

# VOMS CC API

## 1.5.0

Generated by Doxygen 1.7.4

Fri May 25 2012 06:56:52



# 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
3.1.1	Detailed Description . . . . .	5
3.1.2	Field Documentation . . . . .	5
3.1.2.1	name . . . . .	5
3.1.2.2	qualifier . . . . .	5
3.1.2.3	value . . . . .	6
3.2	attributelist Struct Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Field Documentation . . . . .	6
3.2.2.1	attributes . . . . .	6
3.2.2.2	grantor . . . . .	6
3.3	contactdata Struct Reference . . . . .	6
3.3.1	Detailed Description . . . . .	7
3.3.2	Field Documentation . . . . .	7
3.3.2.1	contact . . . . .	7
3.3.2.2	host . . . . .	7
3.3.2.3	nick . . . . .	7
3.3.2.4	port . . . . .	7
3.3.2.5	version . . . . .	7

3.3.2.6	vo	8
3.4	data Struct Reference	8
3.4.1	Detailed Description	8
3.4.2	Field Documentation	8
3.4.2.1	cap	8
3.4.2.2	group	8
3.4.2.3	role	9
3.5	voms Struct Reference	9
3.5.1	Detailed Description	10
3.5.2	Constructor & Destructor Documentation	10
3.5.2.1	voms	10
3.5.2.2	voms	10
3.5.2.3	~voms	10
3.5.3	Member Function Documentation	10
3.5.3.1	GetAC	10
3.5.3.2	GetAttributes	10
3.5.3.3	GetTargets	10
3.5.3.4	operator=	10
3.5.4	Friends And Related Function Documentation	10
3.5.4.1	TranslateVOMS	10
3.5.4.2	vomsdata	10
3.5.5	Field Documentation	10
3.5.5.1	custom	10
3.5.5.2	date1	11
3.5.5.3	date2	11
3.5.5.4	fqn	11
3.5.5.5	serial	11
3.5.5.6	server	11
3.5.5.7	serverca	11
3.5.5.8	siglen	11
3.5.5.9	signature	11
3.5.5.10	std	12
3.5.5.11	type	12
3.5.5.12	uri	12

3.5.5.13	user	12
3.5.5.14	userca	12
3.5.5.15	version	12
3.5.5.16	voname	12
3.6	vomsdata Struct Reference	13
3.6.1	Detailed Description	14
3.6.2	Constructor & Destructor Documentation	14
3.6.2.1	vomsdata	14
3.6.2.2	~vomsdata	14
3.6.2.3	vomsdata	14
3.6.3	Member Function Documentation	14
3.6.3.1	AddTarget	14
3.6.3.2	Contact	15
3.6.3.3	Contact	15
3.6.3.4	ContactRaw	15
3.6.3.5	ContactRaw	16
3.6.3.6	ContactRESTRaw	16
3.6.3.7	DefaultData	16
3.6.3.8	ErrorMessage	16
3.6.3.9	Export	16
3.6.3.10	FindByAlias	17
3.6.3.11	FindByVO	17
3.6.3.12	Import	17
3.6.3.13	ListTargets	17
3.6.3.14	LoadCredentials	18
3.6.3.15	LoadSystemContacts	18
3.6.3.16	LoadUserContacts	18
3.6.3.17	Order	18
3.6.3.18	ResetOrder	18
3.6.3.19	ResetTargets	19
3.6.3.20	Retrieve	19
3.6.3.21	Retrieve	19
3.6.3.22	Retrieve	19
3.6.3.23	Retrieve	19

3.6.3.24	RetrieveFromCred	20
3.6.3.25	RetrieveFromCtx	20
3.6.3.26	RetrieveFromProxy	20
3.6.3.27	ServerErrors	21
3.6.3.28	SetLifetime	21
3.6.3.29	SetRetryCount	21
3.6.3.30	SetVerificationTime	21
3.6.3.31	SetVerificationType	21
3.6.4	Field Documentation	21
3.6.4.1	data	21
3.6.4.2	error	21
3.6.4.3	extra_data	21
3.6.4.4	workvo	22
<b>4</b>	<b>File Documentation</b>	<b>23</b>
4.1	voms_api.h File Reference	23
4.1.1	Define Documentation	24
4.1.1.1	NOGLOBUS	24
4.1.2	Typedef Documentation	24
4.1.2.1	check_sig	24
4.1.2.2	gss_cred_id_t	25
4.1.2.3	gss_ctx_id_t	25
4.1.3	Enumeration Type Documentation	25
4.1.3.1	data_type	25
4.1.3.2	recurse_type	25
4.1.3.3	verify_type	25
4.1.3.4	verror_type	26
4.1.4	Function Documentation	26
4.1.4.1	getVOMSMajorVersionNumber	26
4.1.4.2	getVOMSMinorVersionNumber	26
4.1.4.3	getVOMSPatchVersionNumber	26

