

VOMS C API Reference Manual

1.5.0

Generated by Doxygen 1.2.14

Thu Jan 29 11:36:24 2004

Contents

1	VOMS C API Data Structure Index	1
1.1	VOMS C API Data Structures	1
2	VOMS C API File Index	3
2.1	VOMS C API File List	3
3	VOMS C API Data Structure Documentation	5
3.1	contactdata Struct Reference	5
3.2	data Struct Reference	7
3.3	voms Struct Reference	8
3.4	vomsdata Struct Reference	11
4	VOMS C API File Documentation	13
4.1	voms_apic.h File Reference	13

Chapter 1

VOMS C API Data Structure Index

1.1 VOMS C API Data Structures

Here are the data structures with brief descriptions:

contactdata	5
data (User's characteristics: can be repeated)	7
voms	8
vomsdata	11

Chapter 2

VOMS C API File Index

2.1 VOMS C API File List

Here is a list of all files with brief descriptions:

voms_apic.h	13
---------------------------------------	----

Chapter 3

VOMS C API Data Structure Documentation

3.1 contactdata Struct Reference

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

Data Fields

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

3.1.1 Field Documentation

3.1.1.1 char* contactdata::contact

The subject of the server's certificate
Definition at line 37 of file voms_apic.h.

3.1.1.2 char* contactdata::host

The hostname of the server
Definition at line 36 of file voms_apic.h.

3.1.1.3 char* contactdata::nick

The alias of the server
Definition at line 35 of file voms_apic.h.

3.1.1.4 int contactdata::port

The port on which the server is listening

Definition at line 39 of file voms_apic.h.

3.1.1.5 char* contactdata::reserved

HANDS OFF!

Definition at line 40 of file voms_apic.h.

3.1.1.6 char* contactdata::vo

The VO served by this server

Definition at line 38 of file voms_apic.h.

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

- [voms_apic.h](#)

3.2 data Struct Reference

User's characteristics: can be repeated.

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

Data Fields

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

3.2.1 Detailed Description

User's characteristics: can be repeated.

Definition at line 25 of file voms_apic.h.

3.2.2 Field Documentation

3.2.2.1 char* data::cap

user's capability

Definition at line 28 of file voms_apic.h.

3.2.2.2 char* data::group

user's group

Definition at line 26 of file voms_apic.h.

3.2.2.3 char* data::role

user's role

Definition at line 27 of file voms_apic.h.

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

- [voms_apic.h](#)

3.3 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](#)

3.3.1 Field Documentation

3.3.1.1 char* voms::custom

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

Definition at line 64 of file voms_apic.h.

3.3.1.2 int voms::datalen

Definition at line 65 of file voms_apic.h.

3.3.1.3 char* voms::date1

Beginning of validity of the user info

Definition at line 60 of file voms_apic.h.

3.3.1.4 char* voms::date2

End of validity of the user info

Definition at line 61 of file voms_apic.h.

3.3.1.5 char voms::fqan**

User's attributes in compact format

Definition at line 67 of file voms_apic.h.

3.3.1.6 char* voms::serial

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

Definition at line 68 of file voms_apic.h.

3.3.1.7 char* voms::server

The VOMS server DN, as from its certificate

Definition at line 56 of file voms_apic.h.

3.3.1.8 char* voms::serverca

The CA which signed the VOMS certificate

Definition at line 57 of file voms_apic.h.

3.3.1.9 int voms::siglen

The length of the VOMS server signature

Definition at line 52 of file voms_apic.h.

3.3.1.10 char* voms::signature

The VOMS server signature

Definition at line 53 of file voms_apic.h.

3.3.1.11 struct data voms::std**

User's characteristics

Definition at line 63 of file voms_apic.h.

3.3.1.12 int voms::type

The type of [data](#) returned

Definition at line 62 of file voms_apic.h.

3.3.1.13 char* voms::uri

The URI of the VOMS server

Definition at line 59 of file voms_apic.h.

3.3.1.14 char* voms::user

The user's DN, as from his certificate

Definition at line 54 of file voms_apic.h.

3.3.1.15 char* voms::userca

The CA which signed the user's certificate

Definition at line 55 of file voms_apic.h.

3.3.1.16 int voms::version

Definition at line 66 of file voms_apic.h.

3.3.1.17 char* voms::voname

The name of the VO to which the VOMS belongs

Definition at line 58 of file voms_apic.h.

