

Table of Contents

VOMS Server Installation and Configuration Guide.....	1
Service Overview.....	1
VOMS Node Installation.....	1
Upgrading from a gLite 3.0 VOMS Server.....	1
Migrating the database to the new node.....	1
Migrating the VOMS configuration to the new node.....	2
Upgrading the database schema.....	2
Updating the configuration.....	3
Reconfiguring the VOMS server.....	3
Starting the VOMS server.....	3
Configuring access for gridmap file generation.....	3
Registration of the VOMS host identity as a default admin.....	4
Verification of the ACL configuration.....	4
VOMS Server Configuration.....	4
Configuring the database backend.....	4
Configure (if you use as backend).....	4
Configure Oracle (if you use Oracle as backend).....	4
Glite configuration files.....	4
Setting the Configuration values for the VOMS server.....	5
General configuration of the VOMS server.....	5
VO configuration.....	6
Setting up the VO configuration.....	6
Using the vo-list file locally.....	7
Using the vo-list file remotely.....	7
Including the VO information in the glite-voms-server.cfg.xml file.....	7
Configuring a single VO.....	8
Configuring multiple VOs.....	8
Configuring the VOMS server.....	9
Testing the configuration.....	9
Configuring all the VOs of the VOMS server.....	9
Configuring a single VO of the VOMS server.....	9
Starting the VOMS server.....	9
Configuring access for gridmap file generation.....	10
VOMS Server Configuration options.....	10
Configuring/Updating all VOs.....	10
Starting all VOs.....	10
Stopping all VOs.....	10
Configuring/Adding/Updating a single VO.....	10
Removing a single VO (keeping the database).....	11
Removing a single VO (dropping the database).....	11
Starting a single VO.....	11
Stopping a single VO.....	11
Verifying your configuration.....	11
Checking the status of the VOMS server.....	12
Troubleshooting.....	12
Log information.....	12

VOMS Server Installation and Configuration Guide

Service Overview

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

VOMS Admin provides an intuitive web user interface for daily administration tasks and a SOAP interface for remote clients. (The entire functionality of the VOMS Admin service is accessible via the SOAP interface.) 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. The VOMS server can use MySQL or ORACLE as a backend.

VOMS Node Installation

The general steps to install the OS and the gLite-VOMS 3.1 server are explained at:
<https://twiki.cern.ch/twiki/bin/view/LCG/GenericInstallGuide310>

Upgrading from a gLite 3.0 VOMS Server

This section describes the necessary steps if you are migrating from a 3.0 VOMS server. Please read these instructions carefully before starting the upgrade process!

Besides switching to a new base operating system (from SL3 for the gLite 3.0 release series to SL4 for the gLite 3.1 release series), the gLite 3.1 VOMS server comes with the redesigned version 2 of VOMS-ADMIN which uses a new database schema. The old VOMS databases must therefore be upgraded for the gLite 3.1 VOMS server.

You are strongly advised to backup your database before doing the upgrade!

The database upgrade program is part of the VOMS-ADMIN 2 release and it will be automatically installed on your new server.

Also, the database upgrade will only work if you are using the latest database schema in production from the gLite 3.0 VOMS server.

You should complete successfully all database upgrades before running the VOMS configuration script (glite-voms-server-config.py).

Before proceeding with the database upgrade you should decide whether you want to keep the old database instance or if you want to create a new one. If you run VOMS 3.0 with a MySQL database backend on the same machine you will probably want to use a new database instance on the 3.1 VOMS node as the node also needs to be updated to the new OS. In this case you can copy the dumped database to the new host and then run the database upgrade script against the new instance on the new node.

Migrating the database to the new node

To migrate the database to the new node you need to do the following steps:

- Dump the old database on the old VOMS server and transfer the file to the new node

VOMS < LCG < TWiki

```
mysqldump -h <OLD_HOSTNAME> -u <USER> -p --databases <DB_NAME> > <DUMP_FILE>
```

- Use the database dump on the new machine to create the new database

```
mysql -h <NEW_HOSTNAME> -u <PRIV_USER> -p < <DUMP_FILE>
```

