

VOMS CC API Reference Manual

1.5.0

Generated by Doxygen 1.2.14

Thu Jan 29 11:37:19 2004

Contents

1	VOMS CC API Compound Index	1
1.1	VOMS CC API Compound List	1
2	VOMS CC API File Index	3
2.1	VOMS CC API File List	3
3	VOMS CC API Class 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	12
4	VOMS CC API File Documentation	17
4.1	voms_api.h File Reference	17

Chapter 1

VOMS CC API Compound Index

1.1 VOMS CC API Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

contactdata	5
data (User's characteristics: can be repeated)	7
voms	8
vomsdata	12

Chapter 2

VOMS CC API File Index

2.1 VOMS CC API File List

Here is a list of all files with brief descriptions:

voms_api.h	17
----------------------------	-------	----

Chapter 3

VOMS CC API Class Documentation

3.1 contactdata Struct Reference

```
#include <voms_api.h>
```

Public Attributes

- std::string [nick](#)
- std::string [host](#)
- std::string [contact](#)
- std::string [vo](#)
- int [port](#)

3.1.1 Member Data Documentation

3.1.1.1 std::string contactdata::contact

The subject of the server's certificate
Definition at line 52 of file voms_api.h.

3.1.1.2 std::string contactdata::host

The hostname of the server
Definition at line 51 of file voms_api.h.

3.1.1.3 std::string contactdata::nick

The alias of the server
Definition at line 50 of file voms_api.h.

3.1.1.4 int contactdata::port

The port on which the server is listening

Definition at line 54 of file voms_api.h.

3.1.1.5 std::string contactdata::vo

The VO served by this server

Definition at line 53 of file voms_api.h.

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

- [voms_api.h](#)

3.2 data Struct Reference

User's characteristics: can be repeated.

```
#include <voms_api.h>
```

Public Attributes

- std::string [group](#)
- std::string [role](#)
- std::string [cap](#)

3.2.1 Detailed Description

User's characteristics: can be repeated.

Definition at line 32 of file voms_api.h.

3.2.2 Member Data Documentation

3.2.2.1 std::string data::cap

user's capability

Definition at line 35 of file voms_api.h.

3.2.2.2 std::string data::group

user's group

Definition at line 33 of file voms_api.h.

3.2.2.3 std::string data::role

user's role

Definition at line 34 of file voms_api.h.

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

- [voms_api.h](#)

3.3 voms Struct Reference

```
#include <voms_api.h>
```

Public Methods

- [voms](#) (const voms &)
- [voms](#) ()
- voms & [operator=](#) (const voms &)
- [~voms](#) ()

Public Attributes

- int [version](#)
- int [siglen](#)
- std::string [signature](#)
- std::string [user](#)
- std::string [userca](#)
- std::string [server](#)
- std::string [serverca](#)
- std::string [vname](#)
- std::string [uri](#)
- std::string [date1](#)
- std::string [date2](#)
- [data_type](#) type
- std::vector< [data](#) > [std](#)
- std::string [custom](#)
- std::vector< std::string > [fqan](#)
- std::string [serial](#)

Friends

- class [vomsdata](#)

3.3.1 Constructor & Destructor Documentation

3.3.1.1 `voms::voms (const voms &)`

3.3.1.2 `voms::voms ()`

3.3.1.3 `voms::~~voms ()`

3.3.2 Member Function Documentation

3.3.2.1 `voms& voms::operator= (const voms &)`

3.3.3 Friends And Related Function Documentation

3.3.3.1 `friend class vomsdata [friend]`

Definition at line 58 of file voms_api.h.

3.3.4 Member Data Documentation

3.3.4.1 `std::string voms::custom`

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

Definition at line 72 of file voms_api.h.

3.3.4.2 `std::string voms::date1`

Beginning of validity of the user info

Definition at line 68 of file voms_api.h.

3.3.4.3 `std::string voms::date2`

End of validity of the user info

Definition at line 69 of file voms_api.h.

3.3.4.4 `std::vector<std::string> voms::fqan`

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

Definition at line 74 of file voms_api.h.

3.3.4.5 `std::string voms::serial`

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

Definition at line 75 of file voms_api.h.

3.3.4.6 std::string voms::server

The VOMS server DN, as from its certificate

Definition at line 64 of file voms_api.h.

3.3.4.7 std::string voms::serverca

The CA which signed the VOMS certificate

Definition at line 65 of file voms_api.h.

3.3.4.8 int voms::siglen

The length of the VOMS server signature

Definition at line 60 of file voms_api.h.

