

globus gass copy Reference Manual

4.14

Generated by Doxygen 1.2.18

Tue Aug 11 22:55:21 2009

Contents

1	globus gass copy Main Page	1
2	globus gass copy Data Structure Index	1
3	globus gass copy File Index	2
4	globus gass copy Data Structure Documentation	2
5	globus gass copy File Documentation	6

1 globus gass copy Main Page

The Globus GASS Copy library is motivated by the desire to provide a uniform interface to transfer files specified by different protocols.

The goals in doing this are to:

- Provide a robust way to describe and apply file transfer properties for a variety of protocols. These include the standard HTTP, FTP and GSIFTP options. Some of the new file transfer capabilities in GSIFTP are parallel, striping, authentication and TCP buffer sizing.

- Provide a service to support nonblocking file transfer and handle asynchronous file and network events.

- Provide a simple and portable way to implement file transfers.

Any program that uses Globus GASS Copy functions must include [globusgasscopy.h](#).

2 globus gass copy Data Structure Index

2.1 globus gass copy Data Structures

Here are the data structures with brief descriptions:

globus_gasscopy_glob_stat_t	(Glob expanded entry information)	2
globus_gasscopy_state_s	(The state structure contains all that is required to perform a file transfer from a source to a destination)	3
globus_i_gasscopy_buffer_t	(The buffer structure used for read/write queue entries)	4
globus_i_gasscopy_cancels	(Gass copy cancel struct)	4
globus_i_gasscopy_monitor_t	(The state monitor struct)	4
globus_i_gasscopy_state_target_s	(GASS copy target (e.g)	4

3 globus gass copy File Index

3.1 globus gass copy File List

Here is a list of all documented files with brief descriptions:

globus_gasscopy.c (Globus GASS Copy library)	6
globus_gasscopy.h (Header file for the gass copy library)	17

4 globus gass copy Data Structure Documentation

4.1 globusgasscopy_glob_stat_t Struct Reference

Glob expanded entry information.

Data Fields

```

globusgasscopy_glob_entry_t type
char  unique_id
char  symlink_target
int  mode
int  mdtm
globusoff_t size

```

4.1.1 Detailed Description

Glob expanded entry information.

4.1.2 Field Documentation

4.1.2.1 [globus_gasscopy_glob_entry_t](#) globus_gasscopy_glob_stat_t::type

The file type of the entry.

4.1.2.2 char globus_gasscopy_glob_stat_t::unique_id

A string that uniquely identifies the data that the entry refers to.

A file and a symlink to that file will have the same unique_id. It is NULL for when not available.

4.1.2.3 char globus_gasscopy_glob_stat_t::symlink_target

This points to the full path of the target of a symlink.

It is NULL for non-symlinks or when not available.

4.1.2.4 int globusgasscopy_glob_stat_t::mode

An integer specifying the mode of the file.

It is set to -1 when not available.

4.1.2.5 int globusgasscopy_glob_stat_t::mdtm

An integer specifying the modification time of the file.

It is set to -1 when not available.

4.1.2.6 globus_off_t globus_gasscopy_glob_stat_t::size

A globus_off_t specifying the size of the file.

It is set to -1 when not available.

4.2 globusgasscopy_state.s Struct Reference

The state structure contains all that is required to perform a file transfer from a source to a destination.

Data Fields

globus_i_gasscopy_target_t [source](#)
globus_i_gasscopy_target_t [dest](#)
[globus_i_gasscopy_monitor_t](#) [monitor](#)
globus_mutex_t [mutex](#)
globus_i_gasscopy_cancelstatust [cancel](#)

4.2.1 Detailed Description

The state structure contains all that is required to perform a file transfer from a source to a destination.

4.2.2 Field Documentation

4.2.2.1 globus_i_gasscopy_target_t globus_gasscopy_state.s::source

Source information for the file transfer.

4.2.2.2 globus_i_gasscopy_target_t globus_gasscopy_state.s::dest

Dest information for the file transfer.

4.2.2.3 [globus_i_gasscopy_monitor_t](#) globus_gasscopy_state.s::monitor

Used for signalling from the various callback functions.

4.2.2.4 globus_mutex_t globus_gasscopy_state.s::mutex

coordinates the modifying of the state, aside from the target structures

4.2.2.5 globusi_gasscopy_cancelstatus_t globus_gasscopy_state.s::cancel
indicates the status of the cancel operation.

4.3 globusi_gasscopy_buffer_t Struct Reference

The buffer structure used for read/write queue entries.

4.3.1 Detailed Description

The buffer structure used for read/write queue entries.

4.4 globusi_gasscopy_cancels Struct Reference

gass copy cancel struct

4.4.1 Detailed Description

gass copy cancel struct

4.5 globusi_gasscopy_monitor_t Struct Reference

The state monitor struct.

4.5.1 Detailed Description

The state monitor struct.

4.6 globusi_gasscopy_state_target_s Struct Reference

GASS copy target (e.g.

Data Fields

```

char url
globus_gasscopy_attr_t attr
globus_mutex_t mutex
globus_file_t queue
int n_pending
int n_simultaneous
int n_complete
globusi_gasscopy_targetstatus_t status
globus_gasscopy_url_mode_t mode
union
{
    struct
    {
        g_ftp
    }
    struct

```

```

    globus_gas_transfer_request_t request
  g gass
  structf
    globus_bool_t free_handle
    globus_bool_t seekable
  g io
  g data

```

4.6.1 Detailed Description

GASS copy target (e.g. source, destination) transfer information.

4.6.2 Field Documentation

4.6.2.1 `char globus_i_gasscopy_state.target.s::url`
url for the transfer

4.6.2.2 `globus_gasscopy_attr_t globus_i_gasscopy_state.target.s::attr`
attributes to control the transfer

4.6.2.3 `globus_mutex_t globus_i_gasscopy_state.target.s::mutex`
coordinates the modifying of the target structure

4.6.2.4 `globus_io_t globus_i_gasscopy_state.target.s::queue`
a queue to manage the reading/writing of data buffers

4.6.2.5 `int globus_i_gasscopy_state.target.s::n_pending`
Used for keeping track of reads/writes in the read/write queue.

