

globus gsi syscon g Reference Manual

2.2

Generated by Doxygen 1.2.18

Fri Jun 26 15:27:31 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_systemconfig.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_SYSCONFIGSET_KEY_PERMISSIONS
#define GLOBUS_GSI_SYSCONFIGGET_HOME_DIR
#define GLOBUS_GSI_SYSCONFIGCHECK_KEYFILE
#define GLOBUS_GSI_SYSCONFIGCHECK_CERTFILE
#define GLOBUS_GSI_SYSCONFIGFILE_EXISTS
#define GLOBUS_GSI_SYSCONFIGDIR_EXISTS
```

```

#de ne GLOBUS_GSI_SYSCONFIGGET_CERT_DIR
#de ne GLOBUS_GSI_SYSCONFIGGET_USER_CERT_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGGET_HOST_CERT_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGGET_SERVICE_CERT_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGGET_PROXY_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGGET_SIGNING_POLICY_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGGET_CA_CERT_FILES
#de ne GLOBUS_GSI_SYSCONFIGGET_CURRENT_WORKING_DIR
#de ne GLOBUS_GSI_SYSCONFIGMAKE_ABSOLUTE_PATH_FOR_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGSPLIT_DIR_AND_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGREMOVE_ALL_OWNED_FILES
#de ne GLOBUS_GSI_SYSCONFIGGET_GRIDMAP_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGGET_AUTHZ_CONF_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGGET_GAA_CONF_FILENAME
#de ne GLOBUS_GSI_SYSCONFIGIS_SUPERUSER
#de ne GLOBUS_GSI_SYSCONFIGGET_USER_ID_STRING
#de ne GLOBUS_GSI_SYSCONFIGGET_PROC_ID_STRING
#de ne GLOBUS_GSI_SYSCONFIGGET_USERNAME
#de ne GLOBUS_GSI_SYSCONFIGGET_UNIQUE_PROXY_FILENAME

```

3.1.1 Detailed Description

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

3.1.2 De ne Documentation

3.1.2.1 #de ne GLOBUS_GSI_SYSCONFIG.SET_KEY_PERMISSIONS

Set the correct le permissions on a private key.

See [globusgsi.syscon g.setkey.permissionsunix\(\)](#) and [globusgsi.syscon g.setkey.permissionswin32\(\)](#)

3.1.2.2 #de ne GLOBUS_GSI_SYSCONFIG.GET_HOME_DIR

Get the current users home directory.

See [globusgsi.syscon g.gethomedir_unix\(\)](#) and [globusgsi.syscon g.gethomedir_win32\(\)](#)

3.1.2.3 #de ne GLOBUS_GSI_SYSCONFIG.CHECK_KEYFILE

Check for the correct le permissions on a private key.

See [globusgsi.syscon g.checkkey le_unix\(\)](#) and [globusgsi.syscon g.checkkey le_win32\(\)](#)

3.1.2.4 #de ne GLOBUS_GSI_SYSCONFIG.CHECK_CERTFILE

Check for the correct le permissions on a certi cate.

See [globusgsi.syscon g.checkcert le_unix\(\)](#) and [globusgsi.syscon g.checkcert le_win32\(\)](#)

3.1.2.5 #de ne GLOBUSGSI_SYSCONFIG.FILE_EXISTS

Check whether a given file exists.

See [globusgsi.syscon g le _exists.unix\(\)](#) and [globusgsi.syscon g le _exists.win32\(\)](#)

3.1.2.6 #de ne GLOBUSGSI_SYSCONFIG.DIR_EXISTS

Check whether a given directory exists.

See [globusgsi.syscon g dir _exists.unix\(\)](#) and [globusgsi.syscon g dir _exists.win32\(\)](#)

3.1.2.7 #de ne GLOBUSGSI_SYSCONFIG.GET_CERT_DIR

Determine the location of the trusted certificates directory.

See [globusgsi.syscon g get.cert.dir.unix\(\)](#) and [globusgsi.syscon g get.cert.dir.win32\(\)](#)

3.1.2.8 #de ne GLOBUSGSI_SYSCONFIG.GET_USER_CERT_FILENAME

Determine the location of the users certificate and private key.

See [globusgsi.syscon g get.usercert.filename.unix\(\)](#) and [globusgsi.syscon g get.usercert.filename.win32\(\)](#)

3.1.2.9 #de ne GLOBUSGSI_SYSCONFIG.GET_HOST_CERT_FILENAME

Determine the location of the host certificate and private key.

See [globusgsi.syscon g get.hostcert.filename.unix\(\)](#) and [globusgsi.syscon g get.hostcert.filename.win32\(\)](#)

3.1.2.10 #de ne GLOBUSGSI_SYSCONFIG.GET_SERVICE_CERT_FILENAME

Determine the location of a service certificate and private key.

See [globusgsi.syscon g get.servicecert.filename.unix\(\)](#) and [globusgsi.syscon g get.servicecert.filename.win32\(\)](#)

3.1.2.11 #de ne GLOBUSGSI_SYSCONFIG.GET_PROXY_FILENAME

Determine the location of a proxy certificate and private key.

