

# VOMS C API

1.5.0

Generated by Doxygen 1.7.5

Sun Oct 23 2011 02:02:58



# 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	Field Documentation . . . . .	5
3.1.1.1	name . . . . .	5
3.1.1.2	qualifier . . . . .	5
3.1.1.3	value . . . . .	5
3.2	contactdata Struct Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Field Documentation . . . . .	6
3.2.2.1	contact . . . . .	6
3.2.2.2	host . . . . .	6
3.2.2.3	nick . . . . .	6
3.2.2.4	port . . . . .	6
3.2.2.5	reserved . . . . .	7
3.2.2.6	version . . . . .	7
3.2.2.7	vo . . . . .	7
3.3	data Struct Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Field Documentation . . . . .	7
3.3.2.1	cap . . . . .	7

3.3.2.2	group	8
3.3.2.3	role	8
3.4	voms Struct Reference	8
3.4.1	Field Documentation	8
3.4.1.1	ac	8
3.4.1.2	custom	9
3.4.1.3	datalen	9
3.4.1.4	date1	9
3.4.1.5	date2	9
3.4.1.6	fqan	9
3.4.1.7	holder	9
3.4.1.8	serial	9
3.4.1.9	server	9
3.4.1.10	serverca	10
3.4.1.11	siglen	10
3.4.1.12	signature	10
3.4.1.13	std	10
3.4.1.14	type	10
3.4.1.15	uri	10
3.4.1.16	user	10
3.4.1.17	userca	10
3.4.1.18	version	11
3.4.1.19	voname	11
3.5	vomsdata Struct Reference	11
3.5.1	Field Documentation	11
3.5.1.1	cdir	11
3.5.1.2	data	11
3.5.1.3	extra_data	12
3.5.1.4	extralen	12
3.5.1.5	real	12
3.5.1.6	vdir	12
3.5.1.7	volen	12
3.5.1.8	workvo	12

<b>4</b>	<b>File Documentation</b>	<b>13</b>
4.1	voms_apic.h File Reference	13
4.1.1	Define Documentation	16
4.1.1.1	NOGLOBUS	16
4.1.1.2	RECURSE_CHAIN	16
4.1.1.3	RECURSE_NONE	16
4.1.1.4	TYPE_CUSTOM	16
4.1.1.5	TYPE_NODATA	16
4.1.1.6	TYPE_STD	16
4.1.1.7	VERIFY_CERTLIST	16
4.1.1.8	VERIFY_DATE	16
4.1.1.9	VERIFY_FULL	16
4.1.1.10	VERIFY_ID	17
4.1.1.11	VERIFY_KEY	17
4.1.1.12	VERIFY_NONE	17
4.1.1.13	VERIFY_NOTARGET	17
4.1.1.14	VERIFY_ORDER	17
4.1.1.15	VERIFY_SIGN	17
4.1.1.16	VERR_COMM	17
4.1.1.17	VERR_DIR	17
4.1.1.18	VERR_EXTRAINFO	17
4.1.1.19	VERR_FORMAT	18
4.1.1.20	VERR_IDCHECK	18
4.1.1.21	VERR_MEM	18
4.1.1.22	VERR_NODATA	18
4.1.1.23	VERR_NOEXT	18
4.1.1.24	VERR_NOIDENT	18
4.1.1.25	VERR_NOINIT	18
4.1.1.26	VERR_NONE	18
4.1.1.27	VERR_NOSOCKET	19
4.1.1.28	VERR_NOTAVAIL	19
4.1.1.29	VERR_ORDER	19
4.1.1.30	VERR_PARAM	19
4.1.1.31	VERR_PARSE	19