- Grant ALL PRIVILEGES to the user that VOMS-ADMIN will use to connect to and use the database

```
mysql -h <NEW_HOSTNAME> -u <PRIV_USER> -p  
  
mysql> GRANT ALL PRIVILEGES ON <DATABASE_NAME>  
      TO <VOMS_USER>@localhost  
      IDENTIFIED BY '<VOMS_USER_PASSWORD>';
```

Migrating the VOMS configuration to the new node

- Copy all configuration files *.cfg.xml from /opt/glite/etc/config on the old gLite 3.0 VOMS server node to the new node (in the same place)
- Copy recursively all subdirectories from /var/glite/etc/voms-admin named after the VOs you want to upgrade to the new node (in the same place)
- In each of these subdirectories check the voms.database.properties file. It contains the database connection parameters. Edit these on the new 3.1 server if needed (they should point to the database you want to upgrade).
- Copy the certificates of the VO administrators to the new server (pointed by the value of the voms.admin.certificate VO-level parameter)

Upgrading the database schema

Before you proceed make sure that the database is not in use by someone else as the following steps will modify the database schema! To actually upgrade the database you need to:

- Set the following variables with the appropriate values

```
export GLITE_LOCATION=/opt/glite  
export GLITE_LOCATION_VAR=/var/glite  
export GLITE_LOCATION_LOG=/var/log/glite
```

- In each VO subdirectory located under /var/glite/etc/voms-admin check the voms.database.properties file. It contains the database connection parameters. Edit these on the new 3.1 server if needed (they should point to the database you want to upgrade).

The format of this file will change after the upgrade because VOMS-ADMIN 2 uses Hibernate as a persistence layer.

- For each VO database you want to upgrade check the table named "admins" for rows that have the same value for the column "dn". You have to remove the duplicates manually before proceeding to the next step. After importing the MySQL dump on the new MySQL 4.x server, you could use the following SQL statements to do that. First delete all rows in the admins table that have no corresponding ACL entries:

```
mysql> DELETE FROM admins WHERE adminid NOT IN (SELECT adminid FROM acl);
```

Then use the next query to find duplicates:

VOMS < LCG < TWiki

```
mysql> SELECT adminid, dn, c.ca FROM admins a JOIN ca c ON a.ca = c.cid
        WHERE dn IN (SELECT dn FROM admins GROUP BY dn HAVING COUNT(*) > 1);
```

If this query returns an empty set proceed with the next step. Otherwise you will have to choose which rows to remove (ACLs for the deleted users will not be migrated) and delete them with:

```
mysql>DELETE FROM admins WHERE adminid = <ID>
```

choosing an ID from the `adminid` column of the table returned by the previous `SELECT`

- For each VO database you want to upgrade run:

```
/opt/glite/sbin/voms-admin-configure --vo=<VO_NAME> upgrade
```

Note: In case of an unexpected error during the upgrade of a VO, its corresponding `voms.database.properties` file might be left in the new format. To rerun the upgrade you will have to restore the old one manually.

Updating the configuration

You can continue to use the old configuration files but you will probably need to change the following variables:

- `JAVA_HOME` (in `glite-global.cfg.xml`)
- `CATALINA_HOME` (in `glite-global.cfg.xml`)
- `tomcat.user.name` (in `glite-global.cfg.xml`)
- `tomcat.user.group` (in `glite-global.cfg.xml`)
- `fetch-crl.script` (in `glite-security-utils.cfg.xml`) It should point to the actual location of the `fetch-crl` executable (normally `/usr/sbin/fetch-crl` on gLite VOMS 3.1 host).
- `voms.db.oracleinstantclient.location` (in `glite-voms-server.cfg.xml` if you are using Oracle as backend)
- `voms.admin.certificate` (per-VO parameter) Its value should be a valid path to a PEM file containing the certificate of the VO administrator.

Also make sure you are using the desired database connection parameters for the VO (in the `volist.cfg.xml` file)

Reconfiguring the VOMS server

Reconfigure the VOMS server by executing:

```
/opt/glite/etc/config/scripts/glite-voms-server-config.py --configure
```

Starting the VOMS server

Start the VOMS server by executing:

```
/opt/glite/etc/config/scripts/glite-voms-server-config.py --start
```

Configuring access for gridmap file generation

This step needs an appropriate environment being set up for running the `voms-admin` CLI on the `voms` host. This can be done either by running:

```
[root@vomshost: ~] # . /etc/glite/profile.d/glite_setenv.sh
```

or by restarting the shell session with the server (It is loaded automatically for new sessions with the VOMS host).

