

globus gss assist Reference Manual

4.0

Generated by Doxygen 1.2.18

Fri Jun 26 15:39:15 2009

Contents

1 Globus GSI GSS Assist	1
2 globus gss assist Module Index	1
3 globus gss assist Module Documentation	1

1 Globus GSI GSS Assist

The GSS Assist code provides convenience functions for using the Globus GSS-API.

2 globus gss assist Module Index

2.1 globus gss assist Modules

Here is a list of all modules:

Activation	1
Utility Functions	2
GSI GSS Assist Constants	10
Security Token Transport	11

3 globus gss assist Module Documentation

3.1 Activation

Globus GSI GSS Assist uses standard Globus module activation and deactivation.

De nes

```
#define GLOBUS_GSI_GSSASSIST_MODULE
```

3.1.1 Detailed Description

Globus GSI GSS Assist uses standard Globus module activation and deactivation.

Before any Globus GSS Assist functions are called, the following function must be called:

```
globus_module_activate(GLOBUS_GSI_GSS_ASSIST_MODULE);
```

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

To deactivate Globus GSS Assist, the following function must be called:

```
globus_module_deactivate(GLOBUS_GSI_GSS_ASSIST_MODULE)
```

This function should be called once for each time Globus GSI GSS Assist was activated.

3.1.2 De ne Documentation

3.1.2.1 #de ne GLOBUS_GSI_GSS_ASSIST_MODULE

Module descriptor.

3.2 Utility Functions

Utility functions for GSSAPI.

Accept Security Context

```
OM_uint32 globus_gssassistacceptseccontext(OM_uint32 minor_status, gssctx_id_t contexthandle, const
gsscred_id_t credhandle, char src_namechar, OM_uint32 ret_ags, int user_to_user_ag, int token_status,
gsscred_id_t delegated_credhandle, int(gssassist_get_token)(void *, void *, size_t), void gssassist_get-
context, int(gssassist_send_token)(void *, void *, size_t), void gssassist_sendcontext)
```

Accept Security Context Asynchronous

```
OM_uint32 globus_gssassistacceptseccontextasync(OM_uint32 minor_status, gssctx_id_t context-
handle, const gsscred_id_t credhandle, char src_namechar, OM_uint32 ret_ags, int user_to_user_ag,
void input_buffer, size_t input_buffer_len, void output_bufferp, size_t output_buffer_lenp, gsscred_id_-
t delegated_credhandle)
```

Acquire Credential

```
OM_uint32 globus_gssassistacquirecred(OM_uint32 minor_status, gsscred_usage_t cred_usage, gsscred_-
id_t output_credhandle)
```

Acquire Credential Extension

```
OM_uint32 globus_gssassistacquirecred_ext(OM_uint32 minor_status, char desired_namechar, OM_-
uint32 time_req, const gss_OID_set desired_mechs, gsscred_usage_t cred_usage, gsscred_id_t output_cred-
handle, gss_OID_set actual_mechs, OM_uint32 time_rec)
```

Display Status

```
OM_uint32 globus_gssassistdisplay_status(FILE fp, char comment, OM_uint32 major_status, OM_uint32
minor_status, int token_status)
```

Display Status String

```
OM_uint32 globus_gssassistdisplay_statusstr(char str, char comment, OM_uint32 major_status, OM_-
uint32 minor_status, int token_status)
```

De nes

```
#define NI_MAXHOST 255
```

Functions

```
int globus_gssassistgridmap(char globusidp, char useridp)
int globus_gssassistuserok(char globusid, char userid)
int globus_gssassistmap_local_user(char localUser, char globusidp)
OM_uint32 globus_gssassistimport_sec_context(OM_uint32 minor_status, gssctx_id_t contexthandle, int
token_status, int fdp, FILE fperr)
OM_uint32 globus_gssassistinit_sec_context(OM_uint32 , const gsscred_id_t, gssctx_id_t , char , OM_
uint32, OM_uint32 , int , int( get_token)(void , void , size_t ), void get_arg, int( send_token)(void ,
void , size_t), void send_arg)
OM_uint32 globus_gssassistinit_sec_context_async(OM_uint32 minor_status, const gsscred_id_t cred-
handle, gssctx_id_t contexthandle, char target_namechar, OM_uint32 req_ag, OM_uint32 ret_ag, void
input_buffer, size_t input_buffer_len, void output_bufferp, size_t output_buffer_lenp)
OM_uint32 globus_gssassistwill_handler_restrictions(OM_uint32 minor_status, gssctx_id_t contexthandle)
OM_uint32 globus_gssassistget_unwrap(OM_uint32 minor_status, const gssctx_id_t contexthandle, char
data, size_t length, int token_status, int(gssassist_get_token)(void , void , size_t ), void gssassist-
get_context, FILE fperr)
OM_uint32 globus_gssassistwrap_send(OM_uint32 minor_status, const gssctx_id_t contexthandle, char
data, size_t length, int token_status, int(gssassist_send_token)(void , void , size_t), void gssassistsend-
context, FILE fperr)
```

3.2.1 Detailed Description

Utility functions for GSSAPI.

