

EGEE-II

VOMS ADMIN USER'S GUIDE

Document identifier: VOMS-Admin-Users-Guide.odt

Date: **11/04/08**

Activity:

Lead Partner:

Document status: **DRAFT**

Document link:

Abstract:

Copyright notice:

Copyright © Members of the EGEE-II Collaboration, 2006.

See www.eu-egee.org for details on the copyright holders.

EGEE-II (“Enabling Grids for E-science-II”) is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 6th Framework Programme. EGEE-II began in April 2006 and will run for 2 years.

For more information on EGEE-II, its partners and contributors please see www.eu-egee.org

You are permitted to copy and distribute, for non-profit purposes, verbatim copies of this document containing this copyright notice. This includes the right to copy this document in whole or in part, but without modification, into other documents if you attach the following reference to the copied elements: “Copyright © Members of the EGEE-II Collaboration 2006. See www.eu-egee.org for details”.

Using this document in a way and/or for purposes not foreseen in the paragraph above, requires the prior written permission of the copyright holders.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE EGEE-II COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Trademarks: EGEE and gLite are registered trademarks held by CERN on behalf of the EGEE collaboration. All rights reserved"

Delivery Slip

	Name	Partner/Activity	Date	Signature
From	Andrea Ceccanti			
Reviewed by				
Approved by				

Document Log

Issue	Date	Comment	Author/Partner
0-0			

Document Change Record

Issue	Item	Reason for Change

TABLE OF CONTENTS

1. INTRODUCTION.....	4
1.1. SERVICE ARCHITECTURE.....	4
1.2. INTERACTION WITH OTHER SERVICES.....	4
2. QUICK START GUIDE.....	5
2.1. PREREQUISITES	5
2.1.1. Java.....	5
2.1.2. Tomcat.....	5
2.1.3. Database backend.....	5
2.2. UPGRADING AN EXISTING VOMS ADMIN 1.2.19 INSTALLATION.....	5
2.3. CREATING A NEW VO.....	5
2.3.1. MySQL VO Configuration.....	5
2.3.2. Oracle VO configuration.....	7
2.3.3. Deploying the database.....	8
2.3.4. Starting up the VOMS core service.....	8
2.3.5. Starting up the VOMS Admin service.....	8
2.3.6. Adding yourself as a VO Administrator.....	8
2.3.7. Testing the service.....	9
3. VOMS ADMIN AUTHORIZATION FRAMEWORK.....	10
3.1. ACL INHERITANCE AND DEFAULT ACL.....	10
3.2. VOMS OPERATIONS AND REQUIRED PERMISSIONS.....	11
4. THE VOMS ADMIN WEB INTERFACE.....	14
4.1. THE VO MANAGEMENT SECTION.....	15
4.1.1. Managing user VO membership.....	15
4.1.2. Managing VOMS groups.....	16
4.1.3. Managing VOMS Roles.....	17
4.1.4. VOMS generic attributes.....	17
4.2. THE SUBSCRIPTIONS SECTION.....	18
4.3. THE CONFIGURATION SECTION.....	19
4.4. THE “OTHERS VOS” SECTION.....	19
5. THE VOMS ADMIN COMMAND LINE UTILITIES.....	20
5.1. THE VOMS ADMIN COMMAND LINE CLIENT.....	20
5.1.1. voms-admin commands.....	21
5.2. THE VOMS-ADMIN-CONFIGURE COMMAND.....	26
5.2.1. Removing a VO.....	26
5.2.2. Upgrading an existing voms-admin 1.2.19 VO.....	27
5.3. THE VOMS-DB-DEPLOY.PY COMMAND.....	27
5.4. THE INIT-VOMS-ADMIN.PY COMMAND.....	27
6. VOMS ADMIN CONFIGURATION FILES.....	28

1. INTRODUCTION

The VOMS Admin service is a web application providing tools for administering member databases for VOMS, the Virtual Organization Membership Service.

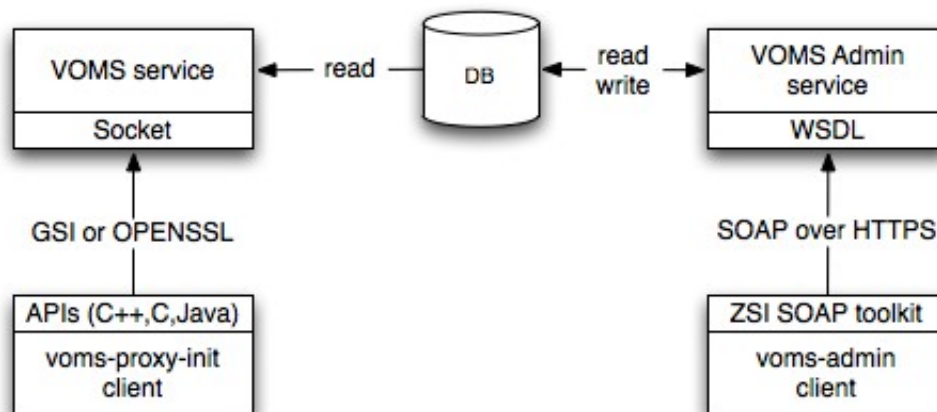
VOMS serves as a central repository for user authorization information, providing support for sorting users into a general group hierarchy, keeping track of their roles, etc. Its functionality may be compared to that of a Kerberos KDC server.

VOMS Admin provides an intuitive web user interface for daily administration tasks, and a SOAP interface for remote clients. The Admin package includes a simple command-line SOAP client that is useful for automating frequently occurring batch operations, or simply to serve as an alternative to the full-blown web interface. It is also useful for bootstrapping the service.

1.1. SERVICE ARCHITECTURE

The figure below show a high-level overview of the main components of a VOMS server. The Admin component implements a comprehensive SOAP application program interface for VO membership management.

The `voms-proxy-init` command contacts the standalone `vomsd` process that queries the authorization database and generates the actual VOMS attribute certificates



1.2. INTERACTION WITH OTHER SERVICES

The VOMS Admin service is not capable of generating VOMS Attribute Certificates itself, so it is not able to provide VO login services. It relies on a separate service (**org.glite.security.voms**) to do this task. In this document, we will refer to this service as the VOMS Core service, or simply core service for brevity.

