

VOMS CC API

1.5.0

Generated by Doxygen 1.8.7

Mon Sep 1 2014 05:56:31

Contents

| | | |
|----------|--|----------|
| 1 | Data Structure Index | 1 |
| 1.1 | Data Structures | 1 |
| 2 | File Index | 3 |
| 2.1 | File List | 3 |
| 3 | Data Structure Documentation | 5 |
| 3.1 | attribute Struct Reference | 5 |
| 3.1.1 | Detailed Description | 5 |
| 3.1.2 | Field Documentation | 5 |
| 3.1.2.1 | name | 5 |
| 3.1.2.2 | qualifier | 5 |
| 3.1.2.3 | value | 5 |
| 3.2 | attributelist Struct Reference | 6 |
| 3.2.1 | Detailed Description | 6 |
| 3.2.2 | Field Documentation | 6 |
| 3.2.2.1 | attributes | 6 |
| 3.2.2.2 | grantor | 6 |
| 3.3 | contactdata Struct Reference | 6 |
| 3.3.1 | Detailed Description | 6 |
| 3.3.2 | Field Documentation | 7 |
| 3.3.2.1 | contact | 7 |
| 3.3.2.2 | host | 7 |
| 3.3.2.3 | nick | 7 |
| 3.3.2.4 | port | 7 |
| 3.3.2.5 | version | 7 |
| 3.3.2.6 | vo | 7 |
| 3.4 | data Struct Reference | 7 |
| 3.4.1 | Detailed Description | 8 |
| 3.4.2 | Field Documentation | 8 |
| 3.4.2.1 | cap | 8 |
| 3.4.2.2 | group | 8 |

| | | |
|----------|--|----|
| 3.4.2.3 | role | 8 |
| 3.5 | voms Struct Reference | 8 |
| 3.5.1 | Detailed Description | 9 |
| 3.5.2 | Constructor & Destructor Documentation | 9 |
| 3.5.2.1 | voms | 9 |
| 3.5.2.2 | voms | 9 |
| 3.5.2.3 | ~voms | 9 |
| 3.5.3 | Member Function Documentation | 9 |
| 3.5.3.1 | GetAC | 9 |
| 3.5.3.2 | GetAttributes | 9 |
| 3.5.3.3 | GetTargets | 9 |
| 3.5.3.4 | operator= | 9 |
| 3.5.4 | Friends And Related Function Documentation | 9 |
| 3.5.4.1 | TranslateVOMS | 9 |
| 3.5.4.2 | vomsdata | 9 |
| 3.5.5 | Field Documentation | 9 |
| 3.5.5.1 | custom | 9 |
| 3.5.5.2 | date1 | 10 |
| 3.5.5.3 | date2 | 10 |
| 3.5.5.4 | fqn | 10 |
| 3.5.5.5 | serial | 10 |
| 3.5.5.6 | server | 10 |
| 3.5.5.7 | serverca | 10 |
| 3.5.5.8 | siglen | 10 |
| 3.5.5.9 | signature | 10 |
| 3.5.5.10 | std | 10 |
| 3.5.5.11 | type | 11 |
| 3.5.5.12 | uri | 11 |
| 3.5.5.13 | user | 11 |
| 3.5.5.14 | userca | 11 |
| 3.5.5.15 | version | 11 |
| 3.5.5.16 | voname | 11 |
| 3.6 | vomsdata Struct Reference | 11 |
| 3.6.1 | Detailed Description | 12 |
| 3.6.2 | Constructor & Destructor Documentation | 12 |
| 3.6.2.1 | vomsdata | 12 |
| 3.6.2.2 | ~vomsdata | 13 |
| 3.6.2.3 | vomsdata | 13 |
| 3.6.3 | Member Function Documentation | 13 |
| 3.6.3.1 | AddTarget | 13 |