3.2.2 De ne Documentation

3.2.2.1 #define NI_MAXHOST 255

Create a GSS Name structure from the given hostname. This function tries to resolve the given host name string to the canonical DNS name for the host.

Parameters:

hostname The host name or numerical address to be resolved and transform into a GSS Name
authorization_hostname The resulting GSS Name

Returns:

GLOBUS_SUCCESS on successful completion, a error object otherwise

3.2.3 Function Documentation

```
OM_uint32 globus_gssassistaccept_sec_context (OM_uint32 minor_status, gssctx_id_t context-
handle, const gsscred_id_t cred_handle, char src_namechar, OM_uint32 ret_ag, int user_to_user_ag,
int token_status, gsscred_id_t delegated_cred_handle, int( gssassist_get_token)(void , void , size_t ), void
gssassist_get_context, int( gssassist_send_token)(void , void , size_t), void gssassistsendcontext)
```

This routine accepts a GSSAPI security context and is called by the `gssapi` keeper. It isolates the GSSAPI from the rest of the gram code.

Initialize a gssapi security connection. Used by the server. The `context_handle` is returned, and there is one for each connection. This routine will take care of the looping and token processing, using the supplied `get_token` and `send_token` routines.

Parameters:

`minor_status` gssapi return code
`context_handle` pointer to returned context.
`cred_handle` the cred handle obtained by `acquire_cred`.
`src_name_char` Pointer to char string representation of the client which contacted the server. Maybe NULL if not wanted. Should be freed when done.
`ret_ags` Pointer to which services are available after the connection is established. Maybe NULL if not wanted. We will also use this to pass in `ags` to the globus version of `gssapi`.
`user_to_user_ag` Pointer to `ag` to be set if the `srcname` is the same as our name. (Following are particular to this assist routine)
`token_status` assist routine get/send token status
`delegated_cred_handle` pointer to be set to the credential delegated by the client if delegation occurs during the security handshake
`gssassist_get_token` a get token routine
`gssassist_get_context` rst arg for the get token routine
`gssassist_send_token` a send token routine
`gssassist_send_context` rst arg for the send token routine

Returns:

GSSS.COMPLETE on success Other gss errors on failure.

3.2.3.2 `OM_uint32 globus_gssassist_accept_sec_context_async(OM_uint32 minor_status, gssctx_id_t context_handle, const gsscred_id_t cred_handle, char src_name_char, OM_uint32 ret_ags, int user_to_user_ag, void input_buffer, size_t input_buffer_len, void output_bufferp, size_t output_buffer_len, gsscred_id_t delegated_cred_handle)`

This is a asynchronous version of the `globus_gssassist_accept_sec_context()` function. Instead of looping itself it passes in and out the read and written buffers and the calling application is responsible for doing the I/O directly.

Parameters:

`minor_status` gssapi return code
`context_handle` pointer to returned context.
`cred_handle` the cred handle obtained by `acquire_cred`.
`src_name_char` Pointer to char string representation of the client which contacted the server. Maybe NULL if not wanted. Should be freed when done.
`ret_ags` Pointer to which services are available after the connection is established. Maybe NULL if not wanted. We will also use this to pass in `ags` to the globus version of `gssapi`.
`user_to_user_ag` Pointer to `ag` to be set if the `srcname` is the same as our name.
`input_buffer` pointer to a buffer received from peer.
`input_buffer_len` length of the buffer `inputbuffer`.
`output_bufferp` pointer to a pointer which will be filled in with a pointer to a allocated block of memory. If non-NULL the contents of this block should be written to the peer where they will be fed into the `gssassist_init_sec_context_async()` function.

`output.buffer_lenp` pointer to an integer which will be filled in with the length of the allocated output buffer pointed to by `output.bufferp`.

`delegated_cred.handle` pointer to be set to the credential delegated by the client if delegation occurs during the security handshake

Returns:

`GSSS.COMPLETE` on successful completion when this function does not need to be called again.

`GSSS.CONTINUE.NEEDED` when `output.bufferp` should be sent to the peer and a new input buffer read and this function called again.

Other gss errors on failure.

