

globus gsi syscon g Reference Manual

2.2

Generated by Doxygen 1.3.5

Fri Jun 26 23:26:24 2009

Contents

1 Globus GSI System Config API	1
2 globus_gsi_sysconfig Module Index	1
3 globus_gsi_sysconfig Module Documentation	1

1 Globus GSI System Config API

This API provides helper functions for detecting installation and environment specific settings applicable to GSI. It also serves as a abstraction layer for OS specific programming details. This is achieved by defining preprocessor symbols that point at the correct platform specific functions. 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 globus_gsi_sysconfig Module Index

2.1 globus_gsi_sysconfig Modules

Here is a list of all modules:

Defines	1
Functions for UNIX platforms	5
Functions for Win32 platforms	14
Functions for all platforms	21
Activation	21
Datatypes	22

3 globus_gsi_sysconfig Module Documentation

3.1 Defines

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

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\(\)](#) and [globus_gsi_sysconfig_set_key_permissions_win32\(\)](#)

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\(\)](#) 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\(\)](#) 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_certificate_unix\(\)](#) and [globus_gsi_sysconfig_check_certificate_win32\(\)](#)

3.1.2.5 #de ne GLOBUS_GSI_SYSCONFIG_FILE_EXISTS

Check whether a given le exists.

See [globus_gsi_syscon g_le_exists_unix\(\)](#) and [globus_gsi_syscon g_le_exists_win32\(\)](#)

3.1.2.6 #de ne GLOBUS_GSI_SYSCONFIG_DIR_EXISTS

Check whether a given directory exists.

See [globus_gsi_syscon g_dir_exists_unix\(\)](#) and [globus_gsi_syscon g_dir_exists_win32\(\)](#)

3.1.2.7 #de ne GLOBUS_GSI_SYSCONFIG_GET_CERT_DIR

Determine the location of the trusted certi cates directory.

See [globus_gsi_syscon g_get_cert_dir_unix\(\)](#) and [globus_gsi_syscon g_get_cert_dir_win32\(\)](#)

3.1.2.8 #de ne GLOBUS_GSI_SYSCONFIG_GET_USER_CERT_FILENAME

Determine the location of the users certi cate and private key.

See [globus_gsi_syscon g_get_user_cert_lename_unix\(\)](#) and [globus_gsi_syscon g_get_user_cert_lename_win32\(\)](#)

3.1.2.9 #de ne GLOBUS_GSI_SYSCONFIG_GET_HOST_CERT_FILENAME

Determine the location of the host certi cate and private key.

See [globus_gsi_syscon g_get_host_cert_lename_unix\(\)](#) and [globus_gsi_syscon g_get_host_cert_lename_win32\(\)](#)

3.1.2.10 #de ne GLOBUS_GSI_SYSCONFIG_GET_SERVICE_CERT_FILENAME

Determine the location of a service certi cate and private key.

See [globus_gsi_syscon g_get_service_cert_lename_unix\(\)](#) and [globus_gsi_syscon g_get_service_cert_lename_win32\(\)](#)

3.1.2.11 #de ne GLOBUS_GSI_SYSCONFIG_GET_PROXY_FILENAME

Determine the location of a proxy certi cate and private key.

See [globus_gsi_syscon g_get_proxy_lename_unix\(\)](#) and [globus_gsi_syscon g_get_proxy_lename_win32\(\)](#)

3.1.2.12 #de ne GLOBUS_GSI_SYSCONFIG_GET_SIGNING_POLICY_FILENAME

Determine the name of the signing policy le for a given CA.

See [globus_gsi_syscon g_get_signing_policy_lename_unix\(\)](#) and [globus_gsi_syscon g_get_signing_policy_lename_win32\(\)](#)

3.1.2.13 #de ne GLOBUS_GSI_SYSCONFIG_GET_CA_CERT_FILES

Get a list of of trusted CA certi cate lenames in a trusted CA certi cate directory.

See [globus_gsi_syscon g_get_ca_cert_les_unix\(\)](#) and [globus_gsi_syscon g_get_ca_cert_les_win32\(\)](#)

3.1.2.14 #de ne GLOBUS_GSI_SYSCONFIG_GET_CURRENT_WORKING_DIR

Get the current working directory.

See [globus_gsi_syscon g_get_current_working_dir_unix\(\)](#) and [globus_gsi_syscon g_get_current_working_dir_win32\(\)](#)

3.1.2.15 #de ne GLOBUS_GSI_SYSCONFIG_MAKE_ABSOLUTE_PATH_FOR_FILENAME

Prepend the current working directory to the give lename.

See [globus_gsi_syscon g_make_absolute_path_for_lename_unix\(\)](#) and [globus_gsi_syscon g_make_absolute_path_for_lename_win32\(\)](#)

3.1.2.16 #de ne GLOBUS_GSI_SYSCONFIG_SPLIT_DIR_AND_FILENAME

Split directory component of path from lename.

See [globus_gsi_syscon g_split_dir_and_lename_unix\(\)](#) and [globus_gsi_syscon g_split_dir_and_lename_win32\(\)](#)

3.1.2.17 #de ne GLOBUS_GSI_SYSCONFIG_REMOVE_ALL_OWNED_FILES

Remove all proxies owned by current uid.

