



v. 1.0 (rev. 14)

# Installation Guide

21 Apri 2005

## TABLE OF CONTENT

<b>1. INTRODUCTION</b> .....	<b>6</b>
1.1. PURPOSE .....	6
1.2. TERMINOLOGY .....	6
<b>2. GLITE DEPLOYMENT</b> .....	<b>7</b>
2.1. SERVICES AND COMPONENTS .....	7
2.2. STANDARD DEPLOYMENT MODEL .....	7
<b>3. GLITE PACKAGES AND DOWNLOADS</b> .....	<b>10</b>
<b>4. THE GLITE CONFIGURATION MODEL</b> .....	<b>11</b>
4.1. THE GLITE CONFIGURATION SCRIPTS .....	11
4.2. THE GLITE CONFIGURATION FILES .....	11
4.2.1. <i>Configuration Parameters Scope</i> .....	11
4.2.2. <i>The Local Service Configuration Files</i> .....	12
4.2.3. <i>The Global Configuration File</i> .....	12
4.2.4. <i>The Site Configuration File</i> .....	14
4.2.5. <i>Internal Configuration</i> .....	15
4.2.6. <i>User environment</i> .....	15
4.2.7. <i>Default Environment Variables</i> .....	16
4.2.8. <i>Configuration Overrides</i> .....	16
<b>5. GLITE SECURITY UTILITIES</b> .....	<b>17</b>
5.1. OVERVIEW .....	17
5.1.1. <i>CA Certificates</i> .....	17
5.1.2. <i>glite-mkgridmap</i> .....	17
5.1.3. <i>fetch-crl</i> .....	17
5.2. INSTALLATION PRE-REQUISITES .....	18
5.3. SECURITY UTILITIES INSTALLATION .....	18
5.4. SECURITY UTILITIES CONFIGURATION .....	18
5.5. SECURITY UTILITIES CONFIGURATION WALKTHROUGH .....	19
<b>6. COMPUTING ELEMENT</b> .....	<b>21</b>
6.1. SERVICE OVERVIEW .....	21
6.2. INSTALLATION PRE-REQUISITES .....	21
6.2.1. <i>Security Settings</i> .....	21
6.2.2. <i>Java JRE/JDK</i> .....	22
6.2.3. <i>Resource Management System</i> .....	22
6.3. COMPUTING ELEMENT SERVICE INSTALLATION .....	22
6.4. COMPUTING ELEMENT SERVICE CONFIGURATION .....	23
6.5. COMPUTING ELEMENT CONFIGURATION WALKTHROUGH .....	27
6.6. MANAGING THE CE SERVICES .....	28
6.7. STARTING THE CE SERVICES AT BOOT .....	29
6.8. WORKSPACE SERVICE TECH-PREVIEW .....	29
<b>7. WORKLOAD MANAGER</b> .....	<b>30</b>
7.1. SERVICE OVERVIEW .....	30
7.2. INSTALLATION PRE-REQUISITES .....	30
7.2.1. <i>Security Settings</i> .....	30
7.3. WORKLOAD MANAGER SYSTEM INSTALLATION .....	30
7.4. WORKLOAD MANAGEMENT SYSTEM CONFIGURATION .....	31
7.5. WORKLOAD MANAGEMENT SYSTEM CONFIGURATION WALKTHROUGH .....	34
7.6. MANAGING THE WMS SERVICES .....	35
7.7. STARTING THE CE SERVICES AT BOOT .....	36
7.8. PUBLISHING WMS SERVICES TO R-GMA .....	36

<b>8. LOGGING AND BOOKKEEPING SERVER.....</b>	<b>37</b>
8.1. SERVICE OVERVIEW.....	37
8.2. INSTALLATION PRE-REQUISITES .....	37
8.2.1. <i>Security Settings</i> .....	37
8.3. LOGGING AND BOOKKEEPING SERVER INSTALLATION.....	38
8.4. LOGGING AND BOOKKEEPING SERVER CONFIGURATION.....	38
8.5. LOGGING AND BOOKKEEPING CONFIGURATION WALKTHROUGH.....	40
<b>9. WORKER NODE.....</b>	<b>42</b>
9.1. SERVICE OVERVIEW.....	42
9.2. INSTALLATION PRE-REQUISITES .....	42
9.2.1. <i>Security Settings</i> .....	42
9.2.2. <i>Java JDK/JRE</i> .....	42
9.2.3. <i>Resource Management System</i> .....	42
9.3. WORKER NODE INSTALLATION.....	42
9.4. WORKER NODE CONFIGURATION.....	43
9.5. WORKER NODE CONFIGURATION WALKTHROUGH.....	44
<b>10. GLITE I/O .....</b>	<b>46</b>
10.1. GLITE I/O SERVER.....	46
10.1.1. <i>Service Overview</i> .....	46
10.1.2. <i>Installation pre-requisites</i> .....	46
10.1.3. <i>gLite I/O Server installation</i> .....	46
10.1.4. <i>gLite I/O Server Configuration</i> .....	47
10.1.5. <i>gLite I/O Server Configuration Walkthrough</i> .....	56
10.2. CLIENT .....	57
10.2.1. <i>Service Overview</i> .....	57
10.2.2. <i>Installation pre-requisites</i> .....	57
10.2.3. <i>gLite I/O Client installation</i> .....	57
10.2.4. <i>gLite I/O Client Configuration</i> .....	58
<b>11. LOCAL TRANSFER SERVICE.....</b>	<b>60</b>
11.1. SERVICE OVERVIEW.....	60
11.2. INSTALLATION PRE-REQUISITES .....	60
11.2.1. <i>Security Settings</i> .....	60
11.2.2. <i>Java JRE/JDK</i> .....	60
11.3. LOCAL TRANSFER SERVICE INSTALLATION.....	60
11.4. LOCAL TRANSFER SERVICE CONFIGURATION.....	61
11.5. LOCAL TRANSFER SERVICE CONFIGURATION WALKTHROUGH .....	64
<b>12. SINGLE CATALOG .....</b>	<b>65</b>
12.1. SERVICE OVERVIEW.....	65
12.2. INSTALLATION PRE-REQUISITES .....	65
12.2.1. <i>Security Settings</i> .....	65
12.2.2. <i>Java JDK</i> .....	65
12.2.3. <i>Oracle JDBC Drivers</i> .....	66
12.3. SINGLE CATALOG INSTALLATION .....	66
12.4. SINGLE CATALOG CONFIGURATION .....	66
12.5. SINGLE CATALOG CONFIGURATION WALKTHROUGH.....	67
<b>13. INFORMATION AND MONITORING SYSTEM (R-GMA).....</b>	<b>69</b>
13.1. SERVICE OVERVIEW.....	69
13.1.1. <i>R-GMA SERVER</i> .....	69
13.1.2. <i>R-GMA CLIENT</i> .....	69
13.2. R-GMA DEPLOYMENT MODULES .....	70
13.2.1. <i>R-GMA Deployment strategy</i> .....	70