2. QUICK START GUIDE

This section provides a step-by-step guide to configure and run VOMS Admin.

2.1. PREREQUISITES

2.1.1. Java

In order to run VOMS Admin you will need Java version 5 SDK installed on your system. Details on how to install Java on a Scientific Linux 4 machine can be found here:

https://edms.cern.ch/file/818502/3.1/gLite_3.1_VOMS_Installation_Configuration_guide.pdf

2.1.2. Tomcat

VOMS Admin is a J2EE Web application that runs on Tomcat 5. Moreover, VOMS Admin leverages the glite Trustmanager to implement X509 authentication, so you will need a Trustmanager-enabled Tomcat instance running.

Details on how to install Tomcat on a Scientific Linux 4 machine can be found here:

https://edms.cern.ch/file/818502/3.1/gLite_3.1_VOMS_Installation_Configuration_guide.pdf

Details on how to install and configure Trustmanager can be found in the Trustmanager documentation.

2.1.3. Database backend

VOMS Admin supports Oracle and MySQL database backends. Details on how to install the database backends on a Scientific Linux 4 machine can be found here:

https://edms.cern.ch/file/818502/3.1/gLite_3.1_VOMS_Installation_Configuration_guide.pdf

2.2. CREATING A NEW VO

Two database backends are currently supported by voms-admin: MySQL and Oracle. You can configure and create a new VO using the `voms-admin-configure` configuration script.

2.2.1. MySQL VO Configuration

The MySQL VO installation procedure depends on whether a MySQL database has already been created for you by you MySQL administrator or you want to create it when configuring voms for the first time.

Usually, you do not have a dedicated MySQL administrator working for you, so you will use voms-admin tools to create the database schema, configure the accounts and deploy the voms database.

If this is the case, you need to run the following command:

```
voms-admin-configure install --dbtype mysql
--vo <VO name>
--createdb
--deploy-database
```

```
--dbauser <MySQL root username>
--dbapwd <MySQL root password>

--dbusername <voms db account username>
--dbpassword <voms db account password>

--port <voms core service port>

--smtp-host <SMTP relay host>
--mail-from <Sender address for service-generated emails>
```

Note that the above command is entered as a single command; it has been broken up into multiple lines for clarity. The command creates and initializes a VOMS database, and configures the VOMS core and admin services that use such database. The required options are described below:

Option name	Meaning
createdb	This option is MySQL specific and is used to specify that the MySQL database for VOMS must be created by the <code>voms-admin-configure</code> script.
deploy-database	This option tells the script that it must create the tables for VOMS and fill in the necessary bootstrap information (e.g., admin accounts, supported CAs, ...)
dbauser, dbapwd	These options are MySQL specific and are used to set the MySQL root user account username and password respectively. These credentials are needed to create the MySQL database for VOMS, and thus required when the <code>createdb</code> option is set. If MySQL is configured with an empty password for the root account, the <code>dbapwd</code> option may be omitted.
dbusername, dbpassword	These options are used to specify the MySQL account that VOMS will use when contacting the database. If the <code>createdb</code> option is set, <code>voms-admin-configure</code> creates the account for you.
port	This option specifies on which port the VOMS core server will listen for requests.
mail-from, smtp-host	These options specify, respectively, the address that must be used for service-generated emails and the SMTP service that must be used to send them.

An example MySQL VO installation command is shown below:

```
$GLITE_LOCATION/sbin/voms-admin-configure install --dbtype mysql \  
--vo test_vo_mysql --createdb --deploy-database \  
--dbauser root --dbapwd pwd \  
--dbusername voms_admin_20 --dbpassword pwd \  
--port 54322 --mail-from ciccio@cnaf.infn.it \  
-smtp-host iris.cnaf.infn.it
```

2.2.2. Oracle VO configuration

Oracle VO configuration is different from MySQL configuration. In Oracle you need to setup the database account for VOMS before launching voms-admin configure. Moreover, Oracle instant client libraries must be installed and configured before running voms-admin configuration.

Once you have configured Oracle stuff, you can install a new Oracle VO using the following command:

```
voms-admin-configure install --dbtype oracle  
--vo <VO name>  
  
--dbname <TNS alias of the database backend>  
--deploy-database  
  
--dbusername <voms db account username>  
--dbpassword <voms db account password>  
  
--port <voms core service port>  
  
--smtp-host <SMTP relay host>  
--mail-from <Sender address for service-generated emails>
```

Note that the above command is entered as a single command; it has been broken up into multiple lines for clarity. This command is indeed very similar to the one used to configure a MySQL VO. The main difference lies in the dbname option, that is used to specify the TNS alias for the Oracle database backend. This TNS alias is needed to build the connection string that VOMS will use to communicate with the database backend. Usually, TNS aliases are maintained in the tnsnames.ora file, located in a directory that is usually exported to applications via the TNS_ADMIN Oracle environment variable. For more information regarding TNS aliases, consult the Oracle online documentation (<http://www.oracle.com/pls/db102/homepage>).

An example Oracle VO installation command is shown below:

```
voms-admin-configure install --dbtype oracle \  
--vo test_vo --dbname test --deploy-database \  

```

```
--dbusername voms_admin_20 --dbpassword pwd \  
--dbhost datatag6.cnaf.infn.it --port 54321 \  
--mail-from ciccio@cnaf.infn.it --smtp host iris.cnaf.infn.it
```

2.2.3. Deploying the database

When configuring a VO for the first time on a machine, `voms-admin-configure` by default tries to deploy the database, unless the `skip-database` option is set. Before overwriting tables and information, `voms-admin-configure` checks whether an existing VOMS database is already deployed. In case one is found, a warning is issued and the database is not touched by the installation procedure.

2.2.4. Starting up the VOMS core service

After a successful configuration, you can start the VOMS core service by typing the following command:

```
$GLITE_LOCATION/etc/init.d/voms start
```

2.2.5. Starting up the VOMS Admin service

You can deploy the just configured VO to Tomcat by typing the following command:

```
$GLITE_LOCATION/etc/init.d/voms-admin start
```

(If you have created other VOs and want to start only one of them, list the VO name to start at the end of the command.)

