

# VOMS C API Reference Manual

## 1.5.0

Generated by Doxygen 1.4.6

Tue Jun 30 11:34:57 2009



# Contents

<b>1</b>	<b>VOMS C API Hierarchical Index</b>	<b>1</b>
1.1	VOMS C API Class Hierarchy . . . . .	1
<b>2</b>	<b>VOMS C API Data Structure Index</b>	<b>3</b>
2.1	VOMS C API Data Structures . . . . .	3
<b>3</b>	<b>VOMS C API File Index</b>	<b>5</b>
3.1	VOMS C API File List . . . . .	5
<b>4</b>	<b>VOMS C API Data Structure Documentation</b>	<b>7</b>
4.1	attribute Struct Reference . . . . .	7
4.2	contactdata Struct Reference . . . . .	8
4.3	data Struct Reference . . . . .	10
4.4	voms Struct Reference . . . . .	11
4.5	vomsdata Struct Reference . . . . .	14
<b>5</b>	<b>VOMS C API File Documentation</b>	<b>17</b>
5.1	voms_apic.h File Reference . . . . .	17



# Chapter 1

## VOMS C API Hierarchical Index

### 1.1 VOMS C API Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

attribute . . . . .	7
contactdata . . . . .	8
data . . . . .	10
voms . . . . .	11
vomsdata . . . . .	14



# Chapter 2

## VOMS C API Data Structure Index

### 2.1 VOMS C API Data Structures

Here are the data structures with brief descriptions:

<a href="#">attribute</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>	14





## Chapter 3

# VOMS C API File Index

### 3.1 VOMS C API File List

Here is a list of all files with brief descriptions:

<a href="#">voms_apic.h</a>	.....	17
-----------------------------	-------	----



## Chapter 4

# VOMS C API Data Structure Documentation

### 4.1 attribute Struct Reference

```
#include <voms_apic.h>
```

#### Data Fields

- const char \* [name](#)
- const char \* [value](#)
- const char \* [qualifier](#)

#### 4.1.1 Detailed Description

Definition at line 38 of file voms\_apic.h.

#### 4.1.2 Field Documentation

##### 4.1.2.1 const char\* [attribute::name](#)

Definition at line 39 of file voms\_apic.h.

##### 4.1.2.2 const char\* [attribute::qualifier](#)

Definition at line 41 of file voms\_apic.h.

##### 4.1.2.3 const char\* [attribute::value](#)

Definition at line 40 of file voms\_apic.h.

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

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

## 4.2 contactdata Struct Reference

```
#include <voms_apic.h>
```

### Data Fields

- char \* [nick](#)
- char \* [host](#)
- char \* [contact](#)
- char \* [vo](#)
- int [port](#)
- char \* [reserved](#)
- int [version](#)

### 4.2.1 Detailed Description

Definition at line 44 of file voms\_apic.h.

### 4.2.2 Field Documentation

#### 4.2.2.1 char\* [contactdata::contact](#)

The subject of the server's certificate

Definition at line 49 of file voms\_apic.h.

#### 4.2.2.2 char\* [contactdata::host](#)

The hostname of the server

Definition at line 48 of file voms\_apic.h.

#### 4.2.2.3 char\* [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 VOMS\_FindBy\* functions. The alias of the server

Definition at line 47 of file voms\_apic.h.

#### 4.2.2.4 int [contactdata::port](#)

The port on which the server is listening

Definition at line 51 of file voms\_apic.h.

#### 4.2.2.5 char\* [contactdata::reserved](#)

HANDS OFF!

Definition at line 52 of file voms\_apic.h.

#### 4.2.2.6 int [contactdata::version](#)

The version of Globus on which this server runs.

Definition at line 53 of file voms\_apic.h.

#### 4.2.2.7 char\* [contactdata::vo](#)

The VO served by this server

Definition at line 50 of file voms\_apic.h.

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

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

## 4.3 data Struct Reference

User's characteristics: can be repeated.

```
#include <voms_apic.h>
```

### Data Fields

- char \* [group](#)
- char \* [role](#)
- char \* [cap](#)

#### 4.3.1 Detailed Description

User's characteristics: can be repeated.

<

Definition at line 32 of file voms\_apic.h.

#### 4.3.2 Field Documentation

##### 4.3.2.1 char\* [data::cap](#)

user's capability

Definition at line 35 of file voms\_apic.h.

##### 4.3.2.2 char\* [data::group](#)

user's group

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

##### 4.3.2.3 char\* [data::role](#)

user's role

Definition at line 34 of file voms\_apic.h.

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

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

## 4.4 voms Struct Reference

```
#include <voms_apic.h>
```

### Data Fields

- int [siglen](#)
- char \* [signature](#)
- char \* [user](#)
- char \* [userca](#)
- char \* [server](#)
- char \* [serverca](#)
- char \* [voname](#)
- char \* [uri](#)
- char \* [date1](#)
- char \* [date2](#)
- int [type](#)
- data \*\* [std](#)
- char \* [custom](#)
- int [datalen](#)
- int [version](#)
- char \*\* [fqan](#)
- char \* [serial](#)
- AC \* [ac](#)
- X509 \* [holder](#)

### 4.4.1 Detailed Description

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

### 4.4.2 Field Documentation

#### 4.4.2.1 AC\* [voms::ac](#)

Definition at line 84 of file voms\_apic.h.

