VOMS CC API Reference Manual

1.5.0

Generated by Doxygen 1.2.18

Tue Jun 30 05:34:26 2009

Contents

Chapter 1

VOMS CC API Data Structure Index

1.1 VOMS CC API Data Structures

Here are the data structures with brief descriptions:

Chapter 2

VOMS CC API File Index

2.1 VOMS CC API File List

Here is a list of all files with brief descriptions:

Chapter 3

VOMS CC API Data Structure Documentation

3.1 attribute Struct Reference

#include <voms_api.h>

Data Fields

- std::string name
- std::string qualifier
- std::string value

3.1.1 Field Documentation

3.1.1.1 std::string attribute::name

attribute's group

Definition at line 46 of file voms_api.h.

3.1.1.2 std::string attribute::qualifier

attribute's qualifier

Definition at line 47 of file voms_api.h.

3.1.1.3 std::string attribute::value

attribute's value

Definition at line 48 of file voms_api.h.

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

- voms_api.h

3.2 attributelist Struct Reference

#include <voms_api.h>

Data Fields

- std::string grantor
- std::vector< attribute > attributes

3.2.1 Field Documentation

3.2.1.1 std::vector< attribute > attributelist::attributes

The attributes themselves.

Definition at line 53 of file voms_api.h.

3.2.1.2 std::string attributelist::grantor

Who granted these attributes.

Definition at line 52 of file voms_api.h.

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

- voms_api.h

3.3 contactdata Struct Reference

`#include <voms_api.h>`

Data Fields

- std::string nick
- std::string host
- std::string contact
- std::string vo
- int port
- int version

3.3.1 Field Documentation

3.3.1.1 std::string contactdata::contact

The subject of the server's certificate

Definition at line 71 of file voms_api.h.

3.3.1.2 std::string contactdata::host

The hostname of the server

Definition at line 70 of file voms_api.h.

3.3.1.3 std::string contactdata::nick

The alias of the server

Definition at line 69 of file voms_api.h.

3.3.1.4 int contactdata::port

The port on which the server is listening

Definition at line 73 of file voms_api.h.

3.3.1.5 int contactdata::version

The version of globus under which the server is running

Definition at line 75 of file voms_api.h.

3.3.1.6 std::string contactdata::vo

The VO served by this server

Definition at line 72 of file voms_api.h.

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

voms_api.h

3.4 data Struct Reference

User's characteristics: can be repeated.

`#include <voms_api.h>`

Data Fields

- std::string group
- std::string role
- std::string cap

3.4.1 Detailed Description

User's characteristics: can be repeated.

Definition at line 37 of file voms_api.h.

3.4.2 Field Documentation

3.4.2.1 std::string data::cap

user's capability

Definition at line 40 of file voms_api.h.

3.4.2.2 std::string data::group

user's group

Definition at line 38 of file voms_api.h.

3.4.2.3 std::string data::role

user's role

Definition at line 39 of file voms_api.h.

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

- voms_api.h

3.5 voms Struct Reference

#include < voms_api.h >

Public Methods

- voms (const voms &)
- voms ()
- voms & operator= (const voms &)
- voms ()
- AC GetAC ()
- std::vector< attributelist > & GetAttributes ()
- std::vector< std::string > GetTargets ()

Data Fields

- int version
- int siglen
- std::string signature
- std::string user
- std::string userca
- std::string server
- std::string serverca
- std::string voname
- std::string uri
- std::string date1
- std::string date2
- datatype type
- std::vector< data > std
- std::string custom
- std::vector< std::string > fqan
- std::string serial

Friends

- class vomsdata
- int TranslateVOMS (struct vomsdatarvd, std::vector< voms > &v, int error)

3.5.1 Constructor & Destructor Documentation

3.5.1.1 voms::voms (const voms &)

3.5.1.2 voms::voms ()

3.5.1.3 voms:: voms ()

3.5.2 Member Function Documentation

3.5.2.1 AC voms::GetAC ()

3.5.2.2 std::vector< attributelist > & voms::GetAttributes ()

Generic attributes

3.5.2.3 std::vector< std::string > voms::GetTargets ()

3.5.2.4 voms& voms::operator= (const voms &)

3.5.3 Friends And Related Function Documentation

3.5.3.1 int TranslateVOMS (struct vomsdatar vd, std::vector< voms > & v, int error) [friend]

3.5.3.2 friend class vomsdata [friend]

Definition at line 79 of file voms_api.h.