4.6.2.6 `int globus_i_gasscopy_state.target.s::n_simultaneous`
Used to limit the number of pending.

4.6.2.7 `int globus_i_gasscopy_state.target.s::n_complete`
Used to compute the offset for ftp writes.

4.6.2.8 `globus_i_gasscopy_target_status_t globus_i_gasscopy_state.target.s::status`
signifies the target has been successfully setup

4.6.2.9 `globus_gasscopy_url_mode_t globus_i_gasscopy_state.target.s::mode`
mode used to identify the below target union struct.

4.6.2.10 `structf ... g globus.i_gasscopy_state.target.s::ftp`

ftp speci c data

4.6.2.11 `globusgasstransfer_request_t globus.i_gasscopy_state.target.s::request`

GASS equivalent of a handle.

4.6.2.12 `structf ... g globus.i_gasscopy_state.target.s::gass`

GASS speci c data.

4.6.2.13 `globusbool_t globus.i_gasscopy_state.target.s::free_handle`

If the IO handle was passed as an argument then FALSE If the IO handle was created internally then TRUE.

4.6.2.14 `globusbool_t globus.i_gasscopy_state.target.s::seekable`

Can `globusio_le_seek()` be performed on this handle?

4.6.2.15 `structf ... g globus.i_gasscopy_state.target.s::io`

IO speci c data.

4.6.2.16 `unionf ... g globus.i_gasscopy_state.target.s::data`

data required to perform each type of transfer

5 globus gass copy File Documentation

5.1 globusgasscopy.c File Reference

Globus GASS Copy library.

Functions

```

globus_result_t globusgasscopy\_handleinit (globusgasscopy_handle_t handle, globusgasscopy_
handle_attr_t attr)
globus_result_t globusgasscopy\_handledestroy(globusgasscopy_handle_t handle)
globus_result_t globusgasscopy\_setbuffer\_length(globusgasscopy_handle_t handle, int length)
globus_result_t globusgasscopy\_getbuffer\_length(globusgasscopy_handle_t handle, int length)
globus_result_t globusgasscopy\_setno\_third\_party\_transfers(globusgasscopy_handle_t handle, globus
bool_t no_third_party_transfers)
globus_result_t globusgasscopy\_getno\_third\_party\_transfers(globusgasscopy_handle_t handle, globus
bool_t no_third_party_transfers)
globus_result_t globusgasscopy\_setallocate(globusgasscopy_handle_t handle, globusbool_t sendallo)
globus_result_t globusgasscopy\_setpartialoffsets (globusgasscopy_handle_t handle, globusoff_t offset,
globusoff_t endoffset)
globus_result_t globusgasscopy\_getpartialoffsets(globusgasscopy_handle_t handle, globusoff_t offset,
globusoff_t endoffset)

```

```

globus_result_t globus_gasscopy_attr_init (globus_gasscopy_attr_t attr)
globus_result_t globus_gasscopy_attr_setftp (globus_gasscopy_attr_t attr, globusftp_client_operationattr_t ftp_attr)
globus_result_t globus_gasscopy_attr_setio (globus_gasscopy_attr_t attr, globusio_attr_t io_attr)
globus_result_t globus_gasscopy_attr_setgass (globus_gasscopy_attr_t attr, globusgasstransferrequestattr_t gassattr)
globus_result_t globus_gasscopy_get_url_mode (char url, globus_gasscopy_url_mode_t mode)
globus_result_t globus_gasscopy_register_performancecb (globus_gasscopy_handle_t handle, globus_gasscopy_performancecb_t callback, void userarg)
globus_result_t globus_gasscopy_get_status (globus_gasscopy_handle_t handle, globus_gasscopy_status_t status)
const char globus_gasscopy_get_statusstring (globus_gasscopy_handle_t handle)
globus_result_t globus_gasscopy_url_to_url (globus_gasscopy_handle_t handle, char sourceurl, globus_gasscopy_attr_t sourceattr, char desturl, globus_gasscopy_attr_t destattr)
globus_result_t globus_gasscopy_url_to_handle (globus_gasscopy_handle_t handle, char sourceurl, globus_gasscopy_attr_t sourceattr, globusio_handle_t desthandle)
globus_result_t globus_gasscopy_handleto_url (globus_gasscopy_handle_t handle, globusio_handle_t sourcehandle, char desturl, globus_gasscopy_attr_t destattr)
globus_result_t globus_gasscopy_registerurl_to_url (globus_gasscopy_handle_t handle, char sourceurl, globus_gasscopy_attr_t sourceattr, char desturl, globus_gasscopy_attr_t destattr, globus_gasscopy_callback_t callbackfunc, void callbackarg)
globus_result_t globus_gasscopy_registerurl_to_handle (globus_gasscopy_handle_t handle, char sourceurl, globus_gasscopy_attr_t sourceattr, globusio_handle_t desthandle, globus_gasscopy_callback_t callbackfunc, void callbackarg)
globus_result_t globus_gasscopy_registerhandleto_url (globus_gasscopy_handle_t handle, globusio_handle_t sourcehandle, char desturl, globus_gasscopy_attr_t destattr, globus_gasscopy_callback_t callbackfunc, void callbackarg)
globus_result_t globus_gasscopy_cacheurl_state (globus_gasscopy_handle_t handle, char url)
globus_result_t globus_gasscopy_ush_url_state (globus_gasscopy_handle_t handle, char url)
globus_result_t globus_gasscopy_setuserpointer (globus_gasscopy_handle_t handle, void userpointer)
globus_result_t globus_gasscopy_getuserpointer (globus_gasscopy_handle_t handle, void userdata)
globus_result_t globus_gasscopy_cancel (globus_gasscopy_handle_t handle, globus_gasscopy_callback_t cancelcallback, void cancelcallbackarg)
globus_result_t globus_l_gasscopy_targetcancel (globus_l_gasscopy_cancel_t cancelinfo)

```

5.1.1 Detailed Description

Globus GASS Copy library.