See [globusgsi.syscon g get.proxy.filename.unix\(\)](#) and [globusgsi.syscon g get.proxy.filename.win32\(\)](#)

3.1.2.12 #de ne GLOBUSGSI_SYSCONFIG.GET_SIGNING_POLICY_FILENAME

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

See [globusgsi.syscon g get.signing.policy.filename.unix\(\)](#) and [globusgsi.syscon g get.signing.policy.filename.win32\(\)](#)

3.1.2.13 #de ne GLOBUSGSI_SYSCONFIG.GET_CA_CERT_FILES

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

See [globusgsi.syscon g get.ca.cert.files.unix\(\)](#) and [globusgsi.syscon g get.ca.cert.files.win32\(\)](#)

3.1.2.14 #de ne GLOBUSGSI_SYSCONFIG.GET_CURRENT_WORKING_DIR

Get the current working directory.

See [globusgsi.syscon g.get.currentworking.dir.unix\(\)](#) and [globusgsi.syscon g.get.currentworking.dir.win32\(\)](#)

3.1.2.15 #de ne GLOBUSGSI_SYSCONFIG.MAKE_ABSOLUTE_PATH_FOR_FILENAME

Prepend the current working directory to the give lename.

See [globusgsi.syscon g.make.absolute.path.for_.lename.unix\(\)](#) and [globusgsi.syscon g.make.absolute.path.for_.lename.win32\(\)](#)

3.1.2.16 #de ne GLOBUSGSI_SYSCONFIG.SPLIT_DIR_AND_FILENAME

Split directory component of path from lename.

See [globusgsi.syscon g.split.dir.and_.lename.unix\(\)](#) and [globusgsi.syscon g.split.dir.and_.lename.win32\(\)](#)

3.1.2.17 #de ne GLOBUSGSI_SYSCONFIG.REMOVE_ALL_OWNED_FILES

Remove all proxies owned by current uid.

See [globusgsi.syscon g.remove.all.owned_.les_.unix\(\)](#) and [globusgsi.syscon g.remove.all.owned_.les_.win32\(\)](#)

3.1.2.18 #de ne GLOBUSGSI_SYSCONFIG.GET_GRIDMAP_FILENAME

Determine the location of the grid map le.

See [globusgsi.syscon g.get.gridmap_.lename.unix\(\)](#) and [globusgsi.syscon g.get.gridmap_.lename.win32\(\)](#)

3.1.2.19 #de ne GLOBUSGSI_SYSCONFIG.GET_AUTHZ_CONF_FILENAME

Determine the location of the authorization callout con g le.

See [globusgsi.syscon g.get.authz.conf_.lename.unix\(\)](#)

3.1.2.20 #de ne GLOBUSGSI_SYSCONFIG.GET_GAA_CONF_FILENAME

Determine the location of the GAA callout con g le.

See [globusgsi.syscon g.get.gaa.conf_.lename.unix\(\)](#)

3.1.2.21 #de ne GLOBUSGSI_SYSCONFIG.IS_SUPERUSER

Determine whether the current user is the super user.

See [globusgsi.syscon g.is.superuser.unix\(\)](#) and [globusgsi.syscon g.is.superuser.win32\(\)](#)

3.1.2.22 #de ne GLOBUSGSI_SYSCONFIG.GET_USER_ID_STRING

Get the current UID in string form.

See [globusgsi.syscon g.get.userid.string.unix\(\)](#) and [globusgsi.syscon g.get.userid.string.win32\(\)](#)

3.1.2.23 #define GLOBUSGSI_SYSCONFIG_GET_PROC_ID_STRING

Get the current PID in string form.

See [globusgsi.syscon g.get.proc.id.string.unix\(\)](#) and [globusgsi.syscon g.get.proc.id.string.win32\(\)](#)

3.1.2.24 #define GLOBUSGSI_SYSCONFIG_GET_USERNAME

Get the current user name.

See [globusgsi.syscon g.get.username.unix\(\)](#) and [globusgsi.syscon g.get.username.win32\(\)](#)

3.1.2.25 #define GLOBUSGSI_SYSCONFIG_GET_UNIQUE_PROXY_FILENAME

Generate a unique proxy file name.

See [globusgsi.syscon g.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

globusresultt [globusgsi.syscon g.setkey.permissions.unix](#) (char lename)

UNIX - Get User ID

globusresultt [globusgsi.syscon g.get.userid.string.unix](#) (char userid_string)

UNIX - Get Username

globusresultt [globusgsi.syscon g.get.username.unix](#) (char username)

UNIX - Get Process ID

globusresultt [globusgsi.syscon g.get.proc.id.string.unix](#) (char proc.id_string)

UNIX - Make Absolute Path

globusresultt [globusgsi.syscon g.make.absolute.path.for.filename.unix](#) (char lename, char absolute-path)

UNIX - Split Directory and Filename

globusresultt [globusgsi.syscon g.split.dir.and.filename.unix](#) (char full_filename, char dir_string, char filename_string)

UNIX - Get Current Working Directory

globusresultt [globusgsi.syscon g.get.current.working.dir.unix](#) (char working_dir)

UNIX - Get HOME Directory