13.2.2. R-GMA Server deployment module .....	72
13.2.3. R-GMA Client deployment module.....	78
13.2.4. R-GMA servicetool deployment module.....	81
13.2.5. R-GMA GadgetIN (GIN) deployment module.....	85
<b>14. VOMS SERVER AND ADMINISTRATION TOOLS .....</b>	<b>89</b>
14.1. SERVICE OVERVIEW.....	89
14.2. INSTALLATION PRE-REQUISITES .....	89
14.2.1. Security Settings.....	89
14.2.2. Java JRE/JDK .....	89
14.3. VOMS SERVER INSTALLATION .....	89
14.4. VOMS SERVER CONFIGURATION .....	90
14.5. VOMS SERVER CONFIGURATION WALKTHROUGH .....	91
<b>15. USER INTERFACE.....</b>	<b>92</b>
15.1. SERVICE OVERVIEW.....	92
15.2. INSTALLATION PRE-REQUISITES .....	92
15.2.1. Security Settings.....	92
15.2.2. Java JRE/JDK .....	92
15.3. UI INSTALLATION.....	92
15.4. UI CONFIGURATION.....	93
15.5. UI CONFIGURATION WALKTHROUGH .....	95
15.6. CONFIGURATION FOR THE UI USERS .....	96
15.7. NOTE .....	96
<b>16. TORQUE .....</b>	<b>97</b>
16.1. SERVICE OVERVIEW.....	97
16.1.1. TORQUE Server Overview .....	97
16.1.2. TORQUE Client Overview.....	97
16.2. INSTALLATION PRE-REQUISITES .....	97
16.3. TORQUE SERVER.....	97
16.3.1. TORQUE Server Installation .....	97
16.3.2. TORQUE Server Service Configuration.....	98
16.3.3. TORQUE Server Configuration Walkthrough.....	104
16.3.4. Managing the TORQUE Server Service.....	105
16.4. TORQUE CLIENT.....	105
16.4.1. TORQUE Client Installation.....	105
16.4.2. TORQUE Client Configuration .....	105
16.4.3. TORQUE Client Configuration Walkthrough .....	106
16.4.4. Managing the TORQUE Client.....	107
<b>17. THE GLITE FUNCTIONAL TEST SUITES.....</b>	<b>108</b>
17.1. OVERVIEW.....	108
17.2. I/O TEST SUITE .....	108
17.2.1. Installation Pre-requisites .....	108
17.2.2. Installation .....	108
17.2.3. Configuration .....	108
17.2.4. Execution.....	109
17.2.5. Test results .....	109
17.3. CATALOG TEST SUITE.....	110
17.3.1. Installation Pre-requisites .....	110
17.3.2. Installation .....	110
17.3.3. Configuration .....	110
17.3.4. Execution.....	110
17.3.5. Test results .....	111
17.4. WMS TEST SUITE .....	111
17.4.1. Installation Pre-requisites .....	112

17.4.2. Installation .....	112
17.4.3. Configuration .....	112
17.4.4. Execution.....	112
17.4.5. Test results .....	113
<b>18. APPENDIX A: SERVICE CONFIGURATION FILE EXAMPLE.....</b>	<b>115</b>
<b>19. APPENDIX B: SITE CONFIGURATION FILE EXAMPLE .....</b>	<b>120</b>

## 1. INTRODUCTION

### 1.1. PURPOSE

This document describes how to install and configure the EGEE middleware known as gLite. The objective is to provide clear instructions for administrators on how to deploy gLite components on machines at their site.

