edg-lcas Reference Manual

Generated by Doxygen 1.2.14

Mon Sep 20 15:14:15 2004

Contents

Chapter 1

LCAS - Local Centre Authorization Service

1.1 Introduction

This document describes the LCAS API and the LCAS plugins. Please check the links above.

1.2 the LCAS Interfaces

1. The interface to the LCAS credential mapping framework is described in Interface to LCAS (library)
2. The LCAS plugins should use the LCAS API described in The API to be used by the LCAS plugins
3. The interface that the plugins should provide to the LCAS framework is described in The interface to the LCAS plugins

1.3 The LCAS plugins

A description of the LCAS plugins can be found here ...

... the basic plugins:

1. allowed users plugin
2. banned users plugin
3. time slots plugin

... the voms-aware plugin:

1. voms plugin

Chapter 2

edg-lcas Module Index

2.1 edg-lcas Modules

Here is a list of all modules:

Chapter 3

edg-lcas Data Structure Index

3.1 edg-lcas Data Structures

Here are the data structures with brief descriptions:

Chapter 4

edg-lcas File Index

4.1 edg-lcas File List

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

Chapter 5

edg-lcas Page Index

5.1 edg-lcas Related Pages

Here is a list of all related documentation pages:

Chapter 6

edg-lcas Module Documentation

6.1 Interface to LCAS (library)

The API is available by including the header lcas.h.

Files

- file lcas.h

  API of the LCAS library.

- file lcas_types.h

  Public header file with typedefs for LCAS.

6.1.1 Detailed Description

The API is available by including the header lcas.h.

6.2 The API to be used by the LCAS plugins

The API is available by including the header lcas_modules.h.

Files

- file lcas_defines.h

  Public header file with common definitions for the LCAS (authorization modules).

- file lcas_log.h

  Logging API for the LCAS plugins and LCAS itself.

- file lcas_modules.h

  The LCAS authorization plugins/modules should ”include” this file.

- file lcas_types.h

  Public header file with typedefs for LCAS.

- file lcas_utils.h

  API for the utilities for the LCAS.

- file lcas_vo_data.h

  LCAS module for creating and accessing VO data structures.

6.2.1 Detailed Description

The API is available by including the header lcas_modules.h.

6.3 The interface to the LCAS plugins

Here the interface is shown that the plugin has to provide to the LCAS. The interface consists of the following functions:

1. plugin_initialize()
2. plugin_confirm_authorization()
3. plugin_terminate()

Chapter 7

edg-lcas Data Structure Documentation

7.1 lcas_cred_id_s Struct Reference

structure representing an LCAS credential.

```
#include <lcas_types.h>
```

Data Fields

- gss_cred_id_t cred
- char * dn

7.1.1 Detailed Description

structure representing an LCAS credential.

Definition at line 48 of file lcas_types.h.

7.1.2 Field Documentation

7.1.2.1 gss_cred_id_t lcas_cred_id_s::cred

the original gss (globus) credential

Definition at line 50 of file lcas_types.h.

7.1.2.2 char* lcas_cred_id_s::dn

the user distinguished name (DN)

Definition at line 51 of file lcas_types.h.

The documentation for this struct was generated from the following file:

- lcas_types.h

7.2 lcas_db_entry_s Struct Reference

LCAS data base element structure.

`#include <_lcas_db_read.h>`

Data Fields

- char pluginname [LCAS_MAXPATHLEN+1]
- char pluginargs [LCAS_MAXARGSTRING+1]
- lcas_db_entry_s * next

7.2.1 Detailed Description

LCAS data base element structure.

Definition at line 43 of file _lcas_db_read.h.

7.2.2 Field Documentation

7.2.2.1 struct lcas_db_entry_s* lcas_db_entry_s::next

handle to next db element

Definition at line 47 of file _lcas_db_read.h.

7.2.2.2 char lcas_db_entry_s::pluginargs[LCAS_MAXARGSTRING+1]

Argument list to be passed to authorization plugin/module

Definition at line 46 of file _lcas_db_read.h.

Referenced by lcas_db_parse_line, lcas_db_read_entries, and PluginInit.

7.2.2.3 char lcas_db_entry_s::pluginname[LCAS_MAXPATHLEN+1]

Name of authorization plugin/module

Definition at line 45 of file _lcas_db_read.h.

Referenced by lcas_db_parse_line, lcas_db_read_entries, and PluginInit.

The documentation for this struct was generated from the following file:

- _lcas_db_read.h

7.3 lcas_plugindl_s Struct Reference

the plugin authorization module structure.

Data Fields

- void ∗ handle
- lcas_proc_t procs [MAXPROCS]
- char pluginname [LCAS_MAXPATHLEN+1]
- char pluginargs [LCAS_MAXARGSTRING+1]
- int argc
- char ∗ argv [LCAS_MAXARGS]
- lcas_plugindl_s ∗ next

7.3.1 Detailed Description

the plugin authorization module structure.

Definition at line 144 of file lcas.c.

7.3.2 Field Documentation

7.3.2.1 int lcas_plugindl_s::argc

number of arguments

Definition at line 150 of file lcas.c.

Referenced by clean_plugin_list, PluginInit, and print_lcas_plugin.

7.3.2.2 char∗ lcas_plugindl_s::argv[LCAS_MAXARGS]

list of arguments

Definition at line 151 of file lcas.c.

Referenced by clean_plugin_list, PluginInit, and print_lcas_plugin.

7.3.2.3 void∗ lcas_plugindl_s::handle

dlopen handle to plugin module

Definition at line 146 of file lcas.c.

Referenced by clean_plugin_list, PluginInit, and print_lcas_plugin.

7.3.2.4 struct lcas_plugindl_s∗ lcas_plugindl_s::next

pointer to the next plugin in the plugin list

Definition at line 152 of file lcas.c.

Referenced by clean_plugin_list, PluginInit, and print_lcas_plugin.

7.3.2.5 char lcas_plugindl_s::pluginargs[LCAS_MAXARGSTRING+1]

argument string

Definition at line 149 of file lcas.c.

Referenced by PluginInit, and print_lcas_plugin.

7.3.2.6 char lcas_plugindl_s::pluginname[LCAS_MAXPATHLEN+1]

name of plugin

Definition at line 148 of file lcas.c.

Referenced by clean_plugin_list, PluginInit, and print_lcas_plugin.

7.3.2.7 lcas_proc_t lcas_plugindl_s::procs[MAXPROCS]

list of handles to interface functions of plugin

Definition at line 147 of file lcas.c.

Referenced by clean_plugin_list, PluginInit, and print_lcas_plugin.

The documentation for this struct was generated from the following file:

- lcas.c

7.4 lcas_vo_data_s Struct Reference

structure that contains the VO information found in the user's gss credential.

`#include <lcas_vo_data.h>`

Data Fields

- char ∗ vo
- char ∗ group
- char ∗ subgroup
- char ∗ role
- char ∗ capability

7.4.1 Detailed Description

structure that contains the VO information found in the user's gss credential.

Definition at line 46 of file lcas_vo_data.h.

7.4.2 Field Documentation

7.4.2.1 char∗ lcas_vo_data_s::capability

the user's capability

Definition at line 52 of file lcas_vo_data.h.

Referenced by lcas_cleanVoData, lcas_copyVoData, lcas_createVoData, lcas_gacl_add_vomsdata, lcas_printVoData, and lcas_stringVoData.

7.4.2.2 char∗ lcas_vo_data_s::group

group within the VO

Definition at line 49 of file lcas_vo_data.h.

Referenced by lcas_cleanVoData, lcas_copyVoData, lcas_createVoData, lcas_gacl_add_vomsdata, lcas_printVoData, and lcas_stringVoData.