3.2.3.3 `OM_uint32 globus.gssassistacquire_cred (OM_uint32 minor_status, gsscred_usage cred_usage, gsscred_id_t output_cred_handle)`

Called once at the start of the process, to obtain the credentials the process is running under. The

Parameters:

`minor_status` pointer for return code

`cred_usage` `GSSC_INITIATE`, `GSSC_ACCEPT`, or `GSSC_BOTH`

`output_cred_handle` Pointer to the returned handle. This needs to be passed to many gss routines.

Returns:

`GSSS.COMPLETE` on success Other GSS return codes

3.2.3.4 `OM_uint32 globus.gssassistacquire_cred_ext (OM_uint32 minor_status, char desired_name_char, OM_uint32 time_req, const gssOID_set desired_mechs, gsscred_usage cred_usage, gsscred_id_t output_cred_handle, gssOID_set actual_mechs, OM_uint32 time_rec)`

Called once at the start of the process, to obtain the credentials the process is running under. All the parameters of the `gssacquire_cred`, except the `desired_name` is a string of the form: `[type:]name`. This will be imported with the type.

Returns:

`GSSS.COMPLETE` on success Other GSS return codes

See also:

`globus.gsi.gssacquire_cred`

3.2.3.5 `OM_uint32 globus.gssassistdisplay_status (FILE fp, char comment, OM_uint32 major_status, OM_uint32 minor_status, int token_status)`

Display the messages for the major and minor status on the file pointed at by `fp`. Takes care of the overloaded major status if there was a problem with the `getoken` or `sendtoken` routines.

Parameters:

`fp` file pointer

`comment` String to print out before other error messages.

`major_status` The major status to display

`minor_status` The minor status to display

`token_status` token status to display

Returns:

0

3.2.3.6 `OM_uint32 globus_gssassistdisplay_status_str (char str, char comment, OM_uint32 major_status, OM_uint32 minor_status, int token_status)`

Display the messages for the major and minor status and return a string with the messages. Takes care of the overloaded `major_status` if there was a problem with the `getoken` or `sendoken` routines.

Parameters:

`str` pointer to `char` for returned string. Must be freed
`comment` String to print out before other error messages.
`major_status` The major status to display
`minor_status` The minor status to display
`token_status` token status to display

Returns:

0

3.2.3.7 `int globusgssassistgridmap (char globusidp, char useridp)`

Routines callable from globus based code to map a globusID to a local unix user

GRIDMAP environment variable pointing at the map file. Defaults to `togridmap`

A gridmap file is required if being run as root. if being run as a user, it is not required, and defaults to the current user who is running the command.

This is the same file used by the `gssapi` but will be used with other gssapi implementations which do not use the gridmap file.

Parameters:

`globusidp` the GSSAPI name from the client who requested authentication
`useridp` the resulting user ID name for the local system

Returns:

0 on success -1 if bad arguments 1 on error

3.2.3.8 `int globusgssassistuserok (char globusid, char userid)`

Check to see if a particular globusid is authorized to access the given local user account.

Parameters:

`globusid` the globus id in string form - this should be the user's subject
`userid` the local account that access is sought for

Returns:

0 on success (authorization allowed) -1 if bad arguments 1 on error

3.2.3.9 `int globusgssassistmap_local_user (char local_user, char globusidp)`

Routine for returning the default globus ID associated with a local user name. This is somewhat of a hack since there is not a guaranteed one-to-one mapping. What we do is look for the first entry in the gridmap file that has the local user as the default login. If the user is not a default on any entry, we find the first entry in which the user exists as a secondary mapping.

Parameters:

local_user the local username to find the DN for
 globusidp the first DN found that reverse maps from the local user

Returns:

0 on success, otherwise an error object identifier is returned. use `globuserror_get` to get the error object from the id. The resulting error object must be freed using `globusobjectfree` when it is no longer needed.

See also:

`globuserror_get` , `globusobjectfree`

3.2.3.10 `OM_uint32 globus_gssassistimport_seccontext (OM_uint32 minor_status, gssctx_id_t context-handle, int token_status, int fdp, FILE fperr)`

Import the security context from a file

Parameters:

minor_status GSSAPI return code. This is a Globus Error code (or `GLOBUS_SUCCESS`) cast to a `OM_uint32` pointer. If an error has occurred, the resulting error (from calling `globuserror_get` on this variable) needs to be freed by the caller
 contexthandle The imported context
 token_status Errors that occurred while reading from the file
 fdp the file descriptor pointing to a file containing the security context
 fperr FILE to write error messages

Returns:

the major status

3.2.3.11 `OM_uint32 globus_gssassistinit_seccontext (OM_uint32 minor_status, const gsscred_id_t cred-handle, gssctx_id_t contexthandle, char targetname, char, OM_uint32 req_flags, OM_uint32 ret_flags, int token_status, int(gssassistgettoken)(void *, void *, size_t), void gssassistgetcontext, int(gssassist-sendtoken)(void *, void *, size_t), void gssassistsendcontext)`

Initialize a gssapi security connection. Used by the client. The contexthandle is returned, and there is one for each connection. This routine will take care of the looping and token processing, using the supplied `globus_gssassistgettoken` and `globus_gssassistsendtoken` routines.