4.1.1.32	VERR_SERVER	19
4.1.1.33	VERR_SERVERCODE	19
4.1.1.34	VERR_SIGN	19
4.1.1.35	VERR_TIME	20
4.1.1.36	VERR_TYPE	20
4.1.1.37	VERR_VERIFY	20
4.1.2	Typedef Documentation	20
4.1.2.1	gss_cred_id_t	20
4.1.2.2	gss_ctx_id_t	20
4.1.3	Function Documentation	20
4.1.3.1	getMajorVersionNumber	20
4.1.3.2	getMinorVersionNumber	20
4.1.3.3	getPatchVersionNumber	20
4.1.3.4	VOMS_AddTarget	20
4.1.3.5	VOMS_Contact	21
4.1.3.6	VOMS_ContactRaw	21
4.1.3.7	VOMS_Copy	22
4.1.3.8	VOMS_CopyAll	22
4.1.3.9	VOMS_DefaultData	22
4.1.3.10	VOMS_Delete	22
4.1.3.11	VOMS_DeleteContacts	23
4.1.3.12	VOMS_Destroy	23
4.1.3.13	VOMS_Duplicate	23
4.1.3.14	VOMS_ErrorMessage	23
4.1.3.15	VOMS_Export	23
4.1.3.16	VOMS_FindByAlias	24
4.1.3.17	VOMS_FindByVO	24
4.1.3.18	VOMS_FreeTargets	25
4.1.3.19	VOMS_FreeTargetsList	25
4.1.3.20	VOMS_GetAC	25
4.1.3.21	VOMS_GetAttribute	25
4.1.3.22	VOMS_GetAttributeGrantor	25
4.1.3.23	VOMS_GetAttributesNumber	25
4.1.3.24	VOMS_GetAttributeSourceHandle	25

4.1.3.25	VOMS_GetAttributeSourcesNumber	25
4.1.3.26	VOMS_GetTargetsList	25
4.1.3.27	VOMS_Import	25
4.1.3.28	VOMS_Init	26
4.1.3.29	VOMS_ListTargets	26
4.1.3.30	VOMS_LoadCredentials	26
4.1.3.31	VOMS_Ordering	26
4.1.3.32	VOMS_ResetOrder	26
4.1.3.33	VOMS_Retrieve	27
4.1.3.34	VOMS_RetrieveEXT	27
4.1.3.35	VOMS_RetrieveFromAC	27
4.1.3.36	VOMS_RetrieveFromCred	28
4.1.3.37	VOMS_RetrieveFromCtx	28
4.1.3.38	VOMS_RetrieveFromFile	28
4.1.3.39	VOMS_RetrieveFromProxy	29
4.1.3.40	VOMS_SetLifetime	29
4.1.3.41	VOMS_SetTimeout	29
4.1.3.42	VOMS_SetVerificationTime	29
4.1.3.43	VOMS_SetVerificationType	29





# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">attribute</a>	5
<a href="#">contactdata</a>	
The type of data returned	6
<a href="#">data</a>	
User's characteristics: can be repeated	7
<a href="#">voms</a>	8
<a href="#">vomsdata</a>	11



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">voms_apic.h</a>	13
-----------------------------	----



## Chapter 3

# Data Structure Documentation

### 3.1 attribute Struct Reference

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

#### Data Fields

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

#### 3.1.1 Field Documentation

##### 3.1.1.1 const char\* attribute::name

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

##### 3.1.1.2 const char\* attribute::qualifier

Definition at line 57 of file voms\_apic.h.

##### 3.1.1.3 const char\* attribute::value

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

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

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

## 3.2 contactdata Struct Reference

The type of data returned.

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

### Data Fields

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

### 3.2.1 Detailed Description

The type of data returned.

### 3.2.2 Field Documentation

#### 3.2.2.1 char\* contactdata::contact

The subject of the server's certificate

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

#### 3.2.2.2 char\* contactdata::host

The hostname of the server

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

#### 3.2.2.3 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 63 of file voms\_apic.h.

#### 3.2.2.4 int contactdata::port

The port on which the server is listening

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

#### 3.2.2.5 char\* contactdata::reserved

HANDS OFF!

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

#### 3.2.2.6 int contactdata::version

The version of Globus on which this server runs.

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

#### 3.2.2.7 char\* contactdata::vo

The VO served by this server

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

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

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

### 3.3 data Struct Reference

User's characteristics: can be repeated.

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

#### Data Fields

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

#### 3.3.1 Detailed Description

User's characteristics: can be repeated.