# 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>	6
<a href="#">contactdata</a>	6
<a href="#">data</a> (User's characteristics: can be repeated. Generic name-value attribute : can be repeated )	8
<a href="#">voms</a>	9
<a href="#">vomsdata</a>	13





# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">voms_api.h</a>	23
----------------------------	----



## 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 63 of file voms\_api.h.

#### 3.1.2 Field Documentation

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

attribute's group

Definition at line 64 of file voms\_api.h.

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

attribute's qualifier

Definition at line 65 of file voms\_api.h.

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

attribute's value

Definition at line 66 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 69 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 71 of file `voms_api.h`.

#### 3.2.2.2 `std::string attributelist::grantor`

Who granted these attributes.

Definition at line 70 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 83 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 89 of file voms\_api.h.

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

The hostname of the server

Definition at line 88 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 87 of file voms\_api.h.

#### 3.3.2.4 int contactdata::port

The port on which the server is listening

Definition at line 91 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 93 of file voms\_api.h.

### 3.3.2.6 std::string contactdata::vo

The VO served by this server

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. Generic name-value attribute : 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. Generic name-value attribute : can be repeated.

Definition at line 55 of file voms\_api.h.

### 3.4.2 Field Documentation

#### 3.4.2.1 std::string data::cap

user's capability

Definition at line 58 of file voms\_api.h.

#### 3.4.2.2 std::string data::group

user's group

Definition at line 56 of file voms\_api.h.

### 3.4.2.3 std::string data::role

user's role

Definition at line 57 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 98 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 99 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 113 of file voms\_api.h.



**3.5.5.2 std::string voms::date1**

Beginning of validity of the user info

Definition at line 109 of file voms\_api.h.

**3.5.5.3 std::string voms::date2**

End of validity of the user info

Definition at line 110 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 115 of file voms\_api.h.

**3.5.5.5 std::string voms::serial**

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

Definition at line 116 of file voms\_api.h.

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

The VOMS server DN, as from its certificate

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

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

The CA which signed the VOMS certificate

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

**3.5.5.8 int voms::siglen**

The length of the VOMS server signature

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

**3.5.5.9 std::string voms::signature**

The VOMS server signature

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

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

User's characteristics

Definition at line 112 of file voms\_api.h.

**3.5.5.11 data\_type voms::type**

The type of data returned

Definition at line 111 of file voms\_api.h.

**3.5.5.12 std::string voms::uri**

The URI of the VOMS server

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

**3.5.5.13 std::string voms::user**

The user's DN, as from his certificate

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

**3.5.5.14 std::string voms::userca**

The CA which signed the user's certificate

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

**3.5.5.15 int voms::version**

0 means data didn't originate from an AC

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

**3.5.5.16 std::string voms::vname**

The name of the VO to which the VOMS belongs

Definition at line 107 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>
```

### Data Structures

- class **Initializer**

### 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](#) (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)
- bool [Retrieve](#) (AC \*ac)
- [~vomsdata](#) ()
- [vomsdata](#) (const [vomsdata](#) &)

- void [SetRetryCount](#) (int retryCount)
- void [SetVerificationTime](#) (time\_t)
- bool [LoadCredentials](#) (X509 \*, EVP\_PKEY \*, STACK\_OF(X509)\*)
- bool [ContactRESTRaw](#) (const std::string &, int, const std::string &, std::string &, int, int)

### Data Fields

- [verror\\_type](#) error
- std::vector< [voms](#) > [data](#)
- std::string [workvo](#)
- std::string [extra\\_data](#)

### 3.6.1 Detailed Description

Definition at line 194 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

<i>voms_dir</i>	The directory which contains the certificate of the VOMS server
<i>cert_dir</i>	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

<i>target</i>	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

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

#### Returns

failure (F) or success (T)

3.6.3.3 `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

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

#### Returns

failure (F) or success (T)

3.6.3.4 `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

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

<i>raw</i>	OUTPUT PARAMETER the answer from the server
<i>version</i>	OUTPUT PARAMETER the version of the answer

**Returns**

failure (F) or success (T)

**3.6.3.5** `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**

<i>hostname</i>	FQDN of the VOMS server
<i>port</i>	the port on which the VOMS server is listening
<i>servsubject</i>	the subject of the server's certificate
<i>command</i>	the command sent to the server
<i>raw</i>	OUTPUT PARAMETER the answer from the server
<i>version</i>	OUTPUT PARAMETER the version of the answer

**Returns**

failure (F) or success (T)

**3.6.3.6** `bool vomsdata::ContactRESTRaw ( const std::string &, int, const std::string &, std::string &, int, int )`

**3.6.3.7** `bool vomsdata::DefaultData ( voms & )`

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

**3.6.3.8** `std::string vomsdata::ErrorMessage ( void )`

Gets a textual description of the error.

**Returns**

A string containing the error message.

**3.6.3.9** `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**

<i>data</i>	The certificate extension
-------------	---------------------------

**Returns**

Failure (F) or Success (T)

**3.6.3.10** `std::vector<contactdata> vomsdata::FindByAlias ( std::string alias )`

Finds servers which share a common alias.

**Parameters**

<i>alias</i>	The alias to look for.
--------------	------------------------

**Returns**

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

**3.6.3.11** `std::vector<contactdata> vomsdata::FindByVO ( std::string vo )`

Finds servers which serve a common VO

**Parameters**

<i>vo</i>	The VO name to look for.
-----------	--------------------------

**Returns**

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

**3.6.3.12** `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**

<i>buffer</i>	contains the data to be converted
---------------	-----------------------------------

**Returns**

Failure (F) or Success (T)

**3.6.3.13** `std::vector<std::string> vomsdata::ListTargets ( void )`

Returns the list of targets.

3.6.3.14 `bool vomldata::LoadCredentials ( X509 *, EVP_PKEY *, STACK_OF(X509)* )`

3.6.3.15 `bool vomldata::LoadSystemContacts ( std::string dir = " " )`

Loads the system wide configuration files.

#### Parameters

<i>dir</i>	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.

3.6.3.16 `bool vomldata::LoadUserContacts ( std::string dir = " " )`

Loads the user-specific configuration files.

#### Parameters

<i>dir</i>	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.

3.6.3.17 `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

<i>att</i>	The attribute to be ordered.
------------	------------------------------

3.6.3.18 `void vomldata::ResetOrder ( void )`

Resets the ordering.



**3.6.3.19 void vomsdata::ResetTargets ( void )**

Resets the target list.

**3.6.3.20 bool vomsdata::Retrieve ( X509\_EXTENSION \* ext )**

Gets VOMS information from the given extension

**Parameters**

<i>ext</i>	The extension to parse.
------------	-------------------------

**Returns**

failure (F) or success (T)

**3.6.3.21 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**

<i>cert</i>	The certificate with the VOMS extensions
<i>chain</i>	The chain of the validation certificates (only the intermediate ones)
<i>how</i>	Recursion type

**Returns**

failure (F) or success (T)

**3.6.3.22 bool vomsdata::Retrieve ( AC \* ac )**

Gets VOMS information from the AC

**Parameters**

<i>ext</i>	The extension to parse.
------------	-------------------------

**Returns**

failure (F) or success (T)

**3.6.3.23 bool vomsdata::Retrieve ( FILE \* file, recurse\_type how )**

Gets VOMS information from a proxy saved as a file.

**Parameters**

<i>file</i>	the file name
<i>how</i>	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.

3.6.3.24 `bool vomldata::RetrieveFromCred ( gss_cred_id_t credential, recurse_type how )`

Gets VOMS information from the given globus credential

**Parameters**

<i>credential</i>	The credential from which to retrieve the certificate.
<i>how</i>	Recursion type

**Returns**

failure (F) or success (T)

3.6.3.25 `bool vomldata::RetrieveFromCtx ( gss_ctx_id_t context, recurse_type how )`

Gets VOMS information from the given globus context

**Parameters**

<i>context</i>	The context from which to retrieve the certificate.
<i>how</i>	Recursion type

**Returns**

failure (F) or success (T)

3.6.3.26 `bool vomldata::RetrieveFromProxy ( recurse_type how )`

Gets VOMS information from an existing globus proxy

**Parameters**

<i>how</i>	Recursion type
------------	----------------

**Returns**

failure (F) or success (T)

3.6.3.27 `std::string vomsdata::ServerErrors ( void )`

Gets the error message returned by the server

3.6.3.28 `void vomsdata::SetLifetime ( int lifetime )`

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

#### Parameters

<i>lifetime</i>	Requested lifetime, in seconds
-----------------	--------------------------------

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

3.6.3.30 `void vomsdata::SetVerificationTime ( time_t )`

3.6.3.31 `void vomsdata::SetVerificationType ( verify_type how )`

Sets the type of verification done on the data.

#### Parameters

<i>how</i>	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 368 of file voms\_api.h.

3.6.4.2 `error_type vomsdata::error`

Error code

Definition at line 213 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 372 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 371 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 <openssl/x509.h>
#include <openssl/bio.h>
#include <sys/types.h>
#include "newformat.h"
```