### 1.2. TERMINOLOGY

#### Glossary

CE	Computing Element
R-GMA	Relational Grid Monitoring Architecture
WMS	Workload Management System
WN	Worker Node
LTS	Local Transfer Service
LB	Logging and Bookkeeping
SC	Single Catalog
UI	User Interface

#### Definitions

Service	A single high-level unit of functionality
Node	A computer where one or more services are deployed

## 2. GLITE DEPLOYMENT

The gLite middleware is a Service Oriented Grid middleware providing services for managing distributed computing and storage resources and the required security, auditing and information services.

The gLite system is composed of a number of high level services that can be installed on individual dedicated computers (nodes) or combined in various ways to satisfy site requirements. This installation guide follows a standard deployment model whereby most of the services are installed on dedicated computers. However, other examples of valid node configuration are also shown.

### 2.1. SERVICES AND COMPONENTS

The following high-level services are part of this release of the gLite middleware:

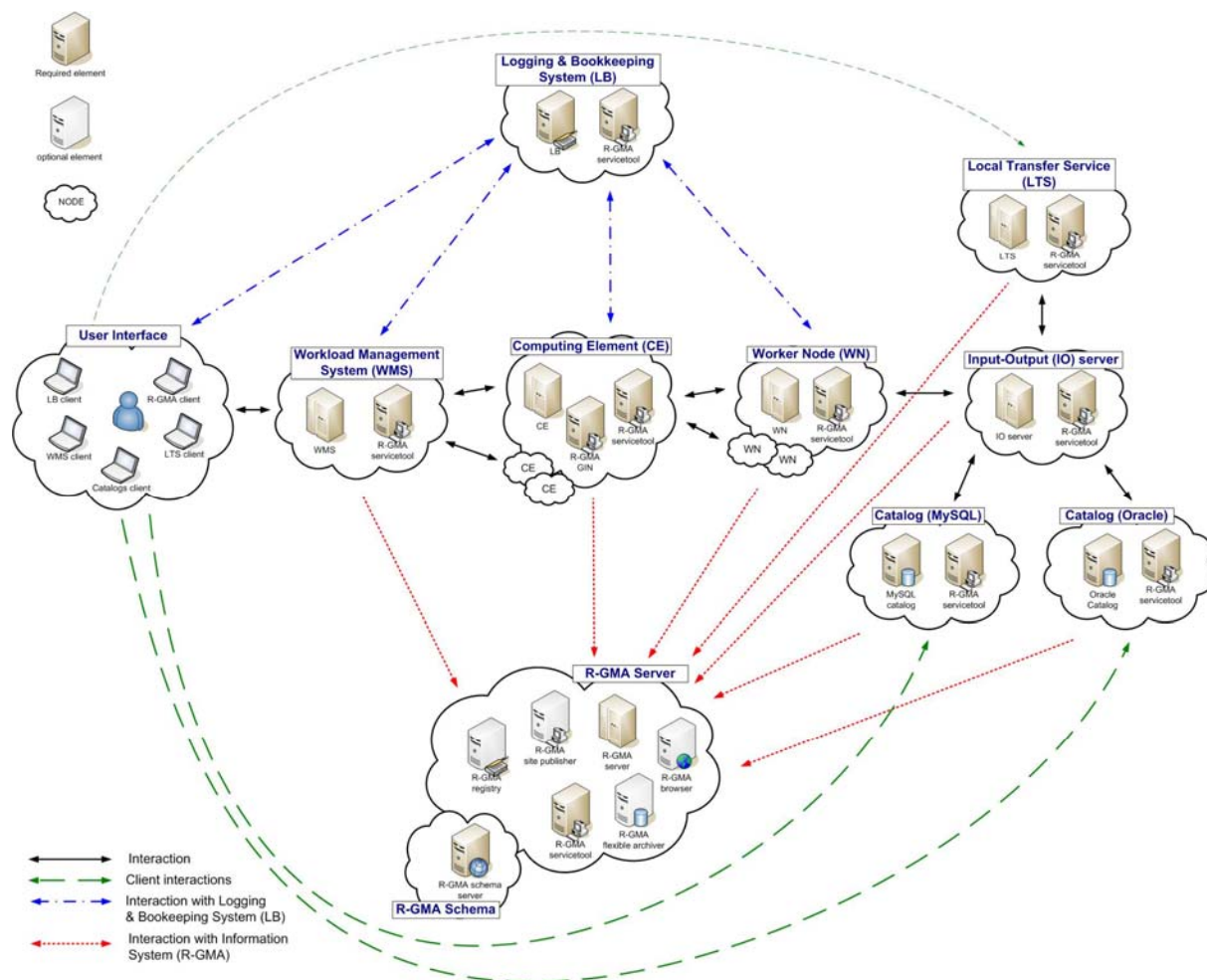
- Authorization, Authentication and Delegation Services (as integral part of the other subsystems)
- Computing Element (CE)
- File & Replica Catalog (called Single Catalog in this release – SC)
- File Transfer and Placement Service (Local Transfer Service)
- gLite I/O Server and Client
- Logging and Bookkeeping Server (LB)
- R-GMA Servers, Client, Site Publisher, Service Tools and Service Discovery
- Standard Worker node (WN, a set of clients and APIs required on a typical worker node installation)
- User Interface
- VOMS and VOMS administration tools
- Workload Manager System (WMS)

### 2.2. STANDARD DEPLOYMENT MODEL

Figure 1: gLite Service Deployment Scenario shows the standard deployment model for these services.