7.4.2.3 char∗ lcas_vo_data_s::role

the user's role

Definition at line 51 of file lcas_vo_data.h.

Referenced by lcas_cleanVoData, lcas_copyVoData, lcas_createVoData, lcas_gacl_add_vomsdata, lcas_printVoData, lcas_stringVoData, and plugin_confirm_authorization.

7.4.2.4 char∗ lcas_vo_data_s::subgroup

subgroup name

Definition at line 50 of file lcas_vo_data.h.

Referenced by lcas_cleanVoData, lcas_copyVoData, lcas_createVoData, lcas_gacl_add_vomsdata, and lcas_printVoData.

7.4.2.5 char* lcas_vo_data_s::vo

name of the VO to which the user belongs

Definition at line 48 of file lcas_vo_data.h.

Referenced by lcas_cleanVoData, lcas_copyVoData, lcas_createVoData, lcas_gacl_add_vomsdata, lcas_printVoData, and lcas_stringVoData.

The documentation for this struct was generated from the following file:

- lcas_vo_data.h

Chapter 8

edg-lcas File Documentation

8.1 _lcas_db_read.h File Reference

Internal header file of LCAS database reader.

```
#include "_lcas_defines.h"
```

Data Structures

- struct lcas_db_entry_s

  LCAS data base element structure.

Typedefs

- typedef lcas_db_entry_s lcas_db_entry_t

  type of LCAS data base element.

Functions

- lcas_db_entry_t * lcas_db_fill_entry (lcas_db_entry_t **plcas_db, lcas_db_entry_t *db_entry)

  Add a database entry to a list.

- lcas_db_entry_t ** lcas_db_read (char *lcas_db_fname)

  Read database from file.

- int lcas_db_clean_list (lcas_db_entry_t **list)

  Clean/remove the database list.

- int lcas_db_clean ()

  Clean/remove the database structure.

8.1.1 Detailed Description

Internal header file of LCAS database reader.

Author:

Martijn Steenbakkers for the EU DataGrid.

This header contains the declarations of the LCAS database reader functions and typedefs.

Definition in file _lcas_db_read.h.

8.1.2 Function Documentation

8.1.2.1 lcas_db_clean ()

Clean/remove the database structure.

Return values:

0 succes

1 failure

Definition at line 545 of file lcas_db_read.c.

8.1.2.2 lcas_db_clean_list (lcas_db_entry_t ∗∗ *list*)

Clean/remove the database list.

Parameters:

list pointer to the database list

Return values:

0 succes.

1 failure.

Definition at line 522 of file lcas_db_read.c.

8.1.2.3 lcas_db_fill_entry (lcas_db_entry_t ∗∗ *list*, lcas_db_entry_t ∗ *entry*)

Add a database entry to a list.

Parameters:

list database list (array of database entry pointers)

entry the database entry to be added

Returns:

a pointer to the newly created database entry in the list or NULL (error)

Definition at line 178 of file lcas_db_read.c.

Referenced by lcas_db_read_entries.

8.1.2.4 lcas_db_read (char ∗ *lcas_db*)

Read database from file.

Parameters:

lcas_db_fname database file.

Returns:

a pointer to the database list

Definition at line 80 of file lcas_db_read.c.

8.2 _lcas_defines.h File Reference

Internal header file with some common defines for LCAS.

```
#include "lcas_defines.h"
```

8.2.1 Detailed Description

Internal header file with some common defines for LCAS.

Author:

Martijn Steenbakkers for the EU DataGrid.

Definition in file _lcas_defines.h.

8.3 _lcas_log.h File Reference

Internal header file for LCAS logging routines.

```
#include "lcas_log.h"
```

Functions

- int lcas_log_open (char ∗path, FILE ∗fp, unsigned short logtype)

  Start logging.

- int lcas_log_close ()

  Stop logging.

8.3.1 Detailed Description

Internal header file for LCAS logging routines.

Author:
Martijn Steenbakkers for the EU DataGrid.

Definition in file _lcas_log.h.

8.3.2 Function Documentation

8.3.2.1 lcas_log_close ()

Stop logging.

Definition at line 253 of file lcas_log.c.

8.3.2.2 lcas_log_open (char ∗ *path*, FILE ∗ *fp*, unsigned short *logtype*)

Start logging.

This function should only be used by the LCAS itself. It opens the logfile and tries to set the debugging level in the following order:

1. Try if DEBUG_LEVEL > 0
2. Try if environment variable LCAS_DEBUG_LEVEL is set and if it is an integer > 0
3. Otherwise set debug_level = 0;

Parameters:

path path of logfile.

fp file pointer to already opened file (or NULL)

logtype DO_USRLOG, DO_SYSLOG

Return values:

- ***0*** succes.
- ***1*** failure.

Definition at line 60 of file lcas_log.c.

8.4 _lcas_utils.h File Reference

Internal header for the LCAS utilities.

```
#include <gssapi.h>
#include "lcas_types.h"
#include "lcas_utils.h"
```

CREDENTIAL FUNCTIONS

- int lcas_fill_cred (char ∗dn, gss_cred_id_t cred, lcas_cred_id_t ∗lcas_credential)

  Fill cedential from distinghuished name and globus credential.

- int lcas_release_cred (lcas_cred_id_t ∗lcas_credential)

  Release the LCAS credential.

OTHER FUNCTIONS

- int lcas_tokenize (const char ∗command, char ∗∗args, int ∗n, char ∗sep)

  Break the argument string up into tokens.

8.4.1 Detailed Description

Internal header for the LCAS utilities.

Author:

Martijn Steenbakkers for the EU DataGrid.

This header contains the declarations of the LCAS utility functions:

1. lcas_fill_cred():
2. lcas_release_cred():
3. lcas_tokenize():

Definition in file _lcas_utils.h.

8.4.2 Function Documentation

8.4.2.1 lcas_fill_cred (char ∗ *dn*, gss_cred_id_t *cred*, lcas_cred_id_t ∗ *lcas_credential*)

Fill cedential from distinghuished name and globus credential.

The LCAS credential only differs from the GLOBUS credential by the extra entry for the dn. This allows (temporarily) the passed delegated GLOBUS credential to be empty.

Parameters:

dn distinguished name

cred GLOBUS credential

lcas_cred pointer to LCAS credential to be filled.

Return values:

0 succes.

1 failure.

Definition at line 53 of file lcas_utils.c.

8.4.2.2 lcas_release_cred (lcas_cred_id_t ∗ *lcas_credential*)

Release the LCAS credential.

Parameters:

lcas_cred pointer to LCAS credential to be released

Return values:

0 succes.

1 failure.

Definition at line 83 of file lcas_utils.c.

8.4.2.3 lcas_tokenize (const char ∗ *command*, char ∗∗ *args*, int ∗ *n*, char ∗ *sep*)

Break the argument string up into tokens.

Breakup the command in to arguments, pointing the args array at the tokens. Replace white space at the end of each token with a null. A token maybe in quotes. (Copied (and modified) from GLOBUS gatekeeper.c)

Parameters:

command the command line to be parsed

args pointer to an array of pointers to be filled

n size of the array, on input, and set to size used on output

sep string of separating characters

Return values:

0 succes

-1 malloc error

-2 too many args

-3 quote not matched

Definition at line 359 of file lcas_utils.c.

8.5 lcas.c File Reference

LCAS - the local centre authorization service.

```
#include "lcas_config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include <dlfcn.h>
#include <gssapi.h>
#include "lcas_types.h"
#include "_lcas_utils.h"
#include "_lcas_defines.h"
#include "_lcas_log.h"
#include "_lcas_db_read.h"
```

Namespaces

- namespace **lcas_proctype_e**

Data Structures

- struct lcas_plugindl_s

  the plugin authorization module structure.