3.3.4.9 std::string voms::signature

The VOMS server signature

Definition at line 61 of file voms_api.h.

3.3.4.10 std::vector<data> voms::std

User's characteristics

Definition at line 71 of file voms_api.h.

3.3.4.11 data_type voms::type

The type of [data](#) returned

Definition at line 70 of file voms_api.h.

3.3.4.12 std::string voms::uri

The URI of the VOMS server

Definition at line 67 of file voms_api.h.

3.3.4.13 std::string voms::user

The user's DN, as from his certificate

Definition at line 62 of file voms_api.h.

3.3.4.14 std::string voms::userca

The CA which signed the user's certificate

Definition at line 63 of file voms_api.h.

3.3.4.15 int voms::version

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

Definition at line 59 of file voms_api.h.

3.3.4.16 std::string voms::voname

The name of the VO to which the VOMS belongs

Definition at line 66 of file voms_api.h.

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

- [voms_api.h](#)

3.4 vomsdata Struct Reference

```
#include <voms_api.h>
```

Public Methods

- [vomsdata](#) (std::string voms_dir="", std::string cert_dir="")
- bool [LoadSystemContacts](#) (std::string dir="")
- bool [LoadUserContacts](#) (std::string dir="")
- std::vector< [contactdata](#) > [FindByAlias](#) (std::string alias)
- std::vector< [contactdata](#) > [FindByVO](#) (std::string vo)
- void [Order](#) (std::string att)
- void [ResetOrder](#) (void)
- void [AddTarget](#) (std::string target)
- std::vector< std::string > [ListTargets](#) (void)
- void [ResetTargets](#) (void)
- std::string [ServerErrors](#) (void)
- bool [Retrieve](#) (X509 *cert, STACK_OF(X509)*chain, [recurse_type](#) how=RECURSE_CHAIN)
- bool [Contact](#) (std::string hostname, int port, std::string servsubject, std::string command)
- bool [ContactRaw](#) (std::string hostname, int port, std::string servsubject, std::string command, std::string &raw, int &version)
- void [SetVerificationType](#) ([verify_type](#) how)
- void [SetLifetime](#) (int lifetime)
- bool [Import](#) (std::string buffer)
- bool [Export](#) (std::string &data)
- bool [DefaultData](#) ([voms](#) &)

Public Attributes

- [verror_type](#) error
- std::vector< [voms](#) > [data](#)
- std::string [workvo](#)
- std::string [extra_data](#)

3.4.1 Constructor & Destructor Documentation

3.4.1.1 vomsdata::vomsdata (std::string voms_dir = "", std::string cert_dir = "")

Parameters:

voms_dir The directory which contains the certificate of the VOMS server

cert_dir The directory which contains the certificate of the CA

If voms_dir is empty, the value of the environment variable X509_VOMS_DIR is taken.

If cert_dir is empty, the value of the environment variable X509_CERT_DIR is taken.

3.4.2 Member Function Documentation

3.4.2.1 void vomsdata::AddTarget (std::string *target*)

Adds a target to the AC.

Parameters:

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

3.4.2.2 bool vomsdata::Contact (std::string *hostname*, int *port*, std::string *servsubject*, std::string *command*)

Contacts a VOMS server to get a certificate

It is the equivalent of the voms_proxy_init command, but without the –include functionality.

Parameters:

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

Returns:

failure (F) or success (T)

3.4.2.3 bool vomsdata::ContactRaw (std::string *hostname*, int *port*, std::string *servsubject*, std::string *command*, std::string & *raw*, int & *version*)

Same as Contact, however it does not start the verification process, and the

Parameters:

raw and *ion* parameters contain the message sent from the server along with the version number of the result

3.4.2.4 bool vomsdata::DefaultData (voms &)

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

3.4.2.5 bool vomsdata::Export (std::string & *data*)

Exports [data](#) from [vomsdata::data](#) to the format used for inclusion into a certificate.

The function doesn't verify the [data](#)

Parameters:

[data](#) The certificate extension

Returns:

Failure (F) or Success (T)

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

Finds servers which share a common alias.

Parameters:

alias The alias to look for.

Returns:

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

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

Finds servers which serve a common VO

Parameters:

vo The VO name to look for.

Returns:

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

3.4.2.8 `bool vomsdata::Import (std::string buffer)`

Converts [data](#) from the format used for inclusion into a certificate to the internal format

The function does verify the [data](#).

Parameters:

buffer contains the [data](#) to be converted

Returns:

Failure (F) or Success (T)

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

Returns the list of targets.

3.4.2.10 `bool vomsdata::LoadSystemContacts (std::string dir = "")`

Loads the system wide configuration files.