| | | |
|----------|--------------------------------|-----------|
| 3.6.3.2 | Contact | 13 |
| 3.6.3.3 | Contact | 13 |
| 3.6.3.4 | ContactRaw | 13 |
| 3.6.3.5 | ContactRaw | 14 |
| 3.6.3.6 | ContactRESTRaw | 15 |
| 3.6.3.7 | DefaultData | 15 |
| 3.6.3.8 | ErrorMessage | 15 |
| 3.6.3.9 | Export | 15 |
| 3.6.3.10 | FindByAlias | 15 |
| 3.6.3.11 | FindByVO | 15 |
| 3.6.3.12 | Import | 16 |
| 3.6.3.13 | ListTargets | 16 |
| 3.6.3.14 | LoadCredentials | 16 |
| 3.6.3.15 | LoadSystemContacts | 16 |
| 3.6.3.16 | LoadUserContacts | 16 |
| 3.6.3.17 | Order | 17 |
| 3.6.3.18 | ResetOrder | 17 |
| 3.6.3.19 | ResetTargets | 17 |
| 3.6.3.20 | Retrieve | 17 |
| 3.6.3.21 | Retrieve | 17 |
| 3.6.3.22 | Retrieve | 17 |
| 3.6.3.23 | Retrieve | 18 |
| 3.6.3.24 | RetrieveFromCred | 19 |
| 3.6.3.25 | RetrieveFromCtx | 19 |
| 3.6.3.26 | RetrieveFromProxy | 19 |
| 3.6.3.27 | ServerErrors | 19 |
| 3.6.3.28 | SetLifetime | 19 |
| 3.6.3.29 | SetRetryCount | 20 |
| 3.6.3.30 | SetVerificationTime | 20 |
| 3.6.3.31 | SetVerificationType | 20 |
| 3.6.4 | Field Documentation | 20 |
| 3.6.4.1 | data | 20 |
| 3.6.4.2 | error | 20 |
| 3.6.4.3 | extra_data | 20 |
| 3.6.4.4 | workvo | 20 |
| 4 | File Documentation | 21 |
| 4.1 | voms_api.h File Reference | 21 |
| 4.1.1 | Macro Definition Documentation | 22 |
| 4.1.1.1 | NOGLOBUS | 22 |

| | | |
|--------------|--|-----------|
| 4.1.2 | Typedef Documentation | 22 |
| 4.1.2.1 | check_sig | 22 |
| 4.1.2.2 | gss_cred_id_t | 22 |
| 4.1.2.3 | gss_ctx_id_t | 22 |
| 4.1.3 | Enumeration Type Documentation | 22 |
| 4.1.3.1 | data_type | 22 |
| 4.1.3.2 | recurse_type | 23 |
| 4.1.3.3 | verify_type | 23 |
| 4.1.3.4 | verror_type | 23 |
| 4.1.4 | Function Documentation | 24 |
| 4.1.4.1 | getVOMSMajorVersionNumber | 24 |
| 4.1.4.2 | getVOMSMinorVersionNumber | 24 |
| 4.1.4.3 | getVOMSPatchVersionNumber | 24 |
| Index | | 25 |

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

| | |
|---|----|
| attribute | 5 |
| attributelist | 6 |
| contactdata | 6 |
| data | |
| User's characteristics: can be repeated. Generic name-value attribute : can be repeated | 7 |
| voms | 8 |
| vomldata | 11 |

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

| | |
|----------------------------|----|
| voms_api.h | 21 |
|----------------------------|----|

Chapter 3

Data Structure Documentation

3.1 attribute Struct Reference

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

Data Fields

- `std::string` [name](#)
- `std::string` [qualifier](#)
- `std::string` [value](#)

3.1.1 Detailed Description

Definition at line 63 of file `voms_api.h`.

3.1.2 Field Documentation

3.1.2.1 `std::string attribute::name`

attribute's group

Definition at line 64 of file `voms_api.h`.

3.1.2.2 `std::string attribute::qualifier`

attribute's qualifier

Definition at line 65 of file `voms_api.h`.

3.1.2.3 `std::string attribute::value`

attribute's value

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

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

- [voms_api.h](#)

3.2 attributelist Struct Reference

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

Data Fields

- std::string [grantor](#)
- std::vector< [attribute](#) > [attributes](#)

3.2.1 Detailed Description

Definition at line 69 of file voms_api.h.