Parameters:

minor_status GSSAPI return code. The new minor status is a globus resultt cast to an `OM_uint32`. If the call was successful, the minor status is equivalent to `GLOBUS_SUCCESS`. Otherwise, it is a globus error object ID that can be passed to `globuserror_get` to get the error object. The error object needs to be freed with `globusobjectfree`.
 cred.handle the cred handle obtained by `globus_gss_acquire_cred`.
 contexthandle pointer to returned context.
 targetname.char char string representation of the server to be contacted.
 req_flags request flags, such as `GSS_DELEG_FLAG` for delegation and the `GSS_MUTUAL_FLAG` for mutual authentication.
 ret_flags Pointer to which services are available after the connection is established. Maybe NULL if not wanted.

The Following are particular to this assist routine:

Parameters:

token.status the assist routine's get/send token status
 gssassistget.token function pointer for getting the token
 gssassistget.context rst argument passed to the gssassistget.token function
 gssassistsend.token function pointer for setting the token
 gssassistsend.context rst argument passed to the gssassistset.token function pointer

Returns:

The major status

3.2.3.12 OM_uint32 globus_gssassistinit_seccontext_async (OM_uint32 minor_status, const gss_cred_id_t cred_handle, gss_ctx_id_t context_handle, char target_name_char, OM_uint32 req_ags, OM_uint32 ret_ags, void input_buffer, size_t input_buffer_len, void output_bufferp, size_t output_buffer_lenp)

This is a asynchronous version of [globus_gssassistinit_seccontext\(\)](#) function. Instead of looping itself it passes in and out the read and written buffers and the calling application is responsible for doing the I/O directly.

Parameters:

minor_status GSSAPI return code. The new minor status is a globus_result_t cast to a OM_uint32. If an error occurred (GSS_ERROR(majorstatus)) the minostatus is a globus error object id. The error object can be obtained via globus_error_get and should be destroyed with globus_object_free when no longer needed. If no error occurred, the minor status is equal to GLOBUS_SUCCESS.
 cred_handle the cred handle obtained by acquire.
 context_handle pointer to returned context.
 target_name_char char string representation of the server to be contacted.
 req_ags request_ags, such as GSS_DELEG_FLAG for delegation and the GSS_MUTUAL_FLAG for mutual authentication.
 ret_ags Pointer to which services are available after the connection is established. Maybe NULL if not wanted.
 input_buffer pointer to a buffer received from peer. Should be NULL on rst call.
 input_buffer_len length of the buffer inputbuffer. Should be zero on rst call.
 output_bufferp pointer to a pointer which will be filled in with a pointer to a allocated block of memory. If non-NULL the contents of this block should be written to the peer where they will be fed into the gssassistinit_seccontext_async() function.
 output_buffer_lenp pointer to an integer which will be filled in with the length of the allocated output buffer pointed to by output_bufferp.

Returns:

GSSS.COMPLETE on successful completion when this function does not need to be called again.

GSSS.CONTINUE.NEEDED when output_bufferp should be sent to the peer and a new input buffer read and this function called again.

Other gss errors on failure.

3.2.3.13 `OM_uint32 globus_gssassistwill_handle_restrictions (OM_uint32 minor_status, gssctx_id_t context_handle)`

Sets the context to handle restrictions

Parameters:

`minor_status` the resulting minor status from setting the context handle
`context_handle` the context handle to set the minor status of

Returns:

the major status from setting the context

3.2.3.14 `OM_uint32 globus_gssassistget_unwrap (OM_uint32 minor_status, const gssctx_id_t context_handle, char *data, size_t length, int token_status, int(gssassistget_token)(void *, void *, size_t), void gssassistget_context, FILE *fperr)`

Gets a token using the specific tokenizing functions, and performs the GSS unwrap of that token

See also:

`gssunwrap`

Parameters:

`minor_status` GSSAPI return code,

See also:

`gssunwrap`

Parameters:

`context_handle` the context
`data` pointer to be set to the unwrapped application data. This must be freed by the caller.
`length` pointer to be set to the length of the data byte array.
`token_status` assist routine get/send token status
`gssassistget_token` a detokenizing routine
`gssassistget_context` rst arg for above routine
`fperr` error stream to print to

Returns:

`GSSS.COMPLETE` on success Other gss errors on failure.