See [globus_gsi_syscon g_remove_all_owned_files_unix\(\)](#) and [globus_gsi_syscon g_remove_all_owned_files_win32\(\)](#)

3.1.2.18 #de ne GLOBUS_GSI_SYSCONFIG_GET_GRIDMAP_FILENAME

Determine the location of the grid map file.

See [globus_gsi_syscon g_get_gridmap_filename_unix\(\)](#) and [globus_gsi_syscon g_get_gridmap_filename_win32\(\)](#)

3.1.2.19 #de ne GLOBUS_GSI_SYSCONFIG_GET_AUTHZ_CONF_FILENAME

Determine the location of the authorization callout config file.

See [globus_gsi_syscon g_get_authz_conf_filename_unix\(\)](#)

3.1.2.20 #de ne GLOBUS_GSI_SYSCONFIG_GET_GAA_CONF_FILENAME

Determine the location of the GAA callout config file.

See [globus_gsi_syscon g_get_gaa_conf_filename_unix\(\)](#)

3.1.2.21 #de ne GLOBUS_GSI_SYSCONFIG_IS_SUPERUSER

Determine whether the current user is the super user.

See [globus_gsi_syscon g_is_superuser_unix\(\)](#) and [globus_gsi_syscon g_is_superuser_win32\(\)](#)

3.1.2.22 #de ne GLOBUS_GSI_SYSCONFIG_GET_USER_ID_STRING

Get the current UID in string form.

See [globus_gsi_syscon g_get_user_id_string_unix\(\)](#) and [globus_gsi_syscon g_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\(\)](#) 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\(\)](#) and [globus_gsi_sysconfig_get_username_win32\(\)](#)

3.1.2.25 #define GLOBUS_GSI_SYSCONFIG_GET_UNIQUE_PROXY_FILENAME

Generate a unique proxy file name.

See [globus_gsi_sysconfig_get_unique_proxy_filename\(\)](#)

3.2 Functions for UNIX platforms

These functions implement the UNIX version of the Globus GSI System Configuration API.

UNIX - Set Key Permissions

- `globus_result_t globus_gsi_sysconfig_set_key_permissions(char filename)`

UNIX - Get User ID

- `globus_result_t globus_gsi_sysconfig_get_user_id_string(char user_id_string)`

UNIX - Get Username

- `globus_result_t globus_gsi_sysconfig_get_username(char username)`

UNIX - Get Process ID

- `globus_result_t globus_gsi_sysconfig_get_proc_id_string(char proc_id_string)`

UNIX - Make Absolute Path

- `globus_result_t globus_gsi_sysconfig_make_absolute_path_for_filename(char filename, char absolute_path)`

UNIX - Split Directory and Filename

- `globus_result_t globus_gsi_sysconfig_split_dir_and_filename(char full_filename, char dir_string, char filename_string)`

UNIX - Get Current Working Directory

- `globus_result_t globus_gsi_sysconfig_get_current_working_dir(char working_dir)`

UNIX - Get HOME Directory

- globus_result_t [globus_gsi_syscon_g_get_home_dir_url\(char home_dir\)](#)

UNIX - File Exists

- globus_result_t [globus_gsi_syscon_g_le_exists_url\(const char lename\)](#)

UNIX - Directory Exists

- globus_result_t [globus_gsi_syscon_g_dir_exists_url\(const char lename\)](#)

UNIX - Check File Status for Key

- globus_result_t [globus_gsi_syscon_g_check_key_le_url\(const char lename\)](#)

UNIX - Check File Status for Cert

- globus_result_t [globus_gsi_syscon_g_check_cert_le_url\(const char lename\)](#)

UNIX - Get Trusted CA Cert Dir

- globus_result_t [globus_gsi_syscon_g_get_cert_dir_url\(char cert_dir\)](#)

UNIX - Get User Certificate and Key Filenames

- globus_result_t [globus_gsi_syscon_g_get_user_cert_lename_url\(char user_cert, char user_key\)](#)

UNIX - Get Host Certificate and Key Filenames

- globus_result_t [globus_gsi_syscon_g_get_host_cert_lename_url\(char host_cert, char host_key\)](#)

UNIX - Get Service Certificate and Key Filenames

- globus_result_t [globus_gsi_syscon_g_get_service_cert_lename_url\(char service_name, char service_cert, char service_key\)](#)

UNIX - Get Proxy Filename

- globus_result_t [globus_gsi_syscon_g_get_proxy_lename_url\(char user_proxy, globus_gsi_proxy_le_type_t proxy_le_type\)](#)

UNIX - Get Signing Policy Filename

- globus_result_t [globus_gsi_syscon_g_get_signing_policy_lename_url\(X509_NAME ca_name, char cert_dir, char signing_policy_lename\)](#)

UNIX - Get CA Cert Filenames

- globus_result_t globus_gsi_sysconfig_get_ca_cert_filenames(char ca_cert_dir, globus_file_t ca_cert_list)

UNIX - Remove all proxies owned by current uid

- globus_result_t globus_gsi_sysconfig_remove_all_owned_files(char default_filename)

UNIX - Check if the current user is root

- globus_result_t globus_gsi_sysconfig_is_superuser(int is_superuser)

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

- globus_result_t globus_gsi_sysconfig_get_gridmap_filename(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(char filename)

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

- globus_result_t globus_gsi_sysconfig_get_authz_lib_conf_filename(char filename)

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

- globus_result_t globus_gsi_sysconfig_get_gaa_conf_filename(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_syscon g_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_syscon g_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_syscon g_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_syscon g_make_absolute_path_for_lename_unix (char lename, char absolute_path)

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

Parameters:

lename the lename 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_syscon g_split_dir_and_lename_unix (char full_lename, char dir_string, char lename_string)

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

Parameters:

full_lename The lename to split. Splits on the last occurrence of '/' where the directory is everything before the last '/', and the lename is everything after.

