globus gsi sysconfig

5.1

Generated by Doxygen 1.7.6.1

Thu Jan 26 2012 14:50:12

Contents

1 Globus GSI System Config API

This API provides helper functions for detecting installation and environment specific settings applicabale to GSI. It also servers as a abstraction layer for OS specific programming details. This is achieves by defining preprocessor symbols that point at the correct platform specific function. **You should never use the platform specific functions directly.**. Any program that uses Globus GSI System Config functions must include "globus_gsi_system_config.h".

2 Module Index

2.1 Modules

Here is a list of all modules:

3 Module Documentation

3.1 Defines

Defines

- #define **GLOBUS_GSI_SYSCONFIG_SET_KEY_PERMISSIONS**
- #define **GLOBUS_GSI_SYSCONFIG_GET_HOME_DIR**
- #define **GLOBUS_GSI_SYSCONFIG_CHECK_KEYFILE**
- #define **GLOBUS_GSI_SYSCONFIG_CHECK_CERTFILE**
- #define **GLOBUS_GSI_SYSCONFIG_FILE_EXISTS**
- #define **GLOBUS_GSI_SYSCONFIG_DIR_EXISTS**
- #define **GLOBUS_GSI_SYSCONFIG_GET_CERT_DIR**
- #define **GLOBUS_GSI_SYSCONFIG_GET_USER_CERT_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_GET_HOST_CERT_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_GET_SERVICE_CERT_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_GET_PROXY_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_GET_SIGNING_POLICY_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_GET_CA_CERT_FILES**
- #define **GLOBUS_GSI_SYSCONFIG_GET_CURRENT_WORKING_DIR**
- #define **GLOBUS_GSI_SYSCONFIG_MAKE_ABSOLUTE_PATH_FOR_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_SPLIT_DIR_AND_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_REMOVE_ALL_OWNED_FILES**
- #define **GLOBUS_GSI_SYSCONFIG_GET_GRIDMAP_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_GET_AUTHZ_CONF_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_GET_GAA_CONF_FILENAME**
- #define **GLOBUS_GSI_SYSCONFIG_IS_SUPERUSER**
- #define **GLOBUS_GSI_SYSCONFIG_GET_USER_ID_STRING**
- #define **GLOBUS_GSI_SYSCONFIG_GET_PROC_ID_STRING**
- #define **GLOBUS_GSI_SYSCONFIG_GET_USERNAME**
- #define **GLOBUS_GSI_SYSCONFIG_GET_UNIQUE_PROXY_FILENAME**

3.1.1 Detailed Description

These precompiler defines allow for a platform (ie Win32 vs UNIX) independent API.

3.1.2 Define Documentation

3.1.2.1 #define GLOBUS_GSI_SYSCONFIG_SET_KEY_PERMISSIONS

Set the correct file permissions on a private key.

See **globus_gsi_sysconfig_set_key_permissions_unix()** (p. 8) and **globus_gsi_sysconfig_set_key_permissions_win32()** (p. 17)

3.1.2.2 #define GLOBUS_GSI_SYSCONFIG_GET_HOME_DIR

Get the current users home directory

See **globus_gsi_sysconfig_get_home_dir_unix()** (p. 10) and globus_gsi_sysconfig_get_home_dir_win32()

3.1.2.3 #define GLOBUS_GSI_SYSCONFIG_CHECK_KEYFILE

Check for the correct file permissions on a private key.

See **globus_gsi_sysconfig_check_keyfile_unix()** (p. 10) and globus_gsi_sysconfig_check_keyfile_win32()

3.1.2.4 #define GLOBUS_GSI_SYSCONFIG_CHECK_CERTFILE

Check for the correct file permissions on a certificate.

See **globus_gsi_sysconfig_check_certfile_unix()** (p. 11) and globus_gsi_sysconfig_check_certfile_win32()

3.1.2.5 #define GLOBUS_GSI_SYSCONFIG_FILE_EXISTS

Check whether a given file exists

See **globus_gsi_sysconfig_file_exists_unix()** (p. 10) and **globus_gsi_sysconfig_file_exists_win32()** (p. 18)

3.1.2.6 #define GLOBUS_GSI_SYSCONFIG_DIR_EXISTS

Check whether a given directory exists

See **globus_gsi_sysconfig_dir_exists_unix()** (p. 10) and **globus_gsi_sysconfig_dir_exists_win32()** (p. 18)

3.1.2.7 #define GLOBUS_GSI_SYSCONFIG_GET_CERT_DIR

Determine the location of the trusted certificates directory

See **globus_gsi_sysconfig_get_cert_dir_unix()** (p. 11) and **globus_gsi_sysconfig_get_cert_dir_win32()** (p. 19)

3.1.2.8 #define GLOBUS_GSI_SYSCONFIG_GET_USER_CERT_FILENAME

Determine the location of the users certificate and private key

See **globus_gsi_sysconfig_get_user_cert_filename_unix()** (p. 11) and **globus_gsi_sysconfig_get_user_cert_filename_win32()** (p. 19)

3.1.2.9 #define GLOBUS_GSI_SYSCONFIG_GET_HOST_CERT_FILENAME

Determine the location of the host certificate and private key

See **globus_gsi_sysconfig_get_host_cert_filename_unix()** (p. 12) and **globus_gsi_sysconfig_get_host_cert_filename_win32()** (p. 20)

3.1.2.10 #define GLOBUS_GSI_SYSCONFIG_GET_SERVICE_CERT_FILENAME

Determine the location of a service certificate and private key

See **globus_gsi_sysconfig_get_service_cert_filename_unix()** (p. 12) and **globus_gsi_sysconfig_get_service_cert_filename_win32()** (p. 20)