3.2.3.15 `OM_uint32 globus_gssassistwrap_send (OM_uint32 minor_status, const gssctx_id_t context_handle, char *data, size_t length, int token_status, int(gssassistsend_token)(void *, void *, size_t), void gssassist_send_context, FILE *fperr)`

Parameters:

`minor_status` GSSAPI return code. If the call was successful, the minor status is equal to `GLIBS_SUCCESS`. Otherwise, it is an error object ID for which `globus_error_get()` and `globus_error_free()` can be used to get and destroy it.
`context_handle` the context.
`data` pointer to application data to wrap and send

length length of the data array
 token.status assist routine get/send token status
 gssassist_send_token a send token routine
 gssassist_send_context rst arg for the send token
 fperr le handle to write error message to.

Returns:

GSSS.COMPLETE on success Other gss errors on failure.

See also:

gss_wrap()

3.3 GSI GSS Assist Constants

Enumerations

```
enum globus_gsi_gssassisterror_t { GLOBUS_GSI_GSSASSIST_ERROR_SUCCESS = 0, GLOBUS_GSI_GSSASSIST_ERROR_WITH_ARGUMENTS = 1, GLOBUS_GSI_GSSASSIST_ERROR_USER_ID_DoesNT_MATCH = 2, GLOBUS_GSI_GSSASSIST_ERROR_IN_GRIDMAP_NO_USER_ENTRY = 3, GLOBUS_GSI_GSSASSIST_ERROR_WITH_GRIDMAP = 4, GLOBUS_GSI_GSSASSIST_ERROR_INVALID_GRIDMAP_FORMAT = 5, GLOBUS_GSI_GSSASSIST_ERROR_ERRNO = 6, GLOBUS_GSI_GSSASSIST_ERROR_WITH_INIT = 7, GLOBUS_GSI_GSSASSIST_ERROR_WITH_WRAP = 8, GLOBUS_GSI_GSSASSIST_ERROR_WITH_TOKEN = 9, GLOBUS_GSI_GSSASSIST_ERROR_EXPORTING_CONTEXT = 10, GLOBUS_GSI_GSSASSIST_ERROR_IMPORTING_CONTEXT = 11, GLOBUS_GSI_GSSASSIST_ERROR_INITIALIZING_CALLOUT_HANDLE = 12, GLOBUS_GSI_GSSASSIST_ERROR_WITH_CALLOUT_CONFIG = 13, GLOBUS_GSI_GSSASSIST_ERROR_CALLOUT_ERROR = 14, GLOBUS_GSI_GSSASSIST_ERROR_GSSAPI_ERROR = 15, GLOBUS_GSI_GSSASSIST_ERROR_GRIDMAP_LOOKUP_FAILED = 16, GLOBUS_GSI_GSSASSIST_ERROR_BUFFER_TOO_SMALL = 17, GLOBUS_GSI_GSSASSIST_ERROR_CANONICALIZING_HOSTNAME = 18 };
```

3.3.1 Enumeration Type Documentation

3.3.1.1 enum globus_gsi_gssassisterror_t

GSI GSS Assist Error codes.

Enumeration values:

GLOBUS_GSI_GSSASSIST_ERROR_SUCCESS Success.

GLOBUS_GSI_GSSASSIST_ERROR_WITH_ARGUMENTS No user entry in gridmap le.

GLOBUS_GSI_GSSASSIST_ERROR_USER_ID_DoesNT_MATCH Error user ID doesn't match.

GLOBUS_GSI_GSSASSIST_ERROR_IN_GRIDMAP_NO_USER_ENTRY Error with arguments passed to function.

GLOBUS_GSI_GSSASSIST_ERROR_WITH_GRIDMAP Error querying gridmap le.

GLOBUS_GSI_GSSASSIST_ERROR_INVALID_GRIDMAP_FORMAT Invalid gridmap le format.

GLOBUS_GSI_GSSASSIST_ERROR_ERRNO System Error.

GLOBUS_GSI_GSSASSIST_ERROR_WITH_INIT Error during context initialization.

GLOBUS_GSI_GSSASSIST_ERROR_WITH_WRAP Error during message wrap.

GLOBUS_GSI_GSSASSIST_ERROR_WITH_TOKEN Error with token.