See also:

See the detailed description of [globusgasscopy.h](#)

5.1.2 Function Documentation

5.1.2.1 `globus_result_t globus_gasscopy_handle_init (globus_gasscopy_handle_t handle, globus_gasscopy_handleattr_t attr)`

Initialize a GASS Copy handle.

A `globus_gasscopy_handle` must be initialized before any transfers may be associated with it. This function initializes a `globus_gasscopy_handle` to be used for doing transfers, this includes initializing a `globusio_client_handle` which

will be used for doing any ftp/gsiftp transfers. The same handle may be used to perform multiple, consecutive transfers. However, there can only be one transfer associated with a particular handle at any given time. After all transfers to be associated with this handle have completed, the handle should be destroyed by calling [globusgasscopy_handle_destroy\(\)](#)

Parameters:

handle The handle to be initialized
attr The handle attributes used to use with this handle

Returns:

This function returns GLOBUS_SUCCESS if successful, or a globus_result_t indicating the error that occurred.

See also:

[globusgasscopy_handledestroy\(\)](#), [globusgasscopy_handleattrinit\(\)](#), [globusftp_client_handleinit\(\)](#)

5.1.2.2 globus_result_t globus_gasscopy_handle_destroy (globus_gasscopy_handle_t handle)

Destroy a GASS Copy handle.

Destroy a [gasscopy_handle](#), which was initialized using [globusgasscopy_handleinit\(\)](#), that will no longer be used for doing transfers. Once the handle is destroyed, no further transfers should be associated with it.

Parameters:

handle The handle to be destroyed

Returns:

This function returns GLOBUS_SUCCESS if successful, or a globus_result_t indicating the error that occurred.

See also:

[globusgasscopy_handleinit\(\)](#), [globusftp_client_handledestroy\(\)](#)

5.1.2.3 globus_result_t globus_gasscopy_setbuffer_length (globus_gasscopy_handle_t handle, int length)

Set the size of the buffer to be used for doing transfers.

This function allows the user to set the size of the buffer that will be used for doing transfers, if this function is not called the buffer size will default to 1M.

Parameters:

handle Set the buffer length for transfers associated with this handle.
length The length, in bytes, to make the buffer.

Returns:

This function returns GLOBUS_SUCCESS if successful, or a globus_result_t indicating the error that occurred.

5.1.2.4 globus_result_t globus_gasscopy_getbuffer_length (globus_gasscopy_handle_t handle, int length)

Get the size of the buffer being used for doing transfers.

This function allows the user to get the size of the buffer that is being used for doing transfers.

Parameters:

- handle Get the buffer length for transfers associated with this handle.
- length The length, in bytes, of the buffer.

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globus_result_t` indicating the error that occurred.

5.1.2.5 `globus_result_t globus_gasscopy_set_no_third_party_transfers (globus_gasscopy_handle_t handle, globus_bool_t no_third_party_transfers)`

Turn third-party transfers on or off.

(They are on by default.)

This function allows the user to turn third-party transfers on or off for ftp to ftp transfers associated with a particular handle. This is often desired if one of the servers involved in the transfer does not allow third-party transfers.

Parameters:

- handle Turn third-party transfers on or off for transfers associated with this handle. They are on by default.
- no_third_party_transfers `GLOBUS_FALSE` if third-party transfers should be used. `GLOBUS_TRUE` if third-party transfers should not be used.

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globus_result_t` indicating the error that occurred.

5.1.2.6 `globus_result_t globus_gasscopy_get_no_third_party_transfers (globus_gasscopy_handle_t handle, globus_bool_t no_third_party_transfers)`

See if third-party transfers are turned on or off.

(They are on by default.)

This function allows the user to see if third-party transfers are turned on or off for ftp to ftp transfers associated with a particular handle. This is often desired if one of the servers involved in the transfer does not allow third-party transfers.

Parameters:

- handle See if third-party transfers are turned on or off for transfers associated with this handle. They are on by default.
- no_third_party_transfers `GLOBUS_FALSE` if third-party transfers should be used. `GLOBUS_TRUE` if third-party transfers should not be used.

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globus_result_t` indicating the error that occurred.

5.1.2.7 `globus_result_t globus_gasscopy_set_allocate (globus_gasscopy_handle_t handle, globus_bool_t send_allo)`

Set allo on or off.

5.1.2.8 `globusresult_t globus_gasscopy_set_partial_offsets (globusgasscopy_handle_t handle, globus_off_t offset, globus_off_t end_offset)`

Set the offsets to be used for doing partial transfers.

This function allows the user to set the offsets that will be used for doing partial transfers. An offset of -1 will disable partial transfers. An end offset of -1 means EOF.

Parameters:

handle Set the offsets for partial transfers associated with this handle.

offset The starting offset for the partial transfer.

end_offset The ending offset for the partial transfer.

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globusresult_t` indicating the error that occurred.

5.1.2.9 `globusresult_t globus_gasscopy_get_partial_offsets (globusgasscopy_handle_t handle, globus_off_t offset, globus_off_t end_offset)`

Get the offsets being used for doing partial transfers.

This function allows the user to get the offsets that are being used for doing partial transfers. An offset of -1 means partial transfers are disabled.

Parameters:

handle Get the offsets for partial transfers associated with this handle.

offset The starting offset for the partial transfer.

end_offset The ending offset for the partial transfer.

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globusresult_t` indicating the error that occurred.

5.1.2.10 `globusresult_t globus_gasscopy_attr_init (globus_gasscopy_attr_t attr)`

Initialize an attribute structure.

The `globus_gasscopy_attr_t` can be used to pass the `globus_gasscopy` library information about how a transfer should be performed. It must first be initialized by calling this function. Then any or all of the following functions may be called to set attributes associated with a particular protocol: `globus_gasscopy_attr_setftp()`, `globus_gasscopy_attr_setgass()`, `globus_gasscopy_attr_setio()`. Any function which takes a `globus_gasscopy_attr_t` as an argument will also accept `GLOBUS_NULL`, in which case the appropriate set of default attributes will be used.

