

# VOMS CC API

## 1.5.0

Generated by Doxygen 1.5.5

Tue Nov 2 22:54:56 2010



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
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
<b>4</b>	<b>File Documentation</b>	<b>25</b>
4.1	voms_api.h File Reference . . . . .	25



# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">attribute</a>	5
<a href="#">attributelist</a>	7
<a href="#">contactdata</a>	8
<a href="#">data</a> (User's characteristics: can be repeated )	10
<a href="#">voms</a>	11
<a href="#">vomsdata</a>	15



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">voms_api.h</a>	25
----------------------------	----





## Chapter 3

# 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 60 of file voms\_api.h.

#### 3.1.2 Field Documentation

##### 3.1.2.1 std::string attribute::name

attribute's group

Definition at line 61 of file voms\_api.h.

##### 3.1.2.2 std::string attribute::qualifier

attribute's qualifier

Definition at line 62 of file voms\_api.h.

##### 3.1.2.3 std::string attribute::value

attribute's value

Definition at line 63 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 66 of file voms\_api.h.

### 3.2.2 Field Documentation

#### 3.2.2.1 std::string attributelist::grantor

Who granted these attributes.

Definition at line 67 of file voms\_api.h.

#### 3.2.2.2 std::vector<attribute> attributelist::attributes

The attributes themselves.

Definition at line 68 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 80 of file voms\_api.h.

### 3.3.2 Field Documentation

#### 3.3.2.1 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 84 of file voms\_api.h.

#### 3.3.2.2 std::string contactdata::host

The hostname of the server

Definition at line 85 of file voms\_api.h.

#### 3.3.2.3 std::string contactdata::contact

The subject of the server's certificate

Definition at line 86 of file voms\_api.h.

#### 3.3.2.4 std::string contactdata::vo

The VO served by this server

Definition at line 87 of file voms\_api.h.

#### 3.3.2.5 int contactdata::port

The port on which the server is listening

Definition at line 88 of file voms\_api.h.

### 3.3.2.6 int contactdata::version

The version of globus under which the server is running

Definition at line 90 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 52 of file `voms_api.h`.

### 3.4.2 Field Documentation

#### 3.4.2.1 `std::string data::group`

user's group

Definition at line 53 of file `voms_api.h`.

#### 3.4.2.2 `std::string data::role`

user's role

Definition at line 54 of file `voms_api.h`.

#### 3.4.2.3 `std::string data::cap`

user's capability

Definition at line 55 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 vomsdata \*vd, std::vector< [voms](#) > &v, int \*error)

#### 3.5.1 Detailed Description

Definition at line 93 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** `voms& voms::operator= (const voms &)`

**3.5.3.2** `AC* voms::GetAC ()`

**3.5.3.3** `std::vector<attributelist>& voms::GetAttributes ()`

Generic attributes

**3.5.3.4** `std::vector<std::string> voms::GetTargets ()`

## 3.5.4 Friends And Related Function Documentation

**3.5.4.1** `friend class vomsdata` [*friend*]

Definition at line 94 of file voms\_api.h.

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

## 3.5.5 Field Documentation

**3.5.5.1** `int voms::version`

0 means [data](#) didn't originate from an AC

Definition at line 95 of file voms\_api.h.

**3.5.5.2** `int voms::siglen`

The length of the VOMS server signature

Definition at line 96 of file voms\_api.h.

**3.5.5.3** `std::string voms::signature`

The VOMS server signature

Definition at line 97 of file voms\_api.h.



**3.5.5.4 std::string voms::user**

The user's DN, as from his certificate

Definition at line 98 of file voms\_api.h.

**3.5.5.5 std::string voms::userca**

The CA which signed the user's certificate

Definition at line 99 of file voms\_api.h.

**3.5.5.6 std::string voms::server**

The VOMS server DN, as from its certificate

Definition at line 100 of file voms\_api.h.