```
globusresultt globusgsi.syscon g.get.homedir.unix (char homedir)
```

UNIX - File Exists

```
globusresultt globusgsi.syscon g.le._exists.unix (const char lename)
```

UNIX - Directory Exists

```
globusresultt globusgsi.syscon g.dir._exists.unix (const char lename)
```

UNIX - Check File Status for Key

```
globusresultt globusgsi.syscon g.checkkey.le._unix (const char lename)
```

UNIX - Check File Status for Cert

```
globusresultt globusgsi.syscon g.checkcert.le._unix (const char lename)
```

UNIX - Get Trusted CA Cert Dir

```
globusresultt globusgsi.syscon g.get.cert.dir.unix (char certdir)
```

UNIX - Get User Certificate and Key Filenames

```
globusresultt globusgsi.syscon g.get.user.cert.lenam.unix (char usercert, char userkey)
```

UNIX - Get Host Certificate and Key Filenames

```
globusresultt globusgsi.syscon g.get.host.cert.lenam.unix (char hostcert, char hostkey)
```

UNIX - Get Service Certificate and Key Filenames

```
globusresultt globusgsi.syscon g.get.service.cert.lenam.unix (char servicename, char servicecert,  
char servicekey)
```

UNIX - Get Proxy Filename

```
globusresultt globusgsi.syscon g.get.proxy.lenam.unix (char userproxy, globusgsi.proxy.le._type.t  
proxy.le._type)
```

UNIX - Get Signing Policy Filename

```
globusresultt globusgsi.syscon g.get.signing.policy.lenam.unix (X509.NAME ca.name, char certdir,  
char signing.policy.lenam)
```

UNIX - Get CA Cert Filenames

```
globusresult_t globus_gsi_syscon_g_get_ca_cert_filenames_unix (char ca_cert_dir, globus_file_t ca_cert_list)
```

UNIX - Remove all proxies owned by current uid

```
globusresult_t globus_gsi_syscon_g_removeall_owned_files_unix (char default_filename)
```

UNIX - Check if the current user is root

```
globusresult_t globus_gsi_syscon_g_is_superuser_unix (int is_superuser)
```

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

```
globusresult_t globus_gsi_syscon_g_get_gridmap_filename_unix (char filename)
```

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

```
globusresult_t globus_gsi_syscon_g_get_authzconf_filename_unix (char filename)
```

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

```
globusresult_t globus_gsi_syscon_g_get_authzlib_conf_filename_unix (char filename)
```

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

```
globusresult_t globus_gsi_syscon_g_get_gaaconf_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 globusresult_t globus_gsi_syscon_g_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 globusresult_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 globusresult_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 globusresult_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 globusresult_t globus_gsi_syscon g_make_absolute_path_for_lename_unix (char lename, char absolutepath)

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

Parameters:

lename the lename to get the absolute path of.

absolutepath 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 globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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 `globusresult_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_le_name_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 `X509USERCERT` and `X509USERKEY`

3.2.2.17 `globus_result_t globus_gsi_syscon_g_get_proxy_lename_unix (char user_proxy, globus_gsi_proxy_len_type_t proxy_len_type)`

Get the proxy cert lename 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_len_type variable is set to GLOBUS_PROXY_OUTPUT (a proxy lename for writing is requested), and the X509_USER_PROXY is set, this will be the resulting value of the user_proxy lename string passed in. If the proxy_len_type is set to GLOBUS_PROXY_INPUT and X509_USER_PROXY is set, but the len 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 len of `/tmp/x509_un<userid>_where` where `<userid>` is some unique string for that user on the host

Parameters:

user_proxy the proxy lename of the user

proxy_len_type Switch for determining whether to return an existing proxy lename or if a lename 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_lename_unix (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

3.2.2.19 `globus_result_t globus_gsi_syscon_g_get_ca_cert_list_unix (char ca_cert_dir, globus_list_t *ca_cert_list)`

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

Parameters:

ca_cert_dir The trusted CA certificate directory to get the lenames from

ca_cert_list The resulting list of CA certificate lenames. This is a globus list structure.

See also:

`globus_list_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_le_unix (char default_le_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_le_name The le_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_le_name_unix (char le_name)

Get the path and le_name of the grid map le.

Parameters:

le_name 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

3.2.2.23 globus_result_t globus_gsi_syscon_g_get_authz_conf_le_name_unix (char le_name)

Get the path and le_name of the authorization callback configuration le

Parameters:

le_name Contains the location of the authorization callback configuration le 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_le_name_unix (char le_name)

Get the path and le_name of the authorization callback configuration le

Parameters:

le_name Contains the location of the authorization callback configuration le 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 file name of the GAA configuration file.