3.2.2 Field Documentation

3.2.2.1 std::vector<attribute> attributelist::attributes

The attributes themselves.

Definition at line 71 of file voms_api.h.

3.2.2.2 std::string attributelist::grantor

Who granted these attributes.

Definition at line 70 of file voms_api.h.

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

- [voms_api.h](#)

3.3 contactdata Struct Reference

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

Data Fields

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

3.3.1 Detailed Description

Definition at line 83 of file voms_api.h.

3.3.2 Field Documentation

3.3.2.1 `std::string contactdata::contact`

The subject of the server's certificate
Definition at line 89 of file voms_api.h.

3.3.2.2 `std::string contactdata::host`

The hostname of the server
Definition at line 88 of file voms_api.h.

3.3.2.3 `std::string contactdata::nick`

< You must never allocate directly this structure. Its `sizeof()` is subject to change without notice. The only supported way to obtain it is via the `FindBy*` functions. The alias of the server
Definition at line 87 of file voms_api.h.

3.3.2.4 `int contactdata::port`

The port on which the server is listening
Definition at line 91 of file voms_api.h.

3.3.2.5 `int contactdata::version`

The version of globus under which the server is running
Definition at line 93 of file voms_api.h.

3.3.2.6 `std::string contactdata::vo`

The VO served by this server
Definition at line 90 of file voms_api.h.

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

- [voms_api.h](#)

3.4 data Struct Reference

User's characteristics: can be repeated. Generic name-value attribute : can be repeated.

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

Data Fields

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

3.4.1 Detailed Description

User's characteristics: can be repeated. Generic name-value attribute : can be repeated.

Definition at line 55 of file voms_api.h.

3.4.2 Field Documentation

3.4.2.1 `std::string data::cap`

user's capability

Definition at line 58 of file voms_api.h.

3.4.2.2 `std::string data::group`

user's group

Definition at line 56 of file voms_api.h.

3.4.2.3 `std::string data::role`

user's role

Definition at line 57 of file voms_api.h.

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

- [voms_api.h](#)

3.5 voms Struct Reference

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

Public Member Functions

- [voms](#) (const [voms](#) &)
- [voms](#) ()
- [voms](#) & [operator=](#) (const [voms](#) &)
- [~voms](#) ()
- AC * [GetAC](#) ()
- `std::vector< attributelist >` & [GetAttributes](#) ()
- `std::vector< std::string >` [GetTargets](#) ()

Data Fields

- int [version](#)
- int [siglen](#)
- `std::string` [signature](#)
- `std::string` [user](#)
- `std::string` [userca](#)
- `std::string` [server](#)
- `std::string` [serverca](#)
- `std::string` [voname](#)

- std::string [uri](#)
- std::string [date1](#)
- std::string [date2](#)
- [data_type](#) type
- std::vector< [data](#) > [std](#)
- std::string [custom](#)
- std::vector< std::string > [fqan](#)
- std::string [serial](#)

Friends

- class [vomsdata](#)
- int [TranslateVOMS](#) (struct vomsdatar *vd, std::vector< [voms](#) > &v, int *error)

3.5.1 Detailed Description

Definition at line 98 of file voms_api.h.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 voms::voms (const voms &)

3.5.2.2 voms::voms ()

3.5.2.3 voms::~~voms ()

3.5.3 Member Function Documentation

3.5.3.1 AC* voms::GetAC ()

3.5.3.2 std::vector<attributelist>& voms::GetAttributes ()

Generic attributes

3.5.3.3 std::vector<std::string> voms::GetTargets ()

3.5.3.4 voms& voms::operator= (const voms &)

3.5.4 Friends And Related Function Documentation

3.5.4.1 int TranslateVOMS (struct vomsdatar * vd, std::vector< [voms](#) > & v, int * error) [\[friend\]](#)

3.5.4.2 friend class [vomsdata](#) [\[friend\]](#)

Definition at line 99 of file voms_api.h.