`dir_string` The directory portion of the `lename` string. If no '/' is found throughout the string, this variable points to NULL. This needs to be freed when no longer needed.

`lename_string` The `lename` portion of the `lename` string. If no '/' is found throughout, this variable is a duplicate of the full `lename` 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_syscon g_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_syscon g_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 occurred, otherwise an error object is returned.

3.2.2.9 `globus_result_t globus_gsi_syscon g_le_exists_unix (const char lename)`

Check if the `le` exists.

Parameters:

`lename` The `lename` of the `le` to check for

Returns:

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

3.2.2.10 `globus_result_t globus_gsi_syscon g_dir_exists_unix (const char lename)`

Check if the directory exists.

Parameters:

`lename` The `lename` 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_syscon g_check_key le_unix (const char lename)

This is a convenience function used to check the status of a private key le. The desired status is only the current user has ownership and read permissions, everyone else should not be able to access it.

Parameters:

lename The name of the le to check the status of

Returns:

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

3.2.2.12 globus_result_t globus_gsi_syscon g_check_cert le_unix (const char lename)

This is a convenience function used to check the status of a certificate le. The desired status is the current user has ownership and read/write permissions, while group and others only have read permissions.

Parameters:

lename The name of the le to check the status of

Returns:

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

3.2.2.13 globus_result_t globus_gsi_syscon g_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_syscon g_get_user_cert_lenam_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 les 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 to the lename of the user certificate
 user_key pointer to the lename 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_syscon_g_get_host_cert_lenam_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\GlobusGSI
3. n< GLOBUS_LOCATION>n\etomhost[certkey].pem
4. n< users home directory>n\globus\host[certkey].pem

Parameters:

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

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_syscon_g_get_service_cert_lenam_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. n\etom\grid-security\{service_name}\{service_name}[certkey].pem
3. GLOBUS_LOCATIONn\etom\{service_name}\{service_name}[certkey].pem So for example, if my service was named: myservice, the location of the certificate would be: GLOBUS_LOCATIONn\etom\myservice\myservicecert.pem
4. n< users home>n\globus\{service_name}\{service_name}[certkey].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 lename
 service_key pointer to the host key lename

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_syscon g_get_proxy_lease_unix (char user_proxy globus\_gsi\_proxy\_lease\_t proxy_lease_t)`

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

1. X509_USER_PROXY environment variable - This environment variable is set by the application at run time for the specific application. If the proxy_lease variable is set to GLOBUS_PROXY_OUTPUT (a proxy lease filename for writing is requested), and the X509_USER_PROXY is set, this will be the resulting value of the user_proxy_lease string passed in. If the proxy_lease 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 lease `/tmp/x509_un<user_id>` where `<user_id>` is some unique string for that user on the host

Parameters:

user_proxy the proxy lease filename of the user

proxy_lease Switch for determining whether to return an existing proxy lease filename or if a lease 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_syscon g_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 signing_policy files of the trusted CA's.

signing_policy_filename The resulting signing_policy filename

Returns:

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID

3.2.2.19 `globus_result_t globus_gsi_syscon g_get_ca_cert_filenames_unix (char ca_cert_dir, globus_fo_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 globus list structure.

See also:

`globus_fo_t`

Returns:

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

3.2.2.20 globus_result_t globus_gsi_syscon g_remove_all_owned_les_unix (char default_les_name)

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_les_name The les_name 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_syscon g_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_syscon g_get_gridmap_les_name_unix (char les_name)

Get the path and les_name of the grid map les_name.

Parameters:

les_name Contains the location of the grid map les_name 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_syscon g_get_authz_conf_les_name_unix (char les_name)

Get the path and les_name of the authorization callback configuration les_name

Parameters:

les_name Contains the location of the authorization callback configuration les_name 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_syscon g_get_authz_lib_conf_les_name_unix (char les_name)

Get the path and les_name of the authorization callback configuration les_name

Parameters:

les_name Contains the location of the authorization callback configuration les_name 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_syscon_g_get_gaa_conf_lename_unix (char lename)

Get the path and le name of the GAA con guration le.

Parameters:

lename Contains the location of the GAA callback con guration le upon successful return

Returns:

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

3.3 Functions for Win32 platforms

These functions implement the Win32 version of the Globus GSI System Con guration API.