Parameters:

`lename` 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

These functions implement the Win32 version of the Globus GSI System Configuration 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_file_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 absolutepath)`

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 usercert, char userkey)`

Win32 - Get Host Certificate and Key Filenames

```
globusresult_t globus_gsi_syscon_g_get_host_cert_lename_win32(char hostcert, char hostkey)
```

Win32 - Get Service Certificate and Key Filenames

```
globusresult_t globus_gsi_syscon_g_get_servicecert_lename_win32(char servicename, char servicecert, char servicekey)
```

Win32 - Get Proxy Filename

```
globusresult_t globus_gsi_syscon_g_get_proxy_lename_win32(char userproxy, globus_gsi_proxy_t proxy, globus_gsi_proxy_t proxy_type)
```

Win32 - Get CA Cert Filenames

```
globusresult_t globus_gsi_syscon_g_get_ca_cert_files_win32(char cacertdir, globus_list_t cacertlist)
```

Win32 - Remove all proxies owned by current uid

```
globusresult_t globus_gsi_syscon_g_removeall_owned_files_win32(char default_lename)
```

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

```
globusresult_t globus_gsi_syscon_g_get_gridmap_lename_win32(char lename)
```

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

```
globusresult_t globus_gsi_syscon_g_get_authzconf_lename_win32(char lename)
```

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

```
globusresult_t globus_gsi_syscon_g_get_gaaconf_lename_win32(char lename)
```

Win32 - Check if the current user is root

```
globusresult_t globus_gsi_syscon_g_is_superuser_win32(int is_superuser)
```