#### 4.4.2.2 char\* [voms::custom](#)

The data returned by an S command

Definition at line 77 of file voms\_apic.h.

#### 4.4.2.3 int [voms::datalen](#)

Definition at line 78 of file voms\_apic.h.

**4.4.2.4 char\* voms::date1**

Beginning of validity of the user info

Definition at line 73 of file voms\_apic.h.

**4.4.2.5 char\* voms::date2**

End of validity of the user info

Definition at line 74 of file voms\_apic.h.

**4.4.2.6 char\*\* voms::fqan**

User's attributes in compact format

Definition at line 80 of file voms\_apic.h.

**4.4.2.7 X509\* voms::holder**

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

**4.4.2.8 char\* voms::serial**

Serial number. Only significant if coming from AC. Null otherwise

Definition at line 81 of file voms\_apic.h.

**4.4.2.9 char\* voms::server**

The VOMS server DN, as from its certificate

Definition at line 69 of file voms\_apic.h.

**4.4.2.10 char\* voms::serverca**

The CA which signed the VOMS certificate

Definition at line 70 of file voms\_apic.h.

**4.4.2.11 int voms::siglen**

The length of the VOMS server signature

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

**4.4.2.12 char\* voms::signature**

The VOMS server signature

Definition at line 66 of file voms\_apic.h.



**4.4.2.13 struct [data\\*\\* voms::std](#)**

User's characteristics

Definition at line 76 of file voms\_apic.h.

**4.4.2.14 int [voms::type](#)**

The type of data returned

Definition at line 75 of file voms\_apic.h.

**4.4.2.15 char\* [voms::uri](#)**

The URI of the VOMS server

Definition at line 72 of file voms\_apic.h.

**4.4.2.16 char\* [voms::user](#)**

The user's DN, as from his certificate

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

**4.4.2.17 char\* [voms::userca](#)**

The CA which signed the user's certificate

Definition at line 68 of file voms\_apic.h.

**4.4.2.18 int [voms::version](#)**

Definition at line 79 of file voms\_apic.h.

**4.4.2.19 char\* [voms::vname](#)**

The name of the VO to which the VOMS belongs

Definition at line 71 of file voms\_apic.h.

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

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

## 4.5 vomsdata Struct Reference

```
#include <voms_apic.h>
```

### Data Fields

- char \* [cdir](#)
- char \* [vdir](#)
- voms \*\* [data](#)
- char \* [workvo](#)
- char \* [extra\\_data](#)
- int [volen](#)
- int [extralen](#)
- vomsdata \* [real](#)

#### 4.5.1 Detailed Description

Definition at line 128 of file voms\_apic.h.

#### 4.5.2 Field Documentation

##### 4.5.2.1 char\* [vomsdata::cdir](#)

Definition at line 129 of file voms\_apic.h.

##### 4.5.2.2 struct 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 131 of file voms\_apic.h.

##### 4.5.2.3 char\* [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 136 of file voms\_apic.h.

##### 4.5.2.4 int [vomsdata::extralen](#)

Definition at line 146 of file voms\_apic.h.

##### 4.5.2.5 struct vomsdata\* [vomsdata::real](#)

Definition at line 148 of file voms\_apic.h.

**4.5.2.6 char\* [vomsdata::vdir](#)**

Definition at line 130 of file voms\_apic.h.

**4.5.2.7 int [vomsdata::volen](#)**

Definition at line 145 of file voms\_apic.h.

**4.5.2.8 char\* [vomsdata::workvo](#)**

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

Definition at line 134 of file voms\_apic.h.

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

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



## Chapter 5

# VOMS C API File Documentation

### 5.1 voms\_apic.h File Reference

```
#include <openssl/x509.h>
#include <time.h>
#include "newformat.h"
```

#### Data Structures

- struct [data](#)  
*User's characteristics: can be repeated.*
- struct [attribute](#)
- struct [contactdata](#)
- struct [voms](#)
- struct [vomsdata](#)

#### Defines

- #define [TYPE\\_NODATA](#) 0
- #define [TYPE\\_STD](#) 1
- #define [TYPE\\_CUSTOM](#) 2
- #define [RECURSE\\_CHAIN](#) 0
- #define [RECURSE\\_NONE](#) 1
- #define [VERIFY\\_FULL](#) 0xffffffff
- #define [VERIFY\\_NONE](#) 0x00000000
- #define [VERIFY\\_DATE](#) 0x00000001
- #define [VERIFY\\_NOTARGET](#) 0x00000002
- #define [VERIFY\\_KEY](#) 0x00000004
- #define [VERIFY\\_SIGN](#) 0x00000008
- #define [VERIFY\\_ORDER](#) 0x00000010
- #define [VERIFY\\_ID](#) 0x00000020
- #define [VERIFY\\_CERTLIST](#) 0x00000040
- #define [VERR\\_NONE](#) 0