Parameters:

attr The attribute structure to be initialized

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globusresult_t` indicating the error that occurred.

See also:

[globus_gasscopy_attr_setftp\(\)](#), [globus_gasscopy_attr_setgass\(\)](#), [globus_gasscopy_attr_setio\(\)](#), [globus_gasscopy_get_url_mode\(\)](#)

5.1.2.11 `globus_result_t globus_gasscopy_attr_setftp (globus_gasscopy_attr_t attr, globus_ftp_client_operationattr_t ftp_attr)`

Set the attributes for ftp/gsftp transfers.

In order to specify attributes for ftp/gsftp transfers, a `globus_ftp_client_operationattr_t` should be initialized and its values set using the appropriate `globus_ftp_client_operationattr` functions. The `globus_ftp_client_operationattr_t` can then be passed to the `globus_gasscopy_attr_t` via this function.

Parameters:

attr A globus.gasscopy attribute structure

ftp_attr The ftp/gsftp attributes to be used

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globus_result_t` indicating the error that occurred.

See also:

[globus_gasscopy_attr_init\(\)](#), [globus_gasscopy_attr_setgass\(\)](#), [globus_gasscopy_attr_setio\(\)](#), [globus_gasscopy_get_url_mode\(\)](#) `globus_ftp_client_operationattr`

5.1.2.12 `globus_result_t globus_gasscopy_attr_setio (globus_gasscopy_attr_t attr, globus_io_attr_t io_attr)`

Set the attributes for le transfers.

In order to specify attributes for le transfers, a `globus_io_attr_t` should be initialized and its values set using the appropriate `globus_io_attr` functions. The `globus_io_attr_t` can then be passed to the `globus_gasscopy_attr_t` via this function.

Parameters:

attr A globus.gasscopy attribute structure

io_attr The le attributes to be used

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globus_result_t` indicating the error that occurred.

See also:

[globus_gasscopy_attr_init\(\)](#), [globus_gasscopy_attr_setgass\(\)](#), [globus_gasscopy_attr_setftp\(\)](#), [globus_gasscopy_get_url_mode\(\)](#) `globusio_attr`

5.1.2.13 `globus_result_t globus_gasscopy_attr_setgass (globus_gasscopy_attr_t attr, globus_gasstransfer_requestattr_t gassattr)`

Set the attributes for http/https transfers.

In order to specify attributes for http/https transfers, a `globus_gasstransfer_requestattr_t` should be initialized and its values set using the appropriate `globus_gasstransfer_requestattr` functions. The `globus_gasstransfer_requestattr_t` can then be passed to the `globus_gasscopy_attr_t` via this function.

Parameters:

attr A globus.gasscopy attribute structure

gassattr The http/https attributes to be used

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globus_result_t` indicating the error that occurred.

See also:

[globusgasscopy_attr_init\(\)](#), [globusgasscopy_attr_setio\(\)](#), [globusgasscopy_attr_setftp\(\)](#), [globusgasscopy_get_url_mode\(\)](#) [globusgasstransferrequestattr](#)

5.1.2.14 `globus_result_t globus_gasscopy_get_url_mode (char url, globus_gasscopy_url_mode_t mode)`

Classify the URL schema into the transfer method that will be used to do the actual transfer.

This function enables the user to determine what protocol will be used to transfer data to/from a particular url. This information can then be used to specify the appropriate attributes when initiating a transfer.

Parameters:

url The URL for schema checking
mode the lled in schema type of the URL param

Returns:

This function returns `GLOBUS_SUCCESS` if successful, or a `globus_result_t` indicating the error that occurred.

See also:

[globusgasscopy_attr_init\(\)](#), [globusgasscopy_attr_setio\(\)](#), [globusgasscopy_attr_setftp\(\)](#), [globusgasscopy_setgass\(\)](#)

5.1.2.15 `globus_result_t globus_gasscopy_register_performance_cb (globus_gasscopy_handle_t handle, globus_gasscopy_performance_cb_t callback, void user_arg)`

Register a performance information callback.

Use this to register a performance information callback. You change or set to `GLOBUS` the callback any time a transfer is not occurring.

Parameters:

handle an initialized gass copy handle for which you would like to see performance info
callback the performance callback
user_arg a user pointer that will be passed to all callbacks for a given handle

Returns:

`GLOBUS_SUCCESS`
error on a NULL or busy handle

See also:

[globusgasscopy_performancecb_t](#)

5.1.2.16 `globus_result_t globus_gasscopy_get_status (globusgasscopy_handle_t handle, globus_gasscopy_status_t status)`

Get the status code of the current transfer.

Get the status of the last transfer to be initiated using the given handle. Only one transfer can be active on a handle at a given time, therefore new transfers may only be initiated when the current status is one of the following: `GLOBUS_GASS_COPY_STATUS_NONE`, `GLOBUS_GASS_COPY_STATUS_DONE_SUCCESS`, `GLOBUS_GASS_COPY_STATUS_DONE_FAILURE`, `GLOBUS_GASS_COPY_STATUS_DONE_CANCELLED`

Parameters:

handle A globus.gasscopy.handle

status Will be one of the following: GLOBUSGASSCOPY_STATUS_NONE (No transfers have been initiated using this handle.) GLOBUSGASSCOPY_STATUS_PENDING (A transfer is currently being set up.) GLOBUSGASSCOPY_STATUS_TRANSFER_IN_PROGRESS (There is currently a transfer in progress.) GLOBUSGASSCOPY_STATUS_CANCEL (The last transfer initiated using this handle has been cancelled by the user before completing, and is in the process of being cleaned up.) GLOBUSGASSCOPY_STATUS_FAILURE (The last transfer initiated using this handle failed, and is in the process of being cleaned up.) GLOBUSGASSCOPY_STATUS_DONE_SUCCESS (The last transfer initiated using this handle has completed successfully.) GLOBUSGASSCOPY_STATUS_DONE_FAILURE (The last transfer initiated using this handle failed and has finished cleaning up.) GLOBUSGASSCOPY_STATUS_DONE_CANCELLED (The last transfer initiated using this handle was cancelled and has finished cleaning up.)