Win32 - Set Key Permissions

- globus_result_t globus_gsi_syscon_g_set_key_permissions_win32 (char lename)

Win32 - File Exists

- globus_result_t globus_gsi_syscon_g_le_exists_win32 (const char lename)

Win32 - Directory Exists

- globus_result_t globus_gsi_syscon_g_dir_exists_win32 (const char lename)

Win32 - Get Current Working Directory

- globus_result_t globus_gsi_syscon_g_get_current_working_dir_win32 (char working_dir)

Win32 - Make Absolute Path

- globus_result_t globus_gsi_syscon_g_make_absolute_path_for_lename_win32 (char lename, char absolute_path)

Win32 - Split Directory and Filename

- globus_result_t globus_gsi_syscon_g_split_dir_and_lename_win32 (char full_lename, char dir_string, char lename_string)

Win32 - Get Trusted CA Cert Dir

- globus_result_t globus_gsi_syscon_g_get_cert_dir_win32 (char cert_dir)

Win32 - Get User Certificate Filename

- globus_result_t globus_gsi_syscon_g_get_user_cert_lename_win32 (char user_cert, char user_key)

Win32 - Get Host Certificate and Key Filenames

- globus_result_t globus_gsi_syscon_g_get_host_cert_lename_win32(char host_cert, char host_key)

Win32 - Get Service Certificate and Key Filenames

- globus_result_t globus_gsi_syscon_g_get_service_cert_lename_win32(char service_name, char service_cert, char service_key)

Win32 - Get Proxy Filename

- globus_result_t globus_gsi_syscon_g_get_proxy_lename_win32(char user_proxy, globus_gsi_proxy_le_type_t proxy_le_type)

Win32 - Get CA Cert Filenames

- globus_result_t globus_gsi_syscon_g_get_ca_cert_lenames_win32(char ca_cert_dir, globus_fo_t ca_cert_list)

Win32 - Remove all proxies owned by current uid

- globus_result_t globus_gsi_syscon_g_remove_all_owned_lenames_win32(char default_lename)

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

- globus_result_t globus_gsi_syscon_g_get_gridmap_lename_win32(char lename)

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

- globus_result_t globus_gsi_syscon_g_get_authz_conf_lename_win32(char lename)

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

- globus_result_t globus_gsi_syscon_g_get_gaa_conf_lename_win32(char lename)

Win32 - Check if the current user is root

- globus_result_t globus_gsi_syscon_g_is_superuser_win32(int is_superuser)

Win32 - Get Signing Policy Filename

- globus_result_t globus_gsi_syscon_g_get_signing_policy_lename_win32(X509_NAME ca_name, char cert_dir, char signing_policy_lename)

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 descriptors.

3.3.2 Function Documentation

3.3.2.1 globus_result_t globus_gsi_syscon g_set_key_permissions_win32 (char* lename)

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

Parameters:

lename

Returns:

GLOBUS_SUCCESS or an error object id

3.3.2.2 globus_result_t globus_gsi_syscon g_file_exists_win32 (const char* lename)

Check that the file exists.

Parameters:

lename 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_syscon g_dir_exists_win32 (const char* lename)

Check that the directory exists.

Parameters:

lename 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_syscon g_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_syscon g_make_absolute_path_for_file_win32 (char* lename, char* absolute_path)

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

Parameters:

lename the file name 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_syscon g_split_dir_and_ lename_win32 (char full_ lename, char dir_string, char lename_string)`

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

Parameters:

full_ lename
dir_string
lename_string

Returns:

3.3.2.7 `globus_result_t globus_gsi_syscon g_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`
3. `%user home directory%\n.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_syscon g_get_user_cert_ lename_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`
3. `<users home directory%\n.globus\usercert.pem` and `<users home directory%\n.globus\userkey.pem`
4. `<users home directory%\n.globus\usercred.p12` - this is a PKCS12 credential

Parameters:

user_cert pointer to the lename of the user certificate
 user_key pointer to the lename 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_lename_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\GlobusGSI
3. < GLOBUS_LOCATION > net\host[cert|key].pem
4. < users home directory > n.globus\host[cert|key].pem

Parameters:

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

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_lename_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\GlobusGSI
3. GLOBUS_LOCATION net\{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 net\myservice\myservicecert.pem
4. < users home directory > n.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 lename
 service_key pointer to the host key lename

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_syscon g_get_proxy_lease_win32 (char user_proxy_lease_type proxy_lease_type)`

Get the proxy lease based on the following search order:

1. X509_USER_PROXY environment variable - This environment variable is set by the application at run time for the specific application. If the proxy_lease_type variable is set to GLOBUS_PROXY_OUTPUT (a proxy lease for writing is requested), and the X509_USER_PROXY is set, this will be the resulting value of the user_proxy_lease string passed in. If the proxy_lease_type is set to GLOBUS_PROXY_INPUT and X509_USER_PROXY is set, but the lease 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_lease_type is GLOBUS_PROXY_OUTPUT, the string set to be the proxy lease will be this registry key's value. If proxy_lease_type is GLOBUS_PROXY_INPUT, and the lease doesn't exist, the function will check the next method for the proxy's lease.
3. Check the default location for the proxy lease. The default location should be set to reside in the temp directory on that host, with the lease taking the format: x509_<user id> where <user id> is some unique string for that user on the host

Parameters:

user_proxy the proxy lease of the user

proxy_lease_type Switch for determining whether to return an existing proxy lease or if a lease 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_syscon g_get_ca_cert_list_win32 (char ca_cert_dir, globus_file_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 globus list structure.

See also:

globus_file_t

Returns:

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

3.3.2.13 `globus_result_t globus_gsi_syscon g_remove_all_owned_leases_win32 (char default_lease)`

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_lease 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_syscon g_get_gridmap_ lename_win32 (char lename)

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

Parameters:

lename Contains the location of the grid map le 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_syscon g_get_authz_conf_ lename_win32 (char lename)

Get the path and le name of the authorization callback con guration le

Parameters:

lename Contains the location of the authorization callback con guration le 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_syscon g_get_gaa_conf_ lename_win32 (char lename)

Get the path and le name of the gaa con g con guration le .