For successful gridmap file generation on the relevant grid hosts, the appropriate access rights should be granted for each VO. This can be done via the voms-admin command line interface. The following command grants the necessary rights to the predefined user “Anyone” (any user that presents a valid certificate issued by a trusted CA) for the top group “/” and its descendants (the TRUE flag triggers recursive operation).

```
voms-admin --vo=<VO_NAME> --nousercert add-ACL-entry /<VO_NAME> ANYONE VOMS_CA 'CONTAINER_READ,ME
```

Registration of the VOMS host identity as a default admin

This step should be performed only if the VOMS host will have a new hostname or for any reasons the subject of its host certificate differs from the one that was installed on the gLite VOMS 3.0 host. For each VO, execute the following command on the VOMS host:

```
/opt/glite/sbin/voms-db-deploy.py add-admin --vo <VO_NAME> --cert /etc/grid-security/hostcert.pem
```

Verification of the ACL configuration

VOMS-ADMIN 2 has a redesigned security model. The global ACL is now deprecated. If you had set up custom ACEs in VOMS-ADMIN 1, inspect the ACLs of the groups hierarchy after the upgrade is complete. Consult the [VOMS-ADMIN-2 Users guide](#) on how the access control mechanism works in the new version.

VOMS Server Configuration

If you upgrade from the gLite 3.0 version, please follow the instructions above.

Configuring the database backend

Configure (if you use as backend)

Make sure that the MySQL administrator password that you have specified in the configuration file matches the password that is set in the MySQL database. The configuration script does not set it for you. If you want to set a MySQL administrator password, you have to issue the following commands as root:

```
/usr/bin/mysqladmin -u root password <yourPassword>;
/usr/bin/mysqladmin -u root -h <yourHostname> password <yourPassword>;
```

where yourHostname is the name of your host and yourPassword is the password that you want to set.

Configure Oracle (if you use Oracle as backend)

Create the necessary users and databases in ORACLE. Please see the ORACLE manuals for details.

Glite configuration files

The configuration files and scripts can be found in `/opt/glite/etc/config`

A set of template configuration files are installed into `/opt/glite/etc/config/templates`

The following configuration files are used:

- glite-global.cfg.xml

It contains global configuration values. Refer to the ‘Advanced Installation and Configuration Guide’ for the values that can be set and for the description about the general configuration.

- glite-security-utils.cfg.xml

It contains the security utils specific configuration values. Refer to ‘Advanced Installation and Configuration Guide’ for the list of parameters and for the description of the security utils.

- glite-voms-server.cfg.xml

It contains the VOMS server specific configuration files. Since multiple instances of the VOMS Server can be installed on the same node (one per VO), some of the parameters refer to individual instances. Each instance is contained in a separate name tag. A default instance is already defined and can be directly configured. Additional instances can be added by simply copying and pasting the section, assigning a name and changing the parameters values as desired.

Setting the Configuration values for the VOMS server

General configuration of the VOMS server

You first need to set the general configuration parameters for the VOMS server. Table 1 shows the list of parameters that can be set. You will find them in the file `glite-voms-server.cfg.xml`.

Parameter	Description
User-defined Parameters	
voms.db.type	Database type to be used. Can be <code>mysql</code> or <code>oracle</code> . This parameter cannot be specified separately per VO. Example: <code>mysql</code> [Type: 'string']
voms.db.host	Hostname of the database server. Put 'localhost' if you run the database on the same machine. This parameter can be specified also separately per VO. Example: <code>localhost</code> [Type: 'string']
voms.admin.smtp.host	Host to which voms-admin-service generated emails should be submitted. Use 'localhost' if you have a fully configured SMTP server running on this host. Otherwise specify the hostname of a working SMTP submission service. This parameter can be specified also separately per VO. Example: <code>localhost</code> [Type: 'string']
voms.mysql.admin.password	MySQL only. Administrator login password for the MySQL database. This parameter can be specified also separately per VO. Example: <code>'verySecret'</code> [Type: 'string']
Advanced Parameters	
glite.installer.verbose	Enable verbose output. [Type: 'boolean'] Default value: <code>true</code>
glite.installer.checkcerts	Enable check of host certificates. [Type: 'boolean'] Default value: <code>true</code>
set.mysql.root.password	If this parameter is true, then the root password of the mysql database is set to the value specified in <code>mysql.root.password</code> if it not yet set. This parameter has no effect if the database root password is already set. It can be used to ease