3.1.2.11 #define GLOBUS_GSI_SYSCONFIG_GET_PROXY_FILENAME

Determine the location of a proxy certificate and private key

See **globus_gsi_sysconfig_get_proxy_filename_unix()** (p. 13) and **globus_gsi_sysconfig_get_proxy_-**

filename_win32() (p. 21)

3.1.2.12 #define GLOBUS_GSI_SYSCONFIG_GET_SIGNING_POLICY_FILENAME

Determine the name of the signing policy file for a given CA

See **globus_gsi_sysconfig_get_signing_policy_filename_unix()** (p. 13) and **globus_gsi_sysconfig_get_signing_policy_filename_win32()** (p. 23)

3.1.2.13 #define GLOBUS_GSI_SYSCONFIG_GET_CA_CERT_FILES

Get a list of of trusted CA certificate filenames in a trusted CA certificate directory.

See **globus_gsi_sysconfig_get_ca_cert_files_unix()** (p. 13) and **globus_gsi_sysconfig_get_ca_cert_files_win32()** (p. 21)

3.1.2.14 #define GLOBUS_GSI_SYSCONFIG_GET_CURRENT_WORKING_DIR

Get the current working directory

See **globus_gsi_sysconfig_get_current_working_dir_unix()** (p. 9) and **globus_gsi_sysconfig_get_current_working_dir_win32()** (p. 18)

3.1.2.15 #define GLOBUS_GSI_SYSCONFIG_MAKE_ABSOLUTE_PATH_FOR_FILENAME

Prepend the current working directory to the give filename

See **globus_gsi_sysconfig_make_absolute_path_for_filename_unix()** (p. 9) and **globus_gsi_sysconfig_make_absolute_path_for_filename_win32()** (p. 18)

3.1.2.16 #define GLOBUS_GSI_SYSCONFIG_SPLIT_DIR_AND_FILENAME

Split directory component of path from filename.

See **globus_gsi_sysconfig_split_dir_and_filename_unix()** (p. 9) and **globus_gsi_sysconfig_split_dir_and_filename_win32()** (p. 19)

3.1.2.17 #define GLOBUS_GSI_SYSCONFIG_REMOVE_ALL_OWNED_FILES

Remove all proxies owned by current uid

See **globus_gsi_sysconfig_remove_all_owned_files_unix()** (p. 14) and **globus_gsi_sysconfig_remove_all_owned_files_win32()** (p. 22)

3.1.2.18 #define GLOBUS_GSI_SYSCONFIG_GET_GRIDMAP_FILENAME

Determine the location of the grid map file.

See **globus_gsi_sysconfig_get_gridmap_filename_unix()** (p. 14) and **globus_gsi_sysconfig_get_gridmap_filename_win32()** (p. 22)

3.1.2.19 #define GLOBUS_GSI_SYSCONFIG_GET_AUTHZ_CONF_FILENAME

Determine the location of the authorization callout config file.

See **globus_gsi_sysconfig_get_authz_conf_filename_unix()** (p. 14)

3.1.2.20 #define GLOBUS_GSI_SYSCONFIG_GET_GAA_CONF_FILENAME

Determine the location of the GAA callout config file.

See **globus_gsi_sysconfig_get_gaa_conf_filename_unix()** (p. 15)

3.1.2.21 #define GLOBUS_GSI_SYSCONFIG_IS_SUPERUSER

Determine whether the current user is the super user

See **globus_gsi_sysconfig_is_superuser_unix()** (p. 14) and **globus_gsi_sysconfig_is_superuser_win32()** (p. 23)

3.1.2.22 #define GLOBUS_GSI_SYSCONFIG_GET_USER_ID_STRING

Get the current UID in string form

See **globus_gsi_sysconfig_get_user_id_string_unix()** (p. 8) and globus_gsi_sysconfig_get_user_id_string_win32()

3.1.2.23 #define GLOBUS_GSI_SYSCONFIG_GET_PROC_ID_STRING

Get the current PID in string form

See **globus_gsi_sysconfig_get_proc_id_string_unix()** (p. 9) and globus_gsi_sysconfig_get_proc_id_string_win32()

3.1.2.24 #define GLOBUS_GSI_SYSCONFIG_GET_USERNAME

Get the current user name

See **globus_gsi_sysconfig_get_username_unix()** (p. 8) and globus_gsi_sysconfig_get_username_win32()

3.1.2.25 #define GLOBUS_GSI_SYSCONFIG_GET_UNIQUE_PROXY_FILENAME

Generate a unqiue proxy file name

See **globus_gsi_sysconfig_get_unique_proxy_filename()** (p. 24)

3.2 Functions for UNIX platforms

UNIX - Set Key Permissions

- globus_result_t **globus_gsi_sysconfig_set_key_permissions_unix** (char *filename)

UNIX - Get User ID

- globus_result_t **globus_gsi_sysconfig_get_user_id_string_unix** (char **user_id_string)

UNIX - Get Username

- globus_result_t **globus_gsi_sysconfig_get_username_unix** (char **username)

UNIX - Get Process ID

- globus_result_t **globus_gsi_sysconfig_get_proc_id_string_unix** (char **proc_id_string)

UNIX - Make Absolute Path

- globus_result_t **globus_gsi_sysconfig_make_absolute_path_for_filename_unix** (char *filename, char **absolute_path)