Parameters:

dir The directory in which the files are stored.

If *dir* is empty, defaults to /opt/edg/etc/vomses.

Returns:

True if all went OK, false otherwise.

3.4.2.11 bool vomsdata::LoadUserContacts (std::string *dir* = "")

Loads the user-specific configuration files.

Parameters:

dir The directory in which the files are stored.

If *dir* is empty, defaults to \$VOMS_USERCONF. If this is empty too, defaults to \$HOME/.edg/vomses, or to ~/.edg/vomses as a last resort.

Returns:

True if all went OK, false otherwise.

3.4.2.12 void vomsdata::Order (std::string *att*)

Sets up the ordering of the results.

Defines the ordering of the [data](#) returned by [Contact\(\)](#). Results are ordered in the same order as the calls to this function.

Parameters:

att The attribute to be ordered.

3.4.2.13 void vomsdata::ResetOrder (void)

Resets the ordering.

3.4.2.14 void vomsdata::ResetTargets (void)

Resets the target list.

3.4.2.15 bool vomsdata::Retrieve (X509 * *cert*, STACK_OF(X509)* *chain*, [recurse_type](#) *how* = RECURSE_CHAIN)

Extracts the VOMS extension from an X.509 certificate. The function doesn't check the validity of the certificates, but it does check the content of the user [data](#).

Parameters:

cert The certificate with the VOMS extensions

chain The chain of the validation certificates (only the intermediate ones)

how Recursion type

Returns:

failure (F) or success (T)

3.4.2.16 std::string vomsdata::ServerErrors (void)

Gets the error message returned by the server

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

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

Parameters:

lifetime Requested lifetime, in seconds

3.4.2.18 void vomsdata::SetVerificationType (*verify_type* *how*)

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

Parameters:

how The type of verification.

3.4.3 Member Data Documentation

3.4.3.1 std::vector<[voms](#)> vomsdata::data

User's info, as in the certificate extension. It may contain [data](#) gathered from more than one VOMS server, Definition at line 276 of file voms_api.h.

3.4.3.2 [verror_type](#) vomsdata::error

Error code

Definition at line 152 of file voms_api.h.

3.4.3.3 std::string vomsdata::extra_data

The [data](#) specified by the user with the `--include` switch.

Note that this field doesn't contain the result of a request to the VOMS server, but instead [data](#) specified by the user.

The reason for the introduction of this extension is to let a user include important [data](#) into his proxy certificate, like, for example, a kerberos ticket

Definition at line 280 of file voms_api.h.

3.4.3.4 std::string vomsdata::workvo

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

Definition at line 279 of file voms_api.h.

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

- [voms_api.h](#)

Chapter 4

VOMS CC API File Documentation

4.1 voms_api.h File Reference

```
#include <string>
#include <vector>
#include <openssl/x509.h>
#include <sys/types.h>
#include "newformat.h"
```

Compounds

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

User's characteristics: can be repeated.

- class **vomsdata::Initializer**
- struct [voms](#)
- struct [vomsdata](#)

Typedefs

- typedef bool(* [check_sig](#))(X509 *, void *, [verror_type](#) &)

Enumerations

- enum [data_type](#) { [TYPE_NODATA](#), [TYPE_STD](#), [TYPE_CUSTOM](#) }

The type of [data](#) returned.

- enum [recurse_type](#) { [RECURSE_CHAIN](#), [RECURSE_NONE](#) }
 - enum [verify_type](#) { [VERIFY_FULL](#) = 0xffffffff, [VERIFY_NONE](#) = 0x00000000, [VERIFY_DATE](#) = 0x00000001, [VERIFY_TARGET](#) = 0x00000002, [VERIFY_KEY](#) = 0x00000004, [VERIFY_SIGN](#) = 0x00000008, [VERIFY_ORDER](#) = 0x00000010, [VERIFY_ID](#) = 0x00000020 }
-

- enum `verror_type` { `VERR_NONE`, `VERR_NOSOCKET`, `VERR_NOIDENT`, `VERR_COMM`, `VERR_PARAM`, `VERR_NOEXT`, `VERR_NOINIT`, `VERR_TIME`, `VERR_IDCHECK`, `VERR_EXTRINFO`, `VERR_FORMAT`, `VERR_NODATA`, `VERR_PARSE`, `VERR_DIR`, `VERR_SIGN`, `VERR_SERVER`, `VERR_MEM`, `VERR_VERIFY`, `VERR_TYPE`, `VERR_ORDER`, `VERR_SERVERCODE` }

Error codes.