Win32 - Get Signing Policy Filename

```
globusresult_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 defines.

3.3.2 Function Documentation

3.3.2.1 globusresult_t globus_gsi_syscon_g_setkey_permissions_win32 (char lename)

Set the le permissions of a le 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 globusresult_t globus_gsi_syscon_g_le_exists_win32 (const char lename)

Check that the le exists.

Parameters:

lename the le to check

Returns:

GLOBUS.SUCCESS (even if the le doesn't exist) - in some abortive cases an error object identi er is returned

3.3.2.3 globusresult_t globus_gsi_syscon_g_dir_exists_win32 (const char lename)

Check that the directory exists.

Parameters:

lename the le to check

Returns:

GLOBUS.SUCCESS if the directory exists, otherwise an error object identi er.

3.3.2.4 globusresult_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 globusresult_t globus_gsi_syscon_g_make_absolute_path_for_lename_win32 (char lename, char absolutepath)

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

Parameters:

lename the lename to get the absolute path of.

absolutepath The resulting absolute path

Returns:

GLOBUS.SUCCESS if no error occurred, otherwise an error object ID is returned

3.3.2.6 globusresult_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 globusresult_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\GlobusGSI
3. <user home directory>\n.globus\ncertificates - 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\nshare\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 globusresult_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 X509USERCERT and X509USERKEY
2. registry keys x509usercert and x509userkey in software\GlobusGSI
3. <users home directory>\n.globus\nusercert.pem and <users home directory>\n.globus\nuserkey.pem
4. <users home directory>\n.globus\nusercred.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_syscon_g_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_USERCERT and X509_USERKEY environment variables
2. registry keys x509usercert and x509userkey in software\Globus\GSI
3. < GLOBUS_LOCATION> netn\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_syscon_g_get_service_cert_lename_win32 (char servicename, char servicecert, char servicekey)

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_USERCERT and X509_USERKEY environment variables
2. registry keys x509usercert and x509userkey in software\Globus\GSI
3. GLOBUS_LOCATION\netn\servicename\servicename[cert|key].pem So for example, if my service was named: myservice, the location of the certificate would be: GLOBUS_LOCATION\netn\myservice\myservicecert.pem
4. < users home directory> n.globus\servicename\servicename[cert|key].pem

Parameters:

servicename The name of the service which allows us to determine the locations of cert and key files to look for
 servicecert pointer to the host certificate lename
 servicekey 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_lename_win32 (char user_proxy, globus_gsi_proxy_len_type_t proxy_len_type)`

Get the proxy cert lename 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_len_type variable is set to GLOBUS_PROXY_OUTPUT (a proxy lename for writing is requested), and the X509_USER_PROXY is set, this will be the resulting value of the user_proxy lename string passed in. If the proxy_len_type is set to GLOBUS_PROXY_INPUT and X509_USER_PROXY is set, but the lename 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: x509userproxy. Just as with the environment variable, if the registry key is set, and proxy_len_type is GLOBUS_PROXY_OUTPUT, the string set to be the proxy lename will be this registry key's value. If proxy_len_type is GLOBUS_PROXY_INPUT, and the lename doesn't exist, the function will check the next method for the proxy's lename.
3. Check the default location for the proxy lename. The default location should be set to reside in the temp directory on that host, with the lename taking the format: x509<user id> where <user id> is some unique string for that user on the host

Parameters:

user_proxy the proxy lename of the user

proxy_len_type Switch for determining whether to return an existing proxy lename or if a lename 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_list_t ca_cert_list)`

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

Parameters:

ca_cert_dir The trusted CA certificate directory to get the lenames from

ca_cert_list The resulting list of CA certificate lenames. This is a globus list structure.

See also:

globus_list

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_list_win32 (char default_lename)`

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_lename The lename 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 the name of the grid map file.

Parameters:

lename 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_syscon_g_get_authz_conf_lename_win32 (char lename)

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

Parameters:

lename 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_syscon_g_get_gaa_conf_lename_win32 (char lename)