Typedefs

- typedef int(∗ lcas_proc_t)()

  this type corresponds to the types of the plugin interface functions.

- typedef lcas_plugindl_s lcas_plugindl_t

  the type definition of the plugin authorization module structure.

Functions

- lcas_plugindl_t ∗ PluginInit (lcas_db_entry_t ∗, lcas_plugindl_t ∗∗)

  Initialize the plugin.

- lcas_proc_t get_procsymbol (void ∗, char ∗)

  get procedure symbol from dlopen-ed library.

- int print_lcas_plugin (int, lcas_plugindl_t ∗)

print the lcas_plugindl_t structure.

- int parse_args_plugin (const char ∗, const char ∗, char ∗∗, int ∗)

  convert plugin argument string into xargc, xargv.

- int clean_plugin_list (lcas_plugindl_t ∗∗)

  clean (free) the list of plugins and call the plugin termination functions.

8.5.1 Detailed Description

LCAS - the local centre authorization service.

Author:

Martijn Steenbakkers for the EU DataGrid.

The interface to the LCAS module is composed of:

1. lcas_init(): To initialize the LCAS module
2. lcas_get_fabric_authorization(): to place an authorization request
3. lcas_term(): To cleanly terminate the module

Definition in file lcas.c.

8.5.2 Function Documentation

8.5.2.1 int clean_plugin_list (lcas_plugindl_t ∗∗ *list*) `[static]`

clean (free) the list of plugins and call the plugin termination functions.

Parameters:

list

list pointer to list of plugins which has to be freeed.

Return values:

0 succes.

1 failure.

Definition at line 832 of file lcas.c.

References lcas_plugindl_s::argc, lcas_plugindl_s::argv, lcas_plugindl_s::handle, lcas_plugindl_s::next, lcas_plugindl_s::pluginname, and lcas_plugindl_s::procs.

8.5.2.2 lcas_proc_t get_procsymbol (void ∗ *handle*, char ∗ *symname*) `[static]`

get procedure symbol from dlopen-ed library.

Parameters:

handle handle of dynamic library

symname name of procedure symbol

Returns:

handle to procedure symbol or NUll

Definition at line 792 of file lcas.c.

References lcas_proc_t.

Referenced by PluginInit.

8.5.2.3 int parse_args_plugin (const char * *name*, const char * *argstring*, char ** *xargv*, int * *xargc*) [static]

convert plugin argument string into xargc, xargv.

Parse the argument string of the plugin and create xargv and xargc

Parameters:

name name of the plugin (goes into xargv[0])

argstring string containing the arguments

xargv array of argument strings (has to be freed later)

xargc number of arguments

Return values:

0 succes.

1 failure.

Definition at line 731 of file lcas.c.

Referenced by PluginInit.

8.5.2.4 lcas_plugindl_t * PluginInit (lcas_db_entry_t * *db_handle*, lcas_plugindl_t ** *list*) [static]

Initialize the plugin.

This function takes a plugin LCAS database entry and performs the following tasks:

- Create entry in (plugin)list
- Open the plugins and check the symbols plugin_init and confirm_authorization
- run plugin_init

Parameters:

db_handle handle to LCAS db (containing pluginname and pluginargs)

list pointer to plugin structure list to which (plugin) module has to be added

Returns:

pointer to newly created plugin structure or NULL in case of failure

Definition at line 479 of file lcas.c.

References lcas_plugindl_s::argc, lcas_plugindl_s::argv, get_procsymbol, lcas_plugindl_s::handle, lcas_proc_t, lcas_plugindl_s::next, parse_args_plugin, lcas_plugindl_s::pluginargs, lcas_db_entry_s::pluginargs, lcas_plugindl_s::pluginname, lcas_db_entry_s::pluginname, and lcas_plugindl_s::procs.

8.5.2.5 int print_lcas_plugin (int *debug_lvl*, lcas_plugindl_t * *plugin*) [static]

print the lcas_plugindl_t structure.

Parameters:

debug_lvl debugging level

plugin plugin structure

Return values:

0 succes.

1 failure.

Definition at line 895 of file lcas.c.

References lcas_plugindl_s::argc, lcas_plugindl_s::argv, lcas_plugindl_s::handle, lcas_plugindl_s::next, lcas_plugindl_s::pluginargs, lcas_plugindl_s::pluginname, and lcas_plugindl_s::procs.

8.5.3 Variable Documentation

8.5.3.1 char* authmods[MAXAUTHMODS][2] [static]

Initial value:

```
 {
                             {(char *) NULL, (char *) NULL},
                             {(char *) NULL, (char *) NULL},
                             {(char *) NULL, (char *) NULL}
}
```

Definition at line 174 of file lcas.c.

8.6 lcas.h File Reference

API of the LCAS library.

```
#include <gssapi.h>
#include "lcas_types.h"
```

Functions

- int lcas_init (FILE *fp)

  Initialize the LCAS module.

- int lcas_term ()

  Terminate the LCAS module.

- int lcas_get_fabric_authorization (gss_cred_id_t user_cred, lcas_request_t request)

  submit authorization request to the LCAS.

8.6.1 Detailed Description

API of the LCAS library.

Author:

Martijn Steenbakkers for the EU DataGrid.

This header contains the declarations of the LCAS library functions:

1. lcas_init(): To initialize the LCAS module
2. lcas_get_fabric_authorization(): to place an authorization request
3. lcas_term(): To cleanly terminate the module

Definition in file lcas.h.

8.6.2 Function Documentation

8.6.2.1 lcas_get_fabric_authorization (gss_cred_id_t *user_cred*, lcas_request_t *request*)

submit authorization request to the LCAS.

This function calls the plugins for authorization.

Parameters:

request authorization request in RSL (later JDL)

user_cred GLOBUS user credential

Return values:

0 initialization succeeded.

1 initialization failed.

Definition at line 941 of file lcas.c.

8.6.2.2 lcas_init (FILE * *fp*)

Initialize the LCAS module.

The function does the following:

- initialize LCAS module.
- setup logging, error handling (not yet).
- read from LCAS database the plugins to be loaded.
- initialize the plugins

Parameters:

fp file handle for logging (from gatekeeper)

Return values:

0 initialization succeeded.

1 initialization failed.

Definition at line 429 of file lcas.c.

8.6.2.3 lcas_term ()

Terminate the LCAS module.

The function does the following:

- terminate the LCAS module
- terminate the plugins

Return values:

0 initialization succeeded.

1 initialization failed.

Definition at line 1070 of file lcas.c.

8.7 lcas_db_read.c File Reference

the LCAS database reader.

```
#include <stdlib.h>
#include <malloc.h>
#include <stdio.h>
#include <string.h>
#include "lcas_log.h"
#include "_lcas_db_read.h"
```

Defines

- #define PAIR_SEP_CHARS ","
- #define VARVAL_SEP_CHARS "="
- #define PAIR_TERMINATOR_CHARS PAIR_SEP_CHARS WHITESPACE_CHARS
- #define VARVAL_TERMINATOR_CHARS VARVAL_SEP_CHARS WHITESPACE_CHARS

Functions

- int lcas_db_read_entries (FILE *)

  Read db entries from stream and fill a lsit of db entries.

- int lcas_db_parse_line (char *, lcas_db_entry_t **)

  Parses database line and fills database structure.

- int lcas_db_parse_pair (char *, char **, char **)

  Parses a database variable-value pair and returns the variable name and its value.

- int lcas_db_parse_string (char **)

  Takes a string and removes prepending and trailing spaces and quotes (unless escaped).

Variables

- lcas_db_entry_t * lcas_db_list = NULL

8.7.1 Detailed Description

the LCAS database reader.

Author:

Martijn Steenbakkers for the EU DataGrid.

Definition in file lcas_db_read.c.