*Error codes.*

- #define [VERR\\_NOSOCKET](#) 1
- #define [VERR\\_NOIDENT](#) 2
- #define [VERR\\_COMM](#) 3
- #define [VERR\\_PARAM](#) 4
- #define [VERR\\_NOEXT](#) 5
- #define [VERR\\_NOINIT](#) 6
- #define [VERR\\_TIME](#) 7
- #define [VERR\\_IDCHECK](#) 8
- #define [VERR\\_EXTRAINFO](#) 9
- #define [VERR\\_FORMAT](#) 10
- #define [VERR\\_NODATA](#) 11
- #define [VERR\\_PARSE](#) 12
- #define [VERR\\_DIR](#) 13
- #define [VERR\\_SIGN](#) 14
- #define [VERR\\_SERVER](#) 15
- #define [VERR\\_MEM](#) 16
- #define [VERR\\_VERIFY](#) 17
- #define [VERR\\_TYPE](#) 18
- #define [VERR\\_ORDER](#) 19
- #define [VERR\\_SERVERCODE](#) 20
- #define [VERR\\_NOTAVAIL](#) 21

## Typedefs

- typedef void \* [gss\\_cred\\_id\\_t](#)
- typedef void \* [gss\\_ctx\\_id\\_t](#)

## Functions

- [contactdata](#) \*\* [VOMS\\_FindByAlias](#) (struct [vomsdata](#) \*vd, char \*alias, char \*system, char \*user, int \*error)
- [contactdata](#) \*\* [VOMS\\_FindByVO](#) (struct [vomsdata](#) \*vd, char \*vo, char \*system, char \*user, int \*error)
- void [VOMS\\_DeleteContacts](#) (struct [contactdata](#) \*\*list)
- [vomsdata](#) \* [VOMS\\_Init](#) (char \*voms, char \*cert)
- [voms](#) \* [VOMS\\_Copy](#) (struct [voms](#) \*v, int \*error)
- [vomsdata](#) \* [VOMS\\_CopyAll](#) (struct [vomsdata](#) \*vd, int \*error)
- void [VOMS\\_Delete](#) (struct [voms](#) \*v)
- int [VOMS\\_AddTarget](#) (struct [vomsdata](#) \*vd, char \*target, int \*error)
- void [VOMS\\_FreeTargets](#) (struct [vomsdata](#) \*vd, int \*error)
- char \* [VOMS\\_ListTargets](#) (struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_SetVerificationType](#) (int type, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_SetLifetime](#) (int length, struct [vomsdata](#) \*vd, int \*error)
- void [VOMS\\_Destroy](#) (struct [vomsdata](#) \*vd)
- int [VOMS\\_ResetOrder](#) (struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_Ordering](#) (char \*order, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_Contact](#) (char \*hostname, int port, char \*servsubject, char \*command, struct [vomsdata](#) \*vd, int \*error)

- int [VOMS\\_ContactRaw](#) (char \*hostname, int port, char \*servsubject, char \*command, void \*\*data, int \*datalen, int \*version, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_Retrieve](#) (X509 \*cert, STACK\_OF(X509)\*chain, int how, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_Import](#) (char \*buffer, int buflen, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_Export](#) (char \*\*buffer, int \*buflen, struct [vomsdata](#) \*vd, int \*error)
- voms \* [VOMS\\_DefaultData](#) (struct [vomsdata](#) \*vd, int \*error)
- char \* [VOMS\\_ErrorMessage](#) (struct [vomsdata](#) \*vd, int error, char \*buffer, int len)
- int [VOMS\\_RetrieveEXT](#) (X509\_EXTENSION \*ext, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_RetrieveFromCred](#) (gss\_cred\_id\_t cred, int how, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_RetrieveFromFile](#) (FILE \*file, int how, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_RetrieveFromCtx](#) (gss\_ctx\_id\_t ctx, int how, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_RetrieveFromProxy](#) (int how, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_GetAttributeSourcesNumber](#) (struct voms \*v, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_GetAttributeSourceHandle](#) (struct voms \*v, int num, struct [vomsdata](#) \*vd, int \*error)
- const char \* [VOMS\\_GetAttributeGrantor](#) (struct voms \*v, int handle, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_GetAttributesNumber](#) (struct voms \*v, int handle, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_GetAttribute](#) (struct voms \*v, int handle, int num, struct [attribute](#) \*at, struct [vomsdata](#) \*vd, int \*error)
- vomsdata \* [VOMS\\_Duplicate](#) (struct [vomsdata](#) \*vd)
- AC \* [VOMS\\_GetAC](#) (struct voms \*v)
- int [getMajorVersionNumber](#) (void)
- int [getMinorVersionNumber](#) (void)
- int [getPatchVersionNumber](#) (void)
- int [VOMS\\_SetVerificationTime](#) (time\_t verificationtime, struct [vomsdata](#) \*vd, int \*error)
- char \*\* [VOMS\\_GetTargetsList](#) (struct voms \*v, struct [vomsdata](#) \*vd, int \*error)
- void [VOMS\\_FreeTargetsList](#) (char \*\*)

## 5.1.1 Define Documentation

### 5.1.1.1 #define RECURSE\_CHAIN 0

Definition at line 89 of file voms\_apic.h.

### 5.1.1.2 #define RECURSE\_NONE 1

Definition at line 90 of file voms\_apic.h.

### 5.1.1.3 #define TYPE\_CUSTOM 2

result of an S command

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

### 5.1.1.4 #define TYPE\_NODATA 0

no data

Definition at line 59 of file voms\_apic.h.

**5.1.1.5 #define TYPE\_STD 1**

group, role, capability triplet

Definition at line 60 of file voms\_apic.h.