3.5.5 Field Documentation

3.5.5.1 std::string voms::custom

The data returned by an S command

Definition at line 113 of file voms_api.h.

3.5.5.2 std::string voms::date1

Beginning of validity of the user info

Definition at line 109 of file voms_api.h.

3.5.5.3 std::string voms::date2

End of validity of the user info

Definition at line 110 of file voms_api.h.

3.5.5.4 std::vector<std::string> voms::fqan

Keeps the data in the compact format

Definition at line 115 of file voms_api.h.

3.5.5.5 std::string voms::serial

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

Definition at line 116 of file voms_api.h.

3.5.5.6 std::string voms::server

The VOMS server DN, as from its certificate

Definition at line 105 of file voms_api.h.

3.5.5.7 std::string voms::serverca

The CA which signed the VOMS certificate

Definition at line 106 of file voms_api.h.

3.5.5.8 int voms::siglen

The length of the VOMS server signature

Definition at line 101 of file voms_api.h.

3.5.5.9 std::string voms::signature

The VOMS server signature

Definition at line 102 of file voms_api.h.

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

User's characteristics

Definition at line 112 of file voms_api.h.

3.5.5.11 data_type voms::type

The type of data returned

Definition at line 111 of file voms_api.h.

3.5.5.12 std::string voms::uri

The URI of the VOMS server

Definition at line 108 of file voms_api.h.

3.5.5.13 std::string voms::user

The user's DN, as from his certificate

Definition at line 103 of file voms_api.h.

3.5.5.14 std::string voms::userca

The CA which signed the user's certificate

Definition at line 104 of file voms_api.h.

3.5.5.15 int voms::version

0 means data didn't originate from an AC

Definition at line 100 of file voms_api.h.

3.5.5.16 std::string voms::voname

The name of the VO to which the VOMS belongs

Definition at line 107 of file voms_api.h.

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

- [voms_api.h](#)

3.6 vomsdata Struct Reference

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

Public Member Functions

- [vomsdata](#) (std::string voms_dir="", std::string cert_dir="")
- bool [LoadSystemContacts](#) (std::string dir="")
- bool [LoadUserContacts](#) (std::string dir="")
- std::vector< [contactdata](#) > [FindByAlias](#) (std::string alias)
- std::vector< [contactdata](#) > [FindByVO](#) (std::string vo)
- void [Order](#) (std::string att)
- void [ResetOrder](#) (void)
- void [AddTarget](#) (std::string target)
- std::vector< std::string > [ListTargets](#) (void)

- void [ResetTargets](#) (void)
- std::string [ServerErrors](#) (void)
- bool [Retrieve](#) (X509 *cert, STACK_OF(X509)*chain, [recurse_type](#) how=[RECURSE_CHAIN](#))
- bool [Contact](#) (std::string hostname, int port, std::string servsubject, std::string command)
- bool [Contact](#) (std::string hostname, int port, std::string servsubject, std::string command, int timeout)
- bool [ContactRaw](#) (std::string hostname, int port, std::string servsubject, std::string command, std::string &raw, int &version)
- bool [ContactRaw](#) (std::string hostname, int port, std::string servsubject, std::string command, std::string &raw, int &version, int timeout)
- void [SetVerificationType](#) ([verify_type](#) how)
- void [SetLifetime](#) (int lifetime)
- bool [Import](#) (std::string buffer)
- bool [Export](#) (std::string &data)
- bool [DefaultData](#) ([voms](#) &)
- std::string [ErrorMessage](#) (void)
- bool [RetrieveFromCtx](#) ([gss_ctx_id_t](#) context, [recurse_type](#) how)
- bool [RetrieveFromCred](#) ([gss_cred_id_t](#) credential, [recurse_type](#) how)
- bool [Retrieve](#) (X509_EXTENSION *ext)
- bool [RetrieveFromProxy](#) ([recurse_type](#) how)
- bool [Retrieve](#) (FILE *file, [recurse_type](#) how)
- bool [Retrieve](#) (AC *ac)
- [~vomsdata](#) ()
- [vomsdata](#) (const [vomsdata](#) &)
- void [SetRetryCount](#) (int retryCount)
- void [SetVerificationTime](#) (time_t)
- bool [LoadCredentials](#) (X509 *, EVP_PKEY *, STACK_OF(X509)*)
- bool [ContactRESTRaw](#) (const std::string &, int, const std::string &, std::string &, int, int)