GLOBUS_GSI_GSSASSIST_ERROR_EXPORTING_CONTEXT Error exporting context.
 GLOBUS_GSI_GSSASSIST_ERROR_IMPORTING_CONTEXT Error importing context.
 GLOBUS_GSI_GSSASSIST_ERROR_INITIALIZING_CALLOUT_HANDLE Error initializing callout handle.
 GLOBUS_GSI_GSSASSIST_ERROR_WITH_CALLOUT_CONFIG Error reading callout configuration.
 GLOBUS_GSI_GSSASSIST_CALLOUT_ERROR Error invoking callout.
 GLOBUS_GSI_GSSASSIST_GSSAPI_ERROR A GSSAPI returned an error.
 GLOBUS_GSI_GSSASSIST_GRIDMAP_LOOKUP_FAILED Gridmap lookup failure.
 GLOBUS_GSI_GSSASSIST_BUFFER_TOO_SMALL Caller provided insufficient buffer space for local identity.
 GLOBUS_GSI_GSSASSIST_ERROR_CANONICALIZING_HOSTNAME Failed to obtain canonical host name.

3.4 Security Token Transport

Token routines using fread and fwrite.

Functions

```

int globusgssassisttoken_getfd(void arg, void bufp, size_t sizep)
int globusgssassisttoken_sendfd(void arg, void buf, size_t size)
int globusgssassisttoken_sendfd_without_length(void arg, void buf, size_t size)
int globusgssassisttoken_sendfd_ex(void arg, void buf, size_t size)
int globusgssassisttoken_getnexus(void arg, void bufp, size_t sizep)
int globusgssassisttoken_sendnexus(void arg, void bufp, size_t sizep)
int globusgssassisttoken_sendnexuswithout_length(void arg, void bufp, size_t sizep)
int globusgssassisttoken_sendnexus_ex(void arg, void bufp, size_t sizep)
  
```

3.4.1 Detailed Description

Token routines using fread and fwrite.

Additional code has been added to detect tokens which are sent without a length field. These can currently be only SSL tokens. This does require some knowledge of the underlying GSSAPI, by the application, but is within the guidelines of the GSSAPI specifications.

The get routine will automatically attempt this test, while a new send routine will check a flag. The old send routine will work as before, sending a 4-byte length.

3.4.2 Function Documentation

3.4.2.1 int globusgssassisttoken_getfd(void arg, void bufp, size_t sizep)

Use an open file descriptor to get a token. This function provides parameter types that allow it to be passed to [Functions](#) and [Utility Functions](#).

Parameters:

arg the FILE stream cast to a void pointer
 bufp the resulting token

sizep the size (number of bytes) read into bufp

Returns:

0 on success, 0 is internal return, < 0 is the -errno

3.4.2.2 int globusgssassisttoken_sendfd (void arg, void buf, size_t size)

Write a token to the open file descriptor. Will write it with a 4 byte length. This function provides parameter types that allow it to be passed [Utility Functions](#) and [Utility Functions](#)

Parameters:

arg the FILE stream to send the token on

buf the token

size the size of the token in bytes

Returns:

0 on success, 0 on error, < 0 on errno error

3.4.2.3 int globusgssassisttoken_sendfd_without_length (void arg, void buf, size_t size)

Write a token to the open file descriptor. Will write it without a length. so as to

3.4.2.4 int globusgssassisttoken_sendfd_ex (void exp, void buf, size_t size)

Write a token to the open file descriptor. will look at the arg to determine if the length field need to be written.

Parameters:

exp the globusgssassistex variable that holds the FILE stream and args to be set

buf the token buffer to send

size size of the token buffer

Returns:

0 on success, 0 on error, < 0 on errno error (-errno)

3.4.2.5 int globusgssassisttoken_get_nexus (void arg, void bufp, size_t sizep)

Use a nexus socket to get the tokens.

Additional code has been added to detect tokens which are sent without a length field. These can currently be only SSL tokens. This does require some knowledge of the underlying GSSAPI, by the application, but is within the guidelines of the GSSAPI specifications.

The get routine will automatically attempt this test, while a new send routine will check a flag. The old send routine will work as before, sending a 4-byte length.

Parameters:

arg the globusio_handle_t to get the token from

bufp the buffer to read the token into

sizep the size of what gets read

Returns:

0 on success, 0 is internal return, < 0 is the -errno returned from nexus

3.4.2.6 int globusgssassisttoken_sendnexus (void arg, void buf, size_t size)

Write a token to the nexus io handle. This function provides parameter types that allow it to be passed to [Utility Functions](#) and [Utility Functions](#)

Parameters:

- arg nexus io handle to send the token on
- buf the token as a buffer
- size the size of the buffer

Returns:

- 0 on success
- 0 on error
- < 0 on errno error (-errno)

3.4.2.7 int globusgssassisttoken_sendnexus_without_length (void arg, void buf, size_t size)

Send a token on a nexus IO handle. Using this function the length is not sent.