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

- [voms_apic.h](#)

3.4 vomsdata Struct Reference

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

Data Fields

- char * [cdir](#)
- char * [vdir](#)
- voms ** [data](#)
- char * [workvo](#)
- char * [extra_data](#)
- int [volen](#)
- int [extralen](#)

3.4.1 Field Documentation

3.4.1.1 char* vomsdata::cdir

Definition at line 113 of file voms_apic.h.

3.4.1.2 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 115 of file voms_apic.h.

3.4.1.3 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 120 of file voms_apic.h.

3.4.1.4 int vomsdata::extralen

Definition at line 130 of file voms_apic.h.

3.4.1.5 char* vomsdata::vdir

Definition at line 114 of file voms_apic.h.

3.4.1.6 int vomsdata::volen

Definition at line 129 of file voms_apic.h.

3.4.1.7 `char* vomsdata::workvo`

The value of the -vo option of the [voms-proxy-init](#) command

Definition at line 118 of file [voms_apic.h](#).

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

- [voms_apic.h](#)

Chapter 4

VOMS C API File Documentation

4.1 voms_apic.h File Reference

```
#include <openssl/x509.h>
```

Data Structures

- struct [contactdata](#)
- struct [data](#)

User's characteristics: can be repeated.

- struct [voms](#)
- struct [vomsdata](#)

Defines

- #define [TYPE_NODATA](#) 0 /*!< no [data](#) */
The type of [data](#) returned.
- #define [TYPE_STD](#) 1 /*!< group, role, capability triplet */
- #define [TYPE_CUSTOM](#) 2 /*!< result of an S command */
- #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 [VERR_NONE](#) 0

Error codes.

- #define `VERR_NOSOCKET` 1 /*!< Socket problem*/
- #define `VERR_NOIDENT` 2 /*!< Cannot identify itself (certificate problem) */
- #define `VERR_COMM` 3 /*!< Server problem */
- #define `VERR_PARAM` 4 /*!< Wrong parameters*/
- #define `VERR_NOEXT` 5 /*!< VOMS extension missing */
- #define `VERR_NOINIT` 6 /*!< Initialization error */
- #define `VERR_TIME` 7 /*!< Error in time checking */
- #define `VERR_IDCHECK` 8 /*!< User `data` in extension different from the real
- #define `VERR_EXTRAINFO` 9 /*!< VO name and URI missing */
- #define `VERR_FORMAT` 10 /*!< Wrong `data` format */
- #define `VERR_NODATA` 11 /*!< Empty extension */
- #define `VERR_PARSE` 12 /*!< Parse error */
- #define `VERR_DIR` 13 /*!< Directory error */
- #define `VERR_SIGN` 14 /*!< Signature error */
- #define `VERR_SERVER` 15 /*!< Unidentifiable VOMS server */
- #define `VERR_MEM` 16 /*!< Memory problems */
- #define `VERR_VERIFY` 17 /*!< Generic verification error*/
- #define `VERR_TYPE` 18 /*!< Returned `data` of unknown type */
- #define `VERR_ORDER` 19 /*!< Ordering different than required */
- #define `VERR_SERVERCODE` 20 /*!< Error from the server */

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)
- int `getMajorVersionNumber` (void)
- int `getMinorVersionNumber` (void)
- int `getPatchVersionNumber` (void)

4.1.1 Define Documentation

4.1.1.1 #define RECURSE_CHAIN 0

Definition at line 74 of file voms_apic.h.

4.1.1.2 #define RECURSE_NONE 1

Definition at line 75 of file voms_apic.h.

4.1.1.3 #define TYPE_CUSTOM 2 /*!< result of an S command */

Definition at line 48 of file voms_apic.h.

4.1.1.4 #define TYPE_NODATA 0 /*!< no data */

The type of [data](#) returned.

Definition at line 46 of file voms_apic.h.

4.1.1.5 #define TYPE_STD 1 /*!< group, role, capability triplet */

Definition at line 47 of file voms_apic.h.

4.1.1.6 #define VERIFY_DATE 0x00000001

Definition at line 79 of file voms_apic.h.

4.1.1.7 #define VERIFY_FULL 0xffffffff

Definition at line 77 of file voms_apic.h.

4.1.1.8 #define VERIFY_ID 0x00000020

Definition at line 84 of file voms_apic.h.

4.1.1.9 #define VERIFY_KEY 0x00000004

Definition at line 81 of file voms_apic.h.

4.1.1.10 #define VERIFY_NONE 0x00000000

Definition at line 78 of file voms_apic.h.