**3.5.5.7 std::string voms::serverca**

The CA which signed the VOMS certificate

Definition at line 101 of file voms\_api.h.

**3.5.5.8 std::string voms::voname**

The name of the VO to which the VOMS belongs

Definition at line 102 of file voms\_api.h.

**3.5.5.9 std::string voms::uri**

The URI of the VOMS server

Definition at line 103 of file voms\_api.h.

**3.5.5.10 std::string voms::date1**

Beginning of validity of the user info

Definition at line 104 of file voms\_api.h.

**3.5.5.11 std::string voms::date2**

End of validity of the user info

Definition at line 105 of file voms\_api.h.

**3.5.5.12 data\_type voms::type**

The type of [data](#) returned

Definition at line 106 of file voms\_api.h.

**3.5.5.13 `std::vector<data> voms::std`**

User's characteristics

Definition at line 107 of file voms\_api.h.

**3.5.5.14 `std::string voms::custom`**

The [data](#) returned by an S command

Definition at line 108 of file voms\_api.h.

**3.5.5.15 `std::vector<std::string> voms::fqan`**

Keeps the [data](#) in the compact format

Definition at line 110 of file voms\_api.h.

**3.5.5.16 `std::string voms::serial`**

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

Definition at line 111 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 [Contact](#) (std::string hostname, int port, std::string servsubject, std::string command, int timeout)
- bool [ContactRaw](#) (std::string hostname, int port, std::string servsubject, std::string command, std::string &raw, int &version)
- bool [ContactRaw](#) (std::string hostname, int port, std::string servsubject, std::string command, std::string &raw, int &version, int timeout)
- 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](#) (void \*context, [recurse\\_type](#) how)
- bool [RetrieveFromCred](#) (void \*credential, [recurse\\_type](#) how)
- 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 \*, EVP\_PKEY \*, STACK\_OF(X509)\*)

### 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 184 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 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.

#### 3.6.3.2 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.

**3.6.3.3 `std::vector<contactdata> vomldata::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.

**3.6.3.4 `std::vector<contactdata> vomldata::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.

**3.6.3.5 `void vomldata::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.

**3.6.3.6 `void vomldata::ResetOrder (void)`**

Resets the ordering.

**3.6.3.7 `void vomldata::AddTarget (std::string target)`**

Adds a target to the AC.

**Parameters:**

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

### 3.6.3.8 `std::vector<std::string> vomsdata::ListTargets (void)`

Returns the list of targets.

### 3.6.3.9 `void vomsdata::ResetTargets (void)`

Resets the target list.

### 3.6.3.10 `std::string vomsdata::ServerErrors (void)`

Gets the error message returned by the server

### 3.6.3.11 `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)

### 3.6.3.12 `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)

### 3.6.3.13 `bool vomsdata::Contact (std::string hostname, int port, std::string servsubject, std::string command, int timeout)`

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)

### 3.6.3.14 `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)

### 3.6.3.15 `bool vomsdata::ContactRaw (std::string hostname, int port, std::string servsubject, std::string command, std::string & raw, int & version, int timeout)`

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)

**3.6.3.16 void vomsdata::SetVerificationType (verify\_type *how*)**

Sets the type of verification done on the [data](#).

**Parameters:**

*how* The type of verification.

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

Set requested lifetime for the [Contact\(\)](#) call.

**Parameters:**

*lifetime* Requested lifetime, in seconds

**3.6.3.18 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)

**3.6.3.19 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)

**3.6.3.20 bool vomsdata::DefaultData (voms &)**

Get the default [data](#) extension from those present in the pseudo certificate



**3.6.3.21 std::string vomsdata::ErrorMessage (void)**

Gets a textual description of the error.

**Returns:**

A string containing the error message.

**3.6.3.22 bool vomsdata::RetrieveFromCtx (void \* *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)

**3.6.3.23 bool vomsdata::RetrieveFromCred (void \* *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)

**3.6.3.24 bool vomsdata::RetrieveFromCtx (gss\_ctx\_id\_t *context*, recurse\_type *how*) [inline]**

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)

Definition at line 432 of file voms\_api.h.