Data Fields

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

3.6.1 Detailed Description

Definition at line 194 of file voms_api.h.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 [vomsdata::vomsdata](#) ([std::string voms_dir](#) = " ", [std::string cert_dir](#) = " ")

Parameters

| | |
|--------------------------|---|
| 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.

3.6.2.2 vomsdata::~vomsdata ()

3.6.2.3 vomsdata::vomsdata (const vomsdata &)

3.6.3 Member Function Documentation

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

Adds a target to the AC.

\param *target* The target to be added. it should be a FQDN.

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

Contacts a VOMS server to get a certificate

It is the equivalent of the voms_proxy_init command, but without the --include functionality.
 \param *hostname* FQDN of the VOMS server
 \param *port* the port on which the VOMS server is listening
 \param *servsubject* the subject of the server's certificate
 \param *command* the command sent to the server
 \return failure (F) or success (T)

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

Contacts a VOMS server to get a certificate

It is the equivalent of the voms_proxy_init command, but without the --include functionality.
 \param *hostname* FQDN of the VOMS server
 \param *port* the port on which the VOMS server is listening
 \param *servsubject* the subject of the server's certificate
 \param *command* the command sent to the server
 \return failure (F) or success (T)

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

Same as Contact, however it does not start the verification process, and the message received from the server is not parsed.

Parameters

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

Returns

failure (F) or success (T)

3.6.3.5 `bool vomldata::ContactRaw (std::string hostname, int port, std::string servsubject, std::string command, std::string & raw, int & version, int timeout)`

Same as `Contact`, however it does not start the verification process, and the message received from the server is not parsed.

Parameters

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

Returns

failure (F) or success (T)

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

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

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

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

Gets a textual description of the error.

Returns

A string containing the error message.

3.6.3.9 `bool vomsdata::Export (std::string & data)`

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

The function doesn't verify the data

Parameters

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

Returns

Failure (F) or Success (T)

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

Finds servers which share a common alias.

Parameters

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

Returns

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

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

Finds servers which serve a common VO

Parameters

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

Returns

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

3.6.3.12 bool vomsdata::Import (std::string *buffer*)

Converts data from the format used for inclusion into a certificate to the internal format

The function does verify the data.

Parameters

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

Returns

Failure (F) or Success (T)

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

Returns the list of targets.

3.6.3.14 bool vomsdata::LoadCredentials (X509 *, EVP_PKEY *, STACK_OF(X509)*)**3.6.3.15 bool vomsdata::LoadSystemContacts (std::string *dir* = " ")**

Loads the system wide configuration files.

Parameters

| | |
|------------|--|
| <i>dir</i> | The directory in which the files are stored. |
|------------|--|

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

Returns

True if all went OK, false otherwise.

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

Loads the user-specific configuration files.

Parameters

| | |
|------------|--|
| <i>dir</i> | The directory in which the files are stored. |
|------------|--|

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

Returns

True if all went OK, false otherwise.

3.6.3.17 void 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.
\param *att* The attribute to be ordered.

3.6.3.18 void vomsdata::ResetOrder (void)

Resets the ordering.

3.6.3.19 void vomsdata::ResetTargets (void)

Resets the target list.

3.6.3.20 bool vomsdata::Retrieve (X509 * *cert*, STACK_OF(X509)* *chain*, recurse_type *how* = RECURSE_CHAIN)

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

Parameters

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

Returns

failure (F) or success (T)

3.6.3.21 bool vomsdata::Retrieve (X509_EXTENSION * *ext*)

Gets VOMS information from the given extension

Parameters

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

Returns

failure (F) or success (T)

3.6.3.22 bool vomsdata::Retrieve (FILE * *file*, recurse_type *how*)

Gets VOMS information from a proxy saved as a file.