3.5.4 Field Documentation

3.5.4.1 std::string voms::custom

The data returned by an S command

Definition at line 93 of file voms_api.h.

3.5.4.2 std::string voms::date1

Beginning of validity of the user info

Definition at line 89 of file voms_api.h.

3.5.4.3 std::string voms::date2

End of validity of the user info

Definition at line 90 of file voms_api.h.

3.5.4.4 std::vector< std::string > voms::fqan

Keeps the data in the compact format

Definition at line 95 of file voms_api.h.

3.5.4.5 std::string voms::serial

Serial number. "0" if coming from non-ac

Definition at line 96 of file voms_api.h.

3.5.4.6 std::string voms::server

The VOMS server DN, as from its certificate

Definition at line 85 of file voms_api.h.

3.5.4.7 std::string voms::serverca

The CA which signed the VOMS certificate

Definition at line 86 of file voms_api.h.

3.5.4.8 int voms::siglen

The length of the VOMS server signature

Definition at line 81 of file voms_api.h.

3.5.4.9 std::string voms::signature

The VOMS server signature

Definition at line 82 of file voms_api.h.

3.5.4.10 std::vector< data > voms::std

User's characteristics

Definition at line 92 of file voms_api.h.

3.5.4.11 data_type voms::type

The type of data returned

Definition at line 91 of file voms_api.h.

3.5.4.12 std::string voms::uri

The URI of the VOMS server

Definition at line 88 of file voms_api.h.

3.5.4.13 std::string voms::user

The user's DN, as from his certificate

Definition at line 83 of file voms_api.h.

3.5.4.14 std::string voms::userca

The CA which signed the user's certificate

Definition at line 84 of file voms_api.h.

3.5.4.15 int voms::version

0 means data didn't originate from an AC

Definition at line 80 of file voms_api.h.

3.5.4.16 std::string voms::voname

The name of the VO to which the VOMS belongs

Definition at line 87 of file voms_api.h.

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

- voms_api.h

3.6 vomsdata Struct Reference

#include <voms_api.h>

Public Methods

vomsdata (std::string voms_dir="", std::string cert_dir="")
bool LoadSystemContacts (std::string dir="")
bool LoadUserContacts (std::string dir="")
std::vector< contactdata > FindByAlias (std::string alias)
std::vector< contactdata > FindByVO (std::string vo)
void Order (std::string att)
void ResetOrder (void)
void AddTarget (std::string target)
std::vector< std::string > ListTargets (void)
void ResetTargets (void)
std::string ServerErrors (void)
bool Retrieve (X509 cert, STACK_OF(X509) chain, recurse_type how=RECURSE_CHAIN)
bool Contact (std::string hostname, int port, std::string servsubject, std::string command)
bool ContactRaw (std::string hostname, int port, std::string servsubject, std::string command, std::string &raw, int &version)
void SetVerificationType (verify_type how)
void SetLifetime (int lifetime)
bool Import (std::string buffer)
bool Export (std::string &data)
bool DefaultData (voms &)
std::string ErrorMessage (void)
bool RetrieveFromCtx (gss_ctx_id_t context, recurse_type how)
bool RetrieveFromCred (gss_cred_id_t credential, recurse_type how)
bool Retrieve (X509_EXTENSION ext)
bool RetrieveFromProxy (recurse_type how)
bool Retrieve (FILE file, recurse_type how)
~vomsdata ()
vomsdata (const vomsdata &)
void SetRetryCount (int retryCount)
void SetVerificationTime (time_t)
bool LoadCredentials (X509 cert, STACK_OF(X509) chain, EVP_PKEY key)

Data Fields

verror_type error
std::vector< voms > data
std::string workvo
std::string extra_data

3.6.1 Constructor & Destructor Documentation

3.6.1.1 vomsdata::vomsdata (std::string voms_dir = "", std::string cert_dir = "")

Parameters:

voms_dir The directory which contains the certificate of the VOMS server

cert_dir The directory which contains the certificate of the CA

If voms_dir is empty, the value of the environment variable X509_VOMS_DIR is taken.