Each site has to provide the local services for job and data management as well as information and monitoring:

- Job Management Services
  - A CE interfaces the local resource management system (e.g. LSF, PBS) to the Grid middleware. The currently released CE contains also a CE Monitor service that notifies one or more WMS server of its capabilities and availability in order to receive matching jobs from the WMS (pull model).
  - The Worker Nodes behind the local resource management system host all the necessary clients to interact with the Grid middleware from within a job.
- Data Management Services
  - Data is stored in an SRM based storage system.
  - The gLite I/O Server allows posix-like access to files stored in the SRM
  - The Single Catalog (SC) keeps track of the LFN:GUID:SURL mapping of local files
  - The Local Transfer Service provides file transfer/file placement service (FPS) and it is used for moving files.



**Figure 1: gLite Service Deployment Scenario**

- Information and Monitoring Services
  - The R-GMA Server accepts connection from clients (producers), published for example by services user jobs, and forwards the information to the appropriate consumers. In addition, one or ore additional R-GMA Servers can be configured as schema server (only one per R-GMA domain) and registry servers.

The figure shows the proposed mapping of services onto physical machines. This mapping will give the best performance and service resilience. Smaller sites may however consider mapping multiple services onto the same machine. This is in particular true for the CE and package manager and for the SC and the LTS.

Instead of the distributed deployment of the catalogs (a local catalog and a global catalog) a centralized deployment of just a global catalog can be considered as well. This is actually the configuration supported in the gLite 1.0.



The VO services act on the Grid level and comprise the Security services, Workload Management services, Information and Monitoring services. Each VO should have an instance of these services, physical service instances can mostly be shared among VOs. For some services, even multiple instances per VO can be provided as indicated below:

- Security services
  - The Virtual Organization Membership Service (VOMS) is used for managing the membership and member rights within a VO. VOMS also acts as attribute authority.
  - myProxy is used as secure proxy store
- Workload Management services
  - The Workload Management Service (WMS) is used to submit jobs to the Grid.
  - The Logging and Bookkeeping service (LB) keeps track of the job status information.

The WMS and the LB can be deployed independently but due to their tight interactions it is recommended to deploy them together. Multiple instances of these services may be provided for a VO.

- Information and Monitoring services
  - The R-GMA Registry Servers and Schema Server are used for binding information consumers and producers. There can be more than one Registry Server that can be replicated for resilience reasons.
- Single Catalog (SC)
  - The single catalog is used for browsing the LFN space and to find out the location (sites) where files are stored. This is in particular need by the WMS.
- User Interface
  - The User Interface (UI) combines all the clients that allow the user to directly interact with the Grid services.

### 3. GLITE PACKAGES AND DOWNLOADS

The gLite middleware is currently published in the form of RPM packages and installation scripts from the gLite web site at:

<http://glite.web.cern.ch/glite/packages>

Required external dependencies in RPM format can also be obtained from the gLite project web site at:

<http://glite.web.cern.ch/glite/packages/externals/bin/rhel30/RPMS>

Deployment modules for each high-level gLite component are provided on the web site and are a straightforward way of downloading and installing all the RPMs for a given component. A configuration script is provided with each module to configure, deploy and start the service or services in each high-level module.

Installation and configuration of the gLite services are kept well separated. Therefore the RPMS required to install each service or node can be deployed on the target computers in any suitable way. The use of dedicated RPMS management tools is actually recommended for production environments. Once the RPMS are installed, it is possible to run the configuration scripts to initialize the environment and the services.

gLite is also distributed using the apt package manager. More details on the apt cache address and the required list entries can be found on the main packages page of the gLite web site (<http://glite.web.cern.ch/glite/packages>).

gLite is also available in the form of source and binary tarballs from the gLite web site and from the EGEE CVS server at:

`jra1mw.cvs.cern.ch:/cvs/jra1mw`

The server support authenticated ssh protocol 1 and Kerberos 4 access and anonymous pserver access (username: anonymous).

## 4. THE GLITE CONFIGURATION MODEL

Each gLite deployment module contains a number of RPMS for the necessary internal and external components that make up a service or node (RPMS that are normally part of standard Linux distributions are not included in the gLite installer scripts). In addition, each module contains one or more configuration RPMS providing configuration scripts and files.

Each module contains at least the following configuration RPMS:

Name	Definition
glite-config-x.y.z-r.noarch.rpm	The glite-config RPM contains the global configuration files and scripts required by all gLite modules
glite-<service>-config-x.y.z-r.noarch.rpm	The glite-<service>-config RPM contains the configuration files and scripts required by a particular service, such as ce, wms or rgma

In addition, a mechanism to load remote configuration files from URLs is provided. Refer to the Site Configuration section later in this chapter (§4.2.3).

### 4.1. THE GLITE CONFIGURATION SCRIPTS

All configuration scripts are installed in:

`$GLITE_LOCATION/etc/config/scripts`

where `$GLITE_LOCATION` is the root of the gLite packages installation. By default `$GLITE_LOCATION = /opt/glite`.

The scripts are written in python and follow a naming convention. Each file is called:

`glite-<service>-config.py`

where `<service>` is the name of the service they can configure.

In addition, the same scripts directory contains the gLite Installer library (`gLiteInstallerLib.py`) and a number of helper scripts used to configure various applications required by the gLite services (`globus.py`, `mysql.py`, `tomcat.py`, etc).