4.1.1.11 #define VERIFY_NOTARGET 0x00000002

Definition at line 80 of file voms_apic.h.

4.1.1.12 #define VERIFY_ORDER 0x00000010

Definition at line 83 of file voms_apic.h.

4.1.1.13 #define VERIFY_SIGN 0x00000008

Definition at line 82 of file voms_apic.h.

4.1.1.14 #define VERR_COMM 3 /*!< Server problem */

Definition at line 92 of file voms_apic.h.

4.1.1.15 #define VERR_DIR 13 /*!< Directory error */

Definition at line 103 of file voms_apic.h.

4.1.1.16 #define VERR_EXTRAINFO 9 /*!< VO name and URI missing */

Definition at line 99 of file voms_apic.h.

4.1.1.17 #define VERR_FORMAT 10 /*!< Wrong data format */

Definition at line 100 of file voms_apic.h.

4.1.1.18 #define VERR_IDCHECK 8 /*!< User data in extension different from the real

Definition at line 97 of file voms_apic.h.

4.1.1.19 #define VERR_MEM 16 /*!< Memory problems */

Definition at line 106 of file voms_apic.h.

4.1.1.20 #define VERR_NODATA 11 /*!< Empty extension */

Definition at line 101 of file voms_apic.h.

4.1.1.21 #define VERR_NOEXT 5 /*!< VOMS extension missing */

Definition at line 94 of file voms_apic.h.

4.1.1.22 #define VERR_NOIDENT 2 /*!< Cannot identify itself (certificate problem) */

Definition at line 91 of file voms_apic.h.

4.1.1.23 #define VERR_NOINIT 6 /*!< Initialization error */

Definition at line 95 of file voms_apic.h.

4.1.1.24 #define VERR_NONE 0

Error codes.

Definition at line 89 of file voms_apic.h.

4.1.1.25 #define VERR_NOSOCKET 1 /*!< Socket problem*/

Definition at line 90 of file voms_apic.h.

4.1.1.26 #define VERR_ORDER 19 /*!< Ordering different than required */

Definition at line 109 of file voms_apic.h.

4.1.1.27 #define VERR_PARAM 4 /*!< Wrong parameters*/

Definition at line 93 of file voms_apic.h.

4.1.1.28 #define VERR_PARSE 12 /*!< Parse error */

Definition at line 102 of file voms_apic.h.

4.1.1.29 #define VERR_SERVER 15 /*!< Unidentifiable VOMS server */

Definition at line 105 of file voms_apic.h.

4.1.1.30 #define VERR_SERVERCODE 20 /*!< Error from the server */

Definition at line 110 of file voms_apic.h.

4.1.1.31 #define VERR_SIGN 14 /*!< Signature error */

Definition at line 104 of file voms_apic.h.

4.1.1.32 #define VERR_TIME 7 /*!< Error in time checking */

Definition at line 96 of file voms_apic.h.

4.1.1.33 #define VERR_TYPE 18 /*!< Returned [data](#) of unknown type */

Definition at line 108 of file voms_apic.h.

4.1.1.34 `#define VERR_VERIFY 17 /*!< Generic verification error*/`

Definition at line 107 of file voms_apic.h.

4.1.2 Function Documentation

4.1.2.1 `int getMajorVersionNumber (void)`

4.1.2.2 `int getMinorVersionNumber (void)`

4.1.2.3 `int getPatchVersionNumber (void)`

4.1.2.4 `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)

4.1.2.5 `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)

4.1.2.6 `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.

version is the version number of the *data*.

4.1.2.7 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.

Parameters:

v The structure to copy.

error RETURN PARAMETER: qualifies the error message.

Returns:

NULL (error) or the new *voms* structure.

4.1.2.8 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.

Parameters:

vd The structure to copy.

error RETURN PARAMETER: qualifies the error message.

Returns:

NULL (error) or the new *vomsdata* structure.

4.1.2.9 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.

4.1.2.10 void VOMS_Delete (struct voms * v)

Deletes a *voms* structure

Parameters:

v Pointer to the structure to delete.

4.1.2.11 void VOMS_DeleteContacts (struct [contactdata](#) ** *list*)

Frees a [contactdata](#) vector.

Parameters:

list The vector to free.

Returns:

NONE

4.1.2.12 void VOMS_Destroy (struct [vomsdata](#) * *vd*)

Destroys a proper [vomsdata](#) structure /param *vd* The structure to deallocate.