Provided that Tomcat is running, you should now have a VOMS Admin service deployed, and ready to serve requests. If you forgot to start Tomcat, do it now. You do not need to type in the above command again unless you explicitly undeploy the service later, or create new VOs. Tomcat will automatically remember to run your VOMS Admin service across server reboots.

2.2.6. Adding yourself as a VO Administrator

VOMS-Admin provides two ways of adding yourself as an administrator for VO. You can either add yourself as a VO user and assign to yourself the `VO-Admin` role, or use the `voms-db-deploy.py` command to interact directly with the voms database.

2.2.6.1. Using the `voms-db-deploy.py` script

In case you have root access on the machine where you are configuring VOMS/VOMS-Admin¹, you can use the `voms-db-deploy.py` command to add yourself as administrator.

```
$GLITE_LOCATION/sbin/voms-db-deploy.py add-admin  
--vo <VO name>  
--cert <certificate>
```

¹or you are configuring VOMS/VOMS-Admin to run as user services and you have the necessary permissions to access the configuration files

where VO name is the name of one of the VO you have configured, and certificate is an X509 certificate in PEM format.

2.2.6.2. Using the VOMS Admin client

In case the VO is already activated (to know how a VO can be activated, see Section 2.3.5), you can add yourself as an administrator using the `voms-admin` command:

```
voms-admin --vo <VO name> --usercert <certificate>  
create-user <certificate> assign-role VO VO-Admin
```

where VO name is the name of one of the VO you have configured, and certificate is an X509 certificate in PEM format.

2.2.7. Testing the service

To test whether the `voms-admin` service is active for your VO, you can point your browser to the following URL:

```
https://<voms-admin server hostname>:8443/voms/<VO name>
```

To get a list of all the VOs configured on the host, use the following URL:

```
https://<voms-admin server hostname>:8443/vomses
```

Note that you must have a suitable certificate already imported in your browser to access the `voms-admin` interface.

3. VOMS ADMIN AUTHORIZATION FRAMEWORK

In VOMS-Admin, each operation that access the VOMS database is authorized via the VOMS-Admin Authorization framework. For instance, only authorized admins have the rights to add users or create groups for a specific VO.

More specifically, Access Control Lists (ACLs) are linked to VOMS contexts to enforce authorization decisions on such contexts. In this framework, a **Context** is either a VOMS group, or a VOMS role within a group. Each **Context** as an ACL, which is a set of ACL entries, i.e., (**VOMS Administrator**, **VOMSPermission**) couples.

A **VOMS Administrator** may be:

- A VO administrator registered in the VO VOMS database;
- A VO user;
- A VOMS FQAN;
- Any authenticated user (i.e., any user who presents a certificate issued by a trusted CA).

A **VOMS Permission** is a fixed-length sequence of permission flags that describe the set of permissions a VOMS Administrator has in a specific context. The following table explains in detail the name and meaning of these permission flags:

<i>Permissions</i>	<i>Use</i>
CONTAINER_READ CONTAINER_WRITE	These flags are used to control access to the operations that list/alter the VO internal structure (groups and roles list/creations/deletions, user creations/deletions).
MEMBERSHIP_READ MEMBERSHIP_WRITE	These flags are used to control access to operations that manage/list membership in group and roles.
ATTRIBUTES_READ ATTRIBUTES_WRITE	These flags are used to control access to operations that manage generic attributes (at the user, group, or role level).
ACL_READ ACL_WRITE ACL_DEFAULT	These flags are used to control access to operations that manage VO ACLs and default ACLs
REQUESTS_READ REQUESTS_WRITE	These flags are used to control access to operations that manage subscription requests regarding the VO, group membership, role assignment etc...

Each operation on the VOMS database is authorized according to the above set of permissions, i.e., whenever an administrator tries to execute such operation, its permissions are matched with the operation's set of required permission in order to authorize the operation execution.

3.1. ACL INHERITANCE AND DEFAULT ACL

Children groups, at creation time, inherit parent's group ACL. However, VOMS Admin implements an override mechanism for this behaviour via *Default ACLs*. When the Default ACL is defined for a group, children groups inherit the Default ACL defined at the parent level instead of the parent's group ACL. So, Default ACLs are useful only if an administrator wants the ACL of children groups to be *different* from the one of the parent's group.

3.2. VOMS OPERATIONS AND REQUIRED PERMISSIONS

In the following, we describe the required permissions for the most common voms-admin operations according to this notation:

Symbol	Meaning
/vo	The VO root group
(g,R)	The context identified by role R within group g
(g→ g')	All the voms groups that lie in the path from group g to group g' included according to the parent-child relation defined between voms groups.
parent(g)	Group g's parent group.
r	Read permission
w	Write permission
d	default permission (applies only to ACL permissions)
C:	Container permissions,
M:	Membership permissions
Attrs:	Attributes permissions
Acl:	Acl permissions
Req:	Requests permissions

In the table below, operations are listed on the left, while required permissions, in the form of (Voms context, Voms Permission) couples are listed on the right.

Operation	RequiredPermission
Create/Delete user	(/vo,C:rw M:rw)
Create/Delete Group g	(/vo, C:rw) (/vo → parent(parent(g)), C:r) (parent(g), C:rw)
List subgroups	(/vo → g, C:r)
Create/Delete Role	(/vo, C:rw)