<

#### 3.3.2 Field Documentation

##### 3.3.2.1 char\* data::cap

user's capability

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

### 3.3.2.2 char\* data::group

user's group

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

### 3.3.2.3 char\* data::role

user's role

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

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

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

## 3.4 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](#)
- struct [data](#) \*\* [std](#)
- char \* [custom](#)
- int [datalen](#)
- int [version](#)
- char \*\* [fqan](#)
- char \* [serial](#)
- AC \* [ac](#)
- X509 \* [holder](#)

### 3.4.1 Field Documentation

#### 3.4.1.1 AC\* voms::ac

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



**3.4.1.2 char\* voms::custom**

The data returned by an S command

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

**3.4.1.3 int voms::datalen**

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

**3.4.1.4 char\* voms::date1**

Beginning of validity of the user info

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

**3.4.1.5 char\* voms::date2**

End of validity of the user info

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

**3.4.1.6 char\*\* voms::fqan**

User's attributes in compact format

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

**3.4.1.7 X509\* voms::holder**

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

**3.4.1.8 char\* voms::serial**

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

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

**3.4.1.9 char\* voms::server**

The VOMS server DN, as from its certificate

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

**3.4.1.10 char\* voms::serverca**

The CA which signed the VOMS certificate

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

**3.4.1.11 int voms::siglen**

The length of the VOMS server signature

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

**3.4.1.12 char\* voms::signature**

The VOMS server signature

Definition at line 82 of file voms\_apic.h.

**3.4.1.13 struct data\*\* voms::std**

User's characteristics

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

**3.4.1.14 int voms::type**

The type of data returned

Definition at line 91 of file voms\_apic.h.

**3.4.1.15 char\* voms::uri**

The URI of the VOMS server

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

**3.4.1.16 char\* voms::user**

The user's DN, as from his certificate

Definition at line 83 of file voms\_apic.h.

**3.4.1.17 char\* voms::userca**

The CA which signed the user's certificate

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

#### 3.4.1.18 int voms::version

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

#### 3.4.1.19 char\* voms::voname

The name of the VO to which the VOMS belongs

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

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

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

## 3.5 vomsdata Struct Reference

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

### Data Fields

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

### 3.5.1 Field Documentation

#### 3.5.1.1 char\* vomsdata::cdir

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

#### 3.5.1.2 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 147 of file voms\_apic.h.

### 3.5.1.3 `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 152 of file `voms_apic.h`.

### 3.5.1.4 `int vomsdata::extralen`

Definition at line 162 of file `voms_apic.h`.

### 3.5.1.5 `struct vomsdata* vomsdata::real`

Definition at line 164 of file `voms_apic.h`.

### 3.5.1.6 `char* vomsdata::vdir`

Definition at line 146 of file `voms_apic.h`.

### 3.5.1.7 `int vomsdata::volen`

Definition at line 161 of file `voms_apic.h`.

### 3.5.1.8 `char* vomsdata::workvo`

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

Definition at line 150 of file `voms_apic.h`.

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

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

## Chapter 4

# File Documentation

### 4.1 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](#)  
*The type of data returned.*
- struct [voms](#)
- struct [vomldata](#)

#### Defines

- #define [NOGLOBUS](#)
- #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

- struct [contactdata](#) \*\* [VOMS\\_FindByAlias](#) (struct [vomsdata](#) \*vd, char \*alias, char \*system, char \*user, int \*error)
- struct [contactdata](#) \*\* [VOMS\\_FindByVO](#) (struct [vomsdata](#) \*vd, char \*vo, char \*system, char \*user, int \*error)
- void [VOMS\\_DeleteContacts](#) (struct [contactdata](#) \*\*list)
- struct [vomsdata](#) \* [VOMS\\_Init](#) (char \*voms, char \*cert)
- struct [voms](#) \* [VOMS\\_Copy](#) (struct [voms](#) \*v, int \*error)
- struct [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)
- struct [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\\_RetrieveFromAC](#) (AC \*ac, 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)
- struct [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 \*\*)
- int [VOMS\\_SetTimeout](#) (int t, struct [vomsdata](#) \*vd, int \*error)
- int [VOMS\\_LoadCredentials](#) (X509 \*cert, EVP\_PKEY \*pkey, STACK\_OF(X509)\*chain, struct [vomsdata](#) \*vd, int \*error)

### 4.1.1 Define Documentation

#### 4.1.1.1 #define NOGLOBUS

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

#### 4.1.1.2 #define RECURSE\_CHAIN 0

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

#### 4.1.1.3 #define RECURSE\_NONE 1

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

#### 4.1.1.4 #define TYPE\_CUSTOM 2

result of an S command

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

#### 4.1.1.5 #define TYPE\_NODATA 0

no data

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

#### 4.1.1.6 #define TYPE\_STD 1

group, role, capability triplet

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

#### 4.1.1.7 #define VERIFY\_CERTLIST 0x00000040

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

#### 4.1.1.8 #define VERIFY\_DATE 0x00000001

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

#### 4.1.1.9 #define VERIFY\_FULL 0xffffffff

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



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

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

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

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

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

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

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

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

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

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

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

Definition at line 113 of file voms\_apic.h.