Parameters:

lename Contains the location of the authorization callback con guration le 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_syscon g_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_syscon g_get_signing_policy_ lename_win32 (X509_NAME* ca_name, char* cert_dir, char* signing_policy_ lename)

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 signing_policy files of the trusted CA's.

signing_policy_ lename The resulting signing_policy lename

Returns:

GLOBUS_SUCCESS if no error occurred, otherwise an error object ID

3.4 Functions for all platforms

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

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

Globus GSI System Configuration API uses standard Globus module activation and deactivation.

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 De ne Documentation

3.5.2.1 #define GLOBUS_GSI_SYSCONFIG_MODULE

Module descriptor.

3.6 Datatypes

Enumerations

```
enum globus_gsi_syscon_g_error {
    GLOBUS_GSI_SYSCONFIG_ERROR_SUCCESS = 0,
    GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_DATA = 1,
    GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_STRING = 2,
    GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_KEY_STRING = 3,
    GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_HOME_DATA = 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_FILENAME = 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_PERMISSIONS = 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_PERMISSION = 21,
    GLOBUS_GSI_SYSCONFIG_ERROR_FILE_NOT_OWNED = 22,
    GLOBUS_GSI_SYSCONFIG_ERROR_FILE_IS_DATA = 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_DATA = 30,
    GLOBUS_GSI_SYSCONFIG_ERROR_LAST = 31
}
```

- `enumglobus_gsi_proxy_le_type_t`
`GLOBUS_PROXY_FILE_INPUT`
`GLOBUS_PROXY_FILE_OUTPUT`

3.6.1 Enumeration Type Documentation

3.6.1.1 `enumglobus_gsi_sysconfig_error_t`

GSI System Configuration Error codes.

Enumeration values:

`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 le-name.

`GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_KEY_STRING`Error while generating private key le-name.

`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 le exists.

`GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CERT_FILENAME`Unable to determine the location of the certificate le.

`GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_PROXY_FILENAME`Unable to determine the location of the proxy le.

`GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_DELEG_FILENAME`Unable to determine the location of the delegated proxy le.

`GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CA_CERT_FILENAMES`Unable to generate a list of CA certificate le-names.

`GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_CWD`Error while discovering the current working directory.

`GLOBUS_GSI_SYSCONFIG_ERROR_REMOVING_OWNED_FILES`Failed to remove all proxy les.

`GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_GRIDMAP_FILENAME`Unable to determine the location of the grid map le.

`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 le permissions.

`GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_SIGNING_POLICY`Unable to determine the location of a signing policy le.

`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 le.

`GLOBUS_GSI_SYSCONFIG_ERROR_FILE_NOT_REGULAR`File is not a regular le.

`GLOBUS_GSI_SYSCONFIG_ERROR_FILE_DOES_NOT_EXIST`File does not exist.

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_BAD_PERMISSIONS File has incorrect permissions for operation.

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_NOT_OWNED File is not owned by current user.

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_IS_DIR File is a directory.

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_ZERO_LENGTH File has zero length.

GLOBAL_GSI_SYSCONFIG_INVALID_ARG Invalid argument.

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_HAS_LINKS File has more than one link.

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_HAS_CHANGED File has changed in the meantime.

GLOBAL_GSI_SYSCONFIG_ERROR_GETTING_AUTHZ_LIB_FILENAME Failed to locate the authorization callout library configuration file.

GLOBAL_GSI_SYSCONFIG_ERROR_GETTING_GAA_FILENAME Failed to locate the gaa configuration file.

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_NOT_DIR File is not a directory.

GLOBAL_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.

Enumeration values:

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

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

Index

Activation, [21](#)

Datatypes, [22](#)

Denes, [1](#)

Functions for all platforms, [21](#)

Functions for UNIX platforms, [5](#)

Functions for Win32 platforms, [4](#)

globus_gsi_proxy_le_type_t

globus_gsi_sysconfig_datatypes, [24](#)

globus_gsi_sysconfig_activation

GLOBAL_GSI_SYSCONFIG_MODULE, [22](#)

GLOBAL_GSI_SYSCONFIG_CHECK_CERTFILE

globus_gsi_system_config_denes, [3](#)

globus_gsi_sysconfig_check_cert_unix

globus_gsi_sysconfig_unix, [10](#)

GLOBAL_GSI_SYSCONFIG_CHECK_KEYFILE

globus_gsi_system_config_denes, [3](#)

globus_gsi_sysconfig_check_key_unix

globus_gsi_sysconfig_unix, [9](#)

globus_gsi_sysconfig_datatypes

GLOBAL_GSI_SYSCONFIG_ERROR_

CHECKING_FILE_EXISTS, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

CHECKING_SUPERUSER, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

ERRNO, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

BAD_PERMISSIONS, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

DOES_NOT_EXIST, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

HAS_CHANGED, [24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

HAS_LINKS, [24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

IS_DIR, [24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

NOT_DIR, [24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

NOT_OWNED, [24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

NOT_REGULAR, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_FILE_

ZERO_LENGTH, [24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_AUTHZ_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_AUTHZ_LIB_FILENAME,

[24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_CA_CERT_FILENAMES,

[23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_CERT_DIR, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_CERT_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_CERT_STRING, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_CWD, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_DELEG_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_GAA_FILENAME, [24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_GRIDMAP_FILENAME,

[23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_HOME_DIR, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_KEY_STRING, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_PROXY_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_PW_ENTRY, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