The gLite Installer library and the helper scripts are contained in the `glite-config` RPM. All service scripts are contained in the respective `glite-<service>-config` RPM.

### 4.2. THE GLITE CONFIGURATION FILES

#### 4.2.1. Configuration Parameters Scope

All parameters in the gLite configuration files are categorised in one of three categories:

- **User-defined parameters:** these parameters have a default value of 'changeme' and have to be replaced with valid values before running the configuration scripts. In some cases, some of the user-defined parameters are alternative depending on the deployment scenarios. In this case the unused parameter must be removed from the file or left empty (remove the 'changeme' value)

- **Advanced parameters:** these parameters have always valid default values, but can be changed by a user or system administrator to customize an installation depending on site policies
- **System parameters:** these parameters have always valid default values and represent advanced options that a system administrator can change to modify the behaviour of the system for special usage or for troubleshooting problems. Normally there is no need to modify these parameters and setting them incorrectly may compromised the functionality of the middleware

#### 4.2.2. The Local Service Configuration Files

The gLite configuration files are XML-encoded files containing all the parameters required to configure the gLite services. The configuration files are distributed as templates and are installed in the \$GLITE\_LOCATION/etc/config/templates directory.

The configuration files follow a similar naming convention as the scripts. Each file is called:

glite-<service>.cfg.xml

Each gLite configuration file contains a global section called `<parameters/>` and may contain one or more `<instance/>` sections in case multiple instances of the same service or client can be configured and started on the same node (see the configuration file example in Appendix A). In case multiple instances can be defined for a service, the global `<parameters/>` section applies to all instances of the service or client, while the parameters in each `<instance/>` section are specific to particular named instance and can override the values in the `<parameters/>` section.

The configuration files support variable substitution. The values can be expressed in term of other configuration parameters or environment variables by using the `${}` notation (for example `${GLITE_LOCATION}`).

The templates directory can also contain additional service templates used by the configuration scripts during their execution (like for example the gLite I/O service templates).

***Note:** When using a local configuration model, before running the configuration scripts the corresponding configuration files must be copied from the templates directory to \$GLITE\_LOCATION/etc/config and all the user-defined parameters must be correctly instantiated (refer also to the Configuration Parameters Scope paragraph later in this section). This is not necessary if using the site configuration model (see below)*

#### 4.2.3. The Global Configuration File

The global configuration file `glite-global.cfg.xml` contains all parameters that have gLite-wide scope and are applicable to all gLite services. The parameters in this file are loaded first by the configuration scripts and cannot be overridden by individual service configuration files.

Currently the global configuration file defines the following parameters:

Parameter	Default value	Description
<b>User-defined Parameters</b>		
site.config.url		The URL of the Site Configuration file for this node. The values defined in

		the Site Configuration file are applied first and are be overridden by values specified in the local configuration files. Leave this parameter empty or remove it to use local configuration only.
<b>Advanced Parameters</b>		
GLITE_LOCATION	/opt/glite	
GLITE_LOCATION_VAR	/var/glite	
GLITE_LOCATION_LOG	/var/log/glite	
GLITE_LOCATION_TMP	/tmp/glite	
GLOBUS_LOCATION	/opt/globus	Environment variable pointing to the Globus package.
GPT_LOCATION	/opt/gpt	Environment variable pointing to the GPT package.
JAVA_HOME	/usr/java/j2sdk1.4.2_04	Environment variable pointing to the SUN Java JRE or J2SE package.
CATALINA_HOME	/var/lib/tomcat5	Environment variable pointing to the Jakarta Tomcat package
host.certificate.file	/etc/grid-security/hostcert.pem	The host certificate (public key) file location
host.key.file	/etc/grid-security/hostkey.pem	The host certificate (private key) file location
ca.certificates.dir	/etc/grid-security/certificates	The location where CA certificates are stored
user.certificate.path	.certs	The location of the user certificates relative to the user home directory
host.gridmapfile	/etc/grid-security/gridmapfile	Location of the grid mapfile
host.gridmap.dir	/etc/grid-security/gridmapdir	The location of the account lease information for dynamic allocation
<b>System Parameters</b>		
installer.export.filename	/etc/glite/profile.d/glite_setenv.sh	Full path of the script containing environment definitions This file is automatically generated by

		the configuration script. If it exists, the new values are appended
tomcat.user.name	tomcat4	Name of the user account used to run tomcat.
tomcat.user.group	tomcat4	Group of the user specified in the parameter 'tomcat.user.name'

**Table 1: Global Configuration Parameters**

#### 4.2.4. The Site Configuration File

All gLite configuration scripts implement a mechanism to load configuration information from a remote URL. This mechanism can be used to configure the services from a central location for example to propagate site-wide configuration.

The URL of the configuration file can be specified as the *site.config.url* parameter in the global configuration file of each node or as a command-line parameter when launching a configuration script, for example:

```
glite-ce-config.py --siteconfig=http://server.domain.com/sitename/siteconfig.xml
```

In the latter case, the site configuration file is only used for running the configuration scripts once and all values are discarded afterwards. For normal operations it is necessary to specify the site configuration URL in the *glite-gobal.cfg.xml* file.

The site configuration file can contain a global section called `<parameters/>` and one `<node/>` section for each node to be remotely configured (see the configuration file example in Appendix B). Each `<node/>` section must be qualified with the host name of the target node, for example:

```
<node name="lxb1428.cern.ch">
...
</node>
```

where the host name must be the value of the \$HOSTNAME environment variable on the node. The `<parameters/>` section contains parameters that apply to all nodes referencing the site configuration file.