UNIX - Split Directory and Filename

- globus_result_t **globus_gsi_sysconfig_split_dir_and_filename_unix** (char *full_filename, char **dir_string, char **filename_string)

UNIX - Get Current Working Directory

- globus_result_t **globus_gsi_sysconfig_get_current_working_dir_unix** (char **working_dir)

UNIX - Get HOME Directory

- globus_result_t **globus_gsi_sysconfig_get_home_dir_unix** (char **home_dir)

UNIX - File Exists

- globus_result_t **globus_gsi_sysconfig_file_exists_unix** (const char *filename)

UNIX - Directory Exists

- globus_result_t **globus_gsi_sysconfig_dir_exists_unix** (const char *filename)

UNIX - Check File Status for Key

- globus_result_t **globus_gsi_sysconfig_check_keyfile_unix** (const char *filename)

UNIX - Check File Status for Cert

- globus_result_t **globus_gsi_sysconfig_check_certfile_unix** (const char *filename)

UNIX - Get Trusted CA Cert Dir

- globus_result_t **globus_gsi_sysconfig_get_cert_dir_unix** (char **cert_dir)

UNIX - Get User Certificate and Key Filenames

- globus_result_t **globus_gsi_sysconfig_get_user_cert_filename_unix** (char **user_cert, char **user_key)

UNIX - Get Host Certificate and Key Filenames

- globus_result_t **globus_gsi_sysconfig_get_host_cert_filename_unix** (char **host_cert, char **host_key)

UNIX - Get Service Certificate and Key Filenames

- globus_result_t **globus_gsi_sysconfig_get_service_cert_filename_unix** (char *service_name, char **service_cert, char **service_key)

UNIX - Get Proxy Filename

- globus_result_t **globus_gsi_sysconfig_get_proxy_filename_unix** (char **user_proxy, **globus_gsi_proxy_file_type_t** proxy_file_type)

UNIX - Get Signing Policy Filename

- globus_result_t **globus_gsi_sysconfig_get_signing_policy_filename_unix** (X509_NAME *ca_name, char *cert_dir, char **signing_policy_filename)

UNIX - Get CA Cert Filenames

- globus_result_t **globus_gsi_sysconfig_get_ca_cert_files_unix** (char *ca_cert_dir, globus_fifo_t *ca_cert_list)

UNIX - Remove all proxies owned by current uid

- globus_result_t **globus_gsi_sysconfig_remove_all_owned_files_unix** (char *default_filename)

UNIX - Check if the current user is root

- globus_result_t **globus_gsi_sysconfig_is_superuser_unix** (int *is_superuser)

UNIX - Get the path and file name of the grid map file

- globus_result_t **globus_gsi_sysconfig_get_gridmap_filename_unix** (char **filename)

UNIX - Get the path and file name of the authorization callback configuration file

- globus_result_t **globus_gsi_sysconfig_get_authz_conf_filename_unix** (char **filename)
- globus_result_t **globus_gsi_sysconfig_get_authz_lib_conf_filename_unix** (char **filename)

UNIX - Get the path and file name of the gaa configuration file

- globus_result_t **globus_gsi_sysconfig_get_gaa_conf_filename_unix** (char **filename)

3.2.1 Detailed Description

These functions implement the UNIX version of the Globus GSI System Configuration API. **They should never be called directly, please use the provided platform independent defines.**

3.2.2 Function Documentation

3.2.2.1 globus_result_t globus_gsi_sysconfig_set_key_permissions_unix (char * *filename*)

Set the file permissions of a file to read-write only by the user which are the permissions that should be set for all private keys.

Parameters

filename	

Returns

GLOBUS_SUCCESS or an error object id

3.2.2.2 globus_result_t globus_gsi_sysconfig_get_user_id_string_unix (char ** *user_id_string*)

Get a unique string representing the current user.

This is just the uid converted to a string.

Parameters

user_id_string	A unique string representing the user

Returns

GLOBUS_SUCCESS unless an error occurred

3.2.2.3 globus_result_t globus_gsi_sysconfig_get_username_unix (char ** *username*)

Get the username of the current user.

Parameters

username	This parameter will contain the current user name upon a successful return. It is the users responsibility to free memory allocated for this return value.

Returns

GLOBUS_SUCCESS unless an error occurred

3.2.2.4 globus_result_t globus_gsi_sysconfig_get_proc_id_string_unix (char ** *proc_id_string*)

Get a unique string representing the current process.

This is just the pid converted to a string.

Parameters

proc_id_string	A unique string representing the process

Returns

GLOBUS_SUCCESS unless an error occurred

3.2.2.5 globus_result_t globus_gsi_sysconfig_make_absolute_path_for_filename_unix (char * *filename,* char ** *absolute_path*)

Make the filename into an absolute path string based on the current working directory.

Parameters

filename	the filename to get the absolute path of.
absolute_path	The resulting absolute path. This needs to be freed when no longer needed.

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.2.2.6 globus_result_t globus_gsi_sysconfig_split_dir_and_filename_unix (char * *full_filename,* char ** *dir_string,* char ** *filename_string*)

Split the directory and filename portions of a filename string into two separate strings.

Parameters

full_filename	The filename to split. Splits on the last occurrance of '/' where the directory is everything before the last '/', and the filename is everything after.
dir_string	The directory portion of the filename string. If no '/' is found throughout the string, this variable points to NULL. This needs to be freed when no longer needed.
filename_string	The filename portion of the filename string. If no '/' is found throughout, this variable is a duplicate of the full_filename parameter. This needs to be freed when no longer needed.