See also:

[globusgssassisttoken_get_nexus\(\)](#) for further info.

3.4.2.8 int globusgssassisttoken_sendnexus_ex (void exp, void buf, size_t size)

Write a token to the open le descriptor. will look at the ag to determine if the length eld need to be written.

Parameters:

- exp The globusgssassistex that the wraps the nexus IO handle to send the token on
- buf the buffer holding the token
- size the size of the buffer

Returns:

- 0 on success
- 0 on error
- < 0 on errno error (-errno)

Index

Activation, [1](#)

globus.gsi.gssassist

- globus.gssassistacceptseccontext, [3](#)
- globus.gssassistacceptseccontextasync, [4](#)
- globus.gssassistacquirecred, [5](#)
- globus.gssassistacquirecred.ext, [5](#)
- globus.gssassistdisplay.status, [5](#)
- globus.gssassistdisplay.statusstr, [5](#)
- globus.gssassistget.unwrap, [9](#)
- globus.gssassistgridmap, [6](#)
- globus.gssassistimport.seccontext, [7](#)
- globus.gssassistinit.seccontext, [7](#)
- globus.gssassistinit.seccontextasync, [8](#)
- globus.gssassistmap.localuser, [6](#)
- globus.gssassistuserok, [6](#)
- globus.gssassistwill_handlerrestrictions, [8](#)
- globus.gssassistwrap.send, [9](#)
- NI_MAXHOST, [3](#)

globus.gsi.gssassistactivation

- GLOBUS_GSI_GSSASSIST_MODULE, [2](#)

GLOBUS_GSI_GSSASSIST_BUFFER_TOO_SMALL

- globus.gsi.gssassistconstants, [11](#)

GLOBUS_GSI_GSSASSIST_CALLOUT_ERROR

- globus.gsi.gssassistconstants, [11](#)

globus.gsi.gssassistconstants

- GLOBUS_GSI_GSSASSIST_BUFFER_TOO_SMALL, [11](#)

- GLOBUS_GSI_GSSASSIST_CALLOUT_ERROR, [11](#)

- GLOBUS_GSI_GSSASSIST_ERROR_CANONICALIZING_HOSTNAME, [11](#)

- GLOBUS_GSI_GSSASSIST_ERROR_ERRNO, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_EXPORTING_CONTEXT, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_IMPORTING_CONTEXT, [11](#)

- GLOBUS_GSI_GSSASSIST_ERROR_IN_GRIDMAP_NO_USERENTRY, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_INITIALIZING_CALLOUT_HANDLE, [11](#)

- GLOBUS_GSI_GSSASSIST_ERROR_INVALID_GRIDMAP_FORMAT, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_SUCCESS, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_USER_ID_Doesnt_Match, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_WITH_ARGUMENTS, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_WITH_CALLOUT_CONFIG, [11](#)

- GLOBUS_GSI_GSSASSIST_ERROR_WITH_GRIDMAP, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_WITH_INIT, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_WITH_TOKEN, [10](#)

- GLOBUS_GSI_GSSASSIST_ERROR_WITH_WRAP, [10](#)

- GLOBUS_GSI_GSSASSIST_GRIDMAP_LOOKUP_FAILED, [11](#)

- GLOBUS_GSI_GSSASSIST_GSSAPI_ERROR, [11](#)

globus.gsi.gssassistconstants

- globus.gsi.gssassisterror.t, [10](#)

GLOBUS_GSI_GSSASSIST_ERROR_CANONICALIZING_HOSTNAME

- globus.gsi.gssassistconstants, [11](#)

GLOBUS_GSI_GSSASSIST_ERROR_ERRNO

- globus.gsi.gssassistconstants, [10](#)

GLOBUS_GSI_GSSASSIST_ERROR_EXPORTING_CONTEXT

- globus.gsi.gssassistconstants, [10](#)

GLOBUS_GSI_GSSASSIST_ERROR_IMPORTING_CONTEXT

- globus.gsi.gssassistconstants, [11](#)

GLOBUS_GSI_GSSASSIST_ERROR_IN_GRIDMAP_NO_USERENTRY

- globus.gsi.gssassistconstants, [10](#)

GLOBUS_GSI_GSSASSIST_ERROR_INITIALIZING_CALLOUT_HANDLE

- globus.gsi.gssassistconstants, [11](#)

GLOBUS_GSI_GSSASSIST_ERROR_INVALID_GRIDMAP_FORMAT

- globus.gsi.gssassistconstants, [10](#)

GLOBUS_GSI_GSSASSIST_ERROR_SUCCESS

- globus.gsi.gssassistconstants, [10](#)

globus.gsi.gssassisterror.t

- globus.gsi.gssassistconstants, [10](#)

