

# VOMS CC API Reference Manual

## 1.5.0

Generated by Doxygen 1.2.18

Tue Jun 30 05:34:26 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 . . . . .	6
3.3	contactdata Struct Reference. . . . .	7
3.4	data Struct Reference . . . . .	9
3.5	voms Struct Reference . . . . .	10
3.6	vomsdata Struct Reference. . . . .	14
4	VOMS CC API File Documentation	21
4.1	vomsapi.h File Reference. . . . .	21

---



# Chapter 1

## VOMS CC API Data Structure Index

### 1.1 VOMS CC API Data Structures

Here are the data structures with brief descriptions:

<a href="#">attribute</a>	5
<a href="#">attributelist</a>	6
<a href="#">contactdata</a>	7
<a href="#">data</a> (User's characteristics: can be repeated)	9
<a href="#">voms</a>	10
<a href="#">vomsdata</a>	14



# Chapter 2

## VOMS CC API File Index

### 2.1 VOMS CC API File List

Here is a list of all files with brief descriptions:

<a href="#">voms.api.h</a>	. . . . .	<a href="#">.21</a>
----------------------------	-----------	---------------------





---

## 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::stringgroup  
std::stringrole  
std::stringcap
```

#### 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 vomsdata_rvd, 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 vomsdata\* 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 vomsdir="", std::string certdir="")
bool LoadSystemContact(std::string dir="")
bool LoadUserContact(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 ResetTarget(void)
std::string ServerErrors(void)
bool Retrieve(X509 cert, STACK_OF(X509) chain, recursetype how=RECURSECHAIN)
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(gssctx_id_t context, recursetype how)
bool RetrieveFromCredential(gss_cred_id_t credential, recursetype how)
bool Retrieve(X509_EXTENSION ext)
bool RetrieveFromProxy(recursetype how)
bool Retrieve(FILE *le, recursetype 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 vomsdir = "", std::string certdir = "")`

Parameters:

`vomsdir` The directory which contains the certificate of the VOMS server

`certdir` The directory which contains the certificate of the CA

If `vomsdir` is empty, the value of the environment variable `X509VOMS_DIR` is taken.

If `certdir` is empty, the value of the environment variable `X509CERT_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 `vomsproxy_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 received from 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 containing 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::stringbuffer)`

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 le, recurse.type how)

Gets VOMS information from a proxy saved as a le.

Parameters:

the le

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 (X509EXTENSION 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::RetrieveFromCredential(gsscred_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::RetrieveFromContext(gssctx_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::extradata

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 vomsapi.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

```
structattribute
structattributelist
structcontactdata
structdata
    User's characteristics: can be repeated.

classInitializer
structvoms
structvomsdata
```

#### Typedefs

```
typedef void gsscredid_t
typedef void gssctx_id_t
typedef bool( checksig)(X509 , void , veror.type&)
```

#### Enumerations

```
enumdatatypef TYPE.NODATA, TYPE.STD, TYPE.CUSTOMg
```

---

The type of data returned.

```
enum recurse_type { RECURSECHAIN, RECURSENONE, RECURSEDEEP }
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_
EXTRINFO, 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 bool( check.sig)(X509 , void , [verror\\_type](#) &)

Definition at line 167 of file voms.api.h.

4.1.1.2 typedef void gsscred\_id\_t

Definition at line 25 of file voms.api.h.

4.1.1.3 typedef void gssctx\_id\_t

Definition at line 26 of file voms.api.h.

### 4.1.2 Enumeration Type Documentation

4.1.2.1 enum datatype

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 recursdtype

Enumeration values:

- RECURSE\_CHAIN
- RECURSE\_NONE
- RECURSE\_DEEP

Definition at line 120 of file vomsapi.h.

#### 4.1.2.3 enum verifytype

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 vomsapi.h.

#### 4.1.2.4 enum verrortype

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 Unidenti able VOMS server  
VERR\_MEM Memory problems  
VERR\_VERIFY Generic veri cation 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 le

De nition at line 140 of le 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, [11](#)  
vomsdata  
  vomsdata, [15](#)

AddTarget  
  vomsdata, [15](#)

attribute, [5](#)  
  name, [5](#)  
  qualifier, [5](#)  
  value, [5](#)

attributelist, [6](#)  
  attributes, [6](#)  
  grantor, [6](#)

attributes  
  attributelist, [6](#)

cap  
  data, [9](#)

checksig  
  voms.api.h, [22](#)

Contact  
  vomsdata, [15](#)

contact  
  contactdata, [7](#)

contactdata, [7](#)  
  contact, [7](#)  
  host, [7](#)  
  nick, [7](#)  
  port, [7](#)  
  version, [7](#)  
  vo, [7](#)

ContactRaw  
  vomsdata, [15](#)

custom  
  voms, [11](#)

data, [9](#)  
  cap, [9](#)  
  group, [9](#)  
  role, [9](#)  
  vomsdata, [20](#)

datatype  
  voms.api.h, [22](#)

date1  
  voms, [11](#)

date2  
  voms, [11](#)

DefaultData  
  vomsdata, [16](#)

error  
  vomsdata, [20](#)

ErrorMessage  
  vomsdata, [16](#)

Export  
  vomsdata, [16](#)

extradata  
  vomsdata, [20](#)

FindByAlias  
  vomsdata, [16](#)

FindByVO  
  vomsdata, [16](#)

fqan  
  voms, [11](#)

GetAC  
  voms, [11](#)

GetAttributes  
  voms, [11](#)

getMajorVersionNumber  
  voms.api.h, [24](#)

getMinorVersionNumber  
  voms.api.h, [24](#)

getPatchVersionNumber  
  voms.api.h, [24](#)

GetTargets  
  voms, [11](#)

grantor  
  attributelist, [6](#)

group  
  data, [9](#)

gsscred.id.t  
  voms.api.h, [22](#)

gssctx.id.t  
  voms.api.h, [22](#)

host  
  contactdata, [7](#)

---

---

- Import
  - vomsdata,16
- ListTargets
  - vomsdata,17
- LoadCredentials
  - vomsdata,17
- LoadSystemContacts
  - vomsdata,17
- LoadUserContacts
  - vomsdata,17
- name
  - attribute,5
- nick
  - contactdata,7
- operator=
  - voms,11
- Order
  - vomsdata,17
- port
  - contactdata,7
- quali er
  - attribute,5
- RECURSECHAIN
  - voms.api.h,23
- RECURSEDEEP
  - voms.api.h,23
- RECURSENONE
  - voms.api.h,23
- recursetype
  - voms.api.h,22
- ResetOrder
  - vomsdata,17
- ResetTargets
  - vomsdata,18
- Retrieve
  - vomsdata,18
- RetrieveFromCred
  - vomsdata,18
- RetrieveFromCtx
  - vomsdata,19
- RetrieveFromProxy
  - vomsdata,19
- role
  - data,9
- serial
  - voms,12
- server
  - voms,12
- serverca
  - voms,12
- ServerErrors
  - vomsdata,19
- SetLifetime
  - vomsdata,19
- SetRetryCount
  - vomsdata,19
- SetVeri cationTime
  - vomsdata,20
- SetVeri cationType
  - vomsdata,20
- siglen
  - voms,12
- signature
  - voms,12
- std
  - voms,12
- TranslateVOMS
  - voms,11
- type
  - voms,12
- TYPE\_CUSTOM
  - voms.api.h,22
- TYPE\_NODATA
  - voms.api.h,22
- TYPE\_STD
  - voms.api.h,22
- uri
  - voms,12
- user
  - voms,12
- userca
  - voms,13
- value
  - attribute,5
- VERIFY\_CERTLIST
  - voms.api.h,23
- VERIFY\_DATE
  - voms.api.h,23
- VERIFY\_FULL
  - voms.api.h,23
- VERIFY\_ID
  - voms.api.h,23
- VERIFY\_KEY
  - voms.api.h,23
- VERIFY\_NONE
  - voms.api.h,23
- VERIFY\_ORDER
  - voms.api.h,23
- VERIFY\_SIGN

---

- voms.api.h, [23](#)
- VERIFY\_TARGET
  - voms.api.h, [23](#)
- verify\_type
  - voms.api.h, [23](#)
- VERR\_COMM
  - voms.api.h, [23](#)
- VERR\_DIR
  - voms.api.h, [23](#)
- VERR\_EXTRINFO
  - voms.api.h, [23](#)
- VERR\_FILE
  - voms.api.h, [24](#)
- VERR\_FORMAT
  - voms.api.h, [23](#)
- VERR\_IDCHECK
  - voms.api.h, [23](#)
- VERR\_MEM
  - voms.api.h, [24](#)
- VERR\_NODATA
  - voms.api.h, [23](#)
- VERR\_NOEXT
  - voms.api.h, [23](#)
- VERR\_NOIDENT
  - voms.api.h, [23](#)
- VERR\_NOINIT
  - voms.api.h, [23](#)
- VERR\_NONE
  - voms.api.h, [23](#)
- VERR\_NOSOCKET
  - voms.api.h, [23](#)
- VERR\_NOTAVAIL
  - voms.api.h, [24](#)
- VERR\_ORDER
  - voms.api.h, [24](#)
- VERR\_PARAM
  - voms.api.h, [23](#)
- VERR\_PARSE
  - voms.api.h, [23](#)
- VERR\_SERVER
  - voms.api.h, [23](#)
- VERR\_SERVERCODE
  - voms.api.h, [24](#)
- VERR\_SIGN
  - voms.api.h, [23](#)
- VERR\_TIME
  - voms.api.h, [23](#)
- VERR\_TYPE
  - voms.api.h, [24](#)
- VERR\_VERIFY
  - voms.api.h, [24](#)
- verror\_type
  - voms.api.h, [23](#)
- version
  - contactdata, [7](#)
  - voms, [13](#)
- vo
  - contactdata, [7](#)
- voms, [10](#)
  - voms, [11](#)
  - custom, [11](#)
  - date1, [11](#)
  - date2, [11](#)
  - fqan, [11](#)
  - GetAC, [11](#)
  - GetAttributes, [11](#)
  - GetTargets, [11](#)
  - operator=, [11](#)
  - serial, [12](#)
  - server, [12](#)
  - serverca, [12](#)
  - siglen, [12](#)
  - signature, [12](#)
  - std, [12](#)
  - TranslateVOMS, [11](#)
  - type, [12](#)
  - uri, [12](#)
  - user, [12](#)
  - userca, [13](#)
  - version, [13](#)
  - voms, [11](#)
  - vomsdata, [11](#)
  - voname, [13](#)
- voms.api.h
  - RECURSECHAIN, [23](#)
  - RECURSEDEEP, [23](#)
  - RECURSENONE, [23](#)
  - TYPE\_CUSTOM, [22](#)
  - TYPE\_NODATA, [22](#)
  - TYPE\_STD, [22](#)
  - VERIFY\_CERTLIST, [23](#)
  - VERIFY\_DATE, [23](#)
  - VERIFY\_FULL, [23](#)
  - VERIFY\_ID, [23](#)
  - VERIFY\_KEY, [23](#)
  - VERIFY\_NONE, [23](#)
  - VERIFY\_ORDER, [23](#)
  - VERIFY\_SIGN, [23](#)
  - VERIFY\_TARGET, [23](#)
  - VERR\_COMM, [23](#)
  - VERR\_DIR, [23](#)
  - VERR\_EXTRINFO, [23](#)
  - VERR\_FILE, [24](#)
  - VERR\_FORMAT, [23](#)
  - VERR\_IDCHECK, [23](#)
  - VERR\_MEM, [24](#)
  - VERR\_NODATA, [23](#)
  - VERR\_NOEXT, [23](#)

- VERR\_NOIDENT, [23](#)
- VERR\_NOINIT, [23](#)
- VERR\_NONE, [23](#)
- VERR\_NOCKET, [23](#)
- VERR\_NOTAVAIL, [24](#)
- VERR\_ORDER, [24](#)
- VERR\_PARAM, [23](#)
- VERR\_PARSE, [23](#)
- VERR\_SERVER, [23](#)
- VERR\_SERVERCODE, [24](#)
- VERR\_SIGN, [23](#)
- VERR\_TIME, [23](#)
- VERR\_TYPE, [24](#)
- VERR\_VERIFY, [24](#)
- voms.api.h, [21](#)
  - checksig, [22](#)
  - datatype, [22](#)
  - getMajorVersionNumber, [24](#)
  - getMinorVersionNumber, [24](#)
  - getPatchVersionNumber, [24](#)
  - gsscred\_id\_t, [22](#)
  - gssctx\_id\_t, [22](#)
  - recursetype, [22](#)
  - verify\_type, [23](#)
  - verror\_type, [23](#)
- vomsdata, [14](#)
  - vomsdata, [15](#)
  - AddTarget, [15](#)
  - Contact, [15](#)
  - ContactRaw, [15](#)
  - data, [20](#)
  - DefaultData, [16](#)
  - error, [20](#)
  - ErrorMessage, [16](#)
  - Export, [16](#)
  - extra\_data, [20](#)
  - FindByAlias, [16](#)
  - FindByVO, [16](#)
  - Import, [16](#)
  - ListTargets, [17](#)
  - LoadCredentials, [17](#)
  - LoadSystemContacts, [17](#)
  - LoadUserContacts, [17](#)
  - Order, [17](#)
  - ResetOrder, [17](#)
  - ResetTargets, [18](#)
  - Retrieve, [18](#)
  - RetrieveFromCred, [18](#)
  - RetrieveFromCtx, [19](#)
  - RetrieveFromProxy, [19](#)
  - ServerErrors, [19](#)
  - SetLifetime, [19](#)
  - SetRetryCount, [19](#)
  - SetVerificationTime, [20](#)
  - SetVerificationType, [20](#)
  - voms, [11](#)
  - vomsdata, [15](#)
  - workvo, [20](#)
- voname
  - voms, [13](#)
- workvo
  - vomsdata, [20](#)