Returns:

This function returns GLOBUS_SUCCESS if successful, or a globus_result_t indicating the error that occurred.

5.1.2.17 const char globus_gasscopy_get_status_string (globus_gasscopy_handle_t handle)

Get the status string of the current transfer.

Get the status of the last transfer to be initiated using the given handle. Only one transfer can be active on a handle at a given time, therefore new transfers may only be initiated when the current status is one of the following: GLOBUSGASSCOPY_STATUS_NONE, GLOBUSGASSCOPY_STATUS_DONE_SUCCESS, GLOBUSGASSCOPY_STATUS_DONE_FAILURE, GLOBUSGASSCOPY_STATUS_DONE_CANCELLED

Parameters:

handle A globus.gasscopy.handle

Returns:

Returns a pointer to a character string describing the current status

5.1.2.18 globus_result_t globus_gasscopy_url_to_url (globus_gasscopy_handle_t handle, char sourceurl, globus_gasscopy_attr_t sourceattr, char desturl, globus_gasscopy_attr_t destattr)

Transfer data from source URL to destination URL (blocking).

Parameters:

handle The handle to perform the copy operation

sourceurl transfer data from this URL

sourceattr Attributes describing how the transfer from the source should be done

desturl transfer data to this URL

destattr Attributes describing how the transfer to the destination should be done

Returns:

This function returns GLOBUS_SUCCESS if the transfer was completed successfully, or a result pointing to an object of one of the the following error types:

Return values:

GLOBUSGASSCOPY_ERROR_TYPE_NULL_PARAMETER The handle was equal to GLOBUS_NULL, so the transfer could not processed.

GLOBUSGASSCOPY_ERROR_TYPE_next_error next error description

See also:

[globus_gasscopy_url_to_handle\(\)](#)[globus_gasscopy_handle_to_url\(\)](#)

5.1.2.19 `globus_result_t globus_gasscopy_url_to_handle (globus_gasscopy_handle_t handle, char source-url, globus_gasscopy_attr_t sourceattr, globus_io_handle_t desthandle)`

Transfer data from source URL to an IO handle (blocking).

Parameters:

handle The handle to perform the copy operation
 sourceurl transfer data from this URL
 sourceattr Attributes describing how the transfer from the source should be done
 desthandle transfer data to this IO handle

Returns:

This function returns `GLOBUS_SUCCESS` if the transfer was completed successfully, or a result pointing to an object of one of the the following error types:

Return values:

`GLOBUS_GASSCOPY_ERROR_TYPE_NULL_PARAMETER` The handle was equal to `GLOBUS_NULL`, so the transfer could not processed.
`GLOBUS_GASSCOPY_ERROR_TYPE_next_error` next error description

See also:

[globus_gasscopy_url_to_url\(\)](#) [globus_gasscopy_handleto_url\(\)](#)

5.1.2.20 `globus_result_t globus_gasscopy_handle_to_url (globus_gasscopy_handle_t handle, globus_io_handle_t sourcehandle, char desturl, globus_gasscopy_attr_t destattr)`

Transfer data from an IO handle to destination URL (blocking).

Parameters:

handle The handle to perform the copy operation
 sourcehandle transfer data from this IO handle
 desturl transfer data to this URL
 destattr Attributes describing how the transfer to the destination should be done

Returns:

This function returns `GLOBUS_SUCCESS` if the transfer was completed successfully, or a result pointing to an object of one of the the following error types:

Return values:

`GLOBUS_GASSCOPY_ERROR_TYPE_NULL_PARAMETER` The handle was equal to `GLOBUS_NULL`, so the transfer could not processed.
`GLOBUS_GASSCOPY_ERROR_TYPE_next_error` next error description

See also:

[globus_gasscopy_url_to_url\(\)](#) [globus_gasscopy_url_to_handle\(\)](#)

5.1.2.21 `globus_result_t globus_gasscopy_register_url_to_url (globus_gasscopy_handle_t handle, char sourceurl, globus_gasscopy_attr_t sourceattr, char desturl, globus_gasscopy_attr_t destattr, globus_gasscopy_callback_t callbackfunc, void callbackarg)`

Transfer data from source URL to destination URL (non-blocking).

This functions initiates a transfer from source URL to destination URL, then returns immediately.

When the transfer is completed or if the transfer is aborted, the `callback` will be invoked with the final status of the transfer.

Parameters:

`handle` The handle to perform the copy operation
`sourceurl` transfer data from this URL
`sourceattr` Attributes describing how the transfer form the source should be done
`desturl` transfer data to this URL
`destattr` Attributes describing how the transfer to the destination should be done
`callbackfunc` Callback to be invoked once the transfer is completed.
`callbackarg` Argument to be passed to the `callback`.

Returns:

This function returns `GLOBUS_SUCCESS` if the transfer was initiated successfully, or a result pointing to an object of one of the the following error types:

Return values:

`GLOBUS_GASSCOPY_ERROR_TYPE_NULL_PARAMETER` The handle was equal to `GLOBUS_NULL`, so the transfer could not processed.
`GLOBUS_GASSCOPY_ERROR_TYPE_next_error` next error description

See also:

[globusgasscopy_registerurl_to_handle\(\)](#) [globusgasscopy_registerhandleto_url\(\)](#)

5.1.2.22 `globus_result_t globus_gasscopy_register_url_to_handle (globusgasscopy_handle_t handle, char sourceurl, globus_gasscopy_attr_t sourceattr, globus_io_handle_t desthandle, globus_gasscopy_callback_t callbackfunc, void callbackarg)`

Transfer data from source URL to an IO handle (non-blocking).