Returns

GLOBUS_SUCCESS if no error occurred. Otherwise an error object ID is returned.

3.2.2.7 globus_result_t globus_gsi_sysconfig_get_current_working_dir_unix (char ** *working_dir*)

Get the current working directory on the system.

Parameters

working_dir	The current working directory

Returns

GLOBUS_SUCCESS or an error object identifier

3.2.2.8 globus_result_t globus_gsi_sysconfig_get_home_dir_unix (char ** *home_dir*)

Get the HOME Directory of the current user.

Should be the $HOME environment variable.

Parameters

home_dir	The home directory of the current user

Returns

GLOBUS_SUCCESS if no error occured, otherwise an error object is returned.

3.2.2.9 globus_result_t globus_gsi_sysconfig_file_exists_unix (const char * *filename*)

Check if the file exists.

Parameters

filename	The filename of the file to check for

Returns

GLOBUS_SUCCESS if the file exists and is readable, otherwise an error object identifier

3.2.2.10 globus_result_t globus_gsi_sysconfig_dir_exists_unix (const char * *filename*)

Check if the directory exists.

Parameters

filename	The filename of the directory to check for

Returns

GLOBUS_SUCCESS if the directory exists, otherwise an error object identifier.

3.2.2.11 globus_result_t globus_gsi_sysconfig_check_keyfile_unix (const char * *filename*)

This is a convenience function used to check the status of a private key file.

The desired status is only the current user has ownership and read permissions, everyone else should not be able to access it.

Parameters

filename	The name of the file to check the status of

Returns

GLOBUS_SUCCESS if the status of the file was able to be determined. Otherwise, an error object identifier

3.2.2.12 globus_result_t globus_gsi_sysconfig_check_certfile_unix (const char * *filename*)

This is a convenience function used to check the status of a certificate file.

The desired status is the current user has ownership and read/write permissions, while group and others only have read permissions.

Parameters

filename	The name of the file to check the status of

Returns

GLOBUS_SUCCESS if the status of the file was able to be determined. Otherwise, an error object identifier

3.2.2.13 globus_result_t globus_gsi_sysconfig_get_cert_dir_unix (char ** *cert_dir*)

Get the Trusted Certificate Directory containing the trusted Certificate Authority certificates.

This directory is determined in the order shown below. Failure in one method results in attempting the next.

1. **X509_CERT_DIR environment variable** - if this is set, the trusted certificates will be searched for in that directory. This variable allows the end user to specify the location of trusted certificates.
2. **$HOME/.globus/certificates** - If this directory exists, and the previous methods of determining the trusted certs directory failed, this directory will be used.
3. **/etc/grid-security/certificates** - This location is intended to be independent of the globus installation ($GLOBUS_LOCATION), and is generally only writeable by the host system administrator.
4. **$GLOBUS_LOCATION/share/certificates**

Parameters

cert_dir	The trusted certificates directory

Returns

GLOBUS_SUCCESS if no error occurred, and a sufficient trusted certificates directory was found. Otherwise, an error object identifier returned.

3.2.2.14 globus_result_t globus_gsi_sysconfig_get_user_cert_filename_unix (char ** *user_cert*, char ** *user_key*)

Get the User Certificate Filename based on the current user's environment.

The following locations are searched for cert and key files in order:

1. environment variables X509_USER_CERT and X509_USER_KEY
2. $HOME/.globus/usercert.pem and $HOME/.globus/userkey.pem
3. $HOME/.globus/usercred.p12 - this is a PKCS12 credential

Parameters

user_cert	pointer the filename of the user certificate
user_key	pointer to the filename of the user key

Returns

GLOBUS_SUCCESS if the cert and key files were found in one of the possible locations, otherwise an error object identifier is returned

3.2.2.15 globus_result_t globus_gsi_sysconfig_get_host_cert_filename_unix (char ** *host_cert,* char ** *host_key*)

Get the Host Certificate and Key Filenames based on the current user's environment.

The host cert and key are searched for in the following locations (in order):

1. X509_USER_CERT and X509_USER_KEY environment variables
2. registry keys x509_user_cert and x509_user_key in software\Globus\GSI
3. \<GLOBUS_LOCATION\>\etc\host[cert|key].pem
4. \<users home directory\>\.globus\host[cert|key].pem

Parameters

host_cert	pointer to the host certificate filename
host_key	pointer to the host key filename

Returns

GLOBUS_SUCCESS if the host cert and key were found, otherwise an error object identifier is returned

3.2.2.16 globus_result_t globus_gsi_sysconfig_get_service_cert_filename_unix (char * *service_name,* char ** *service_cert,* char ** *service_key*)

Get the Service Certificate Filename based on the current user's environment.

The host cert and key are searched for in the following locations (in order):

1. X509_USER_CERT and X509_USER_KEY environment variables
2. \/etc\/grid-security\/{service_name}\/{service_name}[cert|key].pem
3. GLOBUS_LOCATION\/etc\/{service_name}\/{service_name}[cert|key].pem So for example, if my service was named: myservice, the location of the certificate would be: GLOBUS_LOCATIO-N\/etc\/myservice\/myservicecert.pem
4. \<users home\>\/.globus\/{service_name}\/{service_name}[cert|key].pem

Parameters

service_name	The name of the service which allows us to determine the locations of cert and key files to look for
service_cert	pointer to the host certificate filename
service_key	pointer to the host key filename

Returns

GLOBUS_SUCCESS if the service cert and key were found, otherwise an error object identifier

