openssl errors, its possible that some errors (including the original error) are missing.

Parameters

base_source	Pointer to the originating globus module.
base_cause	The error object causing the error. This parameter should be NULL in nearly all cases, as the root cause of an error will most likely be in the openssl code itself. The actual cause of the error is determined from the static stack of openssl errors.

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.

2.3.3.10 globus_object_t* globus_error_initialize_openssl_error (globus_object_t * *error,* globus_module_descriptor_t * *base_source,* globus_object_t * *base_cause,* globus_openssl_error_handle_t *openssl_error_handle*)

Initialize a previously allocated error of type GLOBUS_ERROR_TYPE_OPENSSL.

Parameters

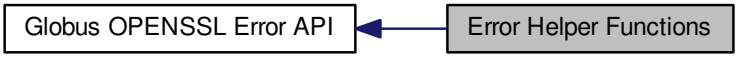
error	The previously allocated error object.
base_source	Pointer to the originating module.
base_cause	The error object causing the error. If this is the original error this paramater may be NULL.
openssl_error_handle	The openssl error handle associated with this error, this parameter should already be initialized to contain the openssl error code associated with the error.

Returns

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

2.4 Error Helper Functions

Collaboration diagram for Error Helper Functions:



OpenSSL Error Match

- globus_bool_t **globus_error_match_openssl_error** (globus_object_t *error, unsigned long library, unsigned long function, unsigned long reason)

Wrap OpenSSL Error

- globus_object_t * **globus_error_wrap_openssl_error** (globus_module_descriptor_t *base_source, int error_type, const char *source_file, const char *source_func, int source_line, const char *format,...)

2.4.1 Detailed Description

Utility functions that deal with Globus OpenSSL Error objects. This section defines utility function for Globus OpenSSL Error objects.

2.4.2 Function Documentation

2.4.2.1 globus_bool_t globus_error_match_openssl_error (globus_object_t * *error,* unsigned long *library,* unsigned long *function,* unsigned long *reason*)

Check whether the error originated from a specific library, from a specific function and is of a specific type.

This function checks whether the error or any of it's causative errors originated from a specific library, specific function and is of a specific type.

Parameters

error	The error object for which to perform the check
library	The library to check for
function	The function to check for
reason	The type to check for

Returns

GLOBUS_TRUE - the error matched GLOBUS_FALSE - the error failed to match

2.4.2.2 globus_object_t* globus_error_wrap_openssl_error (globus_module_descriptor_t * *base_source,* int *error_type,* const char * *source_file,* const char * *source_func,* int *source_line,* const char * *format,* ...)

Wrap the OpenSSL error and create a wrapped globus error object from the error.

This function gets all the openssl errors from the error list, and chains them using the globus error string object. The resulting globus error object is a wrapper to the openssl error at the end of the chain.

Parameters

base_source	The module that the error was generated from
error_type	The type of error encapsulating the openssl error
source_file	Name of file. Use __FILE__
source_func	Name of function. Use _globus_func_name and declare your func with GlobusFuncName(<name>)
source_line	Line number. Use __LINE__
format	format string for the description of the error entry point where the openssl error occurred, should be followed by parameters to fill the format string (like in printf).

Returns

The globus error object. A globus_result_t object can be created using the globus_error_put function

See also

globus_error_put()