VOMS CC API Reference Manual

1.5.0

Generated by Doxygen 1.4.4

Tue Jun 30 05:29:44 2009

Contents

1 VOMS CC API Data Structure Index **1**

1.1 VOMS CC API Data Structures 1

2 VOMS CC API File Index **3**

2.1 VOMS CC API File List 3

3 VOMS CC API Data Structure Documentation **5**

3.1 attribute Struct Reference 5

3.2 attributelist Struct Reference 7

3.3 contactdata Struct Reference 8

3.4 data Struct Reference 10

3.5 voms Struct Reference 11

3.6 vomsdata Struct Reference 15

4 VOMS CC API File Documentation **23**

4.1 voms_api.h File Reference 23

Chapter 1

VOMS CC API Data Structure Index

1.1 VOMS CC API Data Structures

Here are the data structures with brief descriptions:

attribute 5
attributelist 7
contactdata 8
data (User's characteristics: can be repeated) 10
voms 11
vomsdata 15

Chapter 2

VOMS CC API File Index

2.1 VOMS CC API File List

Here is a list of all files with brief descriptions:

voms_api.h 23

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 Detailed Description

Definition at line 45 of file voms_api.h.

3.1.2 Field Documentation

3.1.2.1 std::string attribute::name

attribute's group

Definition at line 46 of file voms_api.h.

3.1.2.2 std::string attribute::qualifier

attribute's qualifier

Definition at line 47 of file voms_api.h.

3.1.2.3 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 Detailed Description

Definition at line 51 of file voms_api.h.

3.2.2 Field Documentation

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

The attributes themselves.

Definition at line 53 of file voms_api.h.

3.2.2.2 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 Detailed Description

Definition at line 65 of file voms_api.h.

3.3.2 Field Documentation

3.3.2.1 std::string contactdata::contact

The subject of the server's certificate

Definition at line 71 of file voms_api.h.

3.3.2.2 std::string contactdata::host

The hostname of the server

Definition at line 70 of file voms_api.h.

3.3.2.3 std::string contactdata::nick

< You must never allocate directly this structure. Its sizeof() is subject to change without notice. The only supported way to obtain it is via the FindBy∗ functions. The alias of the server

Definition at line 69 of file voms_api.h.

3.3.2.4 int contactdata::port

The port on which the server is listening

Definition at line 73 of file voms_api.h.

3.3.2.5 int contactdata::version

The version of globus under which the server is running

Definition at line 75 of file voms_api.h.

3.3.2.6 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 Member Functions

- 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
- data_type type
- std::vector< data > std
- std::string custom
- std::vector< std::string > fqan
- std::string serial

Friends

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

3.5.1 Detailed Description

Definition at line 78 of file voms_api.h.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 voms::voms (const voms &)

3.5.2.2 voms::voms ()

3.5.2.3 voms::~voms ()

3.5.3 Member Function Documentation

3.5.3.1 AC* voms::GetAC ()

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

Generic attributes

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

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

3.5.4 Friends And Related Function Documentation

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

3.5.4.2 friend class vomsdata [friend]

Definition at line 79 of file voms_api.h.

3.5.5 Field Documentation

3.5.5.1 std::string voms::custom

The data returned by an S command

Definition at line 93 of file voms_api.h.

3.5.5.2 std::string voms::date1

Beginning of validity of the user info

Definition at line 89 of file voms_api.h.

3.5.5.3 std::string voms::date2

End of validity of the user info

Definition at line 90 of file voms_api.h.

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

Keeps the data in the compact format

Definition at line 95 of file voms_api.h.

3.5.5.5 std::string voms::serial

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

Definition at line 96 of file voms_api.h.

3.5.5.6 std::string voms::server

The VOMS server DN, as from its certificate

Definition at line 85 of file voms_api.h.

3.5.5.7 std::string voms::serverca

The CA which signed the VOMS certificate

Definition at line 86 of file voms_api.h.