4.1.2.13 int VOMS_Export (char ** *buffer*, int * *buflen*, struct [vomsdata](#) * *vd*, int * *error*)**4.1.2.14 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.

4.1.2.15 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.

4.1.2.16 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.

4.1.2.17 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)

4.1.2.18 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.

4.1.2.19 char* VOMS_ListTargets (struct vomsdata * vd, int * error)**4.1.2.20 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)

4.1.2.21 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)

4.1.2.22 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)

4.1.2.23 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)

4.1.2.24 int VOMS_SetVerificationType (int *type*, struct [vomsdata](#) * *vd*, int * *error*)

Index

cap
 data, [7](#)

cdir
 vomsdata, [11](#)

contact
 contactdata, [5](#)

contactdata, [5](#)
 contact, [5](#)
 host, [5](#)
 nick, [5](#)
 port, [5](#)
 reserved, [6](#)
 vo, [6](#)

custom
 voms, [8](#)

data, [7](#)
 cap, [7](#)
 group, [7](#)
 role, [7](#)
 vomsdata, [11](#)

datalen
 voms, [8](#)

date1
 voms, [8](#)

date2
 voms, [8](#)

extra_data
 vomsdata, [11](#)

extralen
 vomsdata, [11](#)

fqan
 voms, [8](#)

getMajorVersionNumber
 voms_apic.h, [18](#)

getMinorVersionNumber
 voms_apic.h, [18](#)

getPatchVersionNumber
 voms_apic.h, [18](#)

group
 data, [7](#)

host
 contactdata, [5](#)

nick
 contactdata, [5](#)

port
 contactdata, [5](#)

RECURSE_CHAIN
 voms_apic.h, [15](#)

RECURSE_NONE
 voms_apic.h, [15](#)

reserved
 contactdata, [6](#)

role
 data, [7](#)

serial
 voms, [9](#)

server
 voms, [9](#)

serverca
 voms, [9](#)

siglen
 voms, [9](#)

signature
 voms, [9](#)

std
 voms, [9](#)

type
 voms, [9](#)

TYPE_CUSTOM
 voms_apic.h, [15](#)

TYPE_NODATA
 voms_apic.h, [15](#)

TYPE_STD
 voms_apic.h, [15](#)

uri
 voms, [9](#)

user
 voms, [9](#)

userca
 voms, [10](#)

- vdir
 - vomsdata, 11
- VERIFY_DATE
 - voms_apic.h, 15
- VERIFY_FULL
 - voms_apic.h, 15
- VERIFY_ID
 - voms_apic.h, 15
- VERIFY_KEY
 - voms_apic.h, 15
- VERIFY_NONE
 - voms_apic.h, 15
- VERIFY_NOTARGET
 - voms_apic.h, 15
- VERIFY_ORDER
 - voms_apic.h, 15
- VERIFY_SIGN
 - voms_apic.h, 16
- VERR_COMM
 - voms_apic.h, 16
- VERR_DIR
 - voms_apic.h, 16
- VERR_EXTRAINFO
 - voms_apic.h, 16
- VERR_FORMAT
 - voms_apic.h, 16
- VERR_IDCHECK
 - voms_apic.h, 16
- VERR_MEM
 - voms_apic.h, 16
- VERR_NODATA
 - voms_apic.h, 16
- VERR_NOEXT
 - voms_apic.h, 16
- VERR_NOIDENT
 - voms_apic.h, 16
- VERR_NOINIT
 - voms_apic.h, 16
- VERR_NONE
 - voms_apic.h, 17
- VERR_NOCKET
 - voms_apic.h, 17
- VERR_ORDER
 - voms_apic.h, 17
- VERR_PARAM
 - voms_apic.h, 17
- VERR_PARSE
 - voms_apic.h, 17
- VERR_SERVER
 - voms_apic.h, 17
- VERR_SERVERCODE
 - voms_apic.h, 17
- VERR_SIGN
 - voms_apic.h, 17
- VERR_TIME
 - voms_apic.h, 17
- VERR_TYPE
 - voms_apic.h, 17
- VERR_VERIFY
 - voms_apic.h, 17
- version
 - voms, 10
- vo
 - contactdata, 6
- volen
 - vomsdata, 11
- voms, 8
 - custom, 8
 - datalen, 8
 - date1, 8
 - date2, 8
 - fqn, 8
 - serial, 9
 - server, 9
 - serverca, 9
 - siglen, 9
 - signature, 9
 - std, 9
 - type, 9
 - uri, 9
 - user, 9
 - userca, 10
 - version, 10
 - voname, 10
- VOMS_AddTarget
 - voms_apic.h, 18
- voms_apic.h, 13
 - getMajorVersionNumber, 18
 - getMinorVersionNumber, 18
 - getPatchVersionNumber, 18
 - RECURSE_CHAIN, 15
 - RECURSE_NONE, 15
 - TYPE_CUSTOM, 15
 - TYPE_NODATA, 15
 - TYPE_STD, 15
 - VERIFY_DATE, 15
 - VERIFY_FULL, 15
 - VERIFY_ID, 15
 - VERIFY_KEY, 15
 - VERIFY_NONE, 15
 - VERIFY_NOTARGET, 15
 - VERIFY_ORDER, 15
 - VERIFY_SIGN, 16
 - VERR_COMM, 16
 - VERR_DIR, 16
 - VERR_EXTRAINFO, 16
 - VERR_FORMAT, 16
 - VERR_IDCHECK, 16