3.2.2.17 globus_result_t globus_gsi_sysconfig_get_proxy_filename_unix (char ** *user_proxy,* globus_gsi_proxy_file_type_t *proxy_file_type*)

Get the proxy cert filename based on the following search order:

1. X509_USER_PROXY environment variable - This environment variable is set by the at run time for the specific application. If the proxy_file_type variable is set to GLOBUS_PROXY_OUTPUT (a proxy filename for writing is requested), and the X509_USER_PROXY is set, this will be the resulting value of the user_proxy filename string passed in. If the proxy_file_type is set to GLOBUS_PROXY_INPUT and X509_USER_PROXY is set, but the file it points to does not exist, or has some other readability issues, the function will continue checking using the other methods available.
2. Check the default location for the proxy file of \/tmp\/x509_u\<user_id\> where \<user id\> is some unique string for that user on the host

Parameters

user_proxy	the proxy filename of the user
proxy_file_type	Switch for determining whether to return a existing proxy filename or if a filename suitable for creating a proxy should be returned

Returns

GLOBUS_SUCCESS or an error object identifier

3.2.2.18 globus_result_t globus_gsi_sysconfig_get_signing_policy_filename_unix (X509_NAME * *ca_name,* char * *cert_dir,* char ** *signing_policy_filename*)

Get the Signing Policy Filename on the current system, based on the CA's subject name, and the trusted certificates directory.

Parameters

ca_name	The X509 subject name of the CA to get the signing policy of. The hash of the CA is generated from this
cert_dir	The trusted CA certificates directory, containing the singing_policy files of the trusted CA's.
signing_policy_filename	The resulting singing_policy filename

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID

3.2.2.19 globus_result_t globus_gsi_sysconfig_get_ca_cert_files_unix (char * *ca_cert_dir,* globus_fifo_t * *ca_cert_list*)

Gets a list of trusted CA certificate filenames in a trusted CA certificate directory.

Parameters

ca_cert_dir	The trusted CA certificate directory to get the filenames from
ca_cert_list	The resulting list of CA certificate filenames. This is a a globus list structure.

See also

globus_fifo_t

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.2.2.20 globus_result_t globus_gsi_sysconfig_remove_all_owned_files_unix (char * *default_filename*)

Removes all proxies (ie.

all delegated and grid-proxy-init generated proxies) found in the secure tmp directory that are owned by the current user.

Parameters

default_filename	The filename of the default proxy

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.2.2.21 globus_result_t globus_gsi_sysconfig_is_superuser_unix (int * *is_superuser*)

Checks whether the current user is root.

Parameters

is_superuser	1 if the user is the superuser 0 if not

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.2.2.22 globus_result_t globus_gsi_sysconfig_get_gridmap_filename_unix (char ** *filename*)

Get the path and file name of the grid map file.

Parameters

filename	Contains the location of the grid map file upon successful return

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.2.2.23 globus_result_t globus_gsi_sysconfig_get_authz_conf_filename_unix (char ** *filename*)

Get the path and file name of the authorization callback configuration file.

Parameters

filename	Contains the location of the authorization callback configuration file upon successful return

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.2.2.24 globus_result_t globus_gsi_sysconfig_get_authz_lib_conf_filename_unix (char ** *filename*)

Get the path and file name of the authorization callback configuration file.

Parameters

filename	Contains the location of the authorization callback configuration file upon successful return

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.2.2.25 globus_result_t globus_gsi_sysconfig_get_gaa_conf_filename_unix (char ** *filename*)

Get the path and file name of the GAA configuration file.

Parameters

filename	Contains the location of the GAA callback configuration file upon successful return

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3 Functions for Win32 platforms

Win32 - Set Key Permissions

- globus_result_t **globus_gsi_sysconfig_set_key_permissions_win32** (char *filename)

Win32 - File Exists

- globus_result_t **globus_gsi_sysconfig_file_exists_win32** (const char *filename)

Win32 - Directory Exists

- globus_result_t **globus_gsi_sysconfig_dir_exists_win32** (const char *filename)

Win32 - Get Current Working Directory

- globus_result_t **globus_gsi_sysconfig_get_current_working_dir_win32** (char **working_dir)

Win32 - Make Absolute Path

- globus_result_t **globus_gsi_sysconfig_make_absolute_path_for_filename_win32** (char *filename, char **absolute_path)

Win32 - Split Directory and Filename

- globus_result_t **globus_gsi_sysconfig_split_dir_and_filename_win32** (char *full_filename, char **dir_string, char **filename_string)

Win32 - Get Trusted CA Cert Dir

- globus_result_t **globus_gsi_sysconfig_get_cert_dir_win32** (char **cert_dir)

Win32 - Get User Certificate Filename

- globus_result_t **globus_gsi_sysconfig_get_user_cert_filename_win32** (char **user_cert, char **user_key)

Win32 - Get Host Certificate and Key Filenames

- globus_result_t **globus_gsi_sysconfig_get_host_cert_filename_win32** (char **host_cert, char **host_key)

Win32 - Get Service Certificate and Key Filenames

- globus_result_t **globus_gsi_sysconfig_get_service_cert_filename_win32** (char *service_name, char **service_cert, char **service_key)

Win32 - Get Proxy Filename

- globus_result_t **globus_gsi_sysconfig_get_proxy_filename_win32** (char **user_proxy, **globus_gsi_proxy_file_type_t** proxy_file_type)

Win32 - Get CA Cert Filenames

- globus_result_t **globus_gsi_sysconfig_get_ca_cert_files_win32** (char *ca_cert_dir, globus_fifo_t *ca_cert_list)

Win32 - Remove all proxies owned by current uid

- globus_result_t **globus_gsi_sysconfig_remove_all_owned_files_win32** (char *default_filename)

Win32 - Get the path and file name of the grid map file

- globus_result_t **globus_gsi_sysconfig_get_gridmap_filename_win32** (char **filename)
- globus_result_t **globus_gsi_sysconfig_get_authz_conf_filename_win32** (char **filename)

Win32 - Get the path and file name of the gaa config file

- globus_result_t **globus_gsi_sysconfig_get_gaa_conf_filename_win32** (char **filename)

Win32 - Check if the current user is root

- globus_result_t **globus_gsi_sysconfig_is_superuser_win32** (int *is_superuser)

Win32 - Get Signing Policy Filename

- globus_result_t **globus_gsi_sysconfig_get_signing_policy_filename_win32** (X509_NAME *ca_name, char *cert_dir, char **signing_policy_filename)

3.3.1 Detailed Description

These functions implement the Win32 version of the Globus GSI System Configuration API. **They should never be called directly, please use the provided platform independent defines.**

3.3.2 Function Documentation

3.3.2.1 globus_result_t globus_gsi_sysconfig_set_key_permissions_win32 (char * *filename*)

Set the file permissions of a file to read only by the user which are the permissions that should be set for all private keys.

Parameters

filename	

Returns

GLOBUS_SUCCESS or an error object id

3.3.2.2 globus_result_t globus_gsi_sysconfig_file_exists_win32 (const char * *filename*)

Check that the file exists.

Parameters

filename	the file to check

Returns

GLOBUS_SUCCESS (even if the file doesn't exist) - in some abortive cases an error object identifier is returned

3.3.2.3 globus_result_t globus_gsi_sysconfig_dir_exists_win32 (const char * *filename*)

Check that the directory exists.

Parameters

filename	the file to check

Returns

GLOBUS_SUCCESS if the directory exists, otherwise an error object identifier.

3.3.2.4 globus_result_t globus_gsi_sysconfig_get_current_working_dir_win32 (char ** *working_dir*)

Get the current working directory on a windows system.

Parameters

working_dir	The working directory to get

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.5 globus_result_t globus_gsi_sysconfig_make_absolute_path_for_filename_win32 (char * *filename,* char ** *absolute_path*)

Make the filename into an absolute path string based on the current working directory.

Parameters

filename	the filename to get the absolute path of.
absolute_path	The resulting absolute path

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.6 globus_result_t globus_gsi_sysconfig_split_dir_and_filename_win32 (char * *full_filename,* char ** *dir_string,* char ** *filename_string*)

Split the directory and filename portions of a filename string into two separate strings.

Parameters

full_filename	
dir_string	
filename_string	

Returns

3.3.2.7 globus_result_t globus_gsi_sysconfig_get_cert_dir_win32 (char ** *cert_dir*)

Get the Trusted Certificate Directory containing the trusted Certificate Authority certificates.

This directory is determined in the order shown below. Failure in one method results in attempting the next.

1. **X509_CERT_DIR environment variable** - if this is set, the trusted certificates will be searched for in that directory. This variable allows the end user to specify the location of trusted certificates.
2. **x509_cert_dir registry key** - If this registry key is set on windows, the directory it points to should contain the trusted certificates. The path to the registry key is software\Globus\GSI
3. \<**user home directory**>**.globus****certificates** - If this directory exists, and the previous methods of determining the trusted certs directory failed, this directory will be used.
4. **Host Trusted Cert Dir** - This location is intended to be independent of the globus installation ($GLOBUS_LOCATION), and is generally only writeable by the host system administrator.
5. **Globus Install Trusted Cert Dir** - this is $GLOBUS_LOCATION\share\certificates.

Parameters

cert_dir	The trusted certificates directory

Returns

GLOBUS_SUCCESS if no error occurred, and a sufficient trusted certificates directory was found. Otherwise, an error object identifier returned.

3.3.2.8 globus_result_t globus_gsi_sysconfig_get_user_cert_filename_win32 (char ** *user_cert,* char ** *user_key*)

Get the User Certificate Filename based on the current user's environment.

The following locations are searched for cert and key files in order:

1. environment variables X509_USER_CERT and X509_USER_KEY

2. registry keys x509_user_cert and x509_user_key in software\Globus\GSI
3. <users home directory>\.globus\usercert.pem and <users home directory>\.globus\userkey.pem
4. <users home directory>\.globus\usercred.p12 - this is a PKCS12 credential

Parameters

user_cert	pointer the filename of the user certificate
user_key	pointer to the filename of the user key

Returns

GLOBUS_SUCCESS if the cert and key files were found in one of the possible locations, otherwise an error object identifier is returned

3.3.2.9 globus_result_t globus_gsi_sysconfig_get_host_cert_filename_win32 (char ** *host_cert,* char ** *host_key*)

Get the Host Certificate and Key Filenames based on the current user's environment.

The host cert and key are searched for in the following locations (in order):

1. X509_USER_CERT and X509_USER_KEY environment variables
2. registry keys x509_user_cert and x509_user_key in software\Globus\GSI
3. <GLOBUS_LOCATION>\etc\host[cert|key].pem
4. <users home directory>\.globus\host[cert|key].pem

Parameters

host_cert	pointer to the host certificate filename
host_key	pointer to the host key filename

Returns

GLOBUS_SUCCESS if the host cert and key were found, otherwise an error object identifier is returned

3.3.2.10 globus_result_t globus_gsi_sysconfig_get_service_cert_filename_win32 (char * *service_name,* char ** *service_cert,* char ** *service_key*)

Get the Service Certificate Filename based on the current user's environment.