8.7.2 Define Documentation

8.7.2.1 #define PAIR_SEP_CHARS ","

Characters separating variable-value pairs in the lcas database file

Definition at line 40 of file lcas_db_read.c.

Referenced by lcas_db_parse_line.

8.7.2.2 #define PAIR_TERMINATOR_CHARS PAIR_SEP_CHARS WHITESPACE_CHARS

Characters that terminate pairs in the lcas database file. This is a combination of whitespace and separators.

Definition at line 52 of file lcas_db_read.c.

8.7.2.3 #define VARVAL_SEP_CHARS "="

Characters separating variables from values

Definition at line 42 of file lcas_db_read.c.

Referenced by lcas_db_parse_pair.

8.7.2.4 #define VARVAL_TERMINATOR_CHARS VARVAL_SEP_CHARS WHITESPACE_CHARS

Characters that terminate variables and values in the lcas database file. This is a combination of whitespace and separators.

Definition at line 57 of file lcas_db_read.c.

8.7.3 Function Documentation

8.7.3.1 int lcas_db_parse_line (char ∗ *line*, lcas_db_entry_t ∗∗ *entry*) `[static]`

Parses database line and fills database structure.

Parameters:

line database line

entry pointer to a pointer to a database structure (can/should be freed afterwards)

Return values:

1 succes.

0 failure.

Definition at line 241 of file lcas_db_read.c.

References lcas_db_parse_pair, PAIR_SEP_CHARS, lcas_db_entry_s::pluginargs, and lcas_db_entry_s::pluginname.

Referenced by lcas_db_read_entries.

8.7.3.2 int lcas_db_parse_pair (char ∗ *pair*, char ∗∗ *pvar*, char ∗∗ *pval*) `[static]`

Parses a database variable-value pair and returns the variable name and its value.

Parameters:

pair string containing the pair

pvar pointer to the variable string

pval pointer to the value string

Return values:

1 succes.

0 failure.

Definition at line 374 of file lcas_db_read.c.

References lcas_db_parse_string, and VARVAL_SEP_CHARS.

Referenced by lcas_db_parse_line.

8.7.3.3 int lcas_db_parse_string (char ∗∗ *pstring*) `[static]`

Takes a string and removes prepending and trailing spaces and quotes (unless escaped).

Parameters:

pstring pointer to a pointer to a char

Return values:

1 succes.

0 failure.

Definition at line 471 of file lcas_db_read.c.

Referenced by lcas_db_parse_pair.

8.7.3.4 int lcas_db_read_entries (FILE ∗ *dbstream*) `[static]`

Read db entries from stream and fill a lsit of db entries.

Parameters:

dbstream database stream

Returns:

the number of entries found (failure –> negative number)

Definition at line 123 of file lcas_db_read.c.

References lcas_db_fill_entry, lcas_db_parse_line, lcas_db_entry_s::pluginargs, and lcas_db_entry_s::pluginname.

8.7.4 Variable Documentation

8.7.4.1 lcas_db_entry_t∗ lcas_db_list = NULL `[static]`

list of database entries

Definition at line 74 of file lcas_db_read.c.

8.8 lcas_defines.h File Reference

Public header file with common definitions for the LCAS (authorization modules).

8.8.1 Detailed Description

Public header file with common definitions for the LCAS (authorization modules).

Author:

Martijn Steenbakkers for the EU DataGrid.

Here the return values for the LCAS plugins/modules are defined as well as the default locations of the LCAS ”etc”, ”lib” and ”modules” directories.

Definition in file lcas_defines.h.

8.9 lcas_log.c File Reference

Logging routines for LCAS.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdarg.h>
#include <syslog.h>
#include <time.h>
#include <ctype.h>
#include "_lcas_log.h"
```

Variables

- FILE ∗ lcas_logfp = NULL
- int logging_usrlog = 0
- int logging_syslog = 0
- int debug_level = 0
- char ∗ extra_logstr = NULL
- int should_close_lcas_logfp = 0

8.9.1 Detailed Description

Logging routines for LCAS.

Author:

Martijn Steenbakkers for the EU DataGrid.

Definition in file lcas_log.c.

8.9.2 Variable Documentation

8.9.2.1 int debug_level = 0 `[static]`

debugging level

Definition at line 45 of file lcas_log.c.

8.9.2.2 char∗ extra_logstr = NULL `[static]`

string to be included in every log statement

Definition at line 46 of file lcas_log.c.

8.9.2.3 FILE∗ lcas_logfp = NULL `[static]`

stream associated with logfile

Definition at line 41 of file lcas_log.c.

8.9.2.4 int logging_syslog = 0 `[static]`

flag to use syslog

Definition at line 43 of file lcas_log.c.

8.9.2.5 int logging_usrlog = 0 `[static]`

flag to do user logging

Definition at line 42 of file lcas_log.c.

8.9.2.6 int should_close_lcas_logfp = 0 `[static]`

Flag to check if the log stream should be closed

Definition at line 47 of file lcas_log.c.

8.10 lcas_log.h File Reference

Logging API for the LCAS plugins and LCAS itself.

`#include <syslog.h>`

Functions

- int lcas_log (int prty, char ∗fmt,...)

  log information.

- int lcas_log_debug (int debug_lvl, char ∗fmt,...)

  Print debugging information.

- int lcas_log_time (int prty, char ∗fmt,...)

  log information with timestamp.

- int lcas_get_debug_level ()

  Retrieve the debug_level.

8.10.1 Detailed Description

Logging API for the LCAS plugins and LCAS itself.

Author:

Martijn Steenbakkers for the EU DataGrid.

This header contains the declarations of the LCAS logging functions The LCAS plugins can use this API to write output to the LCAS logging devices.

1. lcas_log(): Log to LCAS logging devices.
2. lcas_log_debug(): Produce debugging output.

Definition in file lcas_log.h.

8.10.2 Function Documentation

8.10.2.1 lcas_get_debug_level ()

Retrieve the debug_level.

Returns:

the debug_level

Definition at line 384 of file lcas_log.c.

Referenced by lcas_check_gacl.

8.10.2.2 lcas_log (int *prty*, char ∗ *fmt*, ...)

log information.

This function does the logging for the LCAS and its plugins. Syslog() is called with the specified priority. No syslog() is done if the priority is 0.

Parameters:

prty syslog priority (if 0 don't syslog).

fmt string format

... variable argument list

Return values:

0 succes.

1 failure.

Definition at line 175 of file lcas_log.c.

8.10.2.3 lcas_log_debug (int *debug_lvl*, char ∗ *fmt*, ...)

Print debugging information.

This function prints debugging information (using lcas_log with priority 0) provided debug_lvl <= DEBUG_LEVEL (default is 0).

Parameters:

debug_lvl debugging level

fmt string format

... variable argument list

Return values:

0 succes.

1 failure.

Definition at line 224 of file lcas_log.c.

8.10.2.4 lcas_log_time (int *prty*, char ∗ *fmt*, ...)

log information with timestamp.

This function logs information with a timestamp for LCAS and its plugins. Syslog() is called with the specified priority. No syslog() is done if the priority is 0.

Parameters:

prty syslog priority (if 0 don't syslog).

fmt string format

... variable argument list

Return values:

0 succes.

1 failure.

Definition at line 299 of file lcas_log.c.

8.11 lcas_modules.h File Reference

The LCAS authorization plugins/modules should ”include” this file.

```
#include <gssapi.h>
#include "lcas_utils.h"
#include "lcas_log.h"
#include "lcas_types.h"
#include "lcas_defines.h"
```

8.11.1 Detailed Description

The LCAS authorization plugins/modules should ”include” this file.

Author:

Martijn Steenbakkers for the EU DataGrid.

This file includes the header files that are needed by the LCAS authorization plugins/modules.

Definition in file lcas_modules.h.

8.12 lcas_plugin_example.c File Reference

Interface to the LCAS plugins.