**5.1.1.6 #define VERIFY\_CERTLIST 0x00000040**

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

**5.1.1.7 #define VERIFY\_DATE 0x00000001**

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

**5.1.1.8 #define VERIFY\_FULL 0xffffffff**

Definition at line 92 of file voms\_apic.h.

**5.1.1.9 #define VERIFY\_ID 0x00000020**

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

**5.1.1.10 #define VERIFY\_KEY 0x00000004**

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

**5.1.1.11 #define VERIFY\_NONE 0x00000000**

Definition at line 93 of file voms\_apic.h.

**5.1.1.12 #define VERIFY\_NOTARGET 0x00000002**

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

**5.1.1.13 #define VERIFY\_ORDER 0x00000010**

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

**5.1.1.14 #define VERIFY\_SIGN 0x00000008**

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

**5.1.1.15 #define VERR\_COMM 3**

Server problem

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



**5.1.1.16 #define VERR\_DIR 13**

Directory error

Definition at line 118 of file voms\_apic.h.

**5.1.1.17 #define VERR\_EXTRAINFO 9**

VO name and URI missing

Definition at line 114 of file voms\_apic.h.

**5.1.1.18 #define VERR\_FORMAT 10**

Wrong data format

Definition at line 115 of file voms\_apic.h.

**5.1.1.19 #define VERR\_IDCHECK 8**

User data in extension different from the real ones

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

**5.1.1.20 #define VERR\_MEM 16**

Memory problems

Definition at line 121 of file voms\_apic.h.

**5.1.1.21 #define VERR\_NODATA 11**

Empty extension

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

**5.1.1.22 #define VERR\_NOEXT 5**

VOMS extension missing

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

**5.1.1.23 #define VERR\_NOIDENT 2**

Cannot identify itself (certificate problem)

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

**5.1.1.24 #define VERR\_NOINIT 6**

Initialization error

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

**5.1.1.25 #define VERR\_NONE 0**

Error codes.

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

**5.1.1.26 #define VERR\_NOSOCKET 1**

Socket problem

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

**5.1.1.27 #define VERR\_NOTAVAIL 21**

Method not available

Definition at line 126 of file voms\_apic.h.

**5.1.1.28 #define VERR\_ORDER 19**

Ordering different than required

Definition at line 124 of file voms\_apic.h.

**5.1.1.29 #define VERR\_PARAM 4**

Wrong parameters

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

**5.1.1.30 #define VERR\_PARSE 12**

Parse error

Definition at line 117 of file voms\_apic.h.

**5.1.1.31 #define VERR\_SERVER 15**

Unidentifiable VOMS server

Definition at line 120 of file voms\_apic.h.

**5.1.1.32 #define VERR\_SERVERCODE 20**

Error from the server

Definition at line 125 of file voms\_apic.h.

**5.1.1.33 #define VERR\_SIGN 14**

Signature error

Definition at line 119 of file voms\_apic.h.

#### 5.1.1.34 `#define VERR_TIME 7`

Error in time checking

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

#### 5.1.1.35 `#define VERR_TYPE 18`

Returned data of unknown type

Definition at line 123 of file voms\_apic.h.

#### 5.1.1.36 `#define VERR_VERIFY 17`

Generic verification error

Definition at line 122 of file voms\_apic.h.

### 5.1.2 Typedef Documentation

#### 5.1.2.1 `typedef void* gss_cred_id_t`

Definition at line 19 of file voms\_apic.h.

#### 5.1.2.2 `typedef void* gss_ctx_id_t`

Definition at line 20 of file voms\_apic.h.

### 5.1.3 Function Documentation

#### 5.1.3.1 `int getMajorVersionNumber (void)`

#### 5.1.3.2 `int getMinorVersionNumber (void)`

#### 5.1.3.3 `int getPatchVersionNumber (void)`

#### 5.1.3.4 `int VOMS_AddTarget (struct vomsdata * vd, char * target, int * error)`

Adds a target to the AC.

##### Parameters:

*vd* The vomsdata structure.

*target* The target to add. It should be a FQDN.

*error* RETURN PARAMETER: qualifies the error message.

##### Returns:

failure (0) or success (>0)

### 5.1.3.5 **int VOMS\_Contact** (char \* *hostname*, int *port*, char \* *servsubject*, char \* *command*, struct *vomsdata* \* *vd*, int \* *error*)

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* Command  
*vd* RETURN PARAMETER: contains the data returned by the connection  
*error* RETURN PARAMETER: Qualifies the error message

#### Returns:

failure (0) or success (<>0)

### 5.1.3.6 **int VOMS\_ContactRaw** (char \* *hostname*, int *port*, char \* *servsubject*, char \* *command*, void \*\* *data*, int \* *datalen*, int \* *version*, struct *vomsdata* \* *vd*, int \* *error*)

The same as `VOMS_Contact`, except that instead of starting the verification process, the data is returned as is in the

#### Parameters:

*data* and *datalen* fields.  
*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  
*version* is the version number of the data.  
*vd* RETURN PARAMETER: contains the data returned by the connection  
*error* RETURN PARAMETER: Qualifies the error message

#### Returns:

failure (0) or success (<>0)

### 5.1.3.7 **struct voms\* VOMS\_Copy** (struct *voms* \* *v*, int \* *error*)

Copies a voms structure. N.B: This is the ONLY way to correctly initialize a voms structure as a copy of another voms structure.