This functions initiates a transfer from source URL to an IO handle, then returns immediately.

When the transfer is completed or if the transfer is aborted, the `callback` will be invoked with the final status of the transfer.

Parameters:

`handle` The handle to perform the copy operation
`sourceurl` transfer data from this URL
`sourceattr` Attributes describing how the transfer form the source should be done
`desthandle` transfer data to this IO handle
`callbackfunc` Callback to be invoked once the transfer is completed.
`callbackarg` Argument to be passed to the `callback`.

Returns:

This function returns `GLOBUS_SUCCESS` if the transfer was initiated successfully, or a result pointing to an object of one of the the following error types:

Return values:

`GLOBUS_GASSCOPY_ERROR_TYPE_NULL_PARAMETER` The handle was equal to `GLOBUS_NULL`, so the transfer could not processed.

`GLOBUS_GASSCOPY_ERROR_TYPE_next_error` next error description

See also:

[globus.gasscopy.registerurl.to.url\(\)](#), [globus.gasscopy.registerhandleto.url\(\)](#)

5.1.2.23 `globus_result_t globus_gasscopy_register_handle_to_url(globus_gasscopy_handle_t handle, globus_io_handle_t sourcehandle, char desturl, globus_gasscopy_attr_t destattr, globus_gasscopy_callback_t callbackfunc, void callbackarg)`

Transfer data from an IO handle to destination URL (non-blocking).

This functions initiates a transfer from an IO handle to destination URL, then returns immediately.

When the transfer is completed or if the transfer is aborted, the `callback` will be invoked with the final status of the transfer.

Parameters:

`handle` The handle to perform the copy operation

`sourcehandle` transfer data from this IO handle

`desturl` transfer data to this URL

`destattr` Attributes describing how the transfer to the destination should be done

`callbackfunc` Callback to be invoked once the transfer is completed.

`callbackarg` Argument to be passed to the `callbackfunc`.

Returns:

This function returns `GLOBUS_SUCCESS` if the transfer was initiated successfully, or a result pointing to an object of one of the the following error types:

Return values:

`GLOBUS_GASSCOPY_ERROR_TYPE_NULL_PARAMETER` The handle was equal to `GLOBUS_NULL`, so the transfer could not processed.

`GLOBUS_GASSCOPY_ERROR_TYPE_next_error` next error description

See also:

[globus.gasscopy.registerurl.to.url\(\)](#), [globus.gasscopy.registerurl.to.handle\(\)](#)

5.1.2.24 `globus_result_t globus_gasscopy_cacheurl_state(globusgasscopy_handle_t handle, char url)`

Cache connections to an FTP or GSIFTP server.

Explicitly cache connections to URL server. When an URL is cached, the connection to the URL server will not be closed after a file transfer completes.

Parameters:

`handle` Handle which will contain a cached connection to the URL server.

url The URL of the FTP or GSIFTP server to cache.

Returns:

This function returns GLOBUS_SUCCESS if successful, or a globus_result_t indicating the error that occurred.

5.1.2.25 globus_result_t globus_gasscopy_usch_url_state (globus_gasscopy_handle_t handle, char url)

Remove a cached connection to an FTP or GSIFTP server.

Explicitly remove a cached connection to an FTP or GSIFTP server. If an idle connection to an FTP server exists, it will be closed.

Parameters:

handle Handle which contains a cached connection to the URL server.

url The URL of the FTP or GSIFTP server to remove.

Returns:

This function returns GLOBUS_SUCCESS if successful, or a globus_result_t indicating the error that occurred.

5.1.2.26 globus_result_t globus_gasscopy_set_user_pointer (globus_gasscopy_handle_t handle, void user_pointer)

Set a pointer in the handle to point at user-allocated memory.

5.1.2.27 globus_result_t globus_gasscopy_get_user_pointer (globus_gasscopy_handle_t handle, void user_data)

Get the pointer in the handle that points to user-allocated memory.

5.1.2.28 globus_result_t globus_gasscopy_cancel (globus_gasscopy_handle_t handle, globus_gasscopy_callback_t cancel_callback, void cancel_callback_arg)

Cancel the current transfer associated with this handle,.

5.1.2.29 globus_result_t globus_i_gasscopy_target_cancel (globus_i_gasscopy_cancel_t cancelinfo)

Cancel the source or destination transfer in progress.

5.2 globusgasscopy.h File Reference

Header file for the gass copy library.

Data Structures

struct [globus_gasscopy_glob_stat_t](#)

Glob expanded entry information.

Defines

#define [GLOBUS_GASSCOPY_MODULE](#) (&globus_i_gasscopy_module)

Typedefs

```
typedef void( globusgasscopy_performance_cb_t )(void userarg, globusgasscopy_handle_t handle,
globus_off_t total_bytes, oat instantaneous_throughput, oat avg_throughput)
typedef void( globusgasscopy_glob_entry_cb_t )(const char url, const globusgasscopy_glob_stat_t info_
stat, void userarg)
```

Enumerations

```
enum globusgasscopy_glob_entry_t
```

Functions

```
globus_result_t globusgasscopy_glob_expandurl (globusgasscopy_handle_t handle, const char url,
globusgasscopy_attr_t attr, globusgasscopy_glob_entry_cb_t entry_cb, void userarg)
globus_result_t globusgasscopy_mkdir (globusgasscopy_handle_t handle, char url, globusgasscopy_attr_t
attr)
```

5.2.1 Detailed Description

Header file for the gass copy library.

5.2.2 Define Documentation

5.2.2.1 #define GLOBUSGASS_COPY_MODULE (&globus_i_gasscopy_module)

Module descriptor.

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

```
globus_module_activate(GLOBUS_GASS_COPY_MODULE)
```

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

To deactivate Globus GASS Copy, the following function must be called:

```
globus_module_deactivate(GLOBUS_GASS_COPY_MODULE)
```

This function should be called once for each time Globus GASS Copy was activated.