The `<node/>` sections can contain the same parameters that are defined in the local configuration files. If more than one service is installed on a node, the corresponding `<node/>` section can contain a combination of all parameters of the individual configuration files. For example if a node runs both the WMS and the LB Server services, then the corresponding `<node/>` section in the site configuration file may contain a combination of the parameters contained in the local configuration files for the WMS and the LB Server modules.

If a user-defined parameter (see later in §4.2.1 the definition of parameters scope) is defined in the site configuration file, the same parameter doesn't need to be defined in the local file (it can therefore keep the token value 'changeme' or be removed altogether). However, if a parameter is defined in the local configuration file, it override whatever value is specified in

the site configuration file. If a site configuration file contains all necessary values to configure a node, it is not necessary to create the local configuration files. The only configuration file that must always be present locally in the `/opt/glite/etc/config/` directory is the `glite-global.cfg.xml` file, since it contains the parameter that specifies the URL of the site configuration file.

This mechanism allows distributing a site configuration for all nodes and at the same time gives the possibility of overriding some or all parameters locally in case of need.

New configuration information can be easily propagated simply by publishing a new configuration file and rerunning the service configuration scripts.

In addition, several different models are possible. Instead of having a single configuration file contains all parameters for all nodes, it's possible for example to split the parameters in several files according to specific criteria and point different services to different files. For example it is possible to put all parameters required to configure the Worker Nodes in one file and all parameters for the servers in a separate file, or have a separate file for each node and so on.

#### 4.2.5. Internal Configuration

The configuration scripts and files described above represent the common configuration interfaces of all gLite services. However, since the gLite middleware is a combination of various old and new services, not all services can natively use the common configuration model. Many services come with their own configuration files and formats. Extensive work is being done to make all services use the same model, but until the migration is completed, the common configuration files must be considered as the public configuration interfaces for the system. The configuration scripts do all the necessary work to map the parameters in the public configuration files to parameters in service specific configuration files. In addition, many of the internal configuration files are dynamically created or modified by the public configuration scripts.

The goal is to provide the users with a consistent set of files and scripts that will not change in the future even if the internal behaviour may change. It is therefore recommended whenever possible to use only the common configuration files and scripts and do not modify directly the internal service specific configuration files.

#### 4.2.6. User environment

When anyone of the gLite configuration scripts is run, it creates or modifies a general configuration file called `glite_setenv.sh` in `/etc/glite/profile.d` (the location can be changed using a system-level parameter in the global configuration file).

This file contains all the environment definitions needed to run the gLite services. This file is automatically added to the `.bashrc` file of users under direct control of the middleware, such as service accounts and pool accounts. In addition, if needed the `.bash_profile` file of the accounts is modified to source the `.bashrc` file and to set `BASH_ENV=.bashrc`. The proper environment is therefore created every time an account logs in various ways (interactive, non-interactive or script).

Other users not under control of the middleware can manually source the `glite_setenv.sh` file as required.

In case a gLite service or client is installed using a non-privileged user (if foreseen by the service or client installation), the `glite_setenv.sh` file is created in `$GLITE_LOCATION/etc/profile.d`.

#### 4.2.7. Default Environment Variables

By default the gLite configuration files and scripts define the following environment variables:

GLITE_LOCATION	/opt/glite
GLITE_LOCATION_VAR	/var/glite
GLITE_LOCATION_LOG	/var/log/glite
GLITE_LOCATION_TMP	/tmp/glite
PATH	/opt/glite/bin:/opt/glite/externals/bin:\$PATH
LD_LIBRARY_PATH	/opt/glite/lib:/opt/glite/externals/lib:\$LD_LIBRARY_PATH

The first four variables can be modified in the global configuration file or exported manually before running the configuration scripts. If these variables are already defined in the environment they take priority on the values defined in the configuration files

#### 4.2.8. Configuration Overrides

It is possible to override the values of the parameters in the gLite configuration files by setting appropriate key/value pairs in the following files:

```
/etc/glite/glite.conf
~/.glite/glite.conf
```

The first file has system-wide scope, while the second has user-scope. These files are read by the configuration scripts before the common configuration files and their values take priority on the values defined in the common configuration files.



## 5. GLITE SECURITY UTILITIES

### 5.1. OVERVIEW

The gLite Security Utilities module contains the CA Certificates distributed by the EU Grid PMA. In addition, it contains a number of utilities scripts needed to create or update the local grid mapfile from a VOMS server and periodically update the CA Certificate Revocation Lists.

#### 5.1.1. CA Certificates

The CA Certificate are installed in the default directory

```
/etc/grid-security/certificates
```

This is not configurable at the moment. The installation script downloads the latest available version of the CA RPMS from the gLite software repository.

#### 5.1.2. glite-mkgridmap

The glite-mkgridmap script is used to update the local grid mapfile and its configuration file glite-mkgridmap.conf are installed respectively in

```
$GLITE_LOCATION/sbin
```

and

```
$GLITE_LOCATION/etc
```

The script can be run manually (after customizing its configuration file). Running glite-mkgridmap doesn't preserve the existing grid-mapfile. However, a wrapper script is provided in \$GLITE\_LOCATION/etc/config/scripts/mkgridmap.py to update the grid-mapfile preserving any additional entry in the file not downloaded by glite-mkgridmap.