#### Parameters:

*v* The structure to copy.  
*error* RETURN PARAMETER: qualifies the error message.

#### Returns:

NULL (error) or the new voms structure.

**5.1.3.8 struct vomsdata\* VOMS\_CopyAll (struct vomsdata \* *vd*, int \* *error*)**

Copies a vomsdata structure. N.B: This is the ONLY way to correctly initialize a vomsdata structure as a copy of another vomsdata structure.

**Parameters:**

*vd* The structure to copy.

*error* RETURN PARAMETER: qualifies the error message.

**Returns:**

NULL (error) or the new vomsdata structure.

**5.1.3.9 struct voms\* VOMS\_DefaultData (struct vomsdata \* *vd*, int \* *error*)**

Gets the default attributes from a vomsdata structure.

**Parameters:**

*vd* the vomsdata structure to analyze

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

a pointer to the relevant voms structure. DO NOT modify the fields.

**5.1.3.10 void VOMS\_Delete (struct voms \* *v*)**

Deletes a voms structure

**Parameters:**

*v* Pointer to the structure to delete.

**5.1.3.11 void VOMS\_DeleteContacts (struct contactdata \*\* *list*)**

Frees a contactdata vector.

**Parameters:**

*list* The vector to free.

**Returns:**

NONE

**5.1.3.12 void VOMS\_Destroy (struct vomsdata \* *vd*)**

Destroys a proper vomsdata structure /param *vd* The structure to deallocate.

**5.1.3.13** `struct vomsdata* VOMS_Duplicate (struct vomsdata * vd)`

**5.1.3.14** `char* VOMS_ErrorMessage (struct vomsdata * vd, int error, char * buffer, int len)`

Gets a textual description of the error.

**Parameters:**

*vd* The vomsdata structure to analyze

*error* The error returned by the last function

*buffer* A pointer to a buffer where the error message will be written. If NULL, then memory is allocated by the function, and will have to be free()ed by the caller.

*len* The length of the memory pointed to by the buffer parameter.

**Returns:**

A pointer to the error message.

**5.1.3.15** `int VOMS_Export (char ** buffer, int * buflen, struct vomsdata * vd, int * error)`

Converts data into a test format

**Parameters:**

*buffer* OUTPUT PARAMETER contains the converted data

*buflen* OUTPUT PARAMETER contains the length of buffer

*vd* contains the data to convert

*error* RETURN PARAMETER Qualifies the error message

**Returns:**

failure (0) or success (>0)

**5.1.3.16** `struct contactdata** VOMS_FindByAlias (struct vomsdata * vd, char * alias, char * system, char * user, int * error)`

Gets a list of VOMS servers which share an alias.

**Parameters:**

*vd* The correctly initialized vomsdata structured.

*alias* The alias to look for.

*system* The directory in which to look for the system configuration files. If NULL, defaults to /opt/edc/etc/vomses

*user* The directory in which to look for the user configuration files. Defaults to \$VOMS\_USERCONF if NULL. Again defaults to \$HOME/.edg/vomses if the latter is NULL, or to ~/.edg/vomses as a last resort.

*error* RETURN PARAMETER: qualifies the error message.

**Returns:**

NULL, or a NULL-terminated vector of contactdata structures. The only supported way to free this array is via the VOMS\_DeleteContacts function. Note also that the order in which the servers are returned is unspecified.

**5.1.3.17** struct `contactdata`\*\* VOMS\_FindByVO (struct `vomsdata` \* *vd*, char \* *vo*, char \* *system*, char \* *user*, int \* *error*)

Gets a list of VOMS servers which serve the same VO.

**Parameters:**

*vd* The correctly initialized vomsdata structured.

*vo* The VO to look for.

*system* The directory in which to look for the system configuration files. If NULL, defaults to /opt/edc/etc/vomses

*user* The directory in which to look for the user configuration files. Defaults to \$VOMS\_USERCONF if NULL. Again defaults to \$HOME/.edg/vomses if the latter is NULL, or to ~/.edg/vomses as a last resort.

*error* RETURN PARAMETER: qualifies the error message.

**Returns:**

NULL, or a NULL-terminated vector of contactdata structures. The only supported way to free this array is via the VOMS\_DeleteContacts function. Note also that the order in which the servers are returned is unspecified.

**5.1.3.18** void VOMS\_FreeTargets (struct `vomsdata` \* *vd*, int \* *error*)

Delete the targets from the AC.

**Parameters:**

*vd* The vomsdata structure.

*error* RETURN PARAMETER: qualifies the error message.