```
#include "lcas_config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "lcas_modules.h"
```

Functions

- int plugin_initialize (int argc, char **argv)

  initialize the plugin.

- int plugin_confirm_authorization (lcas_request_t request, lcas_cred_id_t lcas_cred)

  **.

- int plugin_terminate ()

  Whatever is needed to terminate the plugin module goes in here.

8.12.1 Detailed Description

Interface to the LCAS plugins.

Author:
Martijn Steenbakkers for the EU DataGrid.

This file contains the code for an example LCAS plugin and shows the interface the plugin has to provide to the LCAS. The interface consists of the following functions:

1. plugin_initialize()
2. plugin_confirm_authorization()
3. plugin_terminate()

Definition in file lcas_plugin_example.c.

8.12.2 Function Documentation

8.12.2.1 plugin_confirm_authorization (lcas_request_t *request*, lcas_cred_id_t *lcas_cred*)

**.

Ask for authorization by passing the RSL (later JDL) and the user credential. The user credential will contain information on the role the user wants to have. In the RSL (JDL) the user might specify the resources he wants to use. The authorization decision has to be made using this information. The LCAS provides no library for parsing the RSL (JDL).

Parameters:

request LCAS (RSL) request

lcas_cred LCAS credential

Return values:

LCAS_MOD_SUCCESS authorization succeeded

LCAS_MOD_FAIL authorization failed

LCAS_MOD_NOFILE private plugin database could not be found (LCAS will deny authorization)

Definition at line 132 of file lcas_plugin_example.c.

References lcas_request_t.

8.12.2.2 plugin_initialize (int *argc*, char ∗∗ *argv*)

initialize the plugin.

Everything that is needed to initialize the plugin should be put inside this function. Arguments as read from the LCAS database (argc, argv) are passed to the plugin.

Parameters:

argc number of passed arguments.

argv argument list. argv[0] contains the name of the plugin.

Return values:

LCAS_MOD_SUCCESS successful initialization

LCAS_MOD_FAIL failure in the plugin initialization

LCAS_MOD_NOFILE private plugin database could not be found (same effect as LCAS_MOD_FAIL)

Definition at line 84 of file lcas_plugin_example.c.

References plugin_initialize.

Referenced by plugin_initialize.

8.12.2.3 plugin_terminate ()

Whatever is needed to terminate the plugin module goes in here.

Return values:

LCAS_MOD_SUCCESS success

LCAS_MOD_FAIL failure (will result in an authorization failure)

Definition at line 186 of file lcas_plugin_example.c.

8.13 lcas_timeslots.c File Reference

Interface to the LCAS plugins.

```
#include "lcas_config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include <time.h>
#include <values.h>
#include "lcas_modules.h"
```

Functions

- int plugin_confirm_authorization (lcas_request_t request, lcas_cred_id_t lcas_cred)

  **.

8.13.1 Detailed Description

Interface to the LCAS plugins.

Author:
Martijn Steenbakkers for the EU DataGrid.

LCAS plugin that makes authorization decisions based on available time slots Currently it reads a text file that contains the available time slots.

1. plugin_initialize()
2. plugin_confirm_authorization()
3. plugin_terminate()

Definition in file lcas_timeslots.c.

8.13.2 Variable Documentation

8.13.2.1 char∗ days[7] [static]

Initial value:

```
{
    "Sunday",
    "Monday",
    "Tuesday",
    "Wednesday",
    "Thursday",
        "Friday",
        "Saturday"
    }
```

Definition at line 177 of file lcas_timeslots.c.

8.13.2.2 char∗ months[12] `[static]`

Initial value:

```
    {
        "Jan",
        "Feb",
        "Mar",
        "Apr",
        "May",
        "Jun",
        "Jul",
        "Aug",
        "Sep",
        "Oct",
        "Nov",
        "Dec",
    }
```

Definition at line 187 of file lcas_timeslots.c.

8.14 lcas_types.h File Reference

Public header file with typedefs for LCAS.

```
#include <gssapi.h>
```

Data Structures

- struct lcas_cred_id_s

  structure representing an LCAS credential.

Typedefs

- typedef char ∗ lcas_request_t

  Type of the LCAS request expressed in RSL/JDL.

- typedef lcas_cred_id_s lcas_cred_id_t

  Type of LCAS credentials.

8.14.1 Detailed Description

Public header file with typedefs for LCAS.

Author:

Martijn Steenbakkers for the EU DataGrid.

Definition in file lcas_types.h.

8.14.2 Typedef Documentation

8.14.2.1 lcas_request_t

Type of the LCAS request expressed in RSL/JDL.

(Internal) just a string.

Definition at line 38 of file lcas_types.h.

Referenced by plugin_confirm_authorization.

8.15 lcas_userallow.c File Reference

Interface to the LCAS plugins.

```
#include "lcas_config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include "lcas_modules.h"
#include "lcas_gridlist.h"
```

Functions

- int plugin_confirm_authorization (lcas_request_t request, lcas_cred_id_t lcas_cred)

  **.

8.15.1 Detailed Description

Interface to the LCAS plugins.

Author:

Martijn Steenbakkers for the EU DataGrid.

This file contains the code for the plugin that checks a plain text file containing a list of the DN (subjects of X509 certificates) of the users that are *allowed* on the site. The interface consists of the following functions:

1. plugin_initialize()
2. plugin_confirm_authorization()
3. plugin_terminate()

Definition in file lcas_userallow.c.

8.16 lcas_userban.c File Reference

Interface to the LCAS plugins.

```
#include "lcas_config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include "lcas_modules.h"
#include "lcas_gridlist.h"
```

Functions

- int plugin_confirm_authorization (lcas_request_t request, lcas_cred_id_t lcas_cred)

  ***.

8.16.1 Detailed Description

Interface to the LCAS plugins.

Author:

Martijn Steenbakkers for the EU DataGrid.

This file contains the code for the plugin that checks a plain text file containing a list of the DN (subjects of X509 certificates) of the users that are *banned* on the site. The interface consists of the following functions:

1. plugin_initialize()
2. plugin_confirm_authorization()
3. plugin_terminate()

Definition in file lcas_userban.c.

8.17 lcas_utils.c File Reference

the utilities for the LCAS.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <errno.h>
#include <stdarg.h>
#include <gssapi.h>
#include "lcas_defines.h"
#include "lcas_types.h"
```

Functions

- char ∗ cred_to_dn (gss_cred_id_t)

  Get the globus DN from GLOBUS credential (gssapi).

- int fexist (char ∗)

  check the existence of file corresponding to <path>.

8.17.1 Detailed Description

the utilities for the LCAS.

Author:

Martijn Steenbakkers for the EU DataGrid.

Definition in file lcas_utils.c.

8.17.2 Function Documentation

8.17.2.1 char ∗ cred_to_dn (gss_cred_id_t *globus_cred*) `[static]`

Get the globus DN from GLOBUS credential (gssapi).

(copied and modified from GLOBUS gatekeeper.c)

Parameters:

globus_cred GLOBUS credential

Returns:

globus DN string (which may be freed)

Definition at line 143 of file lcas_utils.c.

8.17.2.2 int fexist (char ∗ *path*) `[static]`

check the existence of file corresponding to <path>.

Parameters:

path absolute filename to be checked.

Return values:

1 file exists.

0 failure.

Definition at line 252 of file lcas_utils.c.

8.18 lcas_utils.h File Reference

API for the utilities for the LCAS.

`#include <gssapi.h>`

`#include "lcas_types.h"`

CREDENTIAL FUNCTIONS

- char ∗ lcas_get_dn (lcas_cred_id_t lcas_credential)

  Retrieve user DN from (LCAS) credential.

- gss_cred_id_t lcas_get_gss_cred (lcas_cred_id_t lcas_credential)

  Retrieve globus gss credential from (LCAS) credential.

FILENAME FUNCTIONS

- char ∗ lcas_genfilename (char ∗prefix, char ∗path, char ∗suffix)

  Generate an absolute file name.

- char ∗ lcas_getfexist (int n,...)

  Picks the first existing file in argument list.

- char ∗ lcas_findfile (char ∗name)

  Checks for file in standard directories.

8.18.1 Detailed Description

API for the utilities for the LCAS.

Author:

Martijn Steenbakkers for the EU DataGrid.

This header contains the declarations of the LCAS utility functions:

1. lcas_get_dn():

2. lcas_genfilename():

3. lcas_getfexist():

4. lcas_findfile():

Definition in file lcas_utils.h.

8.18.2 Function Documentation

8.18.2.1 lcas_findfile (char ∗ *name*)

Checks for file in standard directories.

The directories that are checked are:

- current directory
- ”modules”
- LCAS_ETC_HOME
- LCAS_MOD_HOME
- LCAS_LIB_HOME

Parameters:

name string containing the file name

Returns:

pointer to a string containing the absolute path to the file, which has to be freed or NULL.

Definition at line 308 of file lcas_utils.c.

8.18.2.2 lcas_genfilename (char ∗ *prefix*, char ∗ *path*, char ∗ *suffix*)

Generate an absolute file name.

Given a starting prefix, a relative or absolute path, and a suffix an absolute file name is generated. Uses the prefix only if the path is relative. (Copied (and modified) from GLOBUS gatekeeper.c)

Parameters:

prefix string containing the prefix to be prepended.

path relative/absolute path to file name.

suffix string containing the suffix to be appended.

Returns:

pointer to a string containing the absolute path to the file, which has to be freed.

Definition at line 195 of file lcas_utils.c.

8.18.2.3 lcas_get_dn (lcas_cred_id_t *lcas_credential*)

Retrieve user DN from (LCAS) credential.

This function takes an LCAS credential as input and returns the corresponding user distinguished name (DN).

(Internal:) If the GLOBUS credential part of the LCAS credential is empty the user DN is already included in the LCAS credential.

Parameters:

lcas_credential the LCAS credential

Returns:

a string containing the user DN

Definition at line 104 of file lcas_utils.c.

8.18.2.4 lcas_get_gss_cred (lcas_cred_id_t *lcas_credential*)

Retrieve globus gss credential from (LCAS) credential.

This function takes an LCAS credential as input and returns the corresponding globus gss credential.

Parameters:

lcas_credential the LCAS credential

Returns:

globus gss credential

Definition at line 118 of file lcas_utils.c.

8.18.2.5 lcas_getfexist (int *n*, ...)

Picks the first existing file in argument list.

Parameters:

n the number of paths presented in the following argument list.

... variable argument list of paths.

Returns:

filename found or NULL

Definition at line 283 of file lcas_utils.c.

8.19 lcas_vo_data.c File Reference

LCAS utilities for creating and accessing VO data structures.

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>
#include "lcas_vo_data.h"
#include "lcas_log.h"
```

Functions

- lcas_vo_data_t ∗ lcas_createVoData (const char ∗vo, const char ∗group, const char ∗subgroup, const char ∗role, const char ∗capability)

  Create a VoData structure.

- int lcas_deleteVoData (lcas_vo_data_t ∗∗vo_data)

  Delete a VoData structure.

- int lcas_cleanVoData (lcas_vo_data_t ∗vo_data)

  Clean a VoData structure.

- int lcas_copyVoData (lcas_vo_data_t ∗dst_vo_data, const lcas_vo_data_t ∗src_vo_data)

  Copy a VoData structure into an empty VoData structure.

- int lcas_printVoData (int debug_level, const lcas_vo_data_t ∗vo_data)

  Print the contents of a VoData structure.

- int lcas_stringVoData (const lcas_vo_data_t ∗vo_data, char ∗buffer, int nchars)

  Cast a VoData structure into a string.

- char ∗ lcas_parseVostring (char ∗vo_string)

  Strip leading whitespace and check if string != ”NULL”.

8.19.1 Detailed Description

LCAS utilities for creating and accessing VO data structures.

Author:

Martijn Steenbakkers for the EU DataGrid.

The interface is composed of:

1. lcas_createVoData(): create a VoData structure
2. lcas_deleteVoData(): delete a VoData structure

3. lcas_copyVoData(): copy (the contents of) a VoData structure
4. lcas_printVoData(): print the contents of a VoData structure
5. lcas_stringVoData(): cast a VoData structure into a string

Definition in file lcas_vo_data.c.

8.19.2 Function Documentation

8.19.2.1 lcas_cleanVoData (lcas_vo_data_t * *vo_data*)

Clean a VoData structure.

Clean a VoData structure that was previously filled with lcas_copyVoData(). The contents are freed and set to zero.

Parameters:

vo_data a pointer to a VoData structure

Return values:

0 in case of success

-1 in case of failure

Definition at line 192 of file lcas_vo_data.c.

References lcas_vo_data_s::capability, lcas_vo_data_s::group, lcas_cleanVoData, lcas_vo_data_s::role, lcas_vo_data_s::subgroup, and lcas_vo_data_s::vo.

Referenced by lcas_cleanVoData.

8.19.2.2 lcas_copyVoData (lcas_vo_data_t * *dst_vo_data*, const lcas_vo_data_t * *src_vo_data*)

Copy a VoData structure into an empty VoData structure.

Copy a VoData structure into an empty VoData structure which has to exist.

Parameters:

dst_vo_data pointer to a empty VoData structure that should be filled

src_vo_data pointer to the VoData structure that should be copied

Return values:

0 success

-1 failure (either src_vo_data or dst_vo_data was empty)

Definition at line 260 of file lcas_vo_data.c.

References lcas_vo_data_s::capability, lcas_vo_data_s::group, lcas_copyVoData, lcas_vo_data_s::role, lcas_vo_data_s::subgroup, and lcas_vo_data_s::vo.

Referenced by lcas_copyVoData.

8.19.2.3 lcas_createVoData (const char ∗ *vo*, const char ∗ *group*, const char ∗ *subgroup*, const char ∗ *role*, const char ∗ *capability*)

Create a VoData structure.

Create a VoData structure (store a VO, group, (subgroup,) role, capability combination). Allocate the memory. To be freed with lcas_deleteVoData().

Parameters:

- ***vo*** name of the VO
- ***group*** name of the group
- ***subgroup*** name of the subgroup (ignored for the moment)
- ***role*** the role
- ***capability*** the capability (whatever it is)

Returns:

pointer to the VoData structure or NULL

Definition at line 78 of file lcas_vo_data.c.

References lcas_vo_data_s::capability, lcas_vo_data_s::group, lcas_createVoData, lcas_vo_data_s::role, lcas_vo_data_s::subgroup, and lcas_vo_data_s::vo.

Referenced by lcas_createVoData.

8.19.2.4 lcas_deleteVoData (lcas_vo_data_t ∗∗ *vo_data*)

Delete a VoData structure.

Delete a VoData structure that was previously created with lcas_createVoData(). The pointer to the VoData structure is finally set to NULL;

Parameters:

- ***vo_data*** pointer to a pointer to a VoData structure

Return values:

- ***0*** in case of success
- ***-1*** in case of failure

Definition at line 138 of file lcas_vo_data.c.

References lcas_deleteVoData.

Referenced by lcas_deleteVoData.

8.19.2.5 lcas_parseVostring (char ∗ *vo_string*)

Strip leading whitespace and check if string != ”NULL”.

This function is needed because VOMS server fills user credential sometimes with strings like ” NULL”, which is a valid string, but the intention is that the data is empty. A string like this is translated into a NULL pointer by this function.

Parameters:

- ***vo_string*** string of VO credential

Returns:

pointer to the parsed string or NULL

Definition at line 548 of file lcas_vo_data.c.

References lcas_parseVostring.

Referenced by lcas_parseVostring, and lcas_stringVoData.

8.19.2.6 lcas_printVoData (int *debug_level*, const lcas_vo_data_t ∗ *vo_data*)

Print the contents of a VoData structure.

Parameters:

vo_data pointer to a VoData structure

debug_level debug_level for which the contents will be printed

Returns:

0 (always)

Definition at line 321 of file lcas_vo_data.c.

References lcas_vo_data_s::capability, lcas_vo_data_s::group, lcas_printVoData, lcas_vo_data_s::role, lcas_vo_data_s::subgroup, and lcas_vo_data_s::vo.

Referenced by lcas_printVoData.

8.19.2.7 lcas_stringVoData (const lcas_vo_data_t ∗ *vo_data*, char ∗ *buffer*, int *nchars*)

Cast a VoData structure into a string.

The user of this function should create the buffer of size nchars beforehand. In buffer a string like the following will be written: ”/VO=fred/GROUP=fred/flintstone/ROLE=director/CAPABILITY=destroy”

Currently the SUBGROUP entry is ignored. Only if the information is present in the VoData structure, it is added to the string. Both data for VO and GROUP are required (might change).

Parameters:

vo_data pointer to a VoData structure

buffer pointer to character array of size nchars

nchars size of character array

Return values:

0 in case of success

-1 in case of failure

Definition at line 389 of file lcas_vo_data.c.

References lcas_vo_data_s::capability, lcas_vo_data_s::group, lcas_parseVostring, lcas_stringVoData, lcas_vo_data_s::role, and lcas_vo_data_s::vo.

Referenced by lcas_stringVoData.

8.20 lcas_vo_data.h File Reference

LCAS module for creating and accessing VO data structures.

Data Structures

- struct lcas_vo_data_s

  structure that contains the VO information found in the user's gss credential.

8.20.1 Detailed Description

LCAS module for creating and accessing VO data structures.

Author:

Martijn Steenbakkers for the EU DataGrid.

The interface is composed of:

1. lcas_createVoData(): create a VoData structure
2. lcas_deleteVoData(): delete a VoData structure
3. lcas_copyVoData(): copy (the contents of) a VoData structure
4. lcas_printVoData(): print the contents of a VoData structure
5. lcas_stringVoData(): cast a VoData structure into a string

Definition in file lcas_vo_data.h.

8.21 lcas_voms.c File Reference

Interface to the LCAS plugins.

```
#include "lcas_config.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pwd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <libgen.h>
#include <openssl/x509.h>
#include <errno.h>
#include "gssapi.h"
#include "lcas_modules.h"
#include "lcas_voms_utils.h"
#include "lcas_vo_data.h"
#include "lcas_gridlist.h"
#include "voms_apic.h"
#include "gacl.h"
#include "globus_gss_assist.h"
```

Enumerations

- enum authformat_e { NO_FORMAT, SIMPLE_FORMAT, GACL_FORMAT, XACML_FORMAT }

  This enumeration type gives the different plugin symbol/function types.

- enum gacl_use_voms_dn_e { ALWAYS_USE_VOMS_DN, USE_VOMS_DN, DONT_USE_VOMS_DN }

Functions

- int lcas_check_gacl (GACLuser *, char *)

  check the LCAS GACL.

- int lcas_gacl_add_dn (GACLuser **, char *)

  Add the user_dn to gacluser.

- int lcas_gacl_add_vomsdata (GACLuser **, lcas_vo_data_t *, char *)

  Add the VOMS data to the gacl user.

- int plugin_confirm_authorization (lcas_request_t request, lcas_cred_id_t lcas_cred)

  ***.

8.21.1 Detailed Description

Interface to the LCAS plugins.

Author:

Martijn Steenbakkers for the EU DataGrid.

This file contains the code for the voms plugin (extracts the VOMS info from the certificate). The interface consists of the following functions:

1. plugin_initialize()
2. plugin_confirm_authorization()
3. plugin_terminate()

Definition in file lcas_voms.c.

8.21.2 Enumeration Type Documentation

8.21.2.1 enum authformat_e

This enumeration type gives the different plugin symbol/function types.

Enumeration values:

NO_FORMAT this value corresponds to no format at all

SIMPLE_FORMAT this value corresponds to the 'simple' authorization file format

GACL_FORMAT this value corresponds to the 'gacl' authorization file format

XACML_FORMAT this value corresponds to the 'xacml' authorization file format

Definition at line 137 of file lcas_voms.c.

8.21.2.2 enum gacl_use_voms_dn_e

Enumeration values:

ALWAYS_USE_VOMS_DN always include the VOMS dn in the gacl user credential

USE_VOMS_DN create a gacl user credential with and without the VOMS dn

DONT_USE_VOMS_DN do not include the VOMS dn in the gacl user credential

Definition at line 146 of file lcas_voms.c.

8.21.3 Function Documentation

8.21.3.1 int lcas_check_gacl (GACLuser ∗ *gacluser*, char ∗ *gaclfile*) `[static]`

check the LCAS GACL.

Apply the LCAS authorization GACL to the user credentials (VOMS cred. and DN)

Parameters:

gacluser the gacl_user, which consists of his DN and VOMS entries

gaclfile the file containing the LCAS GACL

Return values:

0 success

1 failure

Definition at line 1362 of file lcas_voms.c.

References lcas_get_debug_level.

Referenced by plugin_confirm_authorization.

8.21.3.2 int lcas_gacl_add_dn (GACLuser ∗∗ *pgacluser*, char ∗ *user_dn*) `[static]`

Add the user_dn to gacluser.

Add the user_dn to gacluser. If the gacluser does not exist, it will be created.

Parameters:

pgacluser pointer to the gacl user

user_dn the DN of the user

Return values:

0 success

1 failure

Definition at line 1137 of file lcas_voms.c.

Referenced by plugin_confirm_authorization.

8.21.3.3 int lcas_gacl_add_vomsdata (GACLuser ∗∗ *pgacluser*, lcas_vo_data_t ∗ *lcas_voms_data*, char ∗ *voms_server_dn*) `[static]`

Add the VOMS data to the gacl user.

Add the VOMS data to the gacl user. If the gacluser does not exist, it will be created.

Parameters:

pgacluser pointer to the gacl user

lcas_voms_data the gathered VOMS data structure

voms_server_dn the DN of the VOMS server that signed the VOMS certificate

Return values:

0 success

1 failure

Definition at line 1236 of file lcas_voms.c.

References lcas_vo_data_s::capability, lcas_vo_data_s::group, lcas_vo_data_s::role, lcas_vo_data_s::subgroup, and lcas_vo_data_s::vo.

Referenced by plugin_confirm_authorization.

8.22 lcas_voms_utils.c File Reference

the utilities for the LCAS voms plugin.

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include "lcas_defines.h"
#include "lcas_types.h"
#include "lcas_log.h"
#include <openssl/x509.h>
#include <gssapi.h>
#include "gssapi_openssl.h"
#include "globus_gsi_credential.h"
```

Functions

- X509 * lcas_cred_to_x509 (gss_cred_id_t cred)

  Return the pointer to X509 structure from gss credential.

8.22.1 Detailed Description

the utilities for the LCAS voms plugin.

Author:

Martijn Steenbakkers for the EU DataGrid.

This header contains the definitions of the LCAS utility functions:

1. lcas_cred_to_x509():
2. lcas_cred_to_x509_chain():

Definition in file lcas_voms_utils.c.

8.22.2 Function Documentation

8.22.2.1 lcas_cred_to_x509 (gss_cred_id_t *cred*)

Return the pointer to X509 structure from gss credential.

This function takes a gss credential as input and returns the corresponding X509 structure, which is allocated for this purpose (should be freed)

Parameters:

cred the gss credential

Returns:

a pointer to a X509 struct or NULL

Definition at line 85 of file lcas_voms_utils.c.

References lcas_cred_to_x509.

Referenced by lcas_cred_to_x509.

8.23 lcas_voms_utils.h File Reference

API for the utilities for the LCAS voms plugin.

```
#include <openssl/x509.h>
#include <gssapi.h>
```

8.23.1 Detailed Description

API for the utilities for the LCAS voms plugin.

Author:
Martijn Steenbakkers for the EU DataGrid.

This header contains the declarations of the LCAS utility functions:

1. lcas_cred_to_x509():
2. lcas_cred_to_x509_chain():

Definition in file lcas_voms_utils.h.

Chapter 9

edg-lcas Page Documentation

9.1 allowed users plugin

9.2 SYNOPSIS

lcas_userallow.mod [<allowed user file>]

9.3 DESCRIPTION

This plugin checks a file that contains a list of DNs (subjects of the X509 certificate) of allowed users. If the DN of the user for which the authorization request is made is found in the list, the plugin grants access to the site.

9.4 OPTIONS

9.4.1 [<allowed user file>]

The name of the file that contains the list of allowed user DNs. *NOTE*: Currently this option is ignored; the grid-mapfile is always used.

9.5 RETURN VALUES

- LCAS_MOD_SUCCESS : Success
- LCAS_MOD_NOFILE : the allowed user file could not be found
- LCAS_MOD_FAIL : Failure

9.6 ERRORS

See bugzilla for known errors (`http://marianne.in2p3.fr/datagrid/bugzilla/`)

9.7 SEE ALSO

lcas_userban.mod, lcas_timeslots.mod, lcas_voms.mod,

9.8 banned users plugin

9.9 SYNOPSIS

lcas_userban.mod [<banned user file>]

9.10 DESCRIPTION

This plugin checks a file that contains a list of DNs (subjects of the X509 certificate) of users to be *banned* from the site. If the DN of the user for which the authorization request is made is found in the list, the plugin *denies* access to the site.

9.11 OPTIONS

9.11.1 [<banned user file>]

The name of the file that contains the list of banned user DNs. Default search path: `/opt/edg/lib/lcas/modules`.

9.12 RETURN VALUES

- LCAS_MOD_SUCCESS : Success
- LCAS_MOD_NOFILE : the banned user file could not be found
- LCAS_MOD_FAIL : Failure

9.13 ERRORS

See bugzilla for known errors (`http://marianne.in2p3.fr/datagrid/bugzilla/`)

9.14 SEE ALSO

lcas_userallow.mod, lcas_timeslots.mod, lcas_voms.mod,

9.15 time slots plugin

9.16 SYNOPSIS

lcas_timeslots.mod [<time slots file>]

9.17 DESCRIPTION

Author:

Martijn Steenbakkers for the EU DataGrid.

This plugin makes an authorization decisions based on available time slots. Currently it reads a text file that contains the available time slots.

9.18 OPTIONS

9.18.1 [<time slots file>]

The name of the file that contains the list of time slots. Default search path: `/opt/edg/lib/lcas/modules`. The format of the file is described here:

```
#
# This file contains the time slots for which the fabric
# is available for Grid jobs
# Format:
#        minute1-minute2 hour1-hour2 mday1-mday2 month1-month2 year1-year2 wday1-wday2
# max range:  [0-59]        [0-23]      [1-31]       [1-12]    [1970-...]     [0-6]
#
# wday:
#  0-6 = Sunday-Saturday
#  5-3 = Friday-Wednesday
#
# '*' means the maximum range
# <val>- means from <val> to maximum value
#
# The wall clock time should match at least one time slot for authorization
# The wall clock time matches if:
#           (hour1:minute1)      <= (hour:minute)     <= (hour2:minute2)
#       AND (year1.month1.mday1) <= (year.month.mday) <= (year2.month2.mday2)
#       AND (wday1)              <= (wday)            <= (wday2)
#
# If the fabric is open on working days from 8:30-18:00 h, from 1 July 2002 till 15 January 2003
# the following line should be uncommented:
#      30-0           8-18        1-15         7-1       2002-2003      1-5
#
# If the fabric is open from 18:00-7:00 h, two time slots should be used:
#      18:00-24:00 and 0:00-7:00
#
#       0-0          18-24         *            *            *           *
#       0-0           0-7          *            *            *           *
# If the fabric is always open the following line should be uncommented:
#         *             *           *            *            *           *
```

9.19 RETURN VALUES

- LCAS_MOD_SUCCESS : Success
- LCAS_MOD_NOFILE : the time slots file could not be found
- LCAS_MOD_FAIL : Failure

9.20 ERRORS

See bugzilla for known errors (`http://marianne.in2p3.fr/datagrid/bugzilla/`)

9.21 SEE ALSO

lcas_userallow.mod, lcas_userban.mod, lcas_voms.mod,

9.22 voms plugin

9.23 SYNOPSIS

lcas_voms.mod -vomsdir <vomsdir> -certdir <certdir> -authfile <authorization file> [-authformat <format of the authorization file>]

9.24 DESCRIPTION

This plugin forms the link between the VOMS data found in the user grid credential (X509 certificate) and the lcas system. It will retrieve the VOMS data by using the VOMS API. The VOMS data will be checked against either a (simple) gridmap style file, a GACL-file or an XACML-file in order for the user job to be authorized on the site.

9.25 OPTIONS

9.25.1 -VOMSDIR <vomsdir>

See -vomsdir

9.25.2 -vomsdir <vomsdir>

This is the directory which contains the certificates of the VOMS servers

9.25.3 -CERTDIR <certdir>

See -certdir

9.25.4 -certdir <certdir>

This is the directory which contains the CA certificates

9.25.5 -authfile <authorization file>

In this file the authorization/access control based on VOMS information is specified. The format of this file is ’simple’ (gridmap style), ’gacl’ or ’xacml’, which can be specified explicitly with the option -authformat or will be derived form the suffix of the authorization file (`.gacl` and `.xacml` for ’gacl’ and ’xacml’ formats, otherwise ’simple’).

9.25.6 -authformat <format of the authorization file>

Format of the autorization file, values: `gacl/GACL`, `xacml/XACML` or `simple`.

9.25.7 -gacl_use_voms_dn [yes|no|always]

GACL specific. This option specifies if the voms DN, found in the user certificate, should be included in the user gacl credential. Default is 'yes'. The following arguments are recognized:

- `yes` : For each VO-GROUP-ROLE combination found in the user certificate two gacl credentials are created: one *with* and one *without* the voms DN. In this way the user is also authorized if in the gacl in the authorization file the voms DN is not included (better if it is, though).
- `always` : For each VO-GROUP-ROLE combination found in the user certificate only a gacl credential is created *with* the voms DN.
- `no` : For each VO-GROUP-ROLE combination found in the usercertificate a gacl credential is created *without* the voms DN.

9.25.8 -use_user_dn

If this option is set also user proxies without voms information will be processed. If the user dn of the proxy is present in the gacl or gridmapfile, the user is authorized by this plugin.

9.26 RETURN VALUES

- LCAS_MOD_SUCCESS : Success
- LCAS_MOD_FAIL : Failure

9.27 ERRORS

See bugzilla for known errors (`http://marianne.in2p3.fr/datagrid/bugzilla/`)

9.28 SEE ALSO

lcas_userallow.mod, lcas_userban.mod, lcas_timeslots.mod,

Index