GETTING_SIGNING_POLICY, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_LAST,

[24](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

REMOVING_OWNED_FILES, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

SETTING_PERMS, [23](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

SUCCESS, [23](#)

GLOBAL_GSI_SYSCONFIG_INVALID_ARG,

[24](#)

GLOBAL_PROXY_FILE_INPUT, [24](#)

GLOBAL_PROXY_FILE_OUTPUT, [24](#)

globus_gsi_sysconfig_datatypes

globus_gsi_proxy_le_type, [24](#)

globus_gsi_sysconfig_error, [23](#)

GLOBAL_GSI_SYSCONFIG_DIR_EXISTS

globus_gsi_system_config_denes, [3](#)

globus_gsi_sysconfig_dir_exists_unix

globus_gsi_sysconfig_unix, [9](#)

globus_gsi_sysconfig_dir_exists_win32

globus_gsi_sysconfig_win32, [16](#)

GLOBAL_GSI_SYSCONFIG_ERROR_

CHECKING_FILE_EXISTS
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_-
 CHECKING_SUPERUSER
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_ERRNO
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_-
 BAD_PERMISSIONS
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_-
 DOES_NOT_EXIST
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_-
 HAS_CHANGED
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_-
 HAS_LINKS
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_IS_-
 DIR
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_-
 NOT_DIR
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_-
 NOT_OWNED
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_-
 NOT_REGULAR
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_FILE_-
 ZERO_LENGTH
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 AUTHZ_FILENAME
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 AUTHZ_LIB_FILENAME
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 CA_CERT_FILENAMES
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 CERT_DIR
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 CERT_FILENAME
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 CERT_STRING
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 CWD
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 DELEG_FILENAME
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 GAA_FILENAME
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 GRIDMAP_FILENAME
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 HOME_DIR
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 KEY_STRING
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 PROXY_FILENAME
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 PW_ENTRY
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_-
 SIGNING_POLICY
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_LAST
 globus_gsi_syscon g_datatype⁸⁴
 GLOBUS_GSI_SYSCONFIG_ERROR_-
 REMOVING_OWNED_FILES
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_SETTING_-
 PERMS
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_ERROR_SUCCESS
 globus_gsi_syscon g_datatype⁸³
 globus_gsi_syscon g_error_t
 globus_gsi_syscon g_datatype⁸³
 GLOBUS_GSI_SYSCONFIG_FILE_EXISTS
 globus_gsi_system_con g_de ne⁸³,
 globus_gsi_syscon g_le_exists_unix
 globus_gsi_syscon g_unix⁹
 globus_gsi_syscon g_le_exists_win32
 globus_gsi_syscon g_win32^{1,6}
 GLOBUS_GSI_SYSCONFIG_GET_AUTHZ_-
 CONF_FILENAME
 globus_gsi_system_con g_de ne⁴,
 globus_gsi_syscon g_get_authz_conf_lename_unix
 globus_gsi_syscon g_unix^{1,3}
 globus_gsi_syscon g_get_authz_conf_lename_-
 win32
 globus_gsi_syscon g_win32^{2,0}
 globus_gsi_syscon g_get_authz_lib_conf_lename_-