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

Parameters:

lename 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_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 CAs.

signing_policy_lename The resulting signing policy filename

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_getuniqueproxy_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_getuniqueproxy_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 #de ne 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_ERROR_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 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 name.

GLOBUS_GSI_SYSCONFIG_ERROR_GETTING_KEY_STRING Error while generating private key 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 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 discovering 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 globusgsi_proxy_file_type_t

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

Enumeration values:

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).

Index

Activation, [21](#)

Datatypes, [22](#)

Denes, [1](#)

Functions for all platforms, [21](#)

Functions for UNIX platforms, [5](#)

Functions for Win32 platforms, [4](#)

globusgsi.proxy.le_type.t

globusgsi.syscon.g.datatypes, [23](#)

globusgsi.syscon.g.activation

GLOBAL_GSI_SYSCONFIGMODULE, [22](#)

GLOBAL_GSI_SYSCONFIGCHECK_CERTFILE

globusgsi.systemcon.g.denes, [2](#)

globusgsi.syscon.g.checkcert.le.unix

globusgsi.syscon.g.unix, [10](#)

GLOBAL_GSI_SYSCONFIGCHECK_KEYFILE

globusgsi.systemcon.g.denes, [2](#)

globusgsi.syscon.g.checkkey.le.unix

globusgsi.syscon.g.unix, [9](#)

globusgsi.syscon.g.datatypes

GLOBAL_GSI_SYSCONFIGERROR-

CHECKING_FILE_EXISTS, [22](#)

GLOBAL_GSI_SYSCONFIGERROR-

CHECKING_SUPERUSER, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.ERRNO, [22](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
BAD_PERMISSIONS, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
DOES_NOT_EXIST, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
HAS_CHANGED, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
HAS_LINKS, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
IS_DIR, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
NOT_DIR, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
NOT_OWNED, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
NOT_REGULAR, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.FILE_-
ZERO_LENGTH, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_AUTHZ_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_AUTHZ_LIB_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_CA_CERT_FILENAMES, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_CERT_DIR, [22](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_CERT_FILENAME, [22](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_CERT_STRING, [22](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_CWD, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_DELEG_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_GAA_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_GRIDMAP_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_HOME_DIR, [22](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_KEY_STRING, [22](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_PROXY_FILENAME, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_PW_ENTRY, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
GETTING_SIGNING_POLICY, [23](#)

GLOBAL_GSI_SYSCONFIGERROR.LAST, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
REMOVING_OWNED_FILES, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
SETTING_PERMS, [23](#)

GLOBAL_GSI_SYSCONFIGERROR-
SUCCESS, [22](#)

GLOBAL_GSI_SYSCONFIG.INVALID_ARG, [23](#)

GLOBAL_PROXY_FILE.INPUT, [23](#)

GLOBAL_PROXY_FILE.OUTPUT, [23](#)

globusgsi.syscon.g.datatypes

globusgsi.proxy.le_type.t, [23](#)

globusgsi.syscon.g.error.t, [22](#)

GLOBAL_GSI_SYSCONFIGDIR_EXISTS

globusgsi.systemcon.g.denes, [3](#)

globusgsi.syscon.g.dir.exists.unix

globusgsi.syscon.g.unix, [9](#)

globusgsi.syscon.g.dir.exists.win32

globusgsi.syscon.g.win32, [16](#)

GLOBAL_GSI_SYSCONFIGERROR-

CHECKING_FILE_EXISTS
 globusgsi.syscon g.datatypes²²
 GLOBUS_GSI_SYSCONFIGERROR_CHECKING_SUPERUSER
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_ERRNO
 globusgsi.syscon g.datatypes²²
 GLOBUS_GSI_SYSCONFIGERROR_FILE_BAD_PERMISSIONS
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_FILE_DOES_NOT_EXIST
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_FILE_HAS_CHANGED
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_FILE_HAS_LINKS
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_FILE_IS_DIR
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_FILE_NOT_DIR
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_FILE_NOT_OWNED
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_FILE_NOT_REGULAR
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_FILE_ZERO_LENGTH
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_AUTHZ_FILENAME
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_AUTHZ_LIB_FILENAME