Operation	RequiredPermission
List roles	(/vo, C:r)
Add remove/member to group g	(/vo → parent(g) , C:r) (g, M:rw)
List group g members	(/vo → parent(g) , C:r) (g, M:rw)
Assign/Dismiss role R in group g	(/vo → parent(g) , C:r) ((g,R), M:rw)
List members with role R in group g	(/vo → parent(g) , C:r) ((g,R), M:r)
Set/Delete user attribute	(/vo, Attrs:rw)
List user attributes	(/vo, Attrs:r)
Set/Delete group attributes	(/vo → parent(g) , C:r) (/vo, Attrs: rw) (g, Attrs:rw)
List group attributes	(/vo → parent(g) , C:r) (/vo, Attrs: r) (g, Attrs:r)
Set/Delete Role attributes	(/vo → parent(g) , C:r) (/vo, Attrs: rw) ((g,R), Attrs:rw)
List Role attributes	(/vo → parent(g) , C:r) (/vo, Attrs: r) ((g,R), Attrs:r)
Edit ACL for group g	(/vo → parent(g) , C:r) (g, Acl:rw)
List ACL for group g	(/vo → parent(g) , C:r) (g, Acl:r)
Edit ACL for role R in group g	(/vo → parent(g) , C:r) ((g,R), Acl:rw)
List ACL for role R in group g	(/vo → parent(g) , C:r) ((g,R), Acl:r)
Edit default ACL for group g	(/vo → parent(g) , C:r)

Operation	RequiredPermission
	(g, Acl:rwd)
List default ACL for group g	(/vo → parent(g) , C:r) (g, Acl:rd)
Edit default ACL for role R in group g	(/vo → parent(g) , C:r) ((g,R), Acl:rwd)
List default ACL for role R in group g	(/vo → parent(g) , C:r) ((g,R), Acl:rd)

4. THE VOMS ADMIN WEB INTERFACE

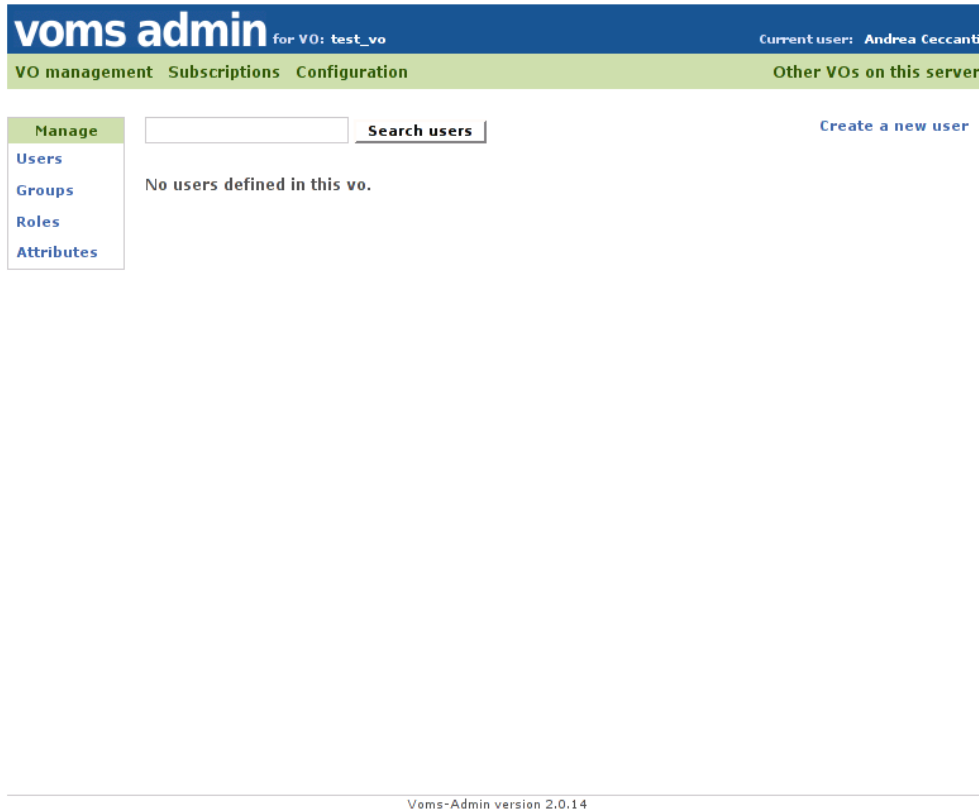


Figure 1: The Voms admin 2.0 web interface

The VOMS-Admin web application provides a usable and intuitive interface towards VO management tasks. A screenshot of the main page of the web application is given in Figure 1.

In the top part of the page, the header provides information about the current user accessing the interface and the name of the VO that is being managed. The green menu bar provides access to the various sections of the web application, while the submenu on the left provides contextual access to subparts of each section.

The *VO management* subsections share a search bar that allows the administrator to search users, groups, roles and generic attributes assignments. Pagination of the search results is also implemented, as shown in the following image:

user14 Test CA,Voms-Admin	delete user
user15 Test CA,Voms-Admin	delete user
user16 Test CA,Voms-Admin	delete user
user17 Test CA,Voms-Admin	delete user

1 - 10 of 28 > last

4.1. THE VO MANAGEMENT SECTION

4.1.1. Managing user VO membership

The user management section of the VOMS-Admin web interface allows administrators to manage group membership, role assignment and generic attributes for VO users. Users access this area by clicking on a user name from the User search page or other parts of the web interface where users can be searched.

The screenshot displays the VOMS-Admin web interface for a user named Andrea Ceccanti. The interface is organized into several sections:

- Header:** "voms admin for VO: test_vo" and "Current user: Andrea Ceccanti".
- Navigation:** "VO management", "Subscriptions", "Configuration", and "Other VOs on this server".
- Manage Panel:** A sidebar with "Users", "Groups", "Roles", and "Attributes".
- User details:** Shows the user's DN & CA, common name, and email address. It includes a "delete this user" link and a "Save changes" button.
- Membership details:** Allows adding the user to a group (e.g., "/test_vo/g1") and assigning a role (e.g., "VO-Admin").
- Generic attributes management:** Allows setting a value for a specific attribute (e.g., "nickname").

At the bottom of the interface, it states "Voms-Admin version 2.0.14".

User details, such as the email address or the user's common name, can be edited by authorized administrators via the *User details* pane.

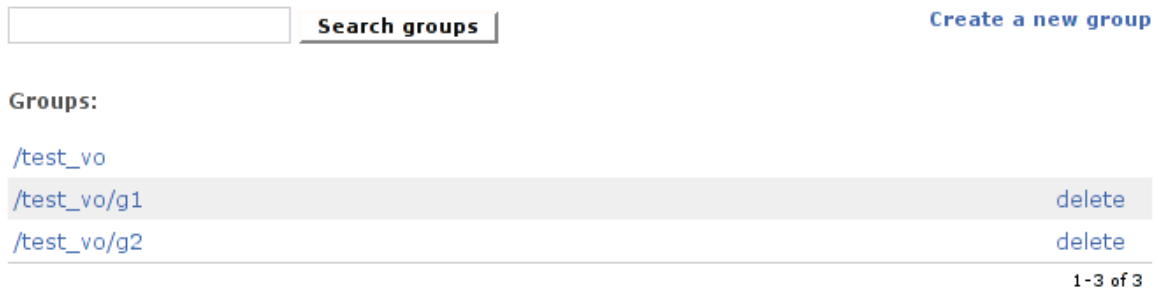
The membership details pane is used to manage group and role membership, while the Generic Attributes management pane allows to set generic attributes for a user. A more detailed explanation of Generic attributes management will be given in section 4.1.4.2.

4.1.2. Managing VOMS groups

The group management section can be accessed by clicking on the “Groups” link in the VO Management contextual submenu.



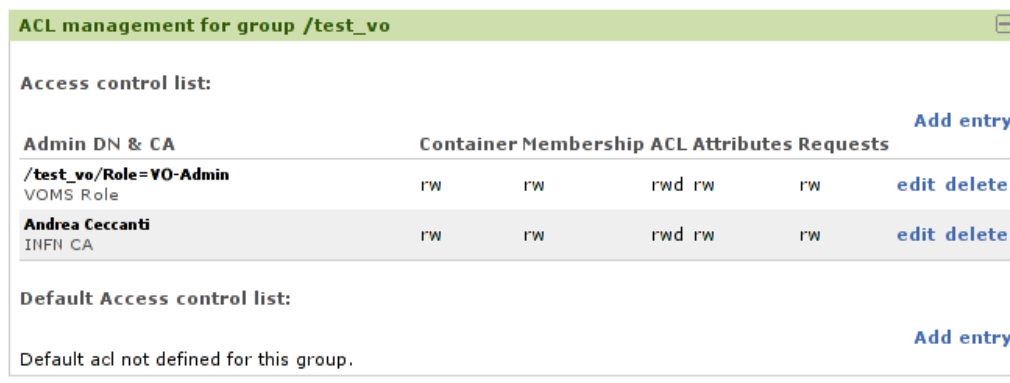
The group management section allows to search, create and remove VO groups:



By clicking on the group name, the Admin access the detailed group management section, that implements Access Control Lists (ACL) and Generic Attributes (GA) management for the group as well as search group members functionality.

4.1.2.1. ACL management

The ACL management pane of the detailed group management page implements management for group's ACL and default ACL. The ACL management pane displays ACL entries in the form of (Voms Administrator, Set of permissions) couples. The display uses the compact representation for VOMS permissions that has been already introduced in section 3.2.



4.1.2.1.1. Managing ACL entries

ACL entries can be added to ACL and default ACLs by clicking on the “add entry” link. Permissions can be set for:

- VO users;
- non VO-users (any trusted identity);
- Anyone having a specific VO role in a specific VO group;
- Anyone belonging to a specific VO group;
- Anyone, i.e., everyone authenticated with a certificate issued by a trusted CA.

Entries added to a group ACL can be propagated to existing children groups' ACLs by ticking the “Propagate to children groups” tick box at the bottom of the page. Similarly, when editing or deleting an ACL entry from a group ACL, it is possible to propagate the deletion or editing to children groups by selecting the “Propagate to children groups” tick box.

4.1.2.1.2. ACL management examples

To grant read only access to any authenticated client (useful to support gridmap file generation) create the following ACL entry on the VO root group and propagate the entry to children contexts:

Admin DN & CA	Container Membership ACL Attributes Requests				Add entry
/test_vo/Role=VO-Admin VOMS Role	rW	rW	rwd rW	rW	edit delete
Andrea Ceccanti INFN CA	rW	rW	rwd rW	rW	edit delete
Any Authenticated User Dummy Certificate Authority	r	r			edit delete



4.1.3. Managing VOMS Roles

The role management section can be accessed by clicking on the “Roles” link in the VO Management contextual submenu.

Like the group management section, this section section allows to search, create and remove VOMS roles. By clicking on the role name, the Admin accesses the detailed role management section where generic attributes management and role members

search functionality is implemented.

4.1.4. VOMS generic attributes

Generic attributes (GAs) are (name, value) pairs that that can be assigned to VO users and that end up in the Attribute Certificate issued by VOMS. GAs extend the range of attributes that VOMS can issue besides Fully Qualified Attributes Names (FQAN), i.e., allow VOMS to issue any kind of VO membership information that can be expressed as (name, value) pairs. Such information can then be leveraged by Grid applications to take authorization decisions.

For their nature, GAs are issued to VO users. VOMS however provides a way to quickly assign GAs to all the VO members that belong to a specific VOMS group or that are assigned a specific VOMS role within a group. For this reason, you find GA management in user, group and role management pages in VOMS Admin.

To assign GA to users, the VO admin must first create the corresponding Generic Attribute class. This Generic Attribute class is used to define the name and possibly a description for the GA. VOMS Admin also implements a configurable uniqueness check on GA values that can be set when creating a GA class. This uniqueness check ensures that two users cannot share the same value for a specific GA. This check is enforced at the GA class level, so you can have GAs that are checked for uniqueness and others that allow users to share the same value for the same GA.



4.1.4.1. Generic Attribute Classes Management

The GA classes management page can be reached by clicking on the “Attributes” link in the VO Management contextual submenu, and then clicking on the “Manage attribute classes” link. GA classes can then be created, specifying the GA name, description and whether uniqueness must be enforced on the GA values assigned directly to users.

4.1.4.2. Managing Generic Attributes for users and groups and roles

Once a GA class has been created, GA values can be assigned to users, groups and role within groups. As mentioned above, when one GA is assigned directly to a user, the (name,value) couple is added by VOMS to the attribute certificate returned to user. When a GA is assigned to a group, or role within a group, such (name, value) pair ends up in the Attribute Certificate of all the VO members belonging to that group.

Generic attributes management
⊞

Attribute:

Attribute value:

[Set an attribute](#)

Attribute list:

Attribute name	Attribute value	
nickname	andrea's nickname	delete

4.1.4.3. Searching GA assignments

VOMS Admin implements search over user GA assignments, so that an administrator can easily know the status of GA assignments. The search functions deal only with GA assigned directly to user, i.e., group and role assignments search and centralized display is currently not supported.

Search user attributes

Manage attribute classes

Attribute name	Attribute value	User DN & CA
nickname	andrea's nickname	Andrea Ceccanti INFN CA,INFN
nickname	user's 15 nickname	user15 Test CA,Voms-Admin

1-2 of 2

4.2. THE SUBSCRIPTIONS SECTION

See	Pending VO Membership requests			
Pending requests	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%; padding: 2px;">user1770 Test CA,Voms-Admin</td> <td style="width: 20%; text-align: center; padding: 2px;">reject</td> <td style="width: 20%; text-align: center; padding: 2px;">approve</td> </tr> </table>	user1770 Test CA,Voms-Admin	reject	approve
user1770 Test CA,Voms-Admin	reject	approve		
Processed requests				

VOMS Admin implements a simple VO registration service. To request VO membership, an aspiring VO member need to point his browser (loaded with his/her certificate) to the following URL:

<https://<voms-admin server hostname>:8443/voms/<VO name>>

If the user's certificate is not yet registered inside the VOMS database, VOMS Admin recognizes the contacting user as an aspiring VO member, and presents a registration form that the user needs to fill out in order to proceed with the registration. After a successful email confirmation by the user, VO administrators receive an email containing information regarding the pending VO membership that points them to the "Subscriptions management section".

Pending VO membership requests can be approved or rejected from this section.

4.3. THE CONFIGURATION SECTION

The Configuration section is comprised of a single page showing configuration information like *vomses* string for the contacted VO or *mkgridmap* example configuration. An example of this page is given below:

Configuration information

VOMS-Admin URL for this vo:

`https://voms112.cern.ch:8443/voms/test`

VOMSES string for this vo:

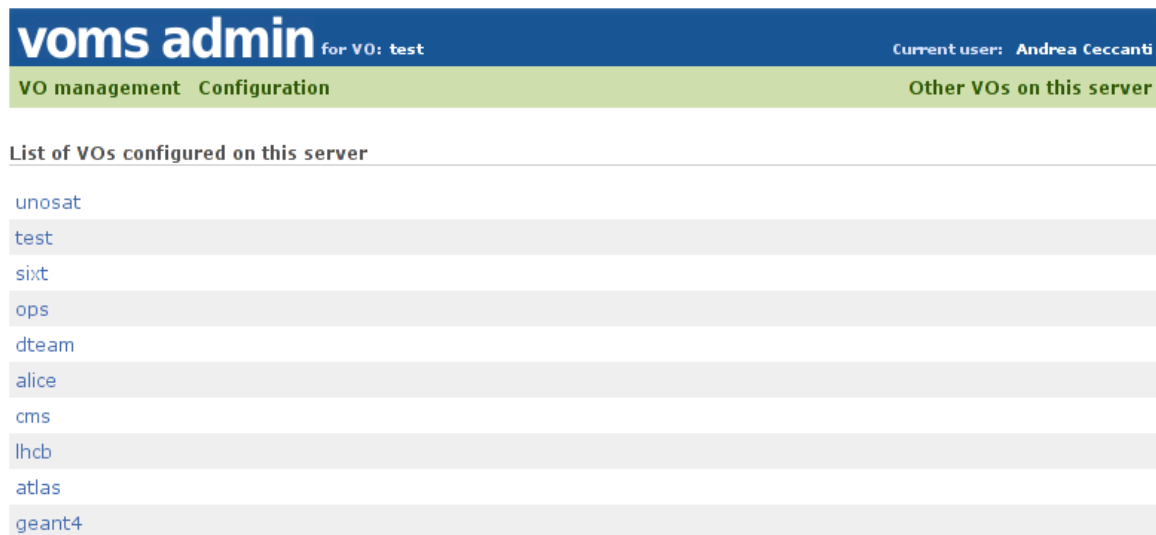
`"test" "voms112.cern.ch" "15019" "/DC=ch/DC=cern/OU=computers/CN=voms112.cern.ch" "test"`

Example Mkgridmap configuration for this vo:

`group voms://voms112.cern.ch:8443/voms/test .test`

4.4. THE "OTHERS VOS" SECTION

This section provides links to the other VOs configured on the server.



The screenshot shows the 'voms admin' interface for VO: test. The current user is Andrea Ceccanti. The interface has two main tabs: 'VO management Configuration' and 'Other VOs on this server'. The 'Other VOs on this server' tab is active, displaying a list of VOs configured on the server:

- unosat
- test
- sixt
- ops
- dteam
- alice
- cms
- lhcb
- atlas
- geant4

5. THE VOMS ADMIN COMMAND LINE UTILITIES

5.1. THE VOMS ADMIN COMMAND LINE CLIENT

VOMS Admin comes with a python command line client utility, called `voms-admin`, that can be used to perform all the operations on the VOMS database that are implemented by the Web interface.

`voms-admin` uses the UNIX effective user ID to choose which X509 credential it must use to connect to a (possibly remote) VOMS Admin instance. When ran as root, `voms-admin` uses the host credentials found in `/etc/gridsecurity`.

When running as a normal user, `voms-admin` does the following:

- If a proxy exists in `/tmp`, the proxy is used,
- otherwise if the `X509_USER_PROXY` environment variable is set, `voms-admin` uses the credentials pointed by such environment variable,
- otherwise if the `X509_USER_CERT` environment variable is set, `voms-admin` uses the credentials pointed by `X509_USER_CERT` and `X509_USER_KEY` environment variables,
- otherwise the `usercert.pem` and `userkey.pem` credentials from the `$HOME/.globus` are used.

A user can get the list of supported commands by typing:

```
voms-admin --list-commands
```

A user can get help about the commands provided by `voms-admin` by typing:

```
voms-admin --help-commands
```

Detailed help about individual commands can be obtained issuing the following command:

```
voms-admin -help-command <command_name>
```

For example, asking help about the `create-user` command produces the following output:

```
andrea@pcceccanti:~$ voms-admin --help-command create-user
create-user CERTIFICATE.PEM

Registers a new user in VOMS.

If you use the --nousercert option, then four parameters are required (DN
CA CN MAIL) to create the user. Otherwise these parameters are extracted
automatically from the certificate.

Examples:

voms-admin --vo test_vo create-user .globus/usercert.pem
voms-admin --nousercert --vo test_vo create-user \
'My DN' 'My CA' 'My CN' 'My Email'
```

In the remainder of this section `voms-admin` commands will be explained in detail. The information reported here is also accessible from the `voms-admin` command leveraging the `-list-commands`, `--help-command` and `-help-commands` options.

5.1.1. VOMS-ADMIN COMMANDS

5.1.1.1. USER MANAGEMENT COMMANDS

list-users

Lists the VO users.

create-user CERTIFICATE.PEM

Registers a new user in VOMS. If you use the `--nousercert` option, then four parameters are required (DN CA CN MAIL) to create the user.

Otherwise these parameters are extracted automatically from the certificate.

Examples:

```
voms-admin --vo test_vo create-user .globus/usercert.pem  
voms-admin --nousercert --vo test_vo create-user 'My DN' 'My CA' 'My CN' 'My  
Email'
```

delete-user USER

Deletes a user from VOMS, including all their attributes and membership information.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the `--nousercert` option is set.

Examples:

```
voms-admin --vo test_vo delete-user .globus/usercert.pem  
voms-admin --nousercert --vo test_vo delete-user 'My DN' 'MY CA'
```

5.1.1.2. GROUP MANAGEMENT COMMANDS

list-groups

Lists all the groups defined in the VO.

list-sub-groups GROUPNAME

List the subgroups of GROUPNAME.

create-group GROUPNAME

Creates a new group named GROUPNAME. Note that the vo root group part of the fully qualified group name can be omitted, i.e., if the group to be created is called `/vo/ciccio`, where `/vo` is the vo root group, this command accepts both the "ciccio" and `"/vo/ciccio"` syntaxes.

delete-group GROUPNAME

Deletes a group.

5.1.1.3. GROUP MEMBERSHIP MANAGEMENT COMMANDS

add-member GROUPNAME USER

Adds USER to the GROUPNAME group.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

remove-member GROUPNAME USER

Removes USER from the GROUPNAME group.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

list-members GROUPNAME

Lists all members of a group.

list-user-groups USER

Lists the groups that USER is a member of.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

5.1.1.4. ROLE MANAGEMENT COMMANDS

list-roles

Lists the roles defined in the VO.

create-role ROLENAME

Creates a new role

delete-role ROLENAME

Deletes a role.

5.1.1.5. ROLE ASSIGNMENT COMMANDS

assign-role GROUPNAME ROLENAME USER

Assigns role ROLENAME to user USER in group GROUPNAME.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

dismiss-role GROUPNAME ROLENAME USER

Dismiss role ROLENAME from user USER in group GROUPNAME.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

list-users-with-role GROUPNAME ROLENAME

Lists all users with ROLENAME in GROUPNAME.

list-user-roles USER

Lists the roles that USER is assigned.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

5.1.1.6. ATTRIBUTE CLASS MANAGEMENT COMMANDS

create-attribute-class CLASSNAME DESCRIPTION UNIQUE

Creates a new generic attribute class named CLASSNAME, with description DESCRIPTION. UNIQUE is a boolean argument. If UNIQUE is true, attribute values assigned to users for this class are checked for uniqueness. Otherwise no checks are performed on user attribute values.

delete-attribute-class CLASSNAME

Removes the generic attribute class CLASSNAME. All the user, group and role attribute mappings will be deleted as well.

list-attribute-classes

Lists the attribute classes defined for the VO.

5.1.1.7. GENERIC ATTRIBUTE ASSIGNMENT COMMANDS

set-user-attribute USER ATTRIBUTE ATTRIBUTE_VALUE

Sets the generic attribute ATTRIBUTE value to ATTRIBUTE_VALUE for user USER.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

delete-user-attribute USER ATTRIBUTE

Deletes the generic attribute ATTRIBUTE value from user USER.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

list-user-attributes USER

Lists the generic attributes defined for user USER.

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

set-group-attribute GROUP ATTRIBUTE ATTRIBUTE_VALUE

Sets the generic attribute ATTRIBUTE value to ATTRIBUTE_VALUE for group GROUP.

set-role-attribute GROUP ROLE ATTRIBUTE ATTRIBUTE_VALUE

Sets the generic attribute ATTRIBUTE value to ATTRIBUTE_VALUE for role ROLE in group GROUP.

delete-group-attribute GROUP ATTRIBUTE

Deletes the generic attribute ATTRIBUTE value from group GROUP.

list-group-attributes GROUP

Lists the generic attributes defined for group GROUP.

list-role-attributes GROUP ROLE

Lists the generic attributes defined for role ROLE in group GROUP.

delete-role-attribute GROUP ROLE ATTRIBUTE

Deletes the generic attribute ATTRIBUTE value from role ROLE in group GROUP.

5.1.1.8. ACL MANAGEMENT COMMANDS

get-ACL CONTEXT

Gets the ACL defined for voms context CONTEXT.

CONTEXT may be either a group (e.g. /groupname) or a qualified role (e.g./groupname/Role=VO-Admin).

get-default-ACL GROUP

Gets the default ACL defined for group GROUP.

add-ACL-entry CONTEXT USER PERMISSION PROPAGATE

Adds an entry to the ACL for CONTEXT assigning PERMISSION to user/admin USER. If PROPAGATE is true, the entry is propagated to children contexts.

CONTEXT may be either a group (e.g. /groupname) or a qualified role (e.g./groupname/Role=VO-Admin).

USER is either an X509 certificate file in PEM format, or a DN, CA couple when the --nousercert option is set.

PERMISSION is a VOMS permission expressed using the VOMS-Admin 2.x format. Allowed permission values are given in section 3.2. Multiple permissions can be assigned by combining them in a comma separated list, e.g.: "CONTAINER_READ, MEMBERSHIP_READ".

Special meaning DN,CA couples (to be used with the --nousercert option set) are listed hereafter:

- If DN is ANYONE and CA is VOMS_CA, an entry will be created that assigns the specified PERMISSION to to any authenticated user (i.e., any client that authenticates with a certificates signed by a trusted CA).
- if CA is GROUP_CA, DN is interpreted as a group and entry will be assigned to members of such group.
- if CA is ROLE_CA, DN is interpreted as a qualified role (i.e., /test_vo/Role=TestRole), the entry will be assigned to VO members that have the given role in the given group.

Examples:

```
voms-admin --vo test_vo add-ACL-entry /test_vo .globus/usercert.pem ALL true
```

The above command grants full rights to the user identified by '.globus/usercert.pem' on the whole VO, since PROPAGATE is true.

```
voms-admin --nousercert --vo test_vo add-ACL-entry /test_vo 'ANYONE' 'VOMS_CA' 'CONTAINER_READ, MEMBERSHIP_READ' true
```

The above command grants READ rights on VO structure and membership to any authenticated user on the whole VO, since PROPAGATE is true.

add-default-ACL-entry GROUP USER PERMISSION

Adds an entry to the default ACL for GROUP assigning PERMISSION to user/admin USER. USER and PERMISSION have the usual meaning (see **add-ACL-entry** help on page 24).

remove-ACL-entry CONTEXT USER PROPAGATE

Removes the entry from the ACL for CONTEXT for user/admin USER. If PROPAGATE is true, the entry is removed also from children contexts.

CONTEXT and USER have the usual meaning (see **add-ACL-entry** help on page 24).

remove-default-ACL-entry GROUP USER

Removes the entry for user/admin USER from the default ACL for GROUP. USER has the usual meaning (see **add-ACL-entry** help on page 24).

5.1.1.9. OTHER COMMANDS

get-vo-name

This command returns the name of the contacted vo.

list-cas

Lists the certificate authorities accepted by the VO.

5.2. THE VOMS-ADMIN-CONFIGURE COMMAND

`voms-admin-configure` is the script that configures `voms-admin` and `voms`. Its usage has already been introduced in Section 2.2. The syntax of the command is:

```
voms-admin-configure COMMAND [OPTIONS]
```

Available commands are:

- `install`: is used to configure a VO
- `remove`: is used to unconfigure a VO
- `upgrade`: is used to upgrade the configuration of a VO installed with an older version of `voms-admin`.

Examples of installation commands have already been given in Section 2.2.

5.2.1. Removing a VO

To remove an already configured VO, the `voms-admin-configure` command is invoked as follows:

```
voms-admin-configure remove --vo VONAME
```

Options for the `remove` command are given in the following table:

<i>Option name</i>	<i>Meaning</i>
<code>--undeploy-database</code>	Undeploys the VOMS database. By default when removing a VO the database is left untouched. All the database content is lost.
<code>--dropdb (MySQL only)</code>	This flag is used to drop the mysql database schema created for MySQL installations using the <code>--createdb</code> option

An example of the `remove` command, and related output, is given:

```

$ voms-admin-configure remove --vo test_vo_mysql --undeploy-database --dropdb
voms-admin-configure, version 2.0.14
Removing vo test_vo_mysql
Dropping mysql db...

WARNING: No password has been specified for the mysql root account! I will
continue the db deployment assuming no password has been set for such account.
  
```

```
VO test_vo_mysql succesfully removed.
```

Information about other voms-admin-configure options can be obtained issuing the following command:

```
voms-admin-configure --help
```

5.2.2. Upgrading an existing voms-admin 1.2.19 VO

To upgrade a VO created (or upgraded) by voms-admin 1.2.19, one just needs to launch the

```
voms-admin-configure upgrade --vo <VO_NAME>
```

Be sure to backup the contents of the database **before** running the upgrade procedure, so if something goes wrong you will not lose any data. You can do the upgrade of the configuration files (without touching the database) by giving the --skip-database option:

```
voms-admin-configure upgrade --vo test_vo --skip-database
```

5.3. THE VOMS-DB-DEPLOY.PY COMMAND

The voms-db-deploy.py command is used to manage the deployment of the VOMS database and to add/remove administrators without requiriing voms-admin VOs to be active.

```

Usage:
voms-db-deploy.py deploy --vo [VONAME]
voms-db-deploy.py undeploy --vo [VONAME]
voms-db-deploy.py upgrade --vo [VONAME]

voms-db-deploy.py add-admin --vo [VONAME] --cert [CERT_FILE]
voms-db-deploy.py add-admin --vo [VONAME] --dn [ADMIN_DN] --ca [ADMIN_CA] \
--email [EMAILADDRESS]

voms-db-deploy.py remove-admin --vo [VONAME] --cert [CERT_FILE]
voms-db-deploy.py remove-admin --vo [VONAME] --dn [ADMIN_DN] --ca [ADMIN_CA]
  
```

5.4. THE INIT-VOMS-ADMIN.PY COMMAND

The init-voms-admin.py (linked by the \$GLITE_LOCATION/etc/init.d/voms-admin) command is used to start, stop and check the status of configured VOs.

Usage:

```
init-voms-admin.py [--context=CONTEXT_FILE] [--use-manager] start [VONAME]
init-voms-admin.py [--use-manager] (stop|reload|status) [VONAME]
init-voms-admin.py [--use-manager] (start-siblings|stop-siblings)
```

VONAME is the name of the vo.

CONTEXT_FILE is a file that contains the web application context descriptor

use-manager uses the tomcat manager application to manage vo apps.

The start-siblings and stop-siblings commands are used to start/stop the siblings webapp independently from other vos.

6. VOMS ADMIN WEB SERVICE APIS

Browsable documentation generated regarding the VOMS Admin Web service APIs can be found in the `$GLITE_LOCATION/share/doc/glite-security-voms-admin-server/web-service-apis/html` directory.