5.2.2.2 #define GASC_SL(s)

Value:

```
globus_common_i18n_get_string( \
    GLOBUS_GASS_COPY_MODULE, \
    s)
```

5.2.3 Typedef Documentation

5.2.3.1 `typedef void(globus_gasscopy_performance_cb_t)(void user_arg, globus_gasscopy_handle_t handle, globus_off_t total_bytes, oat instantaneousthroughput, oat avg_throughput)`

Gass copy transfer performance callback.

This callback is registered with 'globus_gasscopy_registerperformancecb'. It will be called during a transfer to supply performance information on current transfer. Its frequency will be at most one per second, but it is possible to receive no callbacks. This is possible in very short transfers and in ftp transfers in which the server does not provide performance information.

Parameters:

handle the gass copy handle this transfer is occurring on
 user_arg a user pointer registered with 'globus_gasscopy_registerperformancecb'
 total_bytes the total number of bytes transfer so far
 instantaneousthroughput instantaneous rate of transfer (since last callback or start) (bytes / sec)
 avg_throughput the avg thoughput calculated since the start of the transfer (bytes / sec)

Returns:

n/a

5.2.3.2 `typedef void(globus_gasscopy_glob_entry_cb_t)(const char url, const globus_gasscopy_glob_stat_t info_stat, void user_arg)`

Gass copy glob entry callback.

This callback is passed as a parameter to [globus_gasscopy_glob_expandurl\(\)](#). It is called once for each entry that the original expands to.

Parameters:

url The full url to the expanded entry. A directory entry will end in a forward slash '/'.
 stat A pointer to a [globus_gasscopy_glob_stat_t](#) containing information about the entry.
 user_arg The userarg passed to [globus_gasscopy_glob_expand\(\)](#)

See also:

[globus_gasscopy_glob_stat_t](#), [globus_gasscopy_glob_expandurl](#)

5.2.4 Enumeration Type Documentation

5.2.4.1 `enum globus_gasscopy_glob_entry_t`

globbed entry types

5.2.5 Function Documentation

5.2.5.1 `globusresult_t globus_gasscopy_glob_expandurl (globus_gasscopy_handle_t handle, const char url, globus_gasscopy_attr_t attr, globus_gasscopy_glob_entry_cb_t entry_cb, void user_arg)`

Expand globbed url.

This function expands wildcards in a globbed url, and calls [entry_cb](#) on each one.

Parameters:

- handle A gass copy handle to use for the operation.
- url The URL to expand. The URL may be an ftp, gsiftp or le URL. Wildcard characters supported are '?' ']' in the lename portion of the url.
- attr Gass copy attributes for this operation.
- entry.cb Function to call with information about each entry
- user.arg An argument to pass to entry.cb()

Returns:

This function returns an error when any of these conditions are true:

- handle is GLOBUSNULL
- url is GLOBUSNULL
- url cannot be parsed
- url is not a ftp, gsiftp or le url

5.2.5.2 globusresult_t globus_gasscopy_mkdir (globus_gasscopy_handle_t handle, char url, globus_gasscopy_attr_t attr)

Make directory.

This function creates a directory given a ftp or le url.

Parameters:

- handle A gass copy handle to use for the mkdir operation.
- url The URL for the directory to create. The URL may be an ftp, gsiftp or le URL.
- attr Gass copy attributes for this operation.

Returns:

This function returns an error when any of these conditions are true:

- handle is GLOBUSNULL
- url is GLOBUSNULL
- url cannot be parsed
- url is not a ftp, gsiftp or le url
- the directory could not be created