The host cert and key are searched for in the following locations (in order):

1. X509_USER_CERT and X509_USER_KEY environment variables
2. registry keys x509_user_cert and x509_user_key in software\Globus\GSI
3. GLOBUS_LOCATION\etc\{service_name}\{service_name}[cert|key].pem So for example, if my service was named: myservice, the location of the certificate would be: GLOBUS_LOCATION\etc\myservice\myservicecert.-pem
4. <users home>\.globus\{service_name}\{service_name}[cert|key].pem

Parameters

service_name	The name of the service which allows us to determine the locations of cert and key files to look for
service_cert	pointer to the host certificate filename
service_key	pointer to the host key filename

Returns

GLOBUS_SUCCESS if the service cert and key were found, otherwise an error object identifier

3.3.2.11 globus_result_t globus_gsi_sysconfig_get_proxy_filename_win32 (char ** *user_proxy,* globus_gsi_proxy_file_type_t *proxy_file_type*)

Get the proxy cert filename based on the following search order:

1. X509_USER_PROXY environment variable - This environment variable is set by the at run time for the specific application. If the proxy_file_type variable is set to GLOBUS_PROXY_OUTPUT (a proxy filename for writing is requested), and the X509_USER_PROXY is set, this will be the resulting value of the user_proxy filename string passed in. If the proxy_file_type is set to GLOBUS_PROXY_INPUT and X509_USER_PROXY is set, but the file it points to does not exist, or has some other readability issues, the function will continue checking using the other methods available.
2. check the registry key: x509_user_proxy. Just as with the environment variable, if the registry key is set, and proxy_file_type is GLOBUS_PROXY_OUTPUT, the string set to be the proxy filename will be this registry key's value. If proxy_file_type is GLOBUS_PROXY_INPUT, and the file doesn't exist, the function will check the next method for the proxy's filename.
3. Check the default location for the proxy file. The default location should be set to reside in the temp directory on that host, with the filename taking the format: x509_u<user id> where <user id> is some unique string for that user on the host

Parameters

user_proxy	the proxy filename of the user
proxy_file_type	Switch for determining whether to return a existing proxy filename or if a filename suitable for creating a proxy should be returned

Returns

GLOBUS_SUCCESS or an error object identifier

3.3.2.12 globus_result_t globus_gsi_sysconfig_get_ca_cert_files_win32 (char * *ca_cert_dir,* globus_fifo_t * *ca_cert_list*)

Gets a list of trusted CA certificate filenames in a trusted CA certificate directory.

Parameters

ca_cert_dir	The trusted CA certificate directory to get the filenames from
ca_cert_list	The resulting list of CA certificate filenames. This is a a globus list structure.

See also

globus_fifo_t

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.13 globus_result_t globus_gsi_sysconfig_remove_all_owned_files_win32 (char * *default_filename*)

Removes all proxies (ie.

all delegated and grid-proxy-init generated proxies) found in the secure tmp directory that are owned by the current user.

Parameters

default_filename	The filename of the default proxy

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.14 globus_result_t globus_gsi_sysconfig_get_gridmap_filename_win32 (char ** *filename*)

Get the path and file name of the grid map file.

Parameters

filename	Contains the location of the grid map file upon successful return

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.15 globus_result_t globus_gsi_sysconfig_get_authz_conf_filename_win32 (char ** *filename*)

Get the path and file name of the authorization callback configuration file.

Parameters

filename	Contains the location of the authorization callback configuration file upon successful return

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.16 globus_result_t globus_gsi_sysconfig_get_gaa_conf_filename_win32 (char ** *filename*)

Get the path and file name of the gaa config configuration file .

Parameters

filename	Contains the location of the authorization callback configuration file upon successful return

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.17 globus_result_t globus_gsi_sysconfig_is_superuser_win32 (int * *is_superuser*)

Checks whether the current user is root.

Parameters

is_superuser	1 if the user is the superuser 0 if not

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.18 globus_result_t globus_gsi_sysconfig_get_signing_policy_filename_win32 (X509_NAME * *ca_name,* char * *cert_dir,* char ** *signing_policy_filename*)

Get the Signing Policy Filename on the current system, based on the CA's subject name, and the trusted certificates directory.

Parameters

ca_name	The X509 subject name of the CA to get the signing policy of. The hash of the CA is generated from this
cert_dir	The trusted CA certificates directory, containing the singing_policy files of the trusted CA's.
signing_policy_filename	The resulting singing_policy filename

Returns

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID

3.4 Functions for all platforms

Get Unique Proxy Filename

- globus_result_t **globus_gsi_sysconfig_get_unique_proxy_filename** (char **unique_filename)

3.4.1 Detailed Description

These functions are platform independent members of the Globus GSI System Configuration API.

3.4.2 Function Documentation

3.4.2.1 globus_result_t globus_gsi_sysconfig_get_unique_proxy_filename (char ** *unique_filename*)

Get a unique proxy cert filename.

This is mostly used for delegated proxy credentials. Each filename returned is going to be unique for each time the function is called.