#### Data Structures

- struct [data](#)  
*User's characteristics: can be repeated. Generic name-value attribute : can be repeated.*
- struct [attribute](#)
- struct [attributelist](#)
- struct [contactdata](#)
- struct [voms](#)
- struct [vomsdata](#)
- class **vomsdata::Initializer**

#### Defines

- #define [NOGLOBUS](#)

## Typedefs

- typedef void \* [gss\\_cred\\_id\\_t](#)
- typedef void \* [gss\\_ctx\\_id\\_t](#)
- 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\\_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 [getVOMSMajorVersionNumber](#) (void)
- int [getVOMSMinorVersionNumber](#) (void)
- int [getVOMSPatchVersionNumber](#) (void)

### 4.1.1 Define Documentation

#### 4.1.1.1 #define NOGLOBUS

Definition at line 33 of file voms\_api.h.

### 4.1.2 Typedef Documentation

#### 4.1.2.1 typedef bool(\* [check\\_sig](#))(X509 \*, void \*, [verror\\_type](#) &)

Definition at line 190 of file voms\_api.h.

#### 4.1.2.2 typedef void\* gss\_cred\_id\_t

Definition at line 42 of file voms\_api.h.

#### 4.1.2.3 typedef void\* gss\_ctx\_id\_t

Definition at line 43 of file voms\_api.h.

### 4.1.3 Enumeration Type Documentation

#### 4.1.3.1 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 77 of file voms\_api.h.

#### 4.1.3.2 enum recurse\_type

**Enumerator:**

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

Definition at line 143 of file voms\_api.h.

#### 4.1.3.3 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 149 of file voms\_api.h.

#### 4.1.3.4 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 163 of file `voms_api.h`.

#### 4.1.4 Function Documentation

4.1.4.1 `int getVOMSMajorVersionNumber ( void )`

4.1.4.2 `int getVOMSMinorVersionNumber ( void )`

4.1.4.3 `int getVOMSPatchVersionNumber ( void )`



# Index

- ~voms
  - voms, [10](#)
- ~vomsdata
  - vomsdata, [14](#)
- AddTarget
  - vomsdata, [14](#)
- attribute, [5](#)
  - name, [5](#)
  - qualifier, [5](#)
  - value, [5](#)
- attributelist, [6](#)
  - attributes, [6](#)
  - grantor, [6](#)
- attributes
  - attributelist, [6](#)
- cap
  - data, [8](#)
- check\_sig
  - voms\_api.h, [24](#)
- Contact
  - vomsdata, [14](#), [15](#)
- contact
  - contactdata, [7](#)
- contactdata, [6](#)
  - contact, [7](#)
  - host, [7](#)
  - nick, [7](#)
  - port, [7](#)
  - version, [7](#)
  - vo, [7](#)
- ContactRaw
  - vomsdata, [15](#), [16](#)
- ContactRESTRaw
  - vomsdata, [16](#)
- custom
  - voms, [10](#)
- data, [8](#)
  - cap, [8](#)
  - group, [8](#)
  - role, [8](#)
  - vomsdata, [21](#)
- data\_type
  - voms\_api.h, [25](#)
- date1
  - voms, [10](#)
- date2
  - voms, [11](#)
- DefaultData
  - vomsdata, [16](#)
- error
  - vomsdata, [21](#)
- ErrorMessage
  - vomsdata, [16](#)
- Export
  - vomsdata, [16](#)
- extra\_data
  - vomsdata, [21](#)
- FindByAlias
  - vomsdata, [17](#)
- FindByVO
  - vomsdata, [17](#)
- fqan
  - voms, [11](#)
- GetAC
  - voms, [10](#)
- GetAttributes
  - voms, [10](#)
- GetTargets
  - voms, [10](#)
- getVOMSMajorVersionNumber
  - voms\_api.h, [26](#)
- getVOMSMinorVersionNumber
  - voms\_api.h, [26](#)
- getVOMSPatchVersionNumber
  - voms\_api.h, [26](#)
- grantor