If cert_dir is empty, the value of the environment variable X509_CERT_DIR is taken.

3.6.1.2 vomsdata::~vomsdata ()

3.6.1.3 vomsdata::vomsdata (const vomsdata &)

3.6.2 Member Function Documentation

3.6.2.1 void vomsdata::AddTarget (std::string target)

Adds a target to the AC.

Parameters:

target The target to be added. it should be a FQDN.

3.6.2.2 bool vomsdata::Contact (std::string hostname, int port, std::string servsubject, std::string command)

Contacts a VOMS server to get a certificate

It is the equivalent of the voms_proxy_init command, but without the --include functionality.

Parameters:

hostname	FQDN of the VOMS server
port	the port on which the VOMS server is listening
servsubject	the subject of the server's certificate
command	the command sent to the server

Returns:

failure (F) or success (T)

3.6.2.3 bool vomsdata::ContactRaw (std::string hostname, int port, std::string servsubject, std::string command, std::string & raw, int & version)

Same as Contact, however it does not start the verification process, and the message receviedfrom the server is not parsed.

Parameters:

hostname	FQDN of the VOMS server
port	the port on which the VOMS server is listening

servsubject the subject of the server's certificate
command the command sent to the server
raw OUTPUT PARAMETER the answer from the server
version OUTPUT PARAMETER the version of the answer

Returns:
failure (F) or success (T)

3.6.2.4 bool vomsdata::DefaultData (voms &)

Get the default data extension from those present in the pseudo certificate

3.6.2.5 std::string vomsdata::ErrorMessage (void)

Gets a textual description of the error.

Returns:
A string containg the error message.

3.6.2.6 bool vomsdata::Export (std::string & data)

Exports data from vomsdata::data to the format used for inclusion into a certificate.

The function doesn't verify the data

Parameters:
data The certificate extension

Returns:
Failure (F) or Success (T)

3.6.2.7 std::vector< contactdata > vomsdata::FindByAlias (std::string alias)

Finds servers which share a common alias.

Parameters:
alias The alias to look for.

Returns:
The servers found. The order in which they are returned is unspecified.

3.6.2.8 std::vector< contactdata > vomsdata::FindByVO (std::string vo)

Finds servers which serve a common VO

Parameters:
vo The VO name to look for.

Returns:
The servers found. The order in which they are returned is unspecified.

3.6.2.9 bool vomsdata::Import (std::string buffer)

Converts data from the format used for inclusion into a certificate to the internal format

The function does verify the data.

Parameters:
  buffer contains the data to be converted

Returns:
  Failure (F) or Success (T)

3.6.2.10 std::vector< std::string > vomsdata::ListTargets (void)

Returns the list of targets.

3.6.2.11 bool vomsdata::LoadCredentials (X509 cert, STACK_OF(X509) chain, EVP_PKEY key)

3.6.2.12 bool vomsdata::LoadSystemContacts (std::string dir = "")

Loads the system wide configuration files.

Parameters:
  dir The directory in which the files are stored.

If dir is empty, defaults to /opt/edg/etc/vomses.

Returns:
  True if all went OK, false otherwise.

3.6.2.13 bool vomsdata::LoadUserContacts (std::string dir = "")

Loads the user-specific configuration files.

Parameters:
  dir The directory in which the files are stored.

If dir is empty, defaults to $VOMSUSERCONF. If this is empty too, defaults to $HOME/.edg/vomses, or to /.edg/vomses as a last resort.

Returns:
  True if all went OK, false otherwise.