**4.1.1.16 #define VERR\_COMM 3**

Server problem

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

**4.1.1.17 #define VERR\_DIR 13**

Directory error

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

**4.1.1.18 #define VERR\_EXTRAINFO 9**

VO name and URI missing

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

**4.1.1.19 #define VERR\_FORMAT 10**

Wrong data format

Definition at line 131 of file voms\_apic.h.

**4.1.1.20 #define VERR\_IDCHECK 8**

User data in extension different from the real ones

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

**4.1.1.21 #define VERR\_MEM 16**

Memory problems

Definition at line 137 of file voms\_apic.h.

**4.1.1.22 #define VERR\_NODATA 11**

Empty extension

Definition at line 132 of file voms\_apic.h.

**4.1.1.23 #define VERR\_NOEXT 5**

VOMS extension missing

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

**4.1.1.24 #define VERR\_NOIDENT 2**

Cannot identify itself (certificate problem)

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

**4.1.1.25 #define VERR\_NOINIT 6**

Initialization error

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

**4.1.1.26 #define VERR\_NONE 0**

Error codes.

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

**4.1.1.27 #define VERR\_NOSOCKET 1**

Socket problem

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

**4.1.1.28 #define VERR\_NOTAVAIL 21**

Method not available

Definition at line 142 of file voms\_apic.h.

**4.1.1.29 #define VERR\_ORDER 19**

Ordering different than required

Definition at line 140 of file voms\_apic.h.

**4.1.1.30 #define VERR\_PARAM 4**

Wrong parameters

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

**4.1.1.31 #define VERR\_PARSE 12**

Parse error

Definition at line 133 of file voms\_apic.h.

**4.1.1.32 #define VERR\_SERVER 15**

Unidentifiable VOMS server

Definition at line 136 of file voms\_apic.h.

**4.1.1.33 #define VERR\_SERVERCODE 20**

Error from the server

Definition at line 141 of file voms\_apic.h.

**4.1.1.34 #define VERR\_SIGN 14**

Signature error

Definition at line 135 of file voms\_apic.h.

**4.1.1.35 #define VERR\_TIME 7**

Error in time checking

Definition at line 127 of file voms\_apic.h.

**4.1.1.36 #define VERR\_TYPE 18**

Returned data of unknown type

Definition at line 139 of file voms\_apic.h.

**4.1.1.37 #define VERR\_VERIFY 17**

Generic verification error

Definition at line 138 of file voms\_apic.h.

**4.1.2 Typedef Documentation****4.1.2.1 typedef void\* gss\_cred\_id\_t**

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

**4.1.2.2 typedef void\* gss\_ctx\_id\_t**

Definition at line 36 of file voms\_apic.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 )****4.1.3.4 int VOMS\_AddTarget ( struct vomsdata \* *vd*, char \* *target*, int \* *error* )**

Adds a target to the AC.

**Parameters**

<i>vd</i>	The vomsdata structure.
<i>target</i>	The target to add. It should be a FQDN.
<i>error</i>	RETURN PARAMETER: qualifies the error message.

**Returns**

failure (0) or success (<>0)

**4.1.3.5** `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**

<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>	Command
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

**Returns**

failure (0) or success (<>0)

**4.1.3.6** `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\_Contract, except that instead of starting the verification process, the data is returned as is in the

**Parameters**

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

**Returns**

failure (0) or success (<>0)

#### 4.1.3.7 `struct voms* VOMS_Copy ( struct voms * v, int * error )` [read]

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

<i>v</i>	The structure to copy.
<i>error</i>	RETURN PARAMETER: qualifies the error message.

##### Returns

NULL (error) or the new voms structure.