- VERR_MEM, [16](#)
- VERR_NODATA, [16](#)
- VERR_NOEXT, [16](#)
- VERR_NOIDENT, [16](#)
- VERR_NOINIT, [16](#)
- VERR_NONE, [17](#)
- VERR_NOSOCKET, [17](#)
- VERR_ORDER, [17](#)
- VERR_PARAM, [17](#)
- VERR_PARSE, [17](#)
- VERR_SERVER, [17](#)
- VERR_SERVERCODE, [17](#)
- VERR_SIGN, [17](#)
- VERR_TIME, [17](#)
- VERR_TYPE, [17](#)
- VERR_VERIFY, [17](#)
- VOMS_AddTarget, [18](#)
- VOMS_Contact, [18](#)
- VOMS_ContactRaw, [18](#)
- VOMS_Copy, [19](#)
- VOMS_CopyAll, [19](#)
- VOMS_DefaultData, [19](#)
- VOMS_Delete, [19](#)
- VOMS_DeleteContacts, [19](#)
- VOMS_Destroy, [20](#)
- VOMS_Export, [20](#)
- VOMS_FindByAlias, [20](#)
- VOMS_FindByVO, [20](#)
- VOMS_FreeTargets, [21](#)
- VOMS_Import, [21](#)
- VOMS_Init, [21](#)
- VOMS_ListTargets, [21](#)
- VOMS_Ordering, [22](#)
- VOMS_ResetOrder, [22](#)
- VOMS_Retrieve, [22](#)
- VOMS_SetLifetime, [22](#)
- VOMS_SetVerificationType, [23](#)
- VOMS.Contact
 - [voms_apic.h, 18](#)
- VOMS.ContactRaw
 - [voms_apic.h, 18](#)
- VOMS.Copy
 - [voms_apic.h, 19](#)
- VOMS.CopyAll
 - [voms_apic.h, 19](#)
- VOMS.DefaultData
 - [voms_apic.h, 19](#)
- VOMS.Delete
 - [voms_apic.h, 19](#)
- VOMS.DeleteContacts
 - [voms_apic.h, 19](#)
- VOMS.Destroy
 - [voms_apic.h, 20](#)
- VOMS.Export
 - [voms_apic.h, 20](#)
- VOMS.FindByAlias
 - [voms_apic.h, 20](#)
- VOMS.FindByVO
 - [voms_apic.h, 20](#)
- VOMS.FreeTargets
 - [voms_apic.h, 21](#)
- VOMS.Import
 - [voms_apic.h, 21](#)
- VOMS.Init
 - [voms_apic.h, 21](#)
- VOMS.ListTargets
 - [voms_apic.h, 21](#)
- VOMS.Ordering
 - [voms_apic.h, 22](#)
- VOMS.ResetOrder
 - [voms_apic.h, 22](#)
- VOMS.Retrieve
 - [voms_apic.h, 22](#)
- VOMS.SetLifetime
 - [voms_apic.h, 22](#)
- VOMS.SetVerificationType
 - [voms_apic.h, 23](#)
- vomsdata, [11](#)
 - [cdir, 11](#)
 - [data, 11](#)
 - [extra_data, 11](#)
 - [extralen, 11](#)
 - [vdir, 11](#)
 - [volen, 11](#)
 - [workvo, 11](#)
- voname
 - [voms, 10](#)
- workvo
 - [vomsdata, 11](#)