Functions

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

4.1.1 Typedef Documentation

4.1.1.1 typedef bool(* `check_sig`)(X509 *, void *, `verror_type` &)

Definition at line 131 of file `voms_api.h`.

4.1.2 Enumeration Type Documentation

4.1.2.1 enum `data_type`

The type of `data` returned.

Enumeration values:

TYPE_NODATA no `data`

TYPE_STD group, role, capability triplet

TYPE_CUSTOM result of an S command

Definition at line 40 of file `voms_api.h`.

4.1.2.2 enum `recurse_type`

Enumeration values:

RECURSE_CHAIN

RECURSE_NONE

Definition at line 87 of file `voms_api.h`.

4.1.2.3 enum `verify_type`

Enumeration values:

VERIFY_FULL

VERIFY_NONE

VERIFY_DATE

VERIFY_TARGET
VERIFY_KEY
VERIFY_SIGN
VERIFY_ORDER
VERIFY_ID

Definition at line 92 of file voms_api.h.

4.1.2.4 enum verror_type

Error codes.

Enumeration values:

VERR_NONE
VERR_NOSOCKET Socket problem
VERR_NOIDENT Cannot identify itself (certificate problem)
VERR_COMM Server problem
VERR_PARAM Wrong parameters
VERR_NOEXT VOMS extension missing
VERR_NOINIT Initialization error
VERR_TIME Error in time checking
VERR_IDCHECK User [data](#) in extension different from the real ones
VERR_EXTRAINFO VO name and URI missing
VERR_FORMAT Wrong [data](#) format
VERR_NODATA Empty extension
VERR_PARSE Parse error
VERR_DIR Directory error
VERR_SIGN Signature error
VERR_SERVER Unidentifiable VOMS server
VERR_MEM Memory problems
VERR_VERIFY Generic verification error
VERR_TYPE Returned [data](#) of unknown type
VERR_ORDER Ordering different than required
VERR_SERVERCODE Error message from the server

Definition at line 105 of file voms_api.h.

4.1.3 Function Documentation

4.1.3.1 int getMajorVersionNumber (void)

4.1.3.2 int getMinorVersionNumber (void)

4.1.3.3 int getPatchVersionNumber (void)

Index

~voms
 voms, [9](#)

AddTarget
 vomldata, [13](#)

cap
 data, [7](#)

check_sig
 vomls_api.h, [18](#)

Contact
 vomldata, [13](#)

contact
 contactdata, [5](#)

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

ContactRaw
 vomldata, [13](#)

custom
 voms, [9](#)

data, [7](#)
 cap, [7](#)
 group, [7](#)
 role, [7](#)
 vomldata, [16](#)

data_type
 vomls_api.h, [18](#)

date1
 voms, [9](#)

date2
 voms, [9](#)

DefaultData
 vomldata, [13](#)

error
 vomldata, [16](#)

Export
 vomldata, [13](#)

extra_data
 vomldata, [16](#)

FindByAlias
 vomldata, [13](#)

FindByVO
 vomldata, [14](#)

fqn
 voms, [9](#)

getMajorVersionNumber
 vomls_api.h, [19](#)

getMinorVersionNumber
 vomls_api.h, [19](#)

getPatchVersionNumber
 vomls_api.h, [19](#)

group
 data, [7](#)

host
 contactdata, [5](#)

Import
 vomldata, [14](#)

ListTargets
 vomldata, [14](#)

LoadSystemContacts
 vomldata, [14](#)

LoadUserContacts
 vomldata, [14](#)

nick
 contactdata, [5](#)

operator=
 voms, [9](#)

Order
 vomldata, [15](#)

port
 contactdata, [5](#)

RECURSE_CHAIN
 vomls_api.h, [18](#)

RECURSE_NONE
 vomls_api.h, [18](#)

recurse_type
 vomls_api.h, [18](#)