GLOBUS_GSI_GSSASSIST_ERROR_USER_ID_Doesnt_Match

- globus.gsi.gssassistconstants, [10](#)

GLOBUS_GSI_GSSASSIST_ERROR_WITH_ARGUMENTS

- globus.gsi.gssassistconstants, [10](#)

GLOBUS_GSI_GSSASSIST_ERROR_WITH_CALLOUT_CONFIG

- globus.gsi.gss.assistconstants, [11](#)
- GLOBUS.GSI.GSSASSIST.ERRORWITH._GRIDMAP
 - globus.gsi.gss.assistconstants, [10](#)
- GLOBUS.GSI.GSSASSIST.ERRORWITH._INIT
 - globus.gsi.gss.assistconstants, [10](#)
- GLOBUS.GSI.GSSASSIST.ERRORWITH._TOKEN
 - globus.gsi.gss.assistconstants, [10](#)
- GLOBUS.GSI.GSSASSIST.ERRORWITH._WRAP
 - globus.gsi.gss.assistconstants, [10](#)
- GLOBUS.GSI.GSSASSIST.GRIDMAP._LOOKUP_FAILED
 - globus.gsi.gss.assistconstants, [11](#)
- GLOBUS.GSI.GSSASSIST.GSSAPI.ERROR
 - globus.gsi.gss.assistconstants, [11](#)
- GLOBUS.GSI.GSSASSIST.MODULE
 - globus.gsi.gss.assistactivation, [2](#)
- globus.gsi.gss.assisttokens
 - globus.gss.assisttoken.getfd, [11](#)
 - globus.gss.assisttoken.getnexus, [12](#)
 - globus.gss.assisttoken.sendfd, [12](#)
 - globus.gss.assisttoken.sendfd_ex, [12](#)
 - globus.gss.assisttoken.sendfd_without_length, [12](#)
 - globus.gss.assisttoken.sendnexus, [12](#)
 - globus.gss.assisttoken.sendnexus_ex, [13](#)
 - globus.gss.assisttoken.sendnexuswithout_length, [13](#)
- globus.gss.assistacceptseccontext
 - globus.gsi.gss.assist, [3](#)
- globus.gss.assistacceptseccontextasync
 - globus.gsi.gss.assist, [4](#)
- globus.gss.assistacquirecred
 - globus.gsi.gss.assist, [5](#)
- globus.gss.assistacquirecred_ext
 - globus.gsi.gss.assist, [5](#)
- globus.gss.assistdisplay.status
 - globus.gsi.gss.assist, [5](#)
- globus.gss.assistdisplay.statusstr
 - globus.gsi.gss.assist, [5](#)
- globus.gss.assistget.unwrap
 - globus.gsi.gss.assist, [9](#)
- globus.gss.assistgridmap
 - globus.gsi.gss.assist, [6](#)
- globus.gss.assistimport.seccontext
 - globus.gsi.gss.assist, [7](#)
- globus.gss.assistinit.seccontext
 - globus.gsi.gss.assist, [7](#)
- globus.gss.assistinit.seccontextasync
 - globus.gsi.gss.assist, [8](#)
- globus.gss.assistmap.localuser
 - globus.gsi.gss.assist, [6](#)
- globus.gss.assisttoken.getfd
 - globus.gsi.gss.assisttokens, [11](#)
- globus.gss.assisttoken.getnexus
 - globus.gsi.gss.assisttokens, [12](#)
- globus.gss.assisttoken.sendfd
 - globus.gsi.gss.assisttokens, [12](#)
- globus.gss.assisttoken.sendfd_ex
 - globus.gsi.gss.assisttokens, [12](#)
- globus.gss.assisttoken.sendfd_without_length
 - globus.gsi.gss.assisttokens, [12](#)
- globus.gss.assisttoken.sendnexus
 - globus.gsi.gss.assisttokens, [12](#)
- globus.gss.assisttoken.sendnexus_ex
 - globus.gsi.gss.assisttokens, [13](#)
- globus.gss.assisttoken.sendnexuswithout_length
 - globus.gsi.gss.assisttokens, [13](#)
- globus.gss.assistuserok
 - globus.gsi.gss.assist, [6](#)
- globus.gss.assistwill_handlerrestrictions
 - globus.gsi.gss.assist, [8](#)
- globus.gss.assistwrap.send
 - globus.gsi.gss.assist, [9](#)
- GSI GSS Assist Constants, [10](#)
- NI_MAXHOST
 - globus.gsi.gss.assist, [3](#)
- Security Token Transport, [1,1](#)
- Utility Functions, [2](#)