- attributelist, [6](#)
- group
  - data, [8](#)
- gss\_cred\_id\_t
  - voms\_api.h, [24](#)
- gss\_ctx\_id\_t
  - voms\_api.h, [25](#)
- host
  - contactdata, [7](#)
- Import
  - vomsdata, [17](#)
- ListTargets
  - vomsdata, [17](#)
- LoadCredentials
  - vomsdata, [17](#)
- LoadSystemContacts
  - vomsdata, [18](#)
- LoadUserContacts
  - vomsdata, [18](#)
- name
  - attribute, [5](#)
- nick
  - contactdata, [7](#)
- NOGLOBUS
  - voms\_api.h, [24](#)
- operator=
  - voms, [10](#)
- Order
  - vomsdata, [18](#)
- port
  - contactdata, [7](#)
- 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, [25](#)
- ResetOrder
  - vomsdata, [18](#)
- ResetTargets
  - vomsdata, [18](#)
- Retrieve
  - vomsdata, [19](#)
- RetrieveFromCred
  - vomsdata, [20](#)
- RetrieveFromCtx
  - vomsdata, [20](#)
- RetrieveFromProxy
  - vomsdata, [20](#)
- role
  - data, [8](#)
- serial
  - voms, [11](#)
- server
  - voms, [11](#)
- serverca
  - voms, [11](#)
- ServerErrors
  - vomsdata, [20](#)
- SetLifetime
  - vomsdata, [21](#)
- SetRetryCount
  - vomsdata, [21](#)
- SetVerificationTime
  - vomsdata, [21](#)
- SetVerificationType
  - vomsdata, [21](#)
- siglen
  - voms, [11](#)
- signature
  - voms, [11](#)
- std
  - voms, [11](#)
- TranslateVOMS
  - voms, [10](#)
- type
  - voms, [12](#)
- TYPE\_CUSTOM
  - voms\_api.h, [25](#)
- TYPE\_NODATA
  - voms\_api.h, [25](#)
- TYPE\_STD
  - voms\_api.h, [25](#)
- uri
  - voms, [12](#)
- user

- voms, [12](#)
- userca
  - voms, [12](#)
- 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, [26](#)
- VERR\_DIR
  - voms\_api.h, [26](#)
- VERR\_EXTRINFO
  - voms\_api.h, [26](#)
- VERR\_FILE
  - voms\_api.h, [26](#)
- VERR\_FORMAT
  - voms\_api.h, [26](#)
- VERR\_IDCHECK
  - voms\_api.h, [26](#)
- VERR\_MEM
  - voms\_api.h, [26](#)
- VERR\_NODATA
  - voms\_api.h, [26](#)
- VERR\_NOEXT
  - voms\_api.h, [26](#)
- VERR\_NOIDENT
  - voms\_api.h, [26](#)
- VERR\_NOINIT
  - voms\_api.h, [26](#)
- VERR\_NONE
  - voms\_api.h, [26](#)
- VERR\_NOSOCKET
  - voms\_api.h, [26](#)
- VERR\_NOTAVAIL
  - voms\_api.h, [26](#)
- VERR\_ORDER
  - voms\_api.h, [26](#)
- VERR\_PARAM
  - voms\_api.h, [26](#)
- VERR\_PARSE
  - voms\_api.h, [26](#)
- VERR\_SERVER
  - voms\_api.h, [26](#)
- VERR\_SERVERCODE
  - voms\_api.h, [26](#)
- VERR\_SIGN
  - voms\_api.h, [26](#)
- VERR\_TIME
  - voms\_api.h, [26](#)
- VERR\_TYPE
  - voms\_api.h, [26](#)
- VERR\_VERIFY
  - voms\_api.h, [26](#)
- verror\_type
  - voms\_api.h, [25](#)
- version
  - contactdata, [7](#)
  - voms, [12](#)
- vo
  - contactdata, [7](#)
- voms, [9](#)
  - ~voms, [10](#)
  - custom, [10](#)
  - date1, [10](#)
  - date2, [11](#)
  - fqan, [11](#)
  - GetAC, [10](#)
  - GetAttributes, [10](#)
  - GetTargets, [10](#)
  - operator=, [10](#)
  - serial, [11](#)
  - server, [11](#)
  - serverca, [11](#)
  - siglen, [11](#)
  - signature, [11](#)
  - std, [11](#)
  - TranslateVOMS, [10](#)
  - type, [12](#)
  - uri, [12](#)
  - user, [12](#)
  - userca, [12](#)