Parameters

| | |
|-------------|----------------|
| <i>file</i> | the file name |
| <i>how</i> | Recursion type |

Returns

failure (F) or success (T)

Note: Does NOT verify that the proxy is valid. Such verification must be obtained through other means.

3.6.3.23 bool vomsdata::Retrieve (AC * ac)

Gets VOMS information from the AC

Parameters

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

Returns

failure (F) or success (T)

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

Gets VOMS information from the given globus credential

Parameters

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

Returns

failure (F) or success (T)

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

Gets VOMS information from the given globus context

Parameters

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

Returns

failure (F) or success (T)

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

Gets VOMS information from an existing globus proxy

Parameters

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

Returns

failure (F) or success (T)

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

Gets the error message returned by the server

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

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

Parameters

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

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

3.6.3.30 `void vomsdata::SetVerificationTime (time_t)`

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

Sets the type of verification done on the data.

Parameters

| | |
|------------|---------------------------|
| <i>how</i> | The type of verification. |
|------------|---------------------------|

3.6.4 Field Documentation

3.6.4.1 `std::vector<voms> vomsdata::data`

User's info, as in the certificate extension. It may contain data gathered from more than one VOMS server, Definition at line 368 of file voms_api.h.

3.6.4.2 `error_type vomsdata::error`

Error code

Definition at line 213 of file voms_api.h.

3.6.4.3 `std::string vomsdata::extra_data`

The data specified by the user with the `--include` switch.

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

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

Definition at line 372 of file voms_api.h.

3.6.4.4 `std::string vomsdata::workvo`

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

Definition at line 371 of file voms_api.h.

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

- [voms_api.h](#)

Chapter 4

File Documentation

4.1 voms_api.h File Reference

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

Data Structures

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

Macros

- #define [NOGLOBUS](#)

Typedefs

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

Enumerations

- enum [data_type](#) { [TYPE_NODATA](#), [TYPE_STD](#), [TYPE_CUSTOM](#) }
The type of data returned.
- enum [recurse_type](#) { [RECURSE_CHAIN](#), [RECURSE_NONE](#), [RECURSE_DEEP](#) }

- enum `verify_type` {
`VERIFY_FULL` = 0xffffffff, `VERIFY_NONE` = 0x00000000, `VERIFY_DATE` = 0x00000001, `VERIFY_TARGET` = 0x00000002,
`VERIFY_KEY` = 0x00000004, `VERIFY_SIGN` = 0x00000008, `VERIFY_ORDER` = 0x00000010, `VERIFY_ID` = 0x00000020,
`VERIFY_CERTLIST` = 0x00000040 }
 - enum `verror_type` {
`VERR_NONE`, `VERR_NOCKET`, `VERR_NOIDENT`, `VERR_COMM`,
`VERR_PARAM`, `VERR_NOEXT`, `VERR_NOINIT`, `VERR_TIME`,
`VERR_IDCHECK`, `VERR_EXTRINFO`, `VERR_FORMAT`, `VERR_NODATA`,
`VERR_PARSE`, `VERR_DIR`, `VERR_SIGN`, `VERR_SERVER`,
`VERR_MEM`, `VERR_VERIFY`, `VERR_TYPE`, `VERR_ORDER`,
`VERR_SERVERCODE`, `VERR_NOTAVAIL`, `VERR_FILE` }
- Error codes.*

Functions

- int `getVOMSMajorVersionNumber` (void)
- int `getVOMSMinorVersionNumber` (void)
- int `getVOMSPatchVersionNumber` (void)

4.1.1 Macro Definition Documentation

4.1.1.1 #define NOGLOBUS

Definition at line 33 of file `voms_api.h`.

4.1.2 Typedef Documentation

4.1.2.1 typedef bool(* check_sig)(X509 *, void *, verror_type &)

Definition at line 190 of file `voms_api.h`.

4.1.2.2 typedef void* gss_cred_id_t

Definition at line 42 of file `voms_api.h`.

4.1.2.3 typedef void* gss_ctx_id_t

Definition at line 43 of file `voms_api.h`.

4.1.3 Enumeration Type Documentation

4.1.3.1 enum data_type

The type of data returned.

Enumerator

- `TYPE_NODATA`** no data
- `TYPE_STD`** group, role, capability triplet
- `TYPE_CUSTOM`** result of an S command

Definition at line 77 of file `voms_api.h`.

4.1.3.2 enum recurse_type

Enumerator

RECURSE_CHAIN
RECURSE_NONE
RECURSE_DEEP

Definition at line 143 of file voms_api.h.

4.1.3.3 enum verify_type

Enumerator

VERIFY_FULL
VERIFY_NONE
VERIFY_DATE
VERIFY_TARGET
VERIFY_KEY
VERIFY_SIGN
VERIFY_ORDER
VERIFY_ID
VERIFY_CERTLIST

Definition at line 149 of file voms_api.h.

4.1.3.4 enum verror_type

Error codes.

Enumerator

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

VERR_TYPE Returned data of unknown type
VERR_ORDER Ordering different than required
VERR_SERVERCODE Error message from the server
VERR_NOTAVAIL Method not available
VERR_FILE Error reading data from file

Definition at line 163 of file voms_api.h.

4.1.4 Function Documentation

4.1.4.1 int getVOMSMajorVersionNumber (void)

4.1.4.2 int getVOMSMinorVersionNumber (void)

4.1.4.3 int getVOMSPatchVersionNumber (void)

Index

- ~voms
 - voms, [9](#)
- ~vomldata
 - vomldata, [12](#)
- attribute, [5](#)
 - name, [5](#)
 - qualifier, [5](#)
 - value, [5](#)
- attributelist, [6](#)
 - attributes, [6](#)
 - grantor, [6](#)
- attributes
 - attributelist, [6](#)
- cap
 - data, [8](#)
- Contact
 - vomldata, [13](#)
- contact
 - contactdata, [7](#)
- contactdata, [6](#)
 - contact, [7](#)
 - host, [7](#)
 - nick, [7](#)
 - port, [7](#)
 - version, [7](#)
 - vo, [7](#)
- custom
 - voms, [9](#)
- data, [7](#)
 - cap, [8](#)
 - group, [8](#)
 - role, [8](#)
 - vomldata, [20](#)
- date1
 - voms, [9](#)
- date2
 - voms, [10](#)
- error
 - vomldata, [20](#)
- Export
 - vomldata, [15](#)
- fqn
 - voms, [10](#)
- grantor
 - attributelist, [6](#)

- group
 - data, [8](#)
- host
 - contactdata, [7](#)
- Import
 - vomldata, [16](#)
- name
 - attribute, [5](#)
- nick
 - contactdata, [7](#)
- operator=
 - voms, [9](#)
- Order
 - vomldata, [16](#)
- port
 - contactdata, [7](#)
- qualifier
 - attribute, [5](#)
- RECURSE_CHAIN
 - voms_api.h, [23](#)
- RECURSE_DEEP
 - voms_api.h, [23](#)
- RECURSE_NONE
 - voms_api.h, [23](#)
- Retrieve
 - vomldata, [17](#)
- role
 - data, [8](#)
- serial
 - voms, [10](#)
- server
 - voms, [10](#)
- serverca
 - voms, [10](#)
- siglen
 - voms, [10](#)
- signature
 - voms, [10](#)
- std
 - voms, [10](#)
- TYPE_CUSTOM
 - voms_api.h, [22](#)

TYPE_NODATA
 voms_api.h, 22
 TYPE_STD
 voms_api.h, 22
 type
 voms, 10

 uri
 voms, 11
 user
 voms, 11
 userca
 voms, 11

 VERIFY_CERTLIST
 voms_api.h, 23
 VERIFY_DATE
 voms_api.h, 23
 VERIFY_FULL
 voms_api.h, 23
 VERIFY_ID
 voms_api.h, 23
 VERIFY_KEY
 voms_api.h, 23
 VERIFY_NONE
 voms_api.h, 23
 VERIFY_ORDER
 voms_api.h, 23
 VERIFY_SIGN
 voms_api.h, 23
 VERIFY_TARGET
 voms_api.h, 23
 VERR_COMM
 voms_api.h, 23
 VERR_DIR
 voms_api.h, 23
 VERR_EXTRAINFO
 voms_api.h, 23
 VERR_FILE
 voms_api.h, 24
 VERR_FORMAT
 voms_api.h, 23
 VERR_IDCHECK
 voms_api.h, 23
 VERR_MEM
 voms_api.h, 23
 VERR_NODATA
 voms_api.h, 23
 VERR_NOEXT
 voms_api.h, 23
 VERR_NOIDENT
 voms_api.h, 23
 VERR_NOINIT
 voms_api.h, 23
 VERR_NONE
 voms_api.h, 23
 VERR_NOSOCKET
 voms_api.h, 23
 VERR_NOTAVAIL
 voms_api.h, 24
 VERR_ORDER
 voms_api.h, 24
 VERR_PARAM
 voms_api.h, 23
 VERR_PARSE
 voms_api.h, 23
 VERR_SERVER
 voms_api.h, 23
 VERR_SERVERCODE
 voms_api.h, 24
 VERR_SIGN
 voms_api.h, 23
 VERR_TIME
 voms_api.h, 23
 VERR_TYPE
 voms_api.h, 23
 VERR_VERIFY
 voms_api.h, 23
 value
 attribute, 5
 version
 contactdata, 7
 voms, 11
 vo
 contactdata, 7
 voms, 8
 ~voms, 9
 custom, 9
 date1, 9
 date2, 10
 fqan, 10
 operator=, 9
 serial, 10
 server, 10
 serverca, 10
 siglen, 10
 signature, 10
 std, 10
 type, 10
 uri, 11
 user, 11
 userca, 11
 version, 11
 voms, 9
 vomldata, 9
 voname, 11
 voms_api.h
 RECURSE_CHAIN, 23
 RECURSE_DEEP, 23
 RECURSE_NONE, 23
 TYPE_CUSTOM, 22
 TYPE_NODATA, 22
 TYPE_STD, 22
 VERIFY_CERTLIST, 23
 VERIFY_DATE, 23
 VERIFY_FULL, 23
 VERIFY_ID, 23

- VERIFY_KEY, [23](#)
- VERIFY_NONE, [23](#)
- VERIFY_ORDER, [23](#)
- VERIFY_SIGN, [23](#)
- VERIFY_TARGET, [23](#)
- VERR_COMM, [23](#)
- VERR_DIR, [23](#)
- VERR_EXTRINFO, [23](#)
- VERR_FILE, [24](#)
- VERR_FORMAT, [23](#)
- VERR_IDCHECK, [23](#)
- VERR_MEM, [23](#)
- VERR_NODATA, [23](#)
- VERR_NOEXT, [23](#)
- VERR_NOIDENT, [23](#)
- VERR_NOINIT, [23](#)
- VERR_NONE, [23](#)
- VERR_NOSOCKET, [23](#)
- VERR_NOTAVAIL, [24](#)
- VERR_ORDER, [24](#)
- VERR_PARAM, [23](#)
- VERR_PARSE, [23](#)
- VERR_SERVER, [23](#)
- VERR_SERVERCODE, [24](#)
- VERR_SIGN, [23](#)
- VERR_TIME, [23](#)
- VERR_TYPE, [23](#)
- VERR_VERIFY, [23](#)
- vomsdata, [11](#)
 - ~vomsdata, [12](#)
 - Contact, [13](#)
 - data, [20](#)
 - error, [20](#)
 - Export, [15](#)
 - Import, [16](#)
 - Order, [16](#)
 - Retrieve, [17](#)
 - voms, [9](#)
 - vomsdata, [12](#), [13](#)
 - workvo, [20](#)
- voname
 - voms, [11](#)
- workvo
 - vomsdata, [20](#)