#### 4.1.3.8 `struct vomsdata* VOMS_CopyAll ( struct vomsdata * vd, int * error )` [read]

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

<i>vd</i>	The structure to copy.
<i>error</i>	RETURN PARAMETER: qualifies the error message.

##### Returns

NULL (error) or the new vomsdata structure.

#### 4.1.3.9 `struct voms* VOMS_DefaultData ( struct vomsdata * vd, int * error )` [read]

Gets the default attributes from a vomsdata structure.

##### Parameters

<i>vd</i>	the vomsdata structure to analyze
<i>error</i>	RETURN PARAMETER: Qualifies the error message

##### Returns

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

#### 4.1.3.10 `void VOMS_Delete ( struct voms * v )`

Deletes a voms structure

## Parameters

<i>v</i>	Pointer to the structure to delete.
----------	-------------------------------------

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

Frees a contactdata vector.

## Parameters

<i>list</i>	The vector to free.
-------------	---------------------

## Returns

NONE

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

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

4.1.3.13 struct vomsdata\* VOMS\_Duplicate ( struct vomsdata \* *vd* ) [read]4.1.3.14 char\* VOMS\_ErrorMessage ( struct vomsdata \* *vd*, int *error*, char \* *buffer*, int *len* )

Gets a textual description of the error.

## Parameters

<i>vd</i>	The vomsdata structure to analyze
<i>error</i>	The error returned by the last function
<i>buffer</i>	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.
<i>len</i>	The length of the memory pointed to by the buffer parameter.

## Returns

A pointer to the error message.

4.1.3.15 int VOMS\_Export ( char \*\* *buffer*, int \* *buflen*, struct vomsdata \* *vd*, int \* *error* )

Converts data into a test format

## Parameters

<i>buffer</i>	OUTPUT PARAMETER contains the converted data
<i>buflen</i>	OUTPUT PARAMETER contains the length of buffer
<i>vd</i>	contains the data to convert
<i>error</i>	RETURN PARAMETER Qualifies the error message

## Returns

failure (0) or success (>0)

**4.1.3.16** `struct contactdata** VOMS_FindByAlias ( struct vomsdata * vd, char * alias, char * system, char * user, int * error ) [read]`

Gets a list of VOMS servers which share an alias.

## Parameters

<i>vd</i>	The correctly initialized vomsdata structured.
<i>alias</i>	The alias to look for.
<i>system</i>	The directory in which to look for the system configuration files. If NULL, defaults to /opt/edc/etc/vomses
<i>user</i>	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.
<i>error</i>	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.

**4.1.3.17** `struct contactdata** VOMS_FindByVO ( struct vomsdata * vd, char * vo, char * system, char * user, int * error ) [read]`

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

## Parameters

<i>vd</i>	The correctly initialized vomsdata structured.
<i>vo</i>	The VO to look for.
<i>system</i>	The directory in which to look for the system configuration files. If NULL, defaults to /opt/edc/etc/vomses
<i>user</i>	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.
<i>error</i>	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.

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

Delete the targets from the AC.

### Parameters

<i>vd</i>	The vomsdata structure.
<i>error</i>	RETURN PARAMETER: qualifies the error message.

**4.1.3.19** void VOMS\_FreeTargetsList ( char \*\* )

**4.1.3.20** AC\* VOMS\_GetAC ( struct voms \* *v* )

**4.1.3.21** int VOMS\_GetAttribute ( struct voms \* *v*, int *handle*, int *num*, struct attribute \* *at*, struct vomsdata \* *vd*, int \* *error* )

**4.1.3.22** const char\* VOMS\_GetAttributeGrantor ( struct voms \* *v*, int *handle*, struct vomsdata \* *vd*, int \* *error* )

**4.1.3.23** int VOMS\_GetAttributesNumber ( struct voms \* *v*, int *handle*, struct vomsdata \* *vd*, int \* *error* )

**4.1.3.24** int VOMS\_GetAttributeSourceHandle ( struct voms \* *v*, int *num*, struct vomsdata \* *vd*, int \* *error* )

**4.1.3.25** int VOMS\_GetAttributeSourcesNumber ( struct voms \* *v*, struct vomsdata \* *vd*, int \* *error* )

**4.1.3.26** char\*\* VOMS\_GetTargetsList ( struct voms \* *v*, struct vomsdata \* *vd*, int \* *error* )

**4.1.3.27** 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

<i>buffer</i>	contains the data to be converted
<i>buflen</i>	contains the length of buffer
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

**Returns**

failure (0) or success (<>0)

#### 4.1.3.28 struct vomsdata\* VOMS.Init ( char \* *voms*, char \* *cert* ) [read]

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**

<i>voms</i>	The directory which contains the certificates of the VOMS servers
<i>cert</i>	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.