3.5.5.8 int voms::siglen

The length of the VOMS server signature

Definition at line 81 of file voms_api.h.

3.5.5.9 std::string voms::signature

The VOMS server signature

Definition at line 82 of file voms_api.h.

3.5.5.10 std::vector<data> voms::std

User's characteristics

Definition at line 92 of file voms_api.h.

3.5.5.11 data_type voms::type

The type of data returned

Definition at line 91 of file voms_api.h.

3.5.5.12 std::string voms::uri

The URI of the VOMS server

Definition at line 88 of file voms_api.h.

3.5.5.13 std::string voms::user

The user's DN, as from his certificate

Definition at line 83 of file voms_api.h.

3.5.5.14 std::string voms::userca

The CA which signed the user's certificate

Definition at line 84 of file voms_api.h.

3.5.5.15 int voms::version

0 means data didn't originate from an AC

Definition at line 80 of file voms_api.h.

3.5.5.16 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 Member Functions

- 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

Data Structures

- class **Initializer**

3.6.1 Detailed Description

Definition at line 169 of file voms_api.h.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 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.

3.6.2.2 vomsdata::~vomsdata ()

3.6.2.3 vomsdata::vomsdata (const vomsdata &)

3.6.3 Member Function Documentation

3.6.3.1 void vomsdata::AddTarget (std::string *target*)

Adds a target to the AC.

Parameters:

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

3.6.3.2 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)

3.6.3.3 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)

3.6.3.4 bool vomsdata::DefaultData (voms &)

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

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

Gets a textual description of the error.

Returns:

A string containg the error message.

3.6.3.6 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)

3.6.3.7 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.

3.6.3.8 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.

3.6.3.9 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)

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

Returns the list of targets.

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

3.6.3.12 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.

3.6.3.13 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 $VOMS_USERCONF. 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.

3.6.3.14 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.

3.6.3.15 void vomsdata::ResetOrder (void)

Resets the ordering.

3.6.3.16 void vomsdata::ResetTargets (void)

Resets the target list.

3.6.3.17 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.

3.6.3.18 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)

3.6.3.19 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)

3.6.3.20 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)

3.6.3.21 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)

3.6.3.22 bool vomsdata::RetrieveFromProxy (recurse_type *how*)

Gets VOMS information from an existing globus proxy

Parameters:
how Recursion type

Returns:
failure (F) or success (T)

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

Gets the error message returned by the server

3.6.3.24 void vomsdata::SetLifetime (int *lifetime*)

Set requested lifetime for the Contact() call.

Parameters:
lifetime Requested lifetime, in seconds

3.6.3.25 void vomsdata::SetRetryCount (int *retryCount*)

3.6.3.26 void vomsdata::SetVerificationTime (time_t)

3.6.3.27 void vomsdata::SetVerificationType (verify_type *how*)

Sets the type of verification done on the data.

Parameters:
how The type of verification.

3.6.4 Field Documentation

3.6.4.1 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.

3.6.4.2 verror_type vomsdata::error

Error code

Definition at line 188 of file voms_api.h.

3.6.4.3 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.

3.6.4.4 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 data

  User's characteristics: can be repeated.

- struct attribute
- struct attributelist
- struct contactdata
- struct voms
- struct vomsdata
- class **vomsdata::Initializer**

Typedefs

- typedef void * gss_cred_id_t
- typedef void * gss_ctx_id_t

Enumerations

- enum data_type { TYPE_NODATA, TYPE_STD, TYPE_CUSTOM }

  The type of data returned.

- enum recurse_type { RECURSE_CHAIN, RECURSE_NONE, RECURSE_DEEP }
- enum verify_type {

  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 }
- enum verror_type {

  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 }

  Error codes.

Functions

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

4.1.1 Typedef Documentation

4.1.1.1 typedef void* gss_cred_id_t

Definition at line 25 of file voms_api.h.

4.1.1.2 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.

Enumerator:

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

Enumerator:

RECURSE_CHAIN

RECURSE_NONE

RECURSE_DEEP

Definition at line 120 of file voms_api.h.

4.1.2.3 enum verify_type

Enumerator:

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.

Enumerator:

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

~voms
    voms, 12
~vomsdata
    vomsdata, 16

AddTarget
    vomsdata, 16
attribute, 5
    name, 5
    qualifier, 5
    value, 5
attributelist, 7
    attributes, 7
    grantor, 7
attributes
    attributelist, 7

cap
    data, 10
Contact
    vomsdata, 16
contact
    contactdata, 8
contactdata, 8
    contact, 8
    host, 8
    nick, 8
    port, 8
    version, 8
    vo, 8
ContactRaw
    vomsdata, 16
custom
    voms, 12

data, 10
    cap, 10
    group, 10
    role, 10
    vomsdata, 21
data_type
    voms_api.h, 24
date1
    voms, 12
date2
    voms, 12
DefaultData
    vomsdata, 17

error
    vomsdata, 21
ErrorMessage
    vomsdata, 17
Export
    vomsdata, 17
extra_data
    vomsdata, 21

FindByAlias
    vomsdata, 17
FindByVO
    vomsdata, 17
fqan
    voms, 12

GetAC
    voms, 12
GetAttributes
    voms, 12
getMajorVersionNumber
    voms_api.h, 26
getMinorVersionNumber
    voms_api.h, 26
getPatchVersionNumber
    voms_api.h, 26
GetTargets
    voms, 12
grantor
    attributelist, 7
group
    data, 10
gss_cred_id_t
    voms_api.h, 24
gss_ctx_id_t
    voms_api.h, 24

host
    contactdata, 8

Import
    vomsdata, 18

ListTargets
vomsdata, 18
LoadCredentials
vomsdata, 18
LoadSystemContacts
vomsdata, 18
LoadUserContacts
vomsdata, 18

name
attribute, 5
nick
contactdata, 8

operator=
voms, 12
Order
vomsdata, 18

port
contactdata, 8

qualifier
attribute, 5

RECURSE_CHAIN
voms_api.h, 25
RECURSE_DEEP
voms_api.h, 25
RECURSE_NONE
voms_api.h, 25
recurse_type
voms_api.h, 24
ResetOrder
vomsdata, 19
ResetTargets
vomsdata, 19
Retrieve
vomsdata, 19
RetrieveFromCred
vomsdata, 20
RetrieveFromCtx
vomsdata, 20
RetrieveFromProxy
vomsdata, 20
role
data, 10

serial
voms, 13
server
voms, 13
serverca
voms, 13
ServerErrors
vomsdata, 20
SetLifetime
vomsdata, 20
SetRetryCount
vomsdata, 20
SetVerificationTime
vomsdata, 21
SetVerificationType
vomsdata, 21
siglen
voms, 13
signature
voms, 13
std
voms, 13

TranslateVOMS
voms, 12
type
voms, 13
TYPE_CUSTOM
voms_api.h, 24
TYPE_NODATA
voms_api.h, 24
TYPE_STD
voms_api.h, 24

uri
voms, 13
user
voms, 13
userca
voms, 14

value
attribute, 5
VERIFY_CERTLIST
voms_api.h, 25
VERIFY_DATE
voms_api.h, 25
VERIFY_FULL
voms_api.h, 25
VERIFY_ID
voms_api.h, 25
VERIFY_KEY
voms_api.h, 25
VERIFY_NONE
voms_api.h, 25
VERIFY_ORDER
voms_api.h, 25
VERIFY_SIGN
voms_api.h, 25
VERIFY_TARGET
voms_api.h, 25

verify_type
voms_api.h, 25
VERR_COMM
voms_api.h, 25
VERR_DIR
voms_api.h, 25
VERR_EXTRAINFO
voms_api.h, 25
VERR_FILE
voms_api.h, 26
VERR_FORMAT
voms_api.h, 25
VERR_IDCHECK
voms_api.h, 25
VERR_MEM
voms_api.h, 26
VERR_NODATA
voms_api.h, 25
VERR_NOEXT
voms_api.h, 25
VERR_NOIDENT
voms_api.h, 25
VERR_NOINIT
voms_api.h, 25
VERR_NONE
voms_api.h, 25
VERR_NOSOCKET
voms_api.h, 25
VERR_NOTAVAIL
voms_api.h, 26
VERR_ORDER
voms_api.h, 26
VERR_PARAM
voms_api.h, 25
VERR_PARSE
voms_api.h, 25
VERR_SERVER
voms_api.h, 25
VERR_SERVERCODE
voms_api.h, 26
VERR_SIGN
voms_api.h, 25
VERR_TIME
voms_api.h, 25
VERR_TYPE
voms_api.h, 26
VERR_VERIFY
voms_api.h, 26
verror_type
voms_api.h, 25
version
contactdata, 8
voms, 14
vo
contactdata, 8
voms, 11
∼voms, 12
custom, 12
date1, 12
date2, 12
fqan, 12
GetAC, 12
GetAttributes, 12
GetTargets, 12
operator=, 12
serial, 13
server, 13
serverca, 13
siglen, 13
signature, 13
std, 13
TranslateVOMS, 12
type, 13
uri, 13
user, 13
userca, 14
version, 14
voms, 12
vomsdata, 12
voname, 14
voms_api.h
RECURSE_CHAIN, 25
RECURSE_DEEP, 25
RECURSE_NONE, 25
TYPE_CUSTOM, 24
TYPE_NODATA, 24
TYPE_STD, 24
VERIFY_CERTLIST, 25
VERIFY_DATE, 25
VERIFY_FULL, 25
VERIFY_ID, 25
VERIFY_KEY, 25
VERIFY_NONE, 25
VERIFY_ORDER, 25
VERIFY_SIGN, 25
VERIFY_TARGET, 25
VERR_COMM, 25
VERR_DIR, 25
VERR_EXTRAINFO, 25
VERR_FILE, 26
VERR_FORMAT, 25
VERR_IDCHECK, 25
VERR_MEM, 26
VERR_NODATA, 25
VERR_NOEXT, 25
VERR_NOIDENT, 25
VERR_NOINIT, 25
VERR_NONE, 25

VERR_NOSOCKET, 25
VERR_NOTAVAIL, 26
VERR_ORDER, 26
VERR_PARAM, 25
VERR_PARSE, 25
VERR_SERVER, 25
VERR_SERVERCODE, 26
VERR_SIGN, 25
VERR_TIME, 25
VERR_TYPE, 26
VERR_VERIFY, 26
voms_api.h, 23
data_type, 24
getMajorVersionNumber, 26
getMinorVersionNumber, 26
getPatchVersionNumber, 26
gss_cred_id_t, 24
gss_ctx_id_t, 24
recurse_type, 24
verify_type, 25
verror_type, 25
vomsdata, 15
∼vomsdata, 16
AddTarget, 16
Contact, 16
ContactRaw, 16
data, 21
DefaultData, 17
error, 21
ErrorMessage, 17
Export, 17
extra_data, 21
FindByAlias, 17
FindByVO, 17
Import, 18
ListTargets, 18
LoadCredentials, 18
LoadSystemContacts, 18
LoadUserContacts, 18
Order, 18
ResetOrder, 19
ResetTargets, 19
Retrieve, 19
RetrieveFromCred, 20
RetrieveFromCtx, 20
RetrieveFromProxy, 20
ServerErrors, 20
SetLifetime, 20
SetRetryCount, 20
SetVerificationTime, 21
SetVerificationType, 21
voms, 12
vomsdata, 16
workvo, 21
voname
voms, 14
workvo
vomsdata, 21