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_CA_CERT_FILENAMES
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_CERT_DIR
 globusgsi.syscon g.datatypes²²
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_CERT_FILENAME
 globusgsi.syscon g.datatypes²²
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_CERT_STRING
 globusgsi.syscon g.datatypes²²
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_CWD
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_DELEG_FILENAME
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_GAA_FILENAME
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_GRIDMAP_FILENAME
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_HOME_DIR
 globusgsi.syscon g.datatypes²²
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_KEY_STRING
 globusgsi.syscon g.datatypes²²
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_PROXY_FILENAME
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_PW_ENTRY
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_GETTING_SIGNING_POLICY
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_LAST
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_REMOVING_OWNED_FILES
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_SETTING_PERMS
 globusgsi.syscon g.datatypes²³
 GLOBUS_GSI_SYSCONFIGERROR_SUCCESS
 globusgsi.syscon g.datatypes²²
 globusgsi.syscon g.error.t
 globusgsi.syscon g.datatypes²²
 GLOBUS_GSI_SYSCONFIGFILE_EXISTS
 globusgsi.systemcon g.denes,²
 globusgsi.syscon g.le_existsunix
 globusgsi.syscon g.unix,⁹
 globusgsi.syscon g.le_existswin32
 globusgsi.syscon g.win32,¹⁶
 GLOBUS_GSI_SYSCONFIGGET_AUTHZ_CONF_FILENAME
 globusgsi.systemcon g.denes,⁴
 globusgsi.syscon g.get_authz_conf.lename.unix
 globusgsi.syscon g.unix,¹³
 globusgsi.syscon g.get_authz_conf.lename.win32
 globusgsi.syscon g.win32,²⁰
 globusgsi.syscon g.get_authzlib_conf.lename.unix
 globusgsi.syscon g.unix,¹³

GLOBUS.GSI.SYSCONFIGGET.CA.CERT.- FILES

globusgsi.systemcon g.de nes, [3](#)
 globusgsi.syscon g.get.ca.cert.lename.unix
 globusgsi.syscon g.unix, [12](#)
 globusgsi.syscon g.get.ca.cert.lename.win32
 globusgsi.syscon g.win32, [19](#)
 GLOBUS.GSI.SYSCONFIGGET.CERT.DIR
 globusgsi.systemcon g.de nes, [3](#)
 globusgsi.syscon g.get.cert.dir.unix
 globusgsi.syscon g.unix, [10](#)
 globusgsi.syscon g.get.cert.dir.win32
 globusgsi.syscon g.win32, [17](#)
 GLOBUS.GSI.SYSCONFIGGET.CURRENT-
WORKING.DIR
 globusgsi.systemcon g.de nes, [3](#)
 globusgsi.syscon g.get.currentworking.dir.unix
 globusgsi.syscon g.unix, [9](#)
 globusgsi.syscon g.get.currentworking.dir.win32
 globusgsi.syscon g.win32, [16](#)
 GLOBUS.GSI.SYSCONFIGGET.GAA.CONF.-
FILENAME
 globusgsi.systemcon g.de nes, [4](#)
 globusgsi.syscon g.get.gaaconf.lename.unix
 globusgsi.syscon g.unix, [13](#)
 globusgsi.syscon g.get.gaaconf.lename.win32
 globusgsi.syscon g.win32, [20](#)
 GLOBUS.GSI.SYSCONFIGGET.GRIDMAP.-
FILENAME
 globusgsi.systemcon g.de nes, [4](#)
 globusgsi.syscon g.get.gridmap.lename.unix
 globusgsi.syscon g.unix, [13](#)
 globusgsi.syscon g.get.gridmap.lename.win32
 globusgsi.syscon g.win32, [19](#)
 GLOBUS.GSI.SYSCONFIGGET.HOME.DIR
 globusgsi.systemcon g.de nes, [2](#)
 globusgsi.syscon g.get.homedir.unix
 globusgsi.syscon g.unix, [9](#)
 GLOBUS.GSI.SYSCONFIGGET.HOST.CERT.-
FILENAME
 globusgsi.systemcon g.de nes, [3](#)
 globusgsi.syscon g.get.hostcert.lename.unix
 globusgsi.syscon g.unix, [11](#)
 globusgsi.syscon g.get.hostcert.lename.win32
 globusgsi.syscon g.win32, [18](#)
 GLOBUS.GSI.SYSCONFIGGET.PROC.ID.-
STRING
 globusgsi.systemcon g.de nes, [4](#)
 globusgsi.syscon g.get.proc.id.string.unix
 globusgsi.syscon g.unix, [8](#)
 GLOBUS.GSI.SYSCONFIGGET.PROXY.-
FILENAME
 globusgsi.systemcon g.de nes, [3](#)

globusgsi.syscon g.get.proxy.lename.unix
 globusgsi.syscon g.unix, [11](#)
 globusgsi.syscon g.get.proxy.lename.win32
 globusgsi.syscon g.win32, [18](#)
 GLOBUS.GSI.SYSCONFIGGET.SERVICE.-
CERT.FILENAME
 globusgsi.systemcon g.de nes, [3](#)
 globusgsi.syscon g.get.servicecert.lename.unix
 globusgsi.syscon g.unix, [11](#)
 globusgsi.syscon g.get.servicecert.lename.win32
 globusgsi.syscon g.win32, [18](#)
 GLOBUS.GSI.SYSCONFIGGET.SIGNING.-
POLICY.FILENAME
 globusgsi.systemcon g.de nes, [3](#)
 globusgsi.syscon g.get.signing.policy.lename.-
unix
 globusgsi.syscon g.unix, [12](#)
 globusgsi.syscon g.get.signing.policy.lename.-
win32
 globusgsi.syscon g.win32, [20](#)
 GLOBUS.GSI.SYSCONFIGGET.UNIQUE.-
PROXY.FILENAME
 globusgsi.systemcon g.de nes, [5](#)
 globusgsi.syscon g.get.unique.proxy.lename
 globusgsi.syscon g.shared, [21](#)
 GLOBUS.GSI.SYSCONFIGGET.USER.CERT.-
FILENAME
 globusgsi.systemcon g.de nes, [3](#)
 globusgsi.syscon g.get.user.cert.lename.unix
 globusgsi.syscon g.unix, [10](#)
 globusgsi.syscon g.get.user.cert.lename.win32
 globusgsi.syscon g.win32, [17](#)
 GLOBUS.GSI.SYSCONFIGGET.USER.ID.-
STRING
 globusgsi.systemcon g.de nes, [4](#)
 globusgsi.syscon g.get.user.id.string.unix
 globusgsi.syscon g.unix, [7](#)
 GLOBUS.GSI.SYSCONFIGGET.USERNAME
 globusgsi.systemcon g.de nes, [5](#)
 globusgsi.syscon g.get.username.unix
 globusgsi.syscon g.unix, [8](#)