#### 4.1.3.29 char\* VOMS.ListTargets ( struct vomsdata \* *vd*, int \* *error* )

#### 4.1.3.30 int VOMS.LoadCredentials ( X509 \* *cert*, EVP\_PKEY \* *pkey*, STACK\_OF(X509)\* *chain*, struct vomsdata \* *vd*, int \* *error* )

#### 4.1.3.31 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**

<i>order</i>	the group:role attribute.
<i>vd</i>	RETURN PARAMETER: contains the modified data.
<i>error</i>	RETURN PARAMETER: Qualifies the error message

**Returns**

failure (0) or success (<>0)

#### 4.1.3.32 int VOMS.ResetOrder ( struct vomsdata \* *vd*, int \* *error* )

Unsets the return order of the attributes.

## Parameters

<i>vd</i>	RETURN PARAMETER: contains the modified data.
<i>error</i>	RETURN PARAMETER: Qualifies the error message

## Returns

failure (0) or success (<>0)

**4.1.3.33** 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

<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
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

## Returns

failure (0) or success (<>0)

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

Gets VOMS information from the given extension

## Parameters

<i>ext</i>	The extension to parse.
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

## Returns

failure (0) or success (<>0)

**4.1.3.35** int VOMS\_RetrieveFromAC ( AC \* *ac*, struct vomsdata \* *vd*, int \* *error* )

Gets VOMS information from an existing globus proxy

**Parameters**

<i>ac</i>	AC from which to get the credentials
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

**Returns**

failure (0) or success (<>0)

**4.1.3.36** `int VOMS.RetrieveFromCred ( gss_cred_id_t cred, int how, struct vomsdata * vd, int * error )`

Gets VOMS information from the given globus credential

**Parameters**

<i>cred</i>	The credential from which to retrieve the certificate.
<i>how</i>	Recursion type
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

**Returns**

failure (0) or success (<>0)

**4.1.3.37** `int VOMS.RetrieveFromCtx ( gss_ctx_id_t ctx, int how, struct vomsdata * vd, int * error )`

Gets VOMS information from the given globus context

**Parameters**

<i>ctx</i>	The context from which to retrieve the certificate.
<i>how</i>	Recursion type
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

**Returns**

failure (0) or success (<>0)

**4.1.3.38** `int VOMS.RetrieveFromFile ( FILE * file, int how, struct vomsdata * vd, int * error )`

Gets VOMS information from the given globus credential

## Parameters

<i>file</i>	The file from which to retrieve the certificate.
<i>how</i>	Recursion type
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

## Returns

failure (0) or success (<>0)

4.1.3.39 int VOMS\_RetrieveFromProxy ( int *how*, struct vomsdata \* *vd*, int \* *error* )

Gets VOMS information from an existing globus proxy

## Parameters

<i>how</i>	Recursion type
<i>vd</i>	RETURN PARAMETER: contains the data returned by the connection
<i>error</i>	RETURN PARAMETER: Qualifies the error message

## Returns

failure (0) or success (<>0)

4.1.3.40 int VOMS\_SetLifetime ( int *length*, struct vomsdata \* *vd*, int \* *error* )

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

## Parameters

<i>length</i>	Lifetime requested.
<i>vd</i>	RETURN PARAMETER: contains the modified data.
<i>error</i>	RETURN PARAMETER: Qualifies the error message

## Returns

failure (0) or success (<>0)

4.1.3.41 int VOMS\_SetTimeout ( int *t*, struct vomsdata \* *vd*, int \* *error* )4.1.3.42 int VOMS\_SetVerificationTime ( time\_t *verificationtime*, struct vomsdata \* *vd*, int \* *error* )4.1.3.43 int VOMS\_SetVerificationType ( int *type*, struct vomsdata \* *vd*, int \* *error* )