- 5.1.3.19** void VOMS\_FreeTargetsList (char \*\*)
- 5.1.3.20** AC\* VOMS\_GetAC (struct voms \* v)
- 5.1.3.21** int VOMS\_GetAttribute (struct voms \* v, int handle, int num, struct attribute \* at, struct vomsdata \* vd, int \* error)
- 5.1.3.22** const char\* VOMS\_GetAttributeGrantor (struct voms \* v, int handle, struct vomsdata \* vd, int \* error)
- 5.1.3.23** int VOMS\_GetAttributesNumber (struct voms \* v, int handle, struct vomsdata \* vd, int \* error)
- 5.1.3.24** int VOMS\_GetAttributeSourceHandle (struct voms \* v, int num, struct vomsdata \* vd, int \* error)
- 5.1.3.25** int VOMS\_GetAttributeSourcesNumber (struct voms \* v, struct vomsdata \* vd, int \* error)
- 5.1.3.26** char\*\* VOMS\_GetTargetsList (struct voms \* v, struct vomsdata \* vd, int \* error)
- 5.1.3.27** int VOMS\_Import (char \* buffer, int buflen, struct vomsdata \* vd, int \* error)

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
- buflen* contains the length of buffer
- vd* RETURN PARAMETER: contains the data returned by the connection
- error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (>0)

- 5.1.3.28** struct vomsdata\* VOMS\_Init (char \* voms, char \* cert)

Initializes a vomsdata structure for use by the other functions. N.B: This is the ONLY way to correctly initialize a vomsdata structure. It is also forbidden to directly take the sizeof() of this structure.

**Parameters:**

- voms* The directory which contains the certificates of the VOMS servers
- cert* The directory which contains the CA certificates

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

**Returns:**

NULL for failure, or a pointer to a properly initialized structure.



**5.1.3.29** `char* VOMS_ListTargets (struct vomsdata * vd, int * error)`

**5.1.3.30** `int VOMS_Ordering (char * order, struct vomsdata * vd, int * error)`

Further specified the order of the returned attributes. Please do note that calls are cumulative unless `VOMS_ResetOrder()` is called.

**Parameters:**

*order* the group:role attribute.

*vd* RETURN PARAMETER: contains the modified data.

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.31** `int VOMS_ResetOrder (struct vomsdata * vd, int * error)`

Unsets the return order of the attributes.

**Parameters:**

*vd* RETURN PARAMETER: contains the modified data.

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.32** `int VOMS_Retrieve (X509 * cert, STACK_OF(X509)* chain, int how, struct vomsdata * vd, int * error)`

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

*vd* RETURN PARAMETER: contains the data returned by the connection

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.33 int VOMS\_RetrieveEXT (X509\_EXTENSION \* *ext*, struct vomsdata \* *vd*, int \* *error*)**

Gets VOMS information from the given extension

**Parameters:**

*ext* The extension to parse.

*vd* RETURN PARAMETER: contains the data returned by the connection

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.34 int VOMS\_RetrieveFromCred (gss\_cred\_id\_t *cred*, int *how*, struct vomsdata \* *vd*, int \* *error*)**

Gets VOMS information from the given globus credential

**Parameters:**

*cred* The credential from which to retrieve the certificate.

*how* Recursion type

*vd* RETURN PARAMETER: contains the data returned by the connection

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.35 int VOMS\_RetrieveFromCtx (gss\_ctx\_id\_t *ctx*, int *how*, struct vomsdata \* *vd*, int \* *error*)**

Gets VOMS information from the given globus context

**Parameters:**

*ctx* The context from which to retrieve the certificate.

*how* Recursion type

*vd* RETURN PARAMETER: contains the data returned by the connection

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.36 int VOMS\_RetrieveFromFile (FILE \* *file*, int *how*, struct vomsdata \* *vd*, int \* *error*)**

Gets VOMS information from the given globus credential

**Parameters:**

*file* The file from which to retrieve the certificate.

*how* Recursion type

*vd* RETURN PARAMETER: contains the data returned by the connection

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.37 int VOMS\_RetrieveFromProxy (int *how*, struct [vomsdata](#) \* *vd*, int \* *error*)**

Gets VOMS information from an existing globus proxy

**Parameters:**

*how* Recursion type

*vd* RETURN PARAMETER: contains the data returned by the connection

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.38 int VOMS\_SetLifetime (int *length*, struct [vomsdata](#) \* *vd*, int \* *error*)**

Set requested lifetime for [VOMS\\_Contact\(\)](#) calls.

**Parameters:**

*length* Lifetime requested.

*vd* RETURN PARAMETER: contains the modified data.

*error* RETURN PARAMETER: Qualifies the error message

**Returns:**

failure (0) or success (<>0)

**5.1.3.39 int VOMS\_SetVerificationTime (time\_t *verificationtime*, struct [vomsdata](#) \* *vd*, int \* *error*)**

**5.1.3.40 int VOMS\_SetVerificationType (int *type*, struct [vomsdata](#) \* *vd*, int \* *error*)**

# Index

ac  
    voms, [11](#)  
attribute, [7](#)  
    name, [7](#)  
    qualifier, [7](#)  
    value, [7](#)  
  
cap  
    data, [10](#)  
cdir  
    vomldata, [14](#)  
contact  
    contactdata, [8](#)  
contactdata, [8](#)  
    contact, [8](#)  
    host, [8](#)  
    nick, [8](#)  
    port, [8](#)  
    reserved, [8](#)  
    version, [8](#)  
    vo, [9](#)  
custom  
    voms, [11](#)  
  
data, [10](#)  
    cap, [10](#)  
    group, [10](#)  
    role, [10](#)  
    vomldata, [14](#)  
datalen  
    voms, [11](#)  
date1  
    voms, [11](#)  
date2  
    voms, [12](#)  
  
extra\_data  
    vomldata, [14](#)  
extralen  
    vomldata, [14](#)  
  