	automated installation and configuration of the service, if mysql is not managed in some other way [Type: 'boolean'] Default value: false
voms.mysql.admin.name	Administrator login name for the MySQL database. This parameter can be specified also separately per VO. [Type: 'string'] Default value: root
voms.admin.oracle.connection.string	[NEW parameter in 3.0.8] Specifies the long connection string. Leave the value empty is not needed. This parameter can be specified also separately per VO. Example: (DESCRIPTION = (ADDRESS_LIST=(ADDRESS = (PROTOCOL=TCP) (HOST=lxfs5502.cern.ch) (PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=egee102d.cern.ch))) [Type: 'string']
voms.db.mysql.port	Port number of the database server for mysql. This parameter can be specified also separately per VO. Default value: 3306 [Type: 'integer']
voms.db.oracle.port	Port number of the database server for oracle. This parameter can be specified also separately per VO. Default value: 1521 [Type: 'integer']
voms.db.mysql.maxConnections	The number of simultaneous client connections allowed. Increasing this value increases the number of file descriptors that mysqlq requires. Default value: 500 [Type: integer]
voms.logrotate.period	Configure the log rotation period for the vomsd daemon (/var/log/glite*) Default value: daily [Type:enum daily,weekly,monthly)]
voms.logrotate.logNumber	Specify the number of copies to keep Default value: 90 [Type: integer]
System parameters	
voms.db.mysql.library	Defines the MySQL VOMS library location Default value: \${GLITE_LOCATION}/lib/libvomsmysql.so [Type: 'string']
voms.db.oracle.library	Location of the oracle voms libraries. Default value: \${GLITE_LOCATION}/lib/libvomsoracle.so [Type: 'string']

VO configuration

Setting up the VO configuration

To set up a VO, you have to specify the necessary configuration parameters of the VO. Normally this is done via a 'vo-list' file that can be put locally or remote (to share information for several services). A template of such a vo-list file is installed at

```
/opt/glite/etc/config/templates/vo-list.cfg.xml
```

However you can also put this information in the glite-voms-server.cfg.xml file via copy/paste directly. Detailed information on how to use the VO configuration can be found in the 'generic gLite v3.0 VO Configuration Guide'.

Using the vo-list file locally

To use a local vo-list file defining the set of VOs you want to support, you need to do the following steps:

- Copy the template to the configuration directory

```
cp /opt/glite/etc/config/templates/vo-list.cfg.xml /opt/glite/etc/config/.
```

- Make the configuration files aware of the new configuration file. For this we use the standard Xinclude technique of XML. In the file glite-voms-server.cfg.xml replace the token

```
<config>
```

with

```
<config xmlns:xi="http://www.w3.org/2001/XInclude">
```

Like this, the xinclude mechanism is activated.

- Define the file(s) to be included by adding the following tag below (inside) the tag:

```
<xi:include href="/opt/glite/etc/config/vo-list.cfg.xml" xpointer="" />
```

This tells the configuration to include the file specified as href in the configuration. If you want you included other files by repeating this step.

Using the vo-list file remotely

To use a vo-list file remotely follow the same instructions as described in section 4.3.2.2 but add the corresponding parts in your remote configuration file.

Including the VO information in the glite-voms-server.cfg.xml file

If you want to put the information for the vo's directly to the glite-voms-server.cfg.xml file, copy the entries from the vo-list.cfg.xml file

```
<vo name="VO_NAME">
  ...
</vo>
```

and put them inside the tag in the glite-voms-server.cfg.xml file:

```
<config>
  <vo name="VO_NAME">
    ...
  </vo>
  ...
</config>
```

And remove the part

```
<instance service="voms" iterate="volist">
  <parameters/>
</instance>
```

from the file glite-voms-server.cfg.xml.

Configuring a single VO

Each VO is defined via a set of configuration values. Table 2 defines the set of parameters needed per VO to define the settings for a VOMS server.

Parameter	Description
User-defined Parameters	
voms.vo.name	Name of the VO associated with the VOMS instance. Example: EGEE [Type: 'string']
voms.port.number	Port number listening for requests for this VO. Example: 15001 [Type: 'string']
voms.db.name	Database name to be used to store VOMS information. Example: DB_CMS [Type: 'string']
voms.db.user.name	Name of database user. This parameter can be specified also separately per VO. Example: voUser [Type: 'string']
voms.db.user.password	Password of database user defined in 'voms.db.user.name'. This parameter can be specified also separately per VO. Example: verySecret [Type: 'string']
voms.shortfqans	If set to 'true'; the vomsd daemon will write short FQANs in the generated proxies (Role=NULL and Capability=NULL will be stripped) [Type: 'boolean'] Default value: FALSE
voms.proxy.timeout	Sets the length of time that the information inside the voms proxy is valid; measured in secods. [Type: 'integer'] Default value: 86400
VOMS admin specific parameters	
voms.admin.notification.e-mail	E-mail address that is used to send notification mails from the VOMSadmin. Example: name.surname@domain.org [Type: 'string']
voms.admin.certificate	The certificate file (in pem format) of an initial VO administrator. The VO will be set up so that this user has full VO administration privileges. Remove parameter or leave parameter empty if you don't want to create an initial VO administrator. Example: '/your/path/admincert.pem' [Type: 'string']
voms.admin.webRegistration.disable	Disable user registration via the voms-admin web interface. This parameter can be specified separately per VO. Example: true [Type: 'boolean'] Default value: false

Configuring multiple VOs

To configure multiple VOs, you only need to create multiple

```
<vo name="VO_NAME">
```

```
...
</vo>
```

entries in your configuration file via copy/paste and set the configuration parameters for each VO accordingly.

Configuring the VOMS server

After you have set all the configuration parameters as described in section 4.3, you can need to apply the configuration values by running the configuration script of the voms server. You can find the configuration script in `/opt/glite/etc/config/scripts/glite-voms-server-config.py`

You can configure both all VOs together as well a single VO (e.g. if you add another VO to an already configured VOMS server) depending on the options that you give to the configuration script (see section 4.5 for details on the configuration options).

Testing the configuration

It is recommended to first test your configuration for errors by running the script with the option `'-c'` to show what the configuration will configure.

```
./glite-voms-server-config.py -c
```

This will first list you the complete set of specified configuration parameters and at the end show you the main settings (see Figure 1 for an example). This enables you to see if the configuration parameters are set according to your needs. See also section 6 for details on debugging in case of errors/problems.

Configuring all the VOs of the VOMS server

To configure all the VOs of the VOMS server run the VOMS server configuration script with the option `--configure`:

```
/opt/glite/etc/config/scripts/glite-voms-server-config.py --configure
```

Running the configuration script will automatically configure the security utils so there is no need to run the configuration script of the security utils as well.

Check if any error message is displayed and if necessary fix the parameters values and restart the script. If the configuration is successful you should see at the end the message:

```
The gLite VOMS server was successfully configured.
```

Configuring a single VO of the VOMS server

To configure only a single VO of the VOMS server, run the VOMS server configuration script with the option `--configure`:

```
./glite-voms-server-config.py --configure --vo=voName
```

where `voName` is the name of your VO.

Starting the VOMS server

The new VOMS server is started via

```
/opt/glite/etc/config/scripts/glite-voms-server-config.py --start
```

The next step needs an appropriate environment to be set up for running the voms-admin CLI on the voms host. This can be done by running :

```
[root@vomshost: ~] # . /etc/glite/profile.d/glite_setenv.sh
```

It will be done automatically for new sessions with the VOMS host.

Configuring access for gridmap file generation

For successful gridmap file generation on the relevant grid hosts, the appropriate access rights should be granted for each VO. This can be done via the voms-admin command line interface. The following command grants the necessary rights to the predefined user “Anyone” (any user that presents a valid certificate issued by a trusted CA) for the top group “/” and its descendants (the TRUE flag triggers recursive operation).

```
voms-admin --vo=<VO_NAME> --nousercert add-ACL-entry /<VO_NAME> ANYONE VOMS_CA 'CONTAINER_READ,ME
```

VOMS Server Configuration options

Running the VOMS server configuration script with the option `--configure` or `--start` as described in the previous section will configure and update or start all VOs that are specified in your configuration. In addition you can also only operate on a single VO only by using the option

```
--vo=voName
```