3.6.2.14 void vomsdata::Order (std::string att)

Sets up the ordering of the results.

Defines the ordering of the data returned by Contact(). Results are ordered in the same order as the calls to this function.

Parameters:
  att The attribute to be ordered.

3.6.2.15 void vomsdata::ResetOrder (void)

Resets the ordering.

3.6.2.16 void vomsdata::ResetTargets (void)

Resets the target list.

3.6.2.17 bool vomsdata::Retrieve (FILE file, recurse_type how)

Gets VOMS information from a proxy saved as a file.

Parameters:

the file

how Recursion type

Returns:

failure (F) or success (T)

Note: Does NOT verify that the proxy is valid. Such verification must be obtained through other means.

3.6.2.18 bool vomsdata::Retrieve (X509_EXTENSION ext)

Gets VOMS information from the given extension

Parameters:

ext The extension to parse.

Returns:

failure (F) or success (T)

3.6.2.19 bool vomsdata::Retrieve (X509 cert, STACK_OF(X509) chain, recurse_type how = RECURSE_CHAIN)

Extracts the VOMS extension from an X.509 certificate. The function doesn't check the validity of the certificates, but it does check the content of the user data.

Parameters:

cert The certificate with the VOMS extensions

chain The chain of the validation certificates (only the intermediate ones)

how Recursion type

Returns:

failure (F) or success (T)

3.6.2.20 bool vomsdata::RetrieveFromCred (gss_cred_id_t credential, recurse_type how)

Gets VOMS information from the given globus credential

Parameters:

credential The credential from which to retrieve the certificate.

how Recursion type

Returns:

failure (F) or success (T)

3.6.2.21 bool vomsdata::RetrieveFromCtx (gss_ctx_id_t context, recurse_type how)

Gets VOMS information from the given globus context

Parameters:

context The context from which to retrieve the certificate.

how Recursion type

Returns:

failure (F) or success (T)

3.6.2.22 bool vomsdata::RetrieveFromProxy (recurse_type how)

Gets VOMS information from an existing globus proxy

Parameters:

how Recursion type

Returns:

failure (F) or success (T)

3.6.2.23 std::string vomsdata::ServerErrors (void)

Gets the error message returned by the server

3.6.2.24 void vomsdata::SetLifetime (int lifetime)

Set requested lifetime for the Contact() call.

Parameters:

lifetime Requested lifetime, in seconds

3.6.2.25 void vomsdata::SetRetryCount (int retryCount)

3.6.2.26 void vomsdata::SetVerificationTime (time_t)

3.6.2.27 void vomsdata::SetVerificationType (verify_type how)

Sets the type of verification done on the data.

Parameters:
how The type of verification.

3.6.3 Field Documentation

3.6.3.1 std::vector<voms> vomsdata::data

User's info, as in the certificate extension. It may contain data gathered from more than one VOMS server,

Definition at line 316 of file voms_api.h.

3.6.3.2 verror_type vomsdata::error

Error code

Definition at line 188 of file voms_api.h.

3.6.3.3 std::string vomsdata::extra_data

The data specified by the user with the –include switch.

Note that this field doesn't contain the result of a request to the VOMS server, but instead data specified by the user.

The reason for the introduction of this extension is to let a user include important data into his proxy certificate, like, for example, a kerberos ticket

Definition at line 320 of file voms_api.h.

3.6.3.4 std::string vomsdata::workvo

The value of the -vo option of the voms-proxy-init command

Definition at line 319 of file voms_api.h.

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

- voms_api.h

Chapter 4

VOMS CC API File Documentation

4.1 voms_api.h File Reference

#include <fstream>

#include <string>

#include <vector>

#include <openssl/x509.h>

#include <openssl/bio.h>

#include <sys/types.h>

#include "newformat.h"

Data Structures

struct attribute
struct attributelist
struct contactdata
struct data

User's characteristics: can be repeated.

class Initializer
struct voms
struct vomsdata

Typedefs

typedef void gss_cred_id_t
typedef void gss_ctx_id_t
typedef bool(checksig)(X509 , void , verror_type &)

Enumerations

enum datatype { TYPE_NODATA, TYPE_STD, TYPE_CUSTOM }

The type of data returned.