Parameters

unique_filename	the unique filename for a delegated proxy cert

Returns

GLOBUS_SUCCESS or an error object identifier

3.5 Activation

Defines

- #define **GLOBUS_GSI_SYSCONFIG_MODULE**

3.5.1 Detailed Description

Globus GSI System Configuration API uses standard Globus module activation and deactivation. Before any - Globus GSI System Configuration API functions are called, the following function must be called:

```
globus_module_activate(GLOBUS_GSI_SYSCONFIG_MODULE)
```

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

To deactivate Globus GSI System Configuration API, the following function must be called:

```
globus_module_deactivate(GLOBUS_GSI_SYSCONFIG_MODULE)
```

This function should be called once for each time Globus GSI System Configuration API was activated.

3.5.2 Define Documentation

3.5.2.1 #define GLOBUS_GSI_SYSCONFIG_MODULE

Module descriptor.

3.6 Datatypes

Enumerations

- enum **globus_gsi_sysconfig_error_t** { **GLOBUS_GSI_SYSCONFIG_ERROR_SUCCESS** = 0, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_DIR** = 1, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_STRING** = 2, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_KEY_STRING** = 3, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_HOME_DIR** = 4, **GLOBUS_GSI_SYSCONFIG_ERROR_ERRNO** = 5, **GLOBUS_GSI_SYSCONFIG_ERROR_CHECKING_FILE_EXISTS** = 6, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_FILENAME** = 7, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_PROXY_FILENAME** = 8, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_DELEG_FILENAME** = 9, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CA_CERT_FILENAMES** = 10, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CWD** = 11, **GLOBUS_GSI_SYSCONFIG_ERROR_REMOVING_OWNED_FILES** = 12, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_GRIDMAP_FILENAME** = 13, **GLOBUS_GSI_SYSCONFIG_ERROR_CHECKING_SUPERUSER** = 14, **GLOBUS_GSI_SYSCONFIG_ERROR_SETTING_PERMS** = 15, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_SIGNING_POLICY** = 16, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_PW_ENTRY** = 17, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_AUTHZ_FILENAME** = 18, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_NOT_REGULAR** = 19, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_DOES_NOT_EXIST** = 20, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_BAD_PERMISSIONS** = 21, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_NOT_OWNED** = 22, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_IS_DIR** = 23, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_ZERO_LENGTH** = 24, **GLOBUS_GSI_SYSCONFIG_INVALID_ARG** = 25, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_HAS_LINKS** = 26, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_HAS_CHANGED** = 27, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_AUTHZ_LIB_FILENAME** = 28, **GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_GAA_FILENAME** = 29, **GLOBUS_GSI_SYSCONFIG_ERROR_FILE_NOT_DIR** = 30, **GLOBUS_GSI_SYSCONFIG_ERROR_LAST** = 31 }
- enum **globus_gsi_proxy_file_type_t** { **GLOBUS_PROXY_FILE_INPUT**, **GLOBUS_PROXY_FILE_OUTPUT** }

3.6.1 Enumeration Type Documentation

3.6.1.1 enum globus_gsi_sysconfig_error_t

GSI System Config Error codes.

Enumerator:

GLOBUS_GSI_SYSCONFIG_ERROR_SUCCESS Success - never used.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_DIR Unable to determine trusted certificates directory.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_STRING Error while generating certificate filename.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_KEY_STRING Error while generating private key filename.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_HOME_DIR Unable to determine user's home directory.

GLOBUS_GSI_SYSCONFIG_ERROR_ERRNO System Error -- see underlying error for details.

GLOBUS_GSI_SYSCONFIG_ERROR_CHECKING_FILE_EXISTS Unable to determine whether file exists.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_FILENAME Unable to determine the location of the certificate file.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_PROXY_FILENAME Unable to determine the location of the proxy file.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_DELEG_FILENAME Unable to determine the location of the delegated proxy file.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CA_CERT_FILENAMES Unable to generate a list of CA certificate filenames.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CWD Error while dircovering the current working directory.

GLOBUS_GSI_SYSCONFIG_ERROR_REMOVING_OWNED_FILES Failed to remove all proxy files.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_GRIDMAP_FILENAME Unable to determine the location of the grid map file.

GLOBUS_GSI_SYSCONFIG_ERROR_CHECKING_SUPERUSER Failure while checking whether the current user is the super user.

GLOBUS_GSI_SYSCONFIG_ERROR_SETTING_PERMS Error while trying to set file permissions.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_SIGNING_POLICY Unable to determine the location of a signing policy file.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_PW_ENTRY Could not find password entry for user.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_AUTHZ_FILENAME Failed to locate the authorization callout configuration file.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_NOT_REGULAR File is not a regular file.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_DOES_NOT_EXIST File does not exist.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_BAD_PERMISSIONS File has incorrect permissions for operation.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_NOT_OWNED File is not owned by current user.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_IS_DIR File is a directory.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_ZERO_LENGTH File has zero length.

GLOBUS_GSI_SYSCONFIG_INVALID_ARG Invalid argument.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_HAS_LINKS File has more than one link.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_HAS_CHANGED File has changed in the meantime.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_AUTHZ_LIB_FILENAME Failed to locate the authorization callout library configuration file.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_GAA_FILENAME Failed to locate the gaa configuration file.

GLOBUS_GSI_SYSCONFIG_ERROR_FILE_NOT_DIR File is not a directory.

GLOBUS_GSI_SYSCONFIG_ERROR_LAST Last marker - never used.

3.6.1.2 enum globus_gsi_proxy_file_type_t

Enumerator used to keep track of input/output types of filenames.

Enumerator:

GLOBUS_PROXY_FILE_INPUT The proxy filename is intended for reading (it should already exist)

GLOBUS_PROXY_FILE_OUTPUT The proxy filename is intended for writing (it does not need to exist)