 GLOBUS.GSI.SYSCONFIGINVALID_ARG
 globusgsi.syscon g.datatypes, [23](#)
 GLOBUS.GSI.SYSCONFIGIS_SUPERUSER
 globusgsi.systemcon g.de nes, [4](#)
 globusgsi.syscon g.is.superuser.unix
 globusgsi.syscon g.unix, [13](#)
 globusgsi.syscon g.is.superuser.win32
 globusgsi.syscon g.win32, [20](#)
 GLOBUS.GSI.SYSCONFIGMAKE_ABSOLUTE.-
PATH.FOR.FILENAME
 globusgsi.systemcon g.de nes, [4](#)
 globusgsi.syscon g.make.absolute.path.for.-

- lename.unix
- globusgsi.syscon g.unix, 8
- globusgsi.syscon g.makeabsolutepath.for_-
lename.win32
- globusgsi.syscon g.win32, 16
- GLOBUS.GSI.SYSCONFIGMODULE
- globusgsi.syscon g.activation, 22
- GLOBUS.GSI.SYSCONFIGREMOVE.ALL_-
OWNED.FILES
- globusgsi.systemcon g.de nes, 4
- globusgsi.syscon g.removeall_owned les_unix
- globusgsi.syscon g.unix, 12
- globusgsi.syscon g.removeall_owned les_win32
- globusgsi.syscon g.win32, 19
- GLOBUS.GSI.SYSCONFIGSET.KEY_-
PERMISSIONS
- globusgsi.systemcon g.de nes, 2
- globusgsi.syscon g.setkey_permissionsunix
- globusgsi.syscon g.unix, 7
- globusgsi.syscon g.setkey_permissionswin32
- globusgsi.syscon g.win32, 16
- globusgsi.syscon g.shared
- globusgsi.syscon g.getuniqueproxy_-
lename, 21
- GLOBUS.GSI.SYSCONFIGSPLIT.DIR.AND_-
FILENAME
- globusgsi.systemcon g.de nes, 4
- globusgsi.syscon g.split_dir_and lename_unix
- globusgsi.syscon g.unix, 8
- globusgsi.syscon g.split_dir_and lename_win32
- globusgsi.syscon g.win32, 16
- globusgsi.syscon g.unix
- globusgsi.syscon g.checkcert le_unix, 10
- globusgsi.syscon g.checkkey le_unix, 9
- globusgsi.syscon g.dir_existsunix, 9
- globusgsi.syscon g.le_existsunix, 9
- globusgsi.syscon g.get.authzconf lename_-
unix, 13
- globusgsi.syscon g.get.authzlib_conf_-
lename_unix, 13
- globusgsi.syscon g.get.ca.cert les_unix, 12
- globusgsi.syscon g.get.cert.dir_unix, 10
- globusgsi.syscon g.get.currentworking.dir_-
unix, 9
- globusgsi.syscon g.get.gaaconf lename_-
unix, 13
- globusgsi.syscon g.get.gridmap lename_-
unix, 13
- globusgsi.syscon g.get.homedir_unix, 9
- globusgsi.syscon g.get.host.cert lename_-
unix, 11
- globusgsi.syscon g.get.proc.id.string.unix, 8
- globusgsi.syscon g.get.proxy lename_unix,
- 11
- globusgsi.syscon g.get.servicecert lename_-
unix, 11
- globusgsi.syscon g.get.signing.policy_-
lename_unix, 12
- globusgsi.syscon g.get.user.cert lename_-
unix, 10
- globusgsi.syscon g.get.userid.string.unix, 7
- globusgsi.syscon g.get.username_unix, 8
- globusgsi.syscon g.is_superuser_unix, 13
- globusgsi.syscon g.makeabsolutepath.for_-
lename_unix, 8
- globusgsi.syscon g.removeall_owned les_-
unix, 12
- globusgsi.syscon g.setkey_permissionsunix,
- 7
- globusgsi.syscon g.split_dir_and lename_-
unix, 8
- globusgsi.syscon g.win32
- globusgsi.syscon g.dir_existswin32, 16
- globusgsi.syscon g.le_existswin32, 16
- globusgsi.syscon g.get.authzconf lename_-
win32, 20
- globusgsi.syscon g.get.ca.cert les_win32, 19
- globusgsi.syscon g.get.cert.dir_win32, 17
- globusgsi.syscon g.get.currentworking.dir_-
win32, 16
- globusgsi.syscon g.get.gaaconf lename_-
win32, 20
- globusgsi.syscon g.get.gridmap lename_-
win32, 19
- globusgsi.syscon g.get.host.cert lename_-
win32, 18
- globusgsi.syscon g.get.proxy lename_win32,
- 18
- globusgsi.syscon g.get.servicecert lename_-
win32, 18
- globusgsi.syscon g.get.signing.policy_-
lename_win32, 20
- globusgsi.syscon g.get.user.cert lename_-
win32, 17
- globusgsi.syscon g.is_superuserwin32, 20
- globusgsi.syscon g.makeabsolutepath.for_-
lename_win32, 16
- globusgsi.syscon g.removeall_owned les_-
win32, 19
- globusgsi.syscon g.setkey_permissions-
win32, 16
- globusgsi.syscon g.split_dir_and lename_-
win32, 16
- globusgsi.systemcon g.de nes
- GLOBUS.GSI.SYSCONFIGCHECK_-
CERTFILE, 2

GLOBUS_GSI_SYSCONFIGCHECK_-
 KEYFILE, [2](#)
 GLOBUS_GSI_SYSCONFIGDIR_EXISTS, [3](#)
 GLOBUS_GSI_SYSCONFIGFILE_EXISTS, [2](#)
 GLOBUS_GSI_SYSCONFIGGET_AUTHZ_-
 CONF_FILENAME, [4](#)
 GLOBUS_GSI_SYSCONFIGGET_CA_CERT_-
 FILES, [3](#)
 GLOBUS_GSI_SYSCONFIGGET_CERT_DIR,
 [3](#)
 GLOBUS_GSI_SYSCONFIGGET_-
 CURRENT_WORKING_DIR, [3](#)
 GLOBUS_GSI_SYSCONFIGGET_GAA_-
 CONF_FILENAME, [4](#)
 GLOBUS_GSI_SYSCONFIGGET_-
 GRIDMAP_FILENAME, [4](#)
 GLOBUS_GSI_SYSCONFIGGET_HOME_-
 DIR, [2](#)
 GLOBUS_GSI_SYSCONFIGGET_HOST_-
 CERT_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIGGET_PROC_ID_-
 STRING, [4](#)
 GLOBUS_GSI_SYSCONFIGGET_PROXY_-
 FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIGGET_SERVICE_-
 CERT_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIGGET_SIGNING_-
 POLICY_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIGGET_UNIQUE_-
 PROXY_FILENAME, [5](#)
 GLOBUS_GSI_SYSCONFIGGET_USER_-
 CERT_FILENAME, [3](#)
 GLOBUS_GSI_SYSCONFIGGET_USER_ID_-
 STRING, [4](#)
 GLOBUS_GSI_SYSCONFIGGET_-
 USERNAME, [5](#)
 GLOBUS_GSI_SYSCONFIGIS_SUPERUSER,
 [4](#)
 GLOBUS_GSI_SYSCONFIGMAKE_-
 ABSOLUTE_PATH_FOR_FILENAME,
 [4](#)
 GLOBUS_GSI_SYSCONFIGREMOVE_ALL_-
 OWNED_FILES, [4](#)
 GLOBUS_GSI_SYSCONFIGSET_KEY_-
 PERMISSIONS, [2](#)
 GLOBUS_GSI_SYSCONFIGSPLIT_DIR_-
 AND_FILENAME, [4](#)
 GLOBUS_PROXY_FILE_INPUT
 globusgsi.syscon g.datatypes, [23](#)
 GLOBUS_PROXY_FILE_OUTPUT
 globusgsi.syscon g.datatypes, [23](#)