The Security Utilities module configuration script also installs a crontab file in /etc/cron.d that executes the mkgridmap.py script every night at 02:00. The installation of this cron job and the execution of the mkgridmap.py script during the configuration are optional and can be enabled using a configuration parameter (see the configuration walkthrough for more information).

Some services need to run the mkgridmap.py script as part of their initial configuration (this is currently the case for example of the WMS). In this case the installation of the cron job and execution of the script at configuration must be enabled. This is indicated in each case in the appropriate chapter.

#### 5.1.3. fetch-crl

The fetch-crl script is used to update the CA Certificate Revocation Lists. This script is provided by the EU GridPMA organization. It is installed in:

```
/usr/bin
```

The Security Utilities module configuration script installs a crontab file in /etc/cron.d that executes the glite-fetch-crl every four hours. The CRLs are installed in the same directory as the CA certificates, /etc/grid-security/certificates. The module configuration file (glite-security-

utils.cfg.xml) allows specifying an e-mail address to which the errors generated when running the cron job are sent.

## 5.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

## 5.3. SECURITY UTILITIES INSTALLATION

The gLite Security Utilities module is installed as follows:

1. Download from the gLite web site the latest version of the the gLite Security Utilities installation script *glite-security-utils\_installer.sh*
2. Make the file executable (*chmod u+x glite-security-utils\_installer.sh*) and execute it
3. Run the installation script as root. All the required RPMS are downloaded from the gLite software repository in the directory *glite-security-utils* next to the installation script and the installation procedure is started. If some RPMS are already installed, they upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

gLite	in /opt/glite (\$GLITE_LOCATION)
CA Certificates	in /etc/grid-security/certificates

## 5.4. SECURITY UTILITIES CONFIGURATION

1. Copy the global configuration file template *\$GLITE\_LOCATION/etc/config/template/glite-global.cfg.xml* to *\$GLITE\_LOCATION/etc/config*, open it and modify the parameters if required (Table 1)
2. Copy the configuration file template *\$GLITE\_LOCATION/etc/config/template/glite-security-utils.cfg.xml* to *\$GLITE\_LOCATION/etc/config*, open it and set all user-defined parameters (Table 2). You can also modify Advanced and System parameters if required, but it's not normally necessary. All parameters that must be changed have a token value of *changeme*
3. Insert the appropriate entries in the *\$GLITE\_LOCATION/etc/glite-mkgridmap.conf* file, if grid-mapfile management is required
4. Run the configuration script *\$GLITE\_LOCATION/etc/config/scripts/glite-security-utils-config.py*

Parameter	Default value	Description
<b>User-defined Parameters</b>		
cron.mailto		E-mail address to which the stderr of the installed cron jobs is sent

<b>Advanced Parameters</b>		
glite.installer.verbose	true	Produce verbose output when running the script
glite.installer.checkcerts	true	Activate a check for host certificates and stop the script if not available. The certificates are looked for in the location specified by the global parameters <i>host.certificate.file</i> and <i>host.key.file</i>
fetch-crl.cron.tab	00 */4 * * *	The cron tab to use for the fetch-crl cron job.
install.fetch-crl.cron	true	Install the glite-fetch-crl cron job. Possible values are 'true' (install the cron job) or 'false' (do not install the cron job)
install.mkgridmap.cron	false	Install the glite-mkgridmap cron job. Possible values are 'true' (install the cron job) or 'false' (do not install the cron job)
<b>System Parameters</b>		

**Table 2: Security Utilities Configuration Parameters**

## 5.5. SECURITY UTILITIES CONFIGURATION WALKTHROUGH

The Security Utilities configuration script performs the following steps:

1. Sets the following environment variables if not already set:

GLITE\_LOCATION, GLITE\_LOCATION\_VAR, GLITE\_LOCATION\_LOG, GLITE\_LOCATION\_TMP, PATH and LD\_LIBRARY\_PATH to the values specified in the configuration files or already defined in the local environment

2. Creates the file `/etc/cron.d/fetch-crl.cron` containing the following line

```
00 */4 * * * root sh $GLITE_LOCATION/sbin/glite-fetch-crl -o /etc/grid-security/certificates > /dev/null
```

3. Runs the `glite-fetch-crl` once to update the CRLs
4. If required, creates the file `/etc/cron.d/glite-mkgridmap.cron` containing the following line

```
00 02 * * * root python $GLITE_LOCATION/etc/config/scripts/mkgridmap.py > /dev/null
```

5. Runs the `$GLITE_LOCATION/etc/config/scripts/mkgridmap.py` once to update the grid-mapfile
6. Touches the cron daemon spool directory to activate the installed cron jobs



## 6. COMPUTING ELEMENT

### 6.1. SERVICE OVERVIEW

The Computing Element (CE) is the service representing a computing resource. Its main functionality is job management (job submission, job control, etc.). The CE may be used by a generic client: an end-user interacting directly with the Computing Element, or the Workload Manager, which submits a given job to an appropriate CE found by a matchmaking process. For job submission, the CE can work in push model (where the job is pushed to a CE for its execution) or pull model (where the CE asks the Workload Management Service for jobs). Besides job management capabilities, a CE must also provide information describing itself. In the push model this information is published in the information Service, and it is used by the match making engine which matches available resources to queued jobs. In the pull model the CE information is embedded in a "CE availability" message, which is sent by the CE to a Workload Management Service. The matchmaker then uses this information to find a suitable job for the CE.