- enum recurse_type f RECURSE_CHAIN, RECURSE_NONE, RECURSE_DEEP g
- enum verify_type f VERIFY_FULL = 0xffffffff, VERIFY_NONE = 0x00000000, VERIFY_DATE = 0x00000001, VERIFY_TARGET = 0x00000002, VERIFY_KEY = 0x00000004, VERIFY_SIGN = 0x00000008, VERIFY_ORDER = 0x00000010, VERIFY_ID = 0x00000020, VERIFY_CERTLIST = 0x00000040 g
- enum verror_type f VERR_NONE, VERR_NOSOCKET, VERR_NOIDENT, VERR_COMM, VERR_PARAM, VERR_NOEXT, VERR_NOINIT, VERR_TIME, VERR_IDCHECK, VERR_EXTRAINFO, VERR_FORMAT, VERR_NODATA, VERR_PARSE, VERR_DIR, VERR_SIGN, VERR_SERVER, VERR_MEM, VERR_VERIFY, VERR_TYPE, VERR_ORDER, VERR_SERVERCODE, VERR_NOTAVAIL, VERR_FILE g

  Error codes.

Functions

- int getMajorVersionNumber (void)
- int getMinorVersionNumber (void)
- int getPatchVersionNumber (void)

4.1.1 Typedef Documentation

4.1.1.1 typedef bool(check_sig)(X509 , void , verror_type &)

Definition at line 167 of file voms_api.h.

4.1.1.2 typedef void gss_cred_id_t

Definition at line 25 of file voms_api.h.

4.1.1.3 typedef void gss_ctx_id_t

Definition at line 26 of file voms_api.h.

4.1.2 Enumeration Type Documentation

4.1.2.1 enum data_type

The type of data returned.

Enumeration values:

TYPE_NODATA	no data
TYPE_STD	group, role, capability triplet
TYPE_CUSTOM	result of an S command

Definition at line 59 of file voms_api.h.

4.1.2.2 enum recurse_type

Enumeration values:

- RECURSE_CHAIN
- RECURSE_NONE
- RECURSE_DEEP

Definition at line 120 of file voms_api.h.

4.1.2.3 enum verify_type

Enumeration values:

- VERIFY_FULL
- VERIFY_NONE
- VERIFY_DATE
- VERIFY_TARGET
- VERIFY_KEY
- VERIFY_SIGN
- VERIFY_ORDER
- VERIFY_ID
- VERIFY_CERTLIST

Definition at line 126 of file voms_api.h.

4.1.2.4 enum verror_type

Error codes.

Enumeration values:

VERR_NONE	
VERR_NOSOCKET	Socket problem
VERR_NOIDENT	Cannot identify itself (certificate problem)
VERR_COMM	Server problem
VERR_PARAM	Wrong parameters
VERR_NOEXT	VOMS extension missing
VERR_NOINIT	Initialization error
VERR_TIME	Error in time checking
VERR_IDCHECK	User data in extension different from the real ones
VERR_EXTRAINFO	VO name and URI missing
VERR_FORMAT	Wrong data format
VERR_NODATA	Empty extension
VERR_PARSE	Parse error
VERR_DIR	Directory error
VERR_SIGN	Signature error

VERR_SERVER	Unidentifiable VOMS server
VERR_MEM	Memory problems
VERR_VERIFY	Generic verification error
VERR_TYPE	Returned data of unknown type
VERR_ORDER	Ordering different than required
VERR_SERVERCODE	Error message from the server
VERR_NOTAVAIL	Method not available
VERR_FILE	Error reading data from file

Definition at line 140 of file voms_api.h.

4.1.3 Function Documentation

4.1.3.1 int getMajorVersionNumber (void)

4.1.3.2 int getMinorVersionNumber (void)

4.1.3.3 int getPatchVersionNumber (void)

Index