- version, [12](#)
- voms, [10](#)
- vomsdata, [10](#)
- voname, [12](#)
- voms\_api.h
  - RECURSE\_CHAIN, [25](#)
  - RECURSE\_DEEP, [25](#)
  - RECURSE\_NONE, [25](#)
  - TYPE\_CUSTOM, [25](#)
  - TYPE\_NODATA, [25](#)
  - TYPE\_STD, [25](#)
  - 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, [26](#)
  - VERR\_DIR, [26](#)
  - VERR\_EXTRAINFO, [26](#)
  - VERR\_FILE, [26](#)
  - VERR\_FORMAT, [26](#)
  - VERR\_IDCHECK, [26](#)
  - VERR\_MEM, [26](#)
  - VERR\_NODATA, [26](#)
  - VERR\_NOEXT, [26](#)
  - VERR\_NOIDENT, [26](#)
  - VERR\_NOINIT, [26](#)
  - VERR\_NONE, [26](#)
  - VERR\_NOCKET, [26](#)
  - VERR\_NOTAVAIL, [26](#)
  - VERR\_ORDER, [26](#)
  - VERR\_PARAM, [26](#)
  - VERR\_PARSE, [26](#)
  - VERR\_SERVER, [26](#)
  - VERR\_SERVERCODE, [26](#)
  - VERR\_SIGN, [26](#)
  - VERR\_TIME, [26](#)
  - VERR\_TYPE, [26](#)
  - VERR\_VERIFY, [26](#)
- voms\_api.h, [23](#)
  - check\_sig, [24](#)
  - data\_type, [25](#)
  - getVOMSMajorVersionNumber, [26](#)
  - getVOMSMinorVersionNumber, [26](#)
  - getVOMSPatchVersionNumber, [26](#)
  - gss\_cred\_id\_t, [24](#)
  - gss\_ctx\_id\_t, [25](#)
  - NOGLOBUS, [24](#)
  - recurse\_type, [25](#)
  - verify\_type, [25](#)
  - verror\_type, [25](#)
- vomsdata, [13](#)
  - ~vomsdata, [14](#)
  - AddTarget, [14](#)
  - Contact, [14](#), [15](#)
  - ContactRaw, [15](#), [16](#)
  - ContactRESTRaw, [16](#)
  - data, [21](#)
  - DefaultData, [16](#)
  - error, [21](#)
  - ErrorMessage, [16](#)
  - Export, [16](#)
  - extra\_data, [21](#)
  - FindByAlias, [17](#)
  - FindByVO, [17](#)
  - Import, [17](#)
  - ListTargets, [17](#)
  - LoadCredentials, [17](#)
  - LoadSystemContacts, [18](#)
  - LoadUserContacts, [18](#)
  - Order, [18](#)
  - ResetOrder, [18](#)
  - ResetTargets, [18](#)
  - Retrieve, [19](#)
  - RetrieveFromCred, [20](#)
  - RetrieveFromCtx, [20](#)
  - RetrieveFromProxy, [20](#)
  - ServerErrors, [20](#)
  - SetLifetime, [21](#)
  - SetRetryCount, [21](#)
  - SetVerificationTime, [21](#)
  - SetVerificationType, [21](#)
- voms, [10](#)
- vomsdata, [14](#)
- workvo, [22](#)
- voname
  - voms, [12](#)
- workvo
  - vomsdata, [22](#)