The CE uses the R-GMA servicetool to publish information about its services and states to the information services R-GMA. See chapter 13 for more details about R-GMA and the R-GMA servicetool.

### 6.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 6.2.1. Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` (gLite Security Utilities) is installed and configured automatically when installing and configuring the CE (refer to Chapter 5 for more information about the Security Utilities module). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the `glite-fetch-crl`, `glite-mkgridmap` and `mkgridmap.py` scripts and sets up cron jobs that periodically check for updated revocation lists and grid-mapfile entries if required). The use of the `glite-mkgridmap` script is not normally required on the CE node, since VOMS entries are used instead of individual user DN mappings.
2. Install the server host certificate `hostcert.pem` and key `hostkey.pem` in `/etc/grid-security`
3. Install the VOMS Server(s) host certificate in the directory `/etc/grid-security/vomsdir`. This is necessary to allow LCMAPS to extract the VOMS information from the VOMS proxies.
4. The CE Service may require modification to the server firewall settings. The following iptables instructions must be executed.

```
-I <Chain_Name> 1 -m state --state NEW -m tcp -p tcp --dport 2119 -j ACCEPT
```

```
-I <Chain_Name> 2 -m state --state NEW -m tcp -p tcp --dport 3878 -j ACCEPT
-I <Chain_Name> 3 -m state --state NEW -m tcp -p tcp --dport 3879 -j ACCEPT
-I <Chain_Name> 4 -m state --state NEW -m udp -p udp --dport 3879 -j ACCEPT
-I <Chain_Name> 5 -m state --state NEW -m tcp -p tcp --dport 3882 -j ACCEPT
-I <Chain_Name> 6 -m state --state NEW -m udp -p udp --dport 1020 -j ACCEPT
-I <Chain_Name> 7 -m state --state NEW -m udp -p udp --dport 1021 -j ACCEPT
-I <Chain_Name> 8 -m state --state NEW -m udp -p udp --dport 1022 -j ACCEPT
-I <Chain_Name> 9 -m state --state NEW -m udp -p udp --dport 1023 -j ACCEPT
-I <Chain_Name> 10 -m state --state NEW -m tcp -p tcp --dport 32768:65535
```

Please note that the CE configuration script sets the necessary iptables entries automatically. This can be disabled using the `-n` or `--noiptables` option when running the configuration script or by leaving empty or commenting out the `iptables.chain` configuration parameter. If the specified chain doesn't exist, it is created. If the chain exists, the entries are inserted if they do not yet exist.

### 6.2.2. Java JRE/JDK

The Java JRE or JDK are required to run the CE Monitor. This release requires v. 1.4.2 (revision 04 or greater). The JDK version to be used is a configuration parameter in the `glite-global-cfg.xml` file. Please change it according to your version and location (see also sections 4.2.3 and 6.4 for more details).

### 6.2.3. Resource Management System

The Resource Management System must be installed on the CE node or on a separate dedicated node before installing and configuring the CE module. This release of the CE module supports PBS, Torque and LSF.

## 6.3. COMPUTING ELEMENT SERVICE INSTALLATION

1. Download from the gLite web site the latest version of the CE installation script `glite-ce_installer.sh`. It is recommended to download the script in a clean directory
2. Make the script executable (`chmod u+x glite-ce_install.sh`) and execute it
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory `glite-ce` next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

gLite	in <code>/opt/glite</code> ( <code>\$GLITE_LOCATION</code> )
Condor	in <code>/opt/condor-x.y.z</code> (where <code>x.y.z</code> is the current Condor version)
Globus	in <code>/opt/globus</code> ( <code>\$GLOBUS_LOCATION</code> )
Tomcat	in <code>/var/lib/tomcat5</code> (standard JPP location)

5. The gLite CE configuration script is installed in `$GLITE_LOCATION/etc/config/scripts/glite-ce-config.py`. A template configuration file is installed in `$GLITE_LOCATION/etc/config/templates/glite-ce.cfg.xml`
6. The gLite CE installs the R-GMA servicetool to publish its information to the information system R-GMA. The details of the installation of the R-GMA servicetool are described in section 13.2.4.5.

## 6.4. COMPUTING ELEMENT SERVICE CONFIGURATION

1. Copy the global configuration file template

`$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

open it and modify the parameters if required (Table 1).

2. Copy the configuration file template from

`$GLITE_LOCATION/etc/config/templates/glite-ce.cfg.xml`

to

`$GLITE_LOCATION/etc/config/glite-ce.cfg.xml`

and modify the parameters values as necessary. Some parameters have default values, others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The following parameters can be set:

Parameter	Default value	Description
<b>User-defined Parameters</b>		
voms.voname		The names of the VOs that this CE node can serve
voms.vomsnode		The full hostname of the VOMS server responsible for each VO. Even if the same server is responsible for more than one VO, there must be exactly one entry for each VO listed in the 'voms.voname' parameter. For example: 'host.domain.org'
voms.vomsport		The port on the VOMS server listening for request for each VO. This is used in the vomses configuration file. For example: '15000'
voms.vomscertsbj		The subject of the host certificate of the VOMS server for each VO. For example: '/C=ORG/O=DOMAIN/OU=GRID/CN=host.domain.org'
pool.account.basