VOMS C API Reference Manual

1.5.0

Generated by Doxygen 1.5.1

Tue Nov 2 23:00:48 2010

Contents

Chapter 1

VOMS C API Data Structure Index

1.1 VOMS C API Data Structures

Here are the data structures with brief descriptions:

Chapter 2

VOMS C API File Index

2.1 VOMS C API File List

Here is a list of all files with brief descriptions:

Chapter 3

VOMS C API 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 Detailed Description

Definition at line 53 of file voms_apic.h.

3.1.2 Field Documentation

3.1.2.1 const char∗ attribute::name

Definition at line 54 of file voms_apic.h.

3.1.2.2 const char∗ attribute::value

Definition at line 55 of file voms_apic.h.

3.1.2.3 const char∗ attribute::qualifier

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

```
#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

Definition at line 59 of file voms_apic.h.

3.2.2 Field Documentation

3.2.2.1 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 62 of file voms_apic.h.

3.2.2.2 char∗ contactdata::host

The hostname of the server

Definition at line 63 of file voms_apic.h.

3.2.2.3 char∗ contactdata::contact

The subject of the server’s certificate

Definition at line 64 of file voms_apic.h.

3.2.2.4 char∗ contactdata::vo

The VO served by this server

Definition at line 65 of file voms_apic.h.

3.2.2.5 int contactdata::port

The port on which the server is listening

Definition at line 66 of file voms_apic.h.

3.2.2.6 char∗ contactdata::reserved

HANDS OFF!

Definition at line 67 of file voms_apic.h.

3.2.2.7 int contactdata::version

The version of Globus on which this server runs.

Definition at line 68 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.

<

Definition at line 47 of file voms_apic.h.

3.3.2 Field Documentation

3.3.2.1 char∗ data::group

user's group

Definition at line 48 of file voms_apic.h.

3.3.2.2 char∗ data::role

user's role

Definition at line 49 of file voms_apic.h.

3.3.2.3 char∗ data::cap

user's capability

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
- data ∗∗ std
- char ∗ custom
- int datalen
- int version
- char ∗∗ fqan
- char ∗ serial
- AC ∗ ac
- X509 ∗ holder

3.4.1 Detailed Description

Definition at line 79 of file voms_apic.h.

3.4.2 Field Documentation

3.4.2.1 int voms::siglen

The length of the VOMS server signature

Definition at line 80 of file voms_apic.h.

3.4.2.2 char∗ voms::signature

The VOMS server signature

Definition at line 81 of file voms_apic.h.

3.4.2.3 char∗ voms::user

The user's DN, as from his certificate

Definition at line 82 of file voms_apic.h.

3.4.2.4 char∗ voms::userca

The CA which signed the user's certificate

Definition at line 83 of file voms_apic.h.

3.4.2.5 char∗ voms::server

The VOMS server DN, as from its certificate

Definition at line 84 of file voms_apic.h.

3.4.2.6 char∗ voms::serverca

The CA which signed the VOMS certificate

Definition at line 85 of file voms_apic.h.

3.4.2.7 char∗ voms::voname

The name of the VO to which the VOMS belongs

Definition at line 86 of file voms_apic.h.

3.4.2.8 char∗ voms::uri

The URI of the VOMS server

Definition at line 87 of file voms_apic.h.

3.4.2.9 char∗ voms::date1

Beginning of validity of the user info

Definition at line 88 of file voms_apic.h.

3.4.2.10 char∗ voms::date2

End of validity of the user info

Definition at line 89 of file voms_apic.h.

3.4.2.11 int voms::type

The type of data returned

Definition at line 90 of file voms_apic.h.

3.4.2.12 struct data∗∗ voms::std

User's characteristics

Definition at line 91 of file voms_apic.h.

3.4.2.13 char∗ voms::custom

The data returned by an S command

Definition at line 92 of file voms_apic.h.

3.4.2.14 int voms::datalen

Definition at line 93 of file voms_apic.h.

3.4.2.15 int voms::version

Definition at line 94 of file voms_apic.h.

3.4.2.16 char∗∗ voms::fqan

User's attributes in compact format

Definition at line 95 of file voms_apic.h.

3.4.2.17 char∗ voms::serial

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

Definition at line 96 of file voms_apic.h.

3.4.2.18 AC∗ voms::ac

Definition at line 99 of file voms_apic.h.

3.4.2.19 X509∗ voms::holder

Definition at line 100 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
- voms ** data
- char * workvo
- char * extra_data
- int volen
- int extralen
- vomsdata * real

3.5.1 Detailed Description

Definition at line 143 of file voms_apic.h.

3.5.2 Field Documentation

3.5.2.1 char* vomsdata::cdir

Definition at line 144 of file voms_apic.h.

3.5.2.2 char* vomsdata::vdir

Definition at line 145 of file voms_apic.h.

3.5.2.3 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 146 of file voms_apic.h.

3.5.2.4 char* vomsdata::workvo

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

Definition at line 149 of file voms_apic.h.

3.5.2.5 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 151 of file voms_apic.h.

3.5.2.6 int vomsdata::volen

Definition at line 160 of file voms_apic.h.

3.5.2.7 int vomsdata::extralen

Definition at line 161 of file voms_apic.h.

3.5.2.8 struct vomsdata∗ vomsdata::real

Definition at line 163 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 <gssapi.h>
#include <openssl/x509.h>
#include <time.h>
#include "newformat.h"
```

Data Structures

- struct data

  User's characteristics: can be repeated.

- struct attribute
- struct contactdata
- struct voms
- struct vomsdata

Defines

- #define TYPE_NODATA 0
- #define TYPE_STD 1
- #define TYPE_CUSTOM 2
- #define RECURSE_CHAIN 0
- #define RECURSE_NONE 1
- #define VERIFY_FULL 0xffffffff
- #define VERIFY_NONE 0x00000000
- #define VERIFY_DATE 0x00000001
- #define VERIFY_NOTARGET 0x00000002
- #define VERIFY_KEY 0x00000004
- #define VERIFY_SIGN 0x00000008
- #define VERIFY_ORDER 0x00000010
- #define VERIFY_ID 0x00000020

- #define VERIFY_CERTLIST 0x00000040
- #define VERR_NONE 0

  Error codes.

- #define VERR_NOSOCKET 1
- #define VERR_NOIDENT 2
- #define VERR_COMM 3
- #define VERR_PARAM 4
- #define VERR_NOEXT 5
- #define VERR_NOINIT 6
- #define VERR_TIME 7
- #define VERR_IDCHECK 8
- #define VERR_EXTRAINFO 9
- #define VERR_FORMAT 10
- #define VERR_NODATA 11
- #define VERR_PARSE 12
- #define VERR_DIR 13
- #define VERR_SIGN 14
- #define VERR_SERVER 15
- #define VERR_MEM 16
- #define VERR_VERIFY 17
- #define VERR_TYPE 18
- #define VERR_ORDER 19
- #define VERR_SERVERCODE 20
- #define VERR_NOTAVAIL 21

Functions

- contactdata ∗∗ VOMS_FindByAlias (struct vomsdata ∗vd, char ∗alias, char ∗system, char ∗user, int ∗error)
- contactdata ∗∗ VOMS_FindByVO (struct vomsdata ∗vd, char ∗vo, char ∗system, char ∗user, int ∗error)
- void VOMS_DeleteContacts (struct contactdata ∗∗list)
- vomsdata ∗ VOMS_Init (char ∗voms, char ∗cert)
- voms ∗ VOMS_Copy (struct voms ∗v, int ∗error)
- vomsdata ∗ VOMS_CopyAll (struct vomsdata ∗vd, int ∗error)
- void VOMS_Delete (struct voms ∗v)
- int VOMS_AddTarget (struct vomsdata ∗vd, char ∗target, int ∗error)
- void VOMS_FreeTargets (struct vomsdata ∗vd, int ∗error)
- char ∗ VOMS_ListTargets (struct vomsdata ∗vd, int ∗error)
- int VOMS_SetVerificationType (int type, struct vomsdata ∗vd, int ∗error)
- int VOMS_SetLifetime (int length, struct vomsdata ∗vd, int ∗error)
- void VOMS_Destroy (struct vomsdata ∗vd)
- int VOMS_ResetOrder (struct vomsdata ∗vd, int ∗error)
- int VOMS_Ordering (char ∗order, struct vomsdata ∗vd, int ∗error)
- int VOMS_Contact (char ∗hostname, int port, char ∗servsubject, char ∗command, struct vomsdata ∗vd, int ∗error)
- int VOMS_ContactRaw (char ∗hostname, int port, char ∗servsubject, char ∗command, void ∗∗data, int ∗datalen, int ∗version, struct vomsdata ∗vd, int ∗error)

- int VOMS_Retrieve (X509 ∗cert, STACK_OF(X509)∗chain, int how, struct vomsdata ∗vd, int ∗error)
- int VOMS_Import (char ∗buffer, int buflen, struct vomsdata ∗vd, int ∗error)
- int VOMS_Export (char ∗∗buffer, int ∗buflen, struct vomsdata ∗vd, int ∗error)
- voms ∗ VOMS_DefaultData (struct vomsdata ∗vd, int ∗error)
- char ∗ VOMS_ErrorMessage (struct vomsdata ∗vd, int error, char ∗buffer, int len)
- int VOMS_RetrieveEXT (X509_EXTENSION ∗ext, struct vomsdata ∗vd, int ∗error)
- int VOMS_RetrieveFromCred (gss_cred_id_t cred, int how, struct vomsdata ∗vd, int ∗error)
- int VOMS_RetrieveFromFile (FILE ∗file, int how, struct vomsdata ∗vd, int ∗error)
- int VOMS_RetrieveFromCtx (gss_ctx_id_t ctx, int how, struct vomsdata ∗vd, int ∗error)
- int VOMS_RetrieveFromProxy (int how, struct vomsdata ∗vd, int ∗error)
- int VOMS_GetAttributeSourcesNumber (struct voms ∗v, struct vomsdata ∗vd, int ∗error)
- int VOMS_GetAttributeSourceHandle (struct voms ∗v, int num, struct vomsdata ∗vd, int ∗error)
- const char ∗ VOMS_GetAttributeGrantor (struct voms ∗v, int handle, struct vomsdata ∗vd, int ∗error)
- int VOMS_GetAttributesNumber (struct voms ∗v, int handle, struct vomsdata ∗vd, int ∗error)
- int VOMS_GetAttribute (struct voms ∗v, int handle, int num, struct attribute ∗at, struct vomsdata ∗vd, int ∗error)
- vomsdata ∗ VOMS_Duplicate (struct vomsdata ∗vd)
- AC ∗ VOMS_GetAC (struct voms ∗v)
- int getMajorVersionNumber (void)
- int getMinorVersionNumber (void)
- int getPatchVersionNumber (void)
- int VOMS_SetVerificationTime (time_t verificationtime, struct vomsdata ∗vd, int ∗error)
- char ∗∗ VOMS_GetTargetsList (struct voms ∗v, struct vomsdata ∗vd, int ∗error)
- void VOMS_FreeTargetsList (char ∗∗)
- int VOMS_SetTimeout (int t, struct vomsdata ∗vd, int ∗error)

4.1.1 Define Documentation

4.1.1.1 #define RECURSE_CHAIN 0

Definition at line 104 of file voms_apic.h.

4.1.1.2 #define RECURSE_NONE 1

Definition at line 105 of file voms_apic.h.

4.1.1.3 #define TYPE_CUSTOM 2

result of an S command

Definition at line 76 of file voms_apic.h.

4.1.1.4 #define TYPE_NODATA 0

no data

Definition at line 74 of file voms_apic.h.

4.1.1.5 #define TYPE_STD 1

group, role, capability triplet

Definition at line 75 of file voms_apic.h.

4.1.1.6 #define VERIFY_CERTLIST 0x00000040

Definition at line 115 of file voms_apic.h.

4.1.1.7 #define VERIFY_DATE 0x00000001

Definition at line 109 of file voms_apic.h.

4.1.1.8 #define VERIFY_FULL 0xffffffff

Definition at line 107 of file voms_apic.h.

4.1.1.9 #define VERIFY_ID 0x00000020

Definition at line 114 of file voms_apic.h.

4.1.1.10 #define VERIFY_KEY 0x00000004

Definition at line 111 of file voms_apic.h.

4.1.1.11 #define VERIFY_NONE 0x00000000

Definition at line 108 of file voms_apic.h.

4.1.1.12 #define VERIFY_NOTARGET 0x00000002

Definition at line 110 of file voms_apic.h.

4.1.1.13 #define VERIFY_ORDER 0x00000010

Definition at line 113 of file voms_apic.h.

4.1.1.14 #define VERIFY_SIGN 0x00000008

Definition at line 112 of file voms_apic.h.

4.1.1.15 #define VERR_COMM 3

Server problem

Definition at line 122 of file voms_apic.h.

4.1.1.16 #define VERR_DIR 13

Directory error

Definition at line 133 of file voms_apic.h.

4.1.1.17 #define VERR_EXTRAINFO 9

VO name and URI missing

Definition at line 129 of file voms_apic.h.

4.1.1.18 #define VERR_FORMAT 10

Wrong data format

Definition at line 130 of file voms_apic.h.

4.1.1.19 #define VERR_IDCHECK 8

User data in extension different from the real ones

Definition at line 127 of file voms_apic.h.

4.1.1.20 #define VERR_MEM 16

Memory problems

Definition at line 136 of file voms_apic.h.

4.1.1.21 #define VERR_NODATA 11

Empty extension

Definition at line 131 of file voms_apic.h.

4.1.1.22 #define VERR_NOEXT 5

VOMS extension missing

Definition at line 124 of file voms_apic.h.

4.1.1.23 #define VERR_NOIDENT 2

Cannot identify itself (certificate problem)

Definition at line 121 of file voms_apic.h.

4.1.1.24 #define VERR_NOINIT 6

Initialization error

Definition at line 125 of file voms_apic.h.

4.1.1.25 #define VERR_NONE 0

Error codes.

Definition at line 119 of file voms_apic.h.

4.1.1.26 #define VERR_NOSOCKET 1

Socket problem

Definition at line 120 of file voms_apic.h.

4.1.1.27 #define VERR_NOTAVAIL 21

Method not available

Definition at line 141 of file voms_apic.h.

4.1.1.28 #define VERR_ORDER 19

Ordering different than required

Definition at line 139 of file voms_apic.h.

4.1.1.29 #define VERR_PARAM 4

Wrong parameters

Definition at line 123 of file voms_apic.h.

4.1.1.30 #define VERR_PARSE 12

Parse error

Definition at line 132 of file voms_apic.h.

4.1.1.31 #define VERR_SERVER 15

Unidentifiable VOMS server

Definition at line 135 of file voms_apic.h.

4.1.1.32 #define VERR_SERVERCODE 20

Error from the server

Definition at line 140 of file voms_apic.h.

4.1.1.33 #define VERR_SIGN 14

Signature error

Definition at line 134 of file voms_apic.h.

4.1.1.34 #define VERR_TIME 7

Error in time checking

Definition at line 126 of file voms_apic.h.

4.1.1.35 #define VERR_TYPE 18

Returned data of unknown type

Definition at line 138 of file voms_apic.h.

4.1.1.36 #define VERR_VERIFY 17

Generic verification error

Definition at line 137 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.

hostname FQDN of the VOMS server

port the port on which the VOMS server is listening

servsubject the subject of the server's certificate

command the command sent to the server

version is the version number of the data.

vd RETURN PARAMETER: contains the data returned by the connection

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

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 as a copy of another voms structure.

Parameters:

v The structure to copy.

error RETURN PARAMETER: qualifies the error message.

Returns:

NULL (error) or the new voms structure.

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 as a copy of another vomsdata structure.

Parameters:

vd The structure to copy.

error RETURN PARAMETER: qualifies the error message.

Returns:

NULL (error) or the new vomsdata structure.

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 struct vomsdata∗ VOMS_Duplicate (struct vomsdata ∗ *vd*)

4.1.2.14 char∗ VOMS_ErrorMessage (struct vomsdata ∗ *vd*, int *error*, char ∗ *buffer*, int *len*)

Gets a textual description of the error.

Parameters:

vd The vomsdata structure to analyze

error The error returned by the last function

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

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

Returns:

A pointer to the error message.

4.1.2.15 int VOMS_Export (char ∗∗ *buffer*, int ∗ *buflen*, struct vomsdata ∗ *vd*, int ∗ *error*)

Converts data into a test format

Parameters:

buffer OUTPUT PARAMETER contains the converted data

buflen OUTPUT PARAMETER contains the length of buffer

vd contains the data to convert

error RETURN PARAMETER Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.16 struct contactdata∗∗ VOMS_FindByAlias (struct vomsdata ∗ *vd*, char ∗ *alias*, char ∗ *system*, char ∗ *user*, int ∗ *error*)

Gets a list of VOMS servers which share an alias.

Parameters:

vd The correctly initialized vomsdata structured.

alias The alias to look for.

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

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

error RETURN PARAMETER: qualifies the error message.

Returns:

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

4.1.2.17 struct contactdata** VOMS_FindByVO (struct vomsdata * *vd*, char * *vo*, char * *system*, char * *user*, int * *error*)

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

Parameters:

vd The correctly initialized vomsdata structured.

vo The VO to look for.

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

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

error RETURN PARAMETER: qualifies the error message.

Returns:

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

4.1.2.18 void VOMS_FreeTargets (struct vomsdata * *vd*, int * *error*)

Delete the targets from the AC.

Parameters:

vd The vomsdata structure.

error RETURN PARAMETER: qualifies the error message.

4.1.2.19 void VOMS_FreeTargetsList (char ∗∗)

4.1.2.20 AC∗ VOMS_GetAC (struct voms ∗ *v*)

4.1.2.21 int VOMS_GetAttribute (struct voms ∗ *v*, int *handle*, int *num*, struct attribute ∗ *at*, struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.22 const char∗ VOMS_GetAttributeGrantor (struct voms ∗ *v*, int *handle*, struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.23 int VOMS_GetAttributesNumber (struct voms ∗ *v*, int *handle*, struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.24 int VOMS_GetAttributeSourceHandle (struct voms ∗ *v*, int *num*, struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.25 int VOMS_GetAttributeSourcesNumber (struct voms ∗ *v*, struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.26 char∗∗ VOMS_GetTargetsList (struct voms ∗ *v*, struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.27 int VOMS_Import (char ∗ *buffer*, int *buflen*, struct vomsdata ∗ *vd*, int ∗ *error*)

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

The function does verify the data.

Parameters:

buffer contains the data to be converted

buflen contains the length of buffer

vd RETURN PARAMETER: contains the data returned by the connection

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.28 struct vomsdata∗ VOMS_Init (char ∗ *voms*, char ∗ *cert*)

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

Parameters:

voms The directory which contains the certificates of the VOMS servers

cert The directory which contains the CA certificates

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

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

Returns:

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

4.1.2.29 char∗ VOMS_ListTargets (struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.30 int VOMS_Ordering (char ∗ *order*, struct vomsdata ∗ *vd*, int ∗ *error*)

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

Parameters:

order the group:role attribute.

vd RETURN PARAMETER: contains the modified data.

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.31 int VOMS_ResetOrder (struct vomsdata ∗ *vd*, int ∗ *error*)

Unsets the return order of the attributes.

Parameters:

vd RETURN PARAMETER: contains the modified data.

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.32 int VOMS_Retrieve (X509 ∗ *cert*, STACK_OF(X509)∗ *chain*, int *how*, struct vomsdata ∗ *vd*, int ∗ *error*)

Extracts the VOMS extension from an X.509 certificate.

The function doesn't check the validity of the certificates, but it does check the content of the user data.

Parameters:

cert The certificate with the VOMS extensions

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

how Recursion type

vd RETURN PARAMETER: contains the data returned by the connection

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.33 int VOMS_RetrieveEXT (X509_EXTENSION * *ext*, struct vomsdata * *vd*, int * *error*)

Gets VOMS information from the given extension

Parameters:

ext The extension to parse.

vd RETURN PARAMETER: contains the data returned by the connection

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.34 int VOMS_RetrieveFromCred (gss_cred_id_t *cred*, int *how*, struct vomsdata * *vd*, int * *error*)

Gets VOMS information from the given globus credential

Parameters:

cred The credential from which to retrieve the certificate.

how Recursion type

vd RETURN PARAMETER: contains the data returned by the connection

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.35 int VOMS_RetrieveFromCtx (gss_ctx_id_t *ctx*, int *how*, struct vomsdata * *vd*, int * *error*)

Gets VOMS information from the given globus context

Parameters:

ctx The context from which to retrieve the certificate.

how Recursion type

vd RETURN PARAMETER: contains the data returned by the connection

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.36 int VOMS_RetrieveFromFile (FILE ∗ *file*, int *how*, struct vomsdata ∗ *vd*, int ∗ *error*)

Gets VOMS information from the given globus credential

Parameters:

file The file from which to retrieve the certificate.

how Recursion type

vd RETURN PARAMETER: contains the data returned by the connection

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.37 int VOMS_RetrieveFromProxy (int *how*, struct vomsdata ∗ *vd*, int ∗ *error*)

Gets VOMS information from an existing globus proxy

Parameters:

how Recursion type

vd RETURN PARAMETER: contains the data returned by the connection

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.38 int VOMS_SetLifetime (int *length*, struct vomsdata ∗ *vd*, int ∗ *error*)

Set requested lifetime for VOMS_Contact() calls.

Parameters:

length Lifetime requested.

vd RETURN PARAMETER: contains the modified data.

error RETURN PARAMETER: Qualifies the error message

Returns:

failure (0) or success (<>0)

4.1.2.39 int VOMS_SetTimeout (int *t*, struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.40 int VOMS_SetVerificationTime (time_t *verificationtime*, struct vomsdata ∗ *vd*, int ∗ *error*)

4.1.2.41 int VOMS_SetVerificationType (int *type*, struct vomsdata ∗ *vd*, int ∗ *error*)

Index