**3.6.3.25 bool vomldata::RetrieveFromCred (gss\_cred\_id\_t *credential*, recurse\_type *how*)**  
[inline]

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)

Definition at line 439 of file vomldata.h.

**3.6.3.26 bool vomldata::Retrieve (X509\_EXTENSION \* *ext*)**

Gets VOMS information from the given extension

**Parameters:**

*ext* The extension to parse.

**Returns:**

failure (F) or success (T)

**3.6.3.27 bool vomldata::RetrieveFromProxy (recurse\_type *how*)**

Gets VOMS information from an existing globus proxy

**Parameters:**

*how* Recursion type

**Returns:**

failure (F) or success (T)

**3.6.3.28 bool vomldata::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.

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

**3.6.3.30** void vomsdata::SetVerificationTime (time\_t)

**3.6.3.31** bool vomsdata::LoadCredentials (X509 \*, EVP\_PKEY \*, STACK\_OF(X509)\*)

## 3.6.4 Field Documentation

**3.6.4.1** error\_type vomsdata::error

Error code

Definition at line 203 of file voms\_api.h.

**3.6.4.2** 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 358 of file voms\_api.h.

**3.6.4.3** std::string vomsdata::workvo

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

Definition at line 361 of file voms\_api.h.

**3.6.4.4** 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 362 of file voms\_api.h.

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

- [voms\\_api.h](#)



# Chapter 4

## File Documentation

### 4.1 voms\_api.h File Reference

```
#include <fstream>
#include <string>
#include <vector>
#include "gssapi.h"
#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 bool(\* [check\\_sig](#))(X509 \*, void \*, [verror\\_type](#) &)

#### 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_NOCKET`, `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 182 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 74 of file voms\_api.h.

#### 4.1.2.2 enum recurse\_type

Enumerator:

*RECURSE\_CHAIN*  
*RECURSE\_NONE*  
*RECURSE\_DEEP*

Definition at line 135 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 141 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 155 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, [17](#)
- attribute, [5](#)
  - name, [5](#)
  - qualifier, [5](#)
  - value, [5](#)
- attributelist, [7](#)
  - attributes, [7](#)
  - grantor, [7](#)
- attributes
  - attributelist, [7](#)
- cap
  - data, [10](#)
- check\_sig
  - voms\_api.h, [26](#)
- Contact
  - vomsdata, [18](#)
- contact
  - contactdata, [8](#)
- contactdata, [8](#)
  - contact, [8](#)
  - host, [8](#)
  - nick, [8](#)
  - port, [8](#)
  - version, [8](#)
  - vo, [8](#)
- ContactRaw
  - vomsdata, [19](#)
- custom
  - voms, [14](#)
- data, [10](#)
  - cap, [10](#)
  - group, [10](#)
  - role, [10](#)
  - vomsdata, [23](#)
- data\_type
  - voms\_api.h, [26](#)
- date1
  - voms, [13](#)
- date2
  - voms, [13](#)
- DefaultData
  - vomsdata, [20](#)
- error
  - vomsdata, [23](#)
- ErrorMessage
  - vomsdata, [20](#)
- Export
  - vomsdata, [20](#)
- extra\_data
  - vomsdata, [23](#)
- FindByAlias
  - vomsdata, [17](#)
- FindByVO
  - vomsdata, [17](#)
- fqan
  - voms, [14](#)
- GetAC
  - voms, [12](#)
- GetAttributes
  - voms, [12](#)
- getMajorVersionNumber
  - voms\_api.h, [28](#)
- getMinorVersionNumber
  - voms\_api.h, [28](#)
- getPatchVersionNumber
  - voms\_api.h, [28](#)
- GetTargets
  - voms, [12](#)
- grantor
  - attributelist, [7](#)
- group
  - data, [10](#)
- host
  - contactdata, [8](#)
- Import
  - vomsdata, [20](#)
- ListTargets

- vomsdata, 17
- LoadCredentials
  - vomsdata, 23
- LoadSystemContacts
  - vomsdata, 16
- LoadUserContacts
  - vomsdata, 16
- name
  - attribute, 5
- nick
  - contactdata, 8
- operator=
  - voms, 12
- Order
  - vomsdata, 17
- port
  - contactdata, 8
- qualifier
  - attribute, 5
- RECURSE\_CHAIN
  - voms\_api.h, 27
- RECURSE\_DEEP
  - voms\_api.h, 27
- RECURSE\_NONE
  - voms\_api.h, 27
- recurse\_type
  - voms\_api.h, 26
- ResetOrder
  - vomsdata, 17
- ResetTargets
  - vomsdata, 18
- Retrieve
  - vomsdata, 18, 22
- RetrieveFromCred
  - vomsdata, 21
- RetrieveFromCtx
  - vomsdata, 21
- RetrieveFromProxy
  - vomsdata, 22
- role
  - data, 10
- serial
  - voms, 14
- server
  - voms, 13
- serverca
  - voms, 13
- ServerErrors
  - vomsdata, 18
- SetLifetime
  - vomsdata, 20
- SetRetryCount
  - vomsdata, 22
- SetVerificationTime
  - vomsdata, 23
- SetVerificationType
  - vomsdata, 20
- siglen
  - voms, 12
- signature
  - voms, 12
- std
  - voms, 13
- TranslateVOMS
  - voms, 12
- type
  - voms, 13
- TYPE\_CUSTOM
  - voms\_api.h, 26
- TYPE\_NODATA
  - voms\_api.h, 26
- TYPE\_STD
  - voms\_api.h, 26
- uri
  - voms, 13
- user
  - voms, 12
- userca
  - voms, 13
- value
  - attribute, 5
- VERIFY\_CERTLIST
  - voms\_api.h, 27
- VERIFY\_DATE
  - voms\_api.h, 27
- VERIFY\_FULL
  - voms\_api.h, 27
- VERIFY\_ID
  - voms\_api.h, 27
- VERIFY\_KEY
  - voms\_api.h, 27
- VERIFY\_NONE
  - voms\_api.h, 27
- VERIFY\_ORDER
  - voms\_api.h, 27
- VERIFY\_SIGN
  - voms\_api.h, 27
- VERIFY\_TARGET
  - voms\_api.h, 27
- verify\_type

- voms\_api.h, 27
- VERR\_COMM
  - voms\_api.h, 27
- VERR\_DIR
  - voms\_api.h, 27
- VERR\_EXTRINFO
  - voms\_api.h, 27
- VERR\_FILE
  - voms\_api.h, 28
- VERR\_FORMAT
  - voms\_api.h, 27
- VERR\_IDCHECK
  - voms\_api.h, 27
- VERR\_MEM
  - voms\_api.h, 28
- VERR\_NODATA
  - voms\_api.h, 27
- VERR\_NOEXT
  - voms\_api.h, 27
- VERR\_NOIDENT
  - voms\_api.h, 27
- VERR\_NOINIT
  - voms\_api.h, 27
- VERR\_NONE
  - voms\_api.h, 27
- VERR\_NOSOCKET
  - voms\_api.h, 27
- VERR\_NOTAVAIL
  - voms\_api.h, 28
- VERR\_ORDER
  - voms\_api.h, 28
- VERR\_PARAM
  - voms\_api.h, 27
- VERR\_PARSE
  - voms\_api.h, 27
- VERR\_SERVER
  - voms\_api.h, 28
- VERR\_SERVERCODE
  - voms\_api.h, 28
- VERR\_SIGN
  - voms\_api.h, 27
- VERR\_TIME
  - voms\_api.h, 27
- VERR\_TYPE
  - voms\_api.h, 28
- VERR\_VERIFY
  - voms\_api.h, 28
- verror\_type
  - voms\_api.h, 27
- version
  - contactdata, 8
  - voms, 12
- vo
  - contactdata, 8
- voms, 11
  - ~voms, 12
  - custom, 14
  - date1, 13
  - date2, 13
  - fqan, 14
  - GetAC, 12
  - GetAttributes, 12
  - GetTargets, 12
  - operator=, 12
  - serial, 14
  - server, 13
  - serverca, 13
  - siglen, 12
  - signature, 12
  - std, 13
  - TranslateVOMS, 12
  - type, 13
  - uri, 13
  - user, 12
  - userca, 13
  - version, 12
  - voms, 12
  - vomsdata, 12
  - voname, 13
- voms\_api.h
  - RECURSE\_CHAIN, 27
  - RECURSE\_DEEP, 27
  - RECURSE\_NONE, 27
  - TYPE\_CUSTOM, 26
  - TYPE\_NODATA, 26
  - TYPE\_STD, 26
  - VERIFY\_CERTLIST, 27
  - VERIFY\_DATE, 27
  - VERIFY\_FULL, 27
  - VERIFY\_ID, 27
  - VERIFY\_KEY, 27
  - VERIFY\_NONE, 27
  - VERIFY\_ORDER, 27
  - VERIFY\_SIGN, 27
  - VERIFY\_TARGET, 27
  - VERR\_COMM, 27
  - VERR\_DIR, 27
  - VERR\_EXTRINFO, 27
  - VERR\_FILE, 28
  - VERR\_FORMAT, 27
  - VERR\_IDCHECK, 27
  - VERR\_MEM, 28
  - VERR\_NODATA, 27
  - VERR\_NOEXT, 27
  - VERR\_NOIDENT, 27
  - VERR\_NOINIT, 27
  - VERR\_NONE, 27
  - VERR\_NOSOCKET, 27

- VERR\_NOTAVAIL, [28](#)
- VERR\_ORDER, [28](#)
- VERR\_PARAM, [27](#)
- VERR\_PARSE, [27](#)
- VERR\_SERVER, [28](#)
- VERR\_SERVERCODE, [28](#)
- VERR\_SIGN, [27](#)
- VERR\_TIME, [27](#)
- VERR\_TYPE, [28](#)
- VERR\_VERIFY, [28](#)
- voms\_api.h, [25](#)
  - check\_sig, [26](#)
  - data\_type, [26](#)
  - getMajorVersionNumber, [28](#)
  - getMinorVersionNumber, [28](#)
  - getPatchVersionNumber, [28](#)
  - recurse\_type, [26](#)
  - verify\_type, [27](#)
  - verror\_type, [27](#)
- vomsdata, [15](#)
  - ~vomsdata, [16](#)
  - AddTarget, [17](#)
  - Contact, [18](#)
  - ContactRaw, [19](#)
  - data, [23](#)
  - DefaultData, [20](#)
  - error, [23](#)
  - ErrorMessage, [20](#)
  - Export, [20](#)
  - extra\_data, [23](#)
  - FindByAlias, [17](#)
  - FindByVO, [17](#)
  - Import, [20](#)
  - ListTargets, [17](#)
  - LoadCredentials, [23](#)
  - LoadSystemContacts, [16](#)
  - LoadUserContacts, [16](#)
  - Order, [17](#)
  - ResetOrder, [17](#)
  - ResetTargets, [18](#)
  - Retrieve, [18](#), [22](#)
  - RetrieveFromCred, [21](#)
  - RetrieveFromCtx, [21](#)
  - RetrieveFromProxy, [22](#)
  - ServerErrors, [18](#)
  - SetLifetime, [20](#)
  - SetRetryCount, [22](#)
  - SetVerificationTime, [23](#)
  - SetVerificationType, [20](#)
  - voms, [12](#)
  - vomsdata, [16](#)
  - workvo, [23](#)
- voname
  - voms, [13](#)
- workvo
  - vomsdata, [23](#)