Index

- [_GASCSL](#)
 - [globus.gasscopy.h, 18](#)
 - [attr](#)
 - [globus.i_gasscopy.statetargets, 5](#)
 - [cancel](#)
 - [globus.gasscopy.states, 3](#)
 - [data](#)
 - [globus.i_gasscopy.statetargets, 6](#)
 - [dest](#)
 - [globus.gasscopy.states, 3](#)
 - [free.handle](#)
 - [globus.i_gasscopy.statetargets, 6](#)
 - [ftp](#)
 - [globus.i_gasscopy.statetargets, 5](#)
 - [gass](#)
 - [globus.i_gasscopy.statetargets, 6](#)
 - [globus.gasscopy.c, 6](#)
 - [globus.gasscopy.attr.init, 10](#)
 - [globus.gasscopy.attr.setftp, 10](#)
 - [globus.gasscopy.attr.setgass, 11](#)
 - [globus.gasscopy.attr.setio, 11](#)
 - [globus.gasscopy.cacheurl.state, 16](#)
 - [globus.gasscopy.cancel, 17](#)
 - [globus.gasscopy._ush_url.state, 17](#)
 - [globus.gasscopy.getbuffer.length, 8](#)
 - [globus.gasscopy.getno.third.party.transfers, 9](#)
 - [globus.gasscopy.getpartial.offsets, 10](#)
 - [globus.gasscopy.getstatus, 12](#)
 - [globus.gasscopy.getstatusstring, 13](#)
 - [globus.gasscopy.geturl.mode, 12](#)
 - [globus.gasscopy.getuserpointer, 17](#)
 - [globus.gasscopy.handledestroy, 8](#)
 - [globus.gasscopy.handleinit, 7](#)
 - [globus.gasscopy.handleto.url, 14](#)
 - [globus.gasscopy.registerhandleto.url, 16](#)
 - [globus.gasscopy.registerperformancecb, 12](#)
 - [globus.gasscopy.registerurl.to.handle, 15](#)
 - [globus.gasscopy.registerurl.to.url, 14](#)
 - [globus.gasscopy.setallocate, 9](#)
 - [globus.gasscopy.setbuffer.length, 8](#)
 - [globus.gasscopy.setno.third.party.transfers, 9](#)
 - [globus.gasscopy.setpartial.offsets, 9](#)
 - [globus.gasscopy.setuserpointer, 17](#)
 - [globus.gasscopy.url.to.handle, 13](#)
 - [globus.gasscopy.url.to.url, 13](#)
 - [globus.l_gasscopy.targetcancel, 17](#)
 - [globus.gasscopy.h, 17](#)
 - [_GASCSL, 18](#)
 - [globus.gasscopy.glob.entry.cb.t, 19](#)
 - [globus.gasscopy.glob.entry.t, 19](#)
 - [globus.gasscopy.glob.expandurl, 19](#)
 - [globus.gasscopy.mkdir, 20](#)
 - [GLOBUS.GASS.COPY.MODULE, 18](#)
 - [globus.gasscopy.performancecb.t, 19](#)
 - [globus.gasscopy.attr.init](#)
 - [globus.gasscopy.c, 10](#)
 - [globus.gasscopy.attr.setftp](#)
 - [globus.gasscopy.c, 10](#)
 - [globus.gasscopy.attr.setgass](#)
 - [globus.gasscopy.c, 11](#)
 - [globus.gasscopy.attr.setio](#)
 - [globus.gasscopy.c, 11](#)
 - [globus.gasscopy.cacheurl.state](#)
 - [globus.gasscopy.c, 16](#)
 - [globus.gasscopy.cancel](#)
 - [globus.gasscopy.c, 17](#)
 - [globus.gasscopy._ush_url.state](#)
 - [globus.gasscopy.c, 17](#)
 - [globus.gasscopy.getbuffer.length](#)
 - [globus.gasscopy.c, 8](#)
 - [globus.gasscopy.getno.third.party.transfers](#)
 - [globus.gasscopy.c, 9](#)
 - [globus.gasscopy.getpartial.offsets](#)
 - [globus.gasscopy.c, 10](#)
 - [globus.gasscopy.getstatus](#)
 - [globus.gasscopy.c, 12](#)
 - [globus.gasscopy.getstatusstring](#)
 - [globus.gasscopy.c, 13](#)
 - [globus.gasscopy.geturl.mode](#)
 - [globus.gasscopy.c, 12](#)
 - [globus.gasscopy.getuserpointer](#)
 - [globus.gasscopy.c, 17](#)
 - [globus.gasscopy.glob.entry.cb.t](#)
 - [globus.gasscopy.h, 19](#)
 - [globus.gasscopy.glob.entry.t](#)
 - [globus.gasscopy.h, 19](#)
 - [globus.gasscopy.glob.expandurl](#)
 - [globus.gasscopy.h, 19](#)
 - [globus.gasscopy.glob.statt, 2](#)
 - [mdtm, 3](#)
 - [mode, 2](#)
 - [size, 3](#)
 - [symlink.target, 2](#)
 - [type, 2](#)
 - [uniqueid, 2](#)
 - [globus.gasscopy.handledestroy](#)
-

- globus.gasscopy.c, 8
- globus.gasscopy.handleinit
 - globus.gasscopy.c, 7
- globus.gasscopy.handleto_url
 - globus.gasscopy.c, 14
- globus.gasscopy.mkdir
 - globus.gasscopy.h, 20
- GLOBUS.GASSCOPY.MODULE
 - globus.gasscopy.h, 18
- globus.gasscopy.performancecb.t
 - globus.gasscopy.h, 19
- globus.gasscopy.registerhandleto_url
 - globus.gasscopy.c, 16
- globus.gasscopy.registerperformancecb
 - globus.gasscopy.c, 12
- globus.gasscopy.registerurl_to_handle
 - globus.gasscopy.c, 15
- globus.gasscopy.registerurl_to_url
 - globus.gasscopy.c, 14
- globus.gasscopy.setallocate
 - globus.gasscopy.c, 9
- globus.gasscopy.setbuffer.length
 - globus.gasscopy.c, 8
- globus.gasscopy.setno_third_party_transfers
 - globus.gasscopy.c, 9
- globus.gasscopy.setpartialoffsets
 - globus.gasscopy.c, 9
- globus.gasscopy.setuserpointer
 - globus.gasscopy.c, 17
- globus.gasscopy.states, 3
 - cancel, 3
 - dest, 3
 - monitor, 3
 - mutex, 3
 - source, 3
- globus.gasscopy.url_to_handle
 - globus.gasscopy.c, 13
- globus.gasscopy.url_to_url
 - globus.gasscopy.c, 13
- globusi_gasscopy.buffer.t, 4
- globusi_gasscopy.cancels, 4
- globusi_gasscopy.monitor.t, 4
- globusi_gasscopy.statetargets, 4
 - attr, 5
 - data, 6
 - free.handle, 6
 - ftp, 5
 - gass, 6
 - io, 6
 - mode, 5
 - mutex, 5
 - n_complete, 5
 - n_pending, 5
- n_simultaneous, 5
- queue, 5
- request, 6
- seekable, 6
- status, 5
- url, 5
- globusi_gasscopy_targetcancel
 - globus.gasscopy.c, 17
- io
 - globusi_gasscopy.statetargets, 6
- mdtm
 - globus.gasscopy_glob_statt, 3
- mode
 - globus.gasscopy_glob_statt, 2
 - globusi_gasscopy.statetargets, 5
- monitor
 - globus.gasscopy.states, 3
- mutex
 - globus.gasscopy.states, 3
 - globusi_gasscopy.statetargets, 5
- n_complete
 - globusi_gasscopy.statetargets, 5
- n_pending
 - globusi_gasscopy.statetargets, 5
- n_simultaneous
 - globusi_gasscopy.statetargets, 5
- queue
 - globusi_gasscopy.statetargets, 5
- request
 - globusi_gasscopy.statetargets, 6
- seekable
 - globusi_gasscopy.statetargets, 6
- size
 - globus.gasscopy_glob_statt, 3
- source
 - globus.gasscopy.states, 3
- status
 - globusi_gasscopy.statetargets, 5
- symlink.target
 - globus.gasscopy_glob_statt, 2
- type
 - globus.gasscopy_glob_statt, 2
- uniqueid
 - globus.gasscopy_glob_statt, 2
- url
 - globusi_gasscopy.statetargets, 5