fqan  
    voms, [12](#)  
  
getMajorVersionNumber  
    voms\_apic.h, [23](#)  
  
getMinorVersionNumber  
    voms\_apic.h, [23](#)  
getPatchVersionNumber  
    voms\_apic.h, [23](#)  
group  
    data, [10](#)  
gss\_cred\_id\_t  
    voms\_apic.h, [23](#)  
gss\_ctx\_id\_t  
    voms\_apic.h, [23](#)  
  
holder  
    voms, [12](#)  
host  
    contactdata, [8](#)  
  
name  
    attribute, [7](#)  
nick  
    contactdata, [8](#)  
  
port  
    contactdata, [8](#)  
  
qualifier  
    attribute, [7](#)  
  
real  
    vomldata, [14](#)  
RECURSE\_CHAIN  
    voms\_apic.h, [19](#)  
RECURSE\_NONE  
    voms\_apic.h, [19](#)  
reserved  
    contactdata, [8](#)  
role  
    data, [10](#)  
  
serial  
    voms, [12](#)  
server  
    voms, [12](#)  
serverca  
    voms, [12](#)  
siglen  
    voms, [12](#)

signature  
    voms, 12

std  
    voms, 12

type  
    voms, 13

TYPE\_CUSTOM  
    voms\_apic.h, 19

TYPE\_NODATA  
    voms\_apic.h, 19

TYPE\_STD  
    voms\_apic.h, 19

uri  
    voms, 13

user  
    voms, 13

userca  
    voms, 13

value  
    attribute, 7

vdir  
    vomldata, 14

VERIFY\_CERTLIST  
    voms\_apic.h, 20

VERIFY\_DATE  
    voms\_apic.h, 20

VERIFY\_FULL  
    voms\_apic.h, 20

VERIFY\_ID  
    voms\_apic.h, 20

VERIFY\_KEY  
    voms\_apic.h, 20

VERIFY\_NONE  
    voms\_apic.h, 20

VERIFY\_NOTARGET  
    voms\_apic.h, 20

VERIFY\_ORDER  
    voms\_apic.h, 20

VERIFY\_SIGN  
    voms\_apic.h, 20

VERR\_COMM  
    voms\_apic.h, 20

VERR\_DIR  
    voms\_apic.h, 20

VERR\_EXTRINFO  
    voms\_apic.h, 21

VERR\_FORMAT  
    voms\_apic.h, 21

VERR\_IDCHECK  
    voms\_apic.h, 21

VERR\_MEM  
    voms\_apic.h, 21

VERR\_NODATA  
    voms\_apic.h, 21

VERR\_NOEXT  
    voms\_apic.h, 21

VERR\_NOIDENT  
    voms\_apic.h, 21

VERR\_NOINIT  
    voms\_apic.h, 21

VERR\_NONE  
    voms\_apic.h, 21

VERR\_NOSOCKET  
    voms\_apic.h, 22

VERR\_NOTAVAIL  
    voms\_apic.h, 22

VERR\_ORDER  
    voms\_apic.h, 22

VERR\_PARAM  
    voms\_apic.h, 22

VERR\_PARSE  
    voms\_apic.h, 22

VERR\_SERVER  
    voms\_apic.h, 22

VERR\_SERVERCODE  
    voms\_apic.h, 22

VERR\_SIGN  
    voms\_apic.h, 22

VERR\_TIME  
    voms\_apic.h, 22

VERR\_TYPE  
    voms\_apic.h, 23

VERR\_VERIFY  
    voms\_apic.h, 23

version  
    contactdata, 8  
    voms, 13

vo  
    contactdata, 9

volen  
    vomldata, 15

voms, 11  
    ac, 11  
    custom, 11  
    datalen, 11  
    date1, 11  
    date2, 12  
    fqan, 12  
    holder, 12  
    serial, 12  
    server, 12  
    serverca, 12  
    siglen, 12  
    signature, 12  
    std, 12