- unix
- globus_gsi_syscon g_unix^{1,3}
- GLOBUS_GSI_SYSCONFIG_GET_CA_CERT_-
FILES
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_ca_cert_ lename_unix
 - globus_gsi_syscon g_unix^{1,2}
 - globus_gsi_syscon g_get_ca_cert_ lename_win32
 - globus_gsi_syscon g_win32^{1,9}
- GLOBUS_GSI_SYSCONFIG_GET_CERT_DIR
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_cert_dir_unix
 - globus_gsi_syscon g_unix^{1,0}
 - globus_gsi_syscon g_get_cert_dir_win32
 - globus_gsi_syscon g_win32^{1,7}
- GLOBUS_GSI_SYSCONFIG_GET_CURRENT_-
WORKING_DIR
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_current_working_dir_unix
 - globus_gsi_syscon g_unix⁹
 - globus_gsi_syscon g_get_current_working_dir_-
win32
 - globus_gsi_syscon g_win32^{1,6}
- GLOBUS_GSI_SYSCONFIG_GET_GAA_CONF_-
FILENAME
 - globus_gsi_system_con g_de ne⁴,
 - globus_gsi_syscon g_get_gaa_conf_ lename_unix
 - globus_gsi_syscon g_unix^{1,3}
 - globus_gsi_syscon g_get_gaa_conf_ lename_win32
 - globus_gsi_syscon g_win32^{2,0}
- GLOBUS_GSI_SYSCONFIG_GET_GRIDMAP_-
FILENAME
 - globus_gsi_system_con g_de ne⁴,
 - globus_gsi_syscon g_get_gridmap_ lename_unix
 - globus_gsi_syscon g_unix^{1,3}
 - globus_gsi_syscon g_get_gridmap_ lename_win32
 - globus_gsi_syscon g_win32^{1,9}
- GLOBUS_GSI_SYSCONFIG_GET_HOME_DIR
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_home_dir_unix
 - globus_gsi_syscon g_unix⁹
- GLOBUS_GSI_SYSCONFIG_GET_HOST_CERT_-
FILENAME
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_host_cert_ lename_unix
 - globus_gsi_syscon g_unix^{1,1}
 - globus_gsi_syscon g_get_host_cert_ lename_win32
 - globus_gsi_syscon g_win32^{1,8}
- GLOBUS_GSI_SYSCONFIG_GET_PROC_ID_-
STRING
 - globus_gsi_system_con g_de ne⁴,
 - globus_gsi_syscon g_get_proc_id_string_unix
 - globus_gsi_syscon g_unix⁸
- GLOBUS_GSI_SYSCONFIG_GET_PROXY_-
FILENAME
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_proxy_ lename_unix
 - globus_gsi_syscon g_unix^{1,1}
 - globus_gsi_syscon g_get_proxy_ lename_win32
 - globus_gsi_syscon g_win32^{1,8}
- GLOBUS_GSI_SYSCONFIG_GET_SERVICE_-
CERT_FILENAME
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_service_cert_ lename_-
unix
 - globus_gsi_syscon g_unix^{1,1}
 - globus_gsi_syscon g_get_service_cert_ lename_-
win32
 - globus_gsi_syscon g_win32^{1,8}
- GLOBUS_GSI_SYSCONFIG_GET_SIGNING_-
POLICY_FILENAME
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_signing_policy_ lename_-
unix
 - globus_gsi_syscon g_unix^{1,2}
 - globus_gsi_syscon g_get_signing_policy_ lename_-
win32
 - globus_gsi_syscon g_win32^{2,0}
- GLOBUS_GSI_SYSCONFIG_GET_UNIQUE_-
PROXY_FILENAME
 - globus_gsi_system_con g_de ne⁵,
 - globus_gsi_syscon g_get_unique_proxy_ lename
 - globus_gsi_syscon g_share^{2,1}
- GLOBUS_GSI_SYSCONFIG_GET_USER_CERT_-
FILENAME
 - globus_gsi_system_con g_de ne³,
 - globus_gsi_syscon g_get_user_cert_ lename_unix
 - globus_gsi_syscon g_unix^{1,0}
 - globus_gsi_syscon g_get_user_cert_ lename_win32
 - globus_gsi_syscon g_win32^{1,7}
- GLOBUS_GSI_SYSCONFIG_GET_USER_ID_-
STRING
 - globus_gsi_system_con g_de ne⁴,
 - globus_gsi_syscon g_get_user_id_string_unix
 - globus_gsi_syscon g_unix⁷
- GLOBUS_GSI_SYSCONFIG_GET_USERNAME
 - globus_gsi_system_con g_de ne⁵,
 - globus_gsi_syscon g_get_username_unix
 - globus_gsi_syscon g_unix⁸
- GLOBUS_GSI_SYSCONFIG_INVALID_ARG
 - globus_gsi_syscon g_datatype^{6,4}
- GLOBUS_GSI_SYSCONFIG_IS_SUPERUSER
 - globus_gsi_system_con g_de ne⁴,
 - globus_gsi_syscon g_is_superuser_unix
 - globus_gsi_syscon g_unix^{1,3}
 - globus_gsi_syscon g_is_superuser_win32

globus_gsi_syscon g_win32 [20](#)
 GLOBUS_GSI_SYSCONFIG_MAKE_-
 ABSOLUTE_PATH_FOR_FILENAME
 globus_gsi_system_con g_de ne [4](#),
 globus_gsi_syscon g_make_absolute_path_for_-
 lename_unix
 globus_gsi_syscon g_unix [8](#)
 globus_gsi_syscon g_make_absolute_path_for_-
 lename_win32
 globus_gsi_syscon g_win32 [16](#)
 GLOBUS_GSI_SYSCONFIG_MODULE
 globus_gsi_syscon g_activation [22](#)
 GLOBUS_GSI_SYSCONFIG_REMOVE_ALL_-
 OWNED_FILES
 globus_gsi_system_con g_de ne [4](#),
 globus_gsi_syscon g_remove_all_owned_les_unix
 globus_gsi_syscon g_unix [12](#)
 globus_gsi_syscon g_remove_all_owned_les_-
 win32
 globus_gsi_syscon g_win32 [19](#)
 GLOBUS_GSI_SYSCONFIG_SET_KEY_-
 PERMISSIONS
 globus_gsi_system_con g_de ne [3](#),
 globus_gsi_syscon g_set_key_permissions_unix
 globus_gsi_syscon g_unix [7](#),
 globus_gsi_syscon g_set_key_permissions_win32
 globus_gsi_syscon g_win32 [16](#)
 globus_gsi_syscon g_shared
 globus_gsi_syscon g_get_unique_proxy_-
 lename, [21](#)
 GLOBUS_GSI_SYSCONFIG_SPLIT_DIR_AND_-
 FILENAME
 globus_gsi_system_con g_de ne [4](#),
 globus_gsi_syscon g_split_dir_and_ lename_unix
 globus_gsi_syscon g_unix [8](#)
 globus_gsi_syscon g_split_dir_and_ lename_win32
 globus_gsi_syscon g_win32 [16](#)
 globus_gsi_syscon g_unix
 globus_gsi_syscon g_check_cert le_unix [10](#)
 globus_gsi_syscon g_check_key le_unix [9](#),
 globus_gsi_syscon g_dir_exists_unix [8](#),
 globus_gsi_syscon g_le_exists_unix [8](#),
 globus_gsi_syscon g_get_authz_conf_-
 lename_unix, [13](#)
 globus_gsi_syscon g_get_authz_lib_conf_-
 lename_unix, [13](#)
 globus_gsi_syscon g_get_ca_cert_les_un [12](#),
 globus_gsi_syscon g_get_cert_dir_un [10](#)
 globus_gsi_syscon g_get_current_working_-
 dir_unix, [9](#)
 globus_gsi_syscon g_get_gaa_conf_ lename_-
 unix, [13](#)
 globus_gsi_syscon g_get_gridmap_ lename_-
 unix, [13](#)
 globus_gsi_syscon g_get_home_dir_un [8](#),
 globus_gsi_syscon g_get_host_cert_ lename_-
 unix, [11](#)
 globus_gsi_syscon g_get_proc_id_string_unix,
 [8](#)
 globus_gsi_syscon g_get_proxy_ lename_unix,
 [11](#)
 globus_gsi_syscon g_get_service_cert_-
 lename_unix, [11](#)
 globus_gsi_syscon g_get_signing_policy_-
 lename_unix, [12](#)
 globus_gsi_syscon g_get_user_cert_ lename_-
 unix, [10](#)
 globus_gsi_syscon g_get_user_id_string_unix,
 [7](#)
 globus_gsi_syscon g_get_username_un [8](#),
 globus_gsi_syscon g_is_superuser_un [18](#),
 globus_gsi_syscon g_make_absolute_path_-
 for_ lename_unix, [8](#)
 globus_gsi_syscon g_remove_all_owned_les_-
 unix, [12](#)
 globus_gsi_syscon g_set_key_permissions_-
 unix, [7](#)
 globus_gsi_syscon g_split_dir_and_ lename_-
 unix, [8](#)
 globus_gsi_syscon g_win32
 globus_gsi_syscon g_dir_exists_win32 [16](#),
 globus_gsi_syscon g_le_exists_win32 [16](#)
 globus_gsi_syscon g_get_authz_conf_-
 lename_win32, [20](#)
 globus_gsi_syscon g_get_ca_cert_les_win32,
 [19](#)
 globus_gsi_syscon g_get_cert_dir_win32 [17](#),
 globus_gsi_syscon g_get_current_working_-
 dir_win32, [16](#)
 globus_gsi_syscon g_get_gaa_conf_ lename_-
 win32, [20](#)
 globus_gsi_syscon g_get_gridmap_ lename_-
 win32, [19](#)
 globus_gsi_syscon g_get_host_cert_ lename_-
 win32, [18](#)
 globus_gsi_syscon g_get_proxy_ lename_-
 win32, [18](#)
 globus_gsi_syscon g_get_service_cert_-
 lename_win32, [18](#)
 globus_gsi_syscon g_get_signing_policy_-
 lename_win32, [20](#)
 globus_gsi_syscon g_get_user_cert_ lename_-
 win32, [17](#)
 globus_gsi_syscon g_is_superuser_win [30](#),
 globus_gsi_syscon g_make_absolute_path_-
 for_ lename_win32, [16](#)

globus_gsi_syscon g_remove_all_owned_files -
 win32, [19](#)
 globus_gsi_syscon g_set_key_permissions -
 win32, [16](#)
 globus_gsi_syscon g_split_dir_and_filename -
 win32, [16](#)
 globus_gsi_system_configuration
 GLOBUS_GSI_SYSCONFIG_CHECK_CERTFILE, [2](#)
 GLOBUS_GSI_SYSCONFIG_CHECK_KEYFILE, [2](#)
 GLOBUS_GSI_SYSCONFIG_DIR_EXISTS, [3](#)
 GLOBUS_GSI_SYSCONFIG_FILE_EXISTS, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_AUTHZ_CONF_FILENAME, [4](#)
 GLOBUS_GSI_SYSCONFIG_GET_CA_CERT_FILES, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_CERT_DIR, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_CURRENT_WORKING_DIR, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_GAA_CONF_FILENAME, [4](#)
 GLOBUS_GSI_SYSCONFIG_GET_GRIDMAP_FILENAME, [4](#)
 GLOBUS_GSI_SYSCONFIG_GET_HOME_DIR, [2](#)
 GLOBUS_GSI_SYSCONFIG_GET_HOST_CERT_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_PROC_ID_STRING, [4](#)
 GLOBUS_GSI_SYSCONFIG_GET_PROXY_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_SERVICE_CERT_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_SIGNING_POLICY_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_UNIQUE_PROXY_FILENAME, [5](#)
 GLOBUS_GSI_SYSCONFIG_GET_USER_CERT_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIG_GET_USER_ID_STRING, [4](#)
 GLOBUS_GSI_SYSCONFIG_GET_USERNAME, [5](#)
 GLOBUS_GSI_SYSCONFIG_IS_SUPERUSER, [4](#)
 GLOBUS_GSI_SYSCONFIG_MAKE_ABSOLUTE_PATH_FOR_FILENAME, [4](#)
 GLOBUS_GSI_SYSCONFIG_REMOVE_ALL_OWNED_FILES, [4](#)
 GLOBUS_GSI_SYSCONFIG_SET_KEY_PERMISSIONS, [2](#)
 GLOBUS_GSI_SYSCONFIG_SPLIT_DIR_AND_FILENAME, [4](#)
 GLOBUS_PROXY_FILE_INPUT
 globus_gsi_syscon g_datatype, [84](#)
 GLOBUS_PROXY_FILE_OUTPUT
 globus_gsi_syscon g_datatype, [84](#)