- ResetOrder
 - vomsdata, 15
- ResetTargets
 - vomsdata, 15
- Retrieve
 - vomsdata, 15
- role
 - data, 7
- serial
 - voms, 9
- server
 - voms, 9
- serverca
 - voms, 10
- ServerErrors
 - vomsdata, 15
- SetLifetime
 - vomsdata, 15
- SetVerificationType
 - vomsdata, 16
- siglen
 - voms, 10
- signature
 - voms, 10
- std
 - voms, 10
- type
 - voms, 10
- TYPE_CUSTOM
 - voms_api.h, 18
- TYPE_NODATA
 - voms_api.h, 18
- TYPE_STD
 - voms_api.h, 18
- uri
 - voms, 10
- user
 - voms, 10
- userca
 - voms, 10
- VERIFY_DATE
 - voms_api.h, 18
- VERIFY_FULL
 - voms_api.h, 18
- VERIFY_ID
 - voms_api.h, 19
- VERIFY_KEY
 - voms_api.h, 19
- VERIFY_NONE
 - voms_api.h, 18
- VERIFY_ORDER
 - voms_api.h, 19
- VERIFY_SIGN
 - voms_api.h, 19
- VERIFY_TARGET
 - voms_api.h, 18
- verify_type
 - voms_api.h, 18
- VERR_COMM
 - voms_api.h, 19
- VERR_DIR
 - voms_api.h, 19
- VERR_EXTRINFO
 - voms_api.h, 19
- VERR_FORMAT
 - voms_api.h, 19
- VERR_IDCHECK
 - voms_api.h, 19
- VERR_MEM
 - voms_api.h, 19
- VERR_NODATA
 - voms_api.h, 19
- VERR_NOEXT
 - voms_api.h, 19
- VERR_NOIDENT
 - voms_api.h, 19
- VERR_NOINIT
 - voms_api.h, 19
- VERR_NONE
 - voms_api.h, 19
- VERR_NOSOCKET
 - voms_api.h, 19
- VERR_ORDER
 - voms_api.h, 19
- VERR_PARAM
 - voms_api.h, 19
- VERR_PARSE
 - voms_api.h, 19
- VERR_SERVER
 - voms_api.h, 19
- VERR_SERVERCODE
 - voms_api.h, 19
- VERR_SIGN
 - voms_api.h, 19
- VERR_TIME
 - voms_api.h, 19
- VERR_TYPE
 - voms_api.h, 19
- VERR_VERIFY
 - voms_api.h, 19
- verror_type
 - voms_api.h, 19
- version
 - voms, 10

- vo
 - contactdata, 6
- voms, 8
 - ~voms, 9
 - custom, 9
 - date1, 9
 - date2, 9
 - fqan, 9
 - operator=, 9
 - serial, 9
 - server, 9
 - serverca, 10
 - siglen, 10
 - signature, 10
 - std, 10
 - type, 10
 - uri, 10
 - user, 10
 - userca, 10
 - version, 10
 - voms, 9
 - vomsdata, 9
 - voname, 11
- voms_api.h
 - RECURSE_CHAIN, 18
 - RECURSE_NONE, 18
 - TYPE_CUSTOM, 18
 - TYPE_NODATA, 18
 - TYPE_STD, 18
 - VERIFY_DATE, 18
 - VERIFY_FULL, 18
 - VERIFY_ID, 19
 - VERIFY_KEY, 19
 - VERIFY_NONE, 18
 - VERIFY_ORDER, 19
 - VERIFY_SIGN, 19
 - VERIFY_TARGET, 18
 - VERR_COMM, 19
 - VERR_DIR, 19
 - VERR_EXTRAINFO, 19
 - VERR_FORMAT, 19
 - VERR_IDCHECK, 19
 - VERR_MEM, 19
 - VERR_NODATA, 19
 - VERR_NOEXT, 19
 - VERR_NOIDENT, 19
 - VERR_NOINIT, 19
 - VERR_NONE, 19
 - VERR_NOSOCKET, 19
 - VERR_ORDER, 19
 - VERR_PARAM, 19
 - VERR_PARSE, 19
 - VERR_SERVER, 19
 - VERR_SERVERCODE, 19
 - VERR_SIGN, 19
 - VERR_TIME, 19
 - VERR_TYPE, 19
 - VERR_VERIFY, 19
- voms_api.h, 17
 - check_sig, 18
 - data_type, 18
 - getMajorVersionNumber, 19
 - getMinorVersionNumber, 19
 - getPatchVersionNumber, 19
 - recurse_type, 18
 - verify_type, 18
 - verror_type, 19
- vomsdata, 12
 - AddTarget, 13
 - Contact, 13
 - ContactRaw, 13
 - data, 16
 - DefaultData, 13
 - error, 16
 - Export, 13
 - extra_data, 16
 - FindByAlias, 13
 - FindByVO, 14
 - Import, 14
 - ListTargets, 14
 - LoadSystemContacts, 14
 - LoadUserContacts, 14
 - Order, 15
 - ResetOrder, 15
 - ResetTargets, 15
 - Retrieve, 15
 - ServerErrors, 15
 - SetLifetime, 15
 - SetVerificationType, 16
 - voms, 9
 - vomsdata, 12
 - workvo, 16
- voname
 - voms, 11
- workvo
 - vomsdata, 16