- type, 13
- uri, 13
- user, 13
- userca, 13
- version, 13
- voname, 13
- VOMS\_AddTarget
  - voms\_apic.h, 23
- voms\_apic.h, 17
  - getMajorVersionNumber, 23
  - getMinorVersionNumber, 23
  - getPatchVersionNumber, 23
  - gss\_cred\_id\_t, 23
  - gss\_ctx\_id\_t, 23
  - RECURSE\_CHAIN, 19
  - RECURSE\_NONE, 19
  - TYPE\_CUSTOM, 19
  - TYPE\_NODATA, 19
  - TYPE\_STD, 19
  - VERIFY\_CERTLIST, 20
  - VERIFY\_DATE, 20
  - VERIFY\_FULL, 20
  - VERIFY\_ID, 20
  - VERIFY\_KEY, 20
  - VERIFY\_NONE, 20
  - VERIFY\_NOTARGET, 20
  - VERIFY\_ORDER, 20
  - VERIFY\_SIGN, 20
  - VERR\_COMM, 20
  - VERR\_DIR, 20
  - VERR\_EXTRAINFO, 21
  - VERR\_FORMAT, 21
  - VERR\_IDCHECK, 21
  - VERR\_MEM, 21
  - VERR\_NODATA, 21
  - VERR\_NOEXT, 21
  - VERR\_NOIDENT, 21
  - VERR\_NOINIT, 21
  - VERR\_NONE, 21
  - VERR\_NOSOCKET, 22
  - VERR\_NOTAVAIL, 22
  - VERR\_ORDER, 22
  - VERR\_PARAM, 22
  - VERR\_PARSE, 22
  - VERR\_SERVER, 22
  - VERR\_SERVERCODE, 22
  - VERR\_SIGN, 22
  - VERR\_TIME, 22
  - VERR\_TYPE, 23
  - VERR\_VERIFY, 23
  - VOMS\_AddTarget, 23
  - VOMS\_Contact, 23
  - VOMS\_ContactRaw, 24
  - VOMS\_Copy, 24
  - VOMS\_CopyAll, 24
  - VOMS\_DefaultData, 25
  - VOMS\_Delete, 25
  - VOMS\_DeleteContacts, 25
  - VOMS\_Destroy, 25
  - VOMS\_Duplicate, 25
  - VOMS\_ErrorMessage, 26
  - VOMS\_Export, 26
  - VOMS\_FindByAlias, 26
  - VOMS\_FindByVO, 26
  - VOMS\_FreeTargets, 27
  - VOMS\_FreeTargetsList, 27
  - VOMS\_GetAC, 28
  - VOMS\_GetAttribute, 28
  - VOMS\_GetAttributeGrantor, 28
  - VOMS\_GetAttributesNumber, 28
  - VOMS\_GetAttributeSourceHandle, 28
  - VOMS\_GetAttributeSourcesNumber, 28
  - VOMS\_GetTargetsList, 28
  - VOMS\_Import, 28
  - VOMS\_Init, 28
  - VOMS\_ListTargets, 28
  - VOMS\_Ordering, 29
  - VOMS\_ResetOrder, 29
  - VOMS\_Retrieve, 29
  - VOMS\_RetrieveEXT, 29
  - VOMS\_RetrieveFromCred, 30
  - VOMS\_RetrieveFromCtx, 30
  - VOMS\_RetrieveFromFile, 30
  - VOMS\_RetrieveFromProxy, 31
  - VOMS\_SetLifetime, 31
  - VOMS\_SetVerificationTime, 31
  - VOMS\_SetVerificationType, 31
  - VOMS\_Contact
    - voms\_apic.h, 23
  - VOMS\_ContactRaw
    - voms\_apic.h, 24
  - VOMS\_Copy
    - voms\_apic.h, 24
  - VOMS\_CopyAll
    - voms\_apic.h, 24
  - VOMS\_DefaultData
    - voms\_apic.h, 25
  - VOMS\_Delete
    - voms\_apic.h, 25
  - VOMS\_DeleteContacts
    - voms\_apic.h, 25
  - VOMS\_Destroy
    - voms\_apic.h, 25
  - VOMS\_Duplicate
    - voms\_apic.h, 25
  - VOMS\_ErrorMessage
    - voms\_apic.h, 26
  - VOMS\_Export

- voms\_apic.h, [26](#)
- VOMS\_FindByAlias
  - voms\_apic.h, [26](#)
- VOMS\_FindByVO
  - voms\_apic.h, [26](#)
- VOMS\_FreeTargets
  - voms\_apic.h, [27](#)
- VOMS\_FreeTargetsList
  - voms\_apic.h, [27](#)
- VOMS\_GetAC
  - voms\_apic.h, [28](#)
- VOMS\_GetAttribute
  - voms\_apic.h, [28](#)
- VOMS\_GetAttributeGrantor
  - voms\_apic.h, [28](#)
- VOMS\_GetAttributesNumber
  - voms\_apic.h, [28](#)
- VOMS\_GetAttributeSourceHandle
  - voms\_apic.h, [28](#)
- VOMS\_GetAttributeSourcesNumber
  - voms\_apic.h, [28](#)
- VOMS\_GetTargetsList
  - voms\_apic.h, [28](#)
- VOMS\_Import
  - voms\_apic.h, [28](#)
- VOMS\_Init
  - voms\_apic.h, [28](#)
- VOMS\_ListTargets
  - voms\_apic.h, [28](#)
- VOMS\_Ordering
  - voms\_apic.h, [29](#)
- VOMS\_ResetOrder
  - voms\_apic.h, [29](#)
- VOMS\_Retrieve
  - voms\_apic.h, [29](#)
- VOMS\_RetrieveEXT
  - voms\_apic.h, [29](#)
- VOMS\_RetrieveFromCred
  - voms\_apic.h, [30](#)
- VOMS\_RetrieveFromCtx
  - voms\_apic.h, [30](#)
- VOMS\_RetrieveFromFile
  - voms\_apic.h, [30](#)
- VOMS\_RetrieveFromProxy
  - voms\_apic.h, [31](#)
- VOMS\_SetLifetime
  - voms\_apic.h, [31](#)
- VOMS\_SetVerificationTime
  - voms\_apic.h, [31](#)
- VOMS\_SetVerificationType
  - voms\_apic.h, [31](#)
- vomsdata, [14](#)
  - cdir, [14](#)
  - data, [14](#)
  - extra\_data, [14](#)
  - extralen, [14](#)
  - real, [14](#)
  - vdir, [14](#)
  - volen, [15](#)
  - workvo, [15](#)
- voname
  - voms, [13](#)
- workvo
  - vomsdata, [15](#)