Configuring/Updating all VOs

To configure or update the configuration information of all the VOs you can run

```
./glite-voms-server-config.py --configure
```

Starting all VOs

To start the VOMS server for all the VOs you can run

```
./glite-voms-server-config.py --start
```

Stopping all VOs

To stop the VOMS server for all VOs, you can run

```
./glite-voms-server-config.py --stop
```

Configuring/Adding/Updating a single VO

To add a new VO (without affecting the status/information of other VOS, e.g. without stopping the server for the other VOs), you can run the configuration script via

```
./glite-voms-server-config.py --configure --vo=voName
```

where voName is the name of your VO. In this way only the given VO will be added (if not yet present) or stopped and updated (if already present) without affecting the other VOs.

Removing a single VO (keeping the database)

If you want to remove a VO (without affecting the status/information of other VOS, e.g. without stopping the server for the other VOs), and keeping the database, you can run the configuration script via

```
./glite-voms-server-config.py --remove --vo=voName
```

where voName is the name of your VO. In this way the specified VO will be removed without affecting the other VOs.

If you want to remove the VO permanently, you will also need to remove the configuration information of the VO – otherwise the next time you run with the configuration script on a global level (without the option --vo) with the option –configure it will add it again.

Removing a single VO (dropping the database)

If you want to remove a VO (without affecting the status/information of other VOS, e.g. without stopping the server for the other VOs) together with its database, you can run the configuration script via

```
./glite-voms-server-config.py --remove --remove-db --vo=voName
```

where voName is the name of your VO. In this way the specified VO and its database will be removed without affecting the other VOs. Pay attention this operation is not reversible. If you want to remove the VO permanently, you will also need to remove the configuration information of the VO – otherwise the next time you run with the configuration script on a global level (without the option --vo) with the option –configure it will add it again.

Starting a single VO

If you want to start only one VO (without affecting the status of other VOS, e.g. without starting the server for the other VOs), you can run the configuration script via

```
./glite-voms-server-config.py --start --vo=voName
```

where voName is the name of your VO. In this way the specified VO will be started without affecting the other VOs.

Stopping a single VO

If you want to stop only one VO (without affecting the status of other VOS, e.g. without stopping the server for the other VOs), you can run the configuration script via

```
./glite-voms-server-config.py --stop --vo=voName
```

where voName is the name of your VO. In this way the specified VO will be stopped without affecting the other VOs.

Verifying your configuration

To verify which configuration your VOMS server will be using and which VOs will be configured together with their corresponding settings, you can run the configuration script with the option –c

```
./glite-voms-server-config.py -c
```

giving you an overview of the global configuration parameters, the most important VOMS specific parameters

as well as the different VOs and their parameters.

Checking the status of the VOMS server

To check the status of the VOMS server you can run

```
./glite-voms-server-config.py --status
```

Troubleshooting

If you are having trouble installing, configuring, starting or running the VOMS server, please first check that you have installed all necessary dependencies (see section). Next make sure that your configuration is correct. The best way to check this is to run the configuration script with the option `-c`

```
/opt/glite/etc/config/scripts/glite-voms-server-config.py -c
```

that will prompt you with the list of the most important configuration parameters. You should see for example all the VOs appearing there that you want to set up/have set up. If you have a syntax mistake in your configuration files, you can run the linux command `xmllint` to check proper xml syntax of your files, e.g.

```
xmllint /opt/glite/etc/config/glite-voms-server.cfg.xml
```

that will show you all the syntax errors in the configuration files.

For MySQL and Oracle debugging, please see the corresponding MySQL and Oracle manuals.

Finally, try to have a look at the log files to understand what went wrong during service startup and running.

Log information

The following log files are used by the VOMS server:

- `/var/log/glite/voms.<voName>` for the VO `<voName>` of the VOMS service
- `glite-fetch-crl-cron.log` for the fetch-crl
- `/var/log/tomcat5/catalina.out` for tomcat main process
- `voms-admin.<voName>.log` for the VO `<voName>` of the voms-admin service
- `glite-security-trustmanager.log` for the security trustmanager

Please consult the MySQL and Oracle manuals for the corresponding log files of the DB backends.

This topic: LCG > VOMS

Topic revision: r5 - 2009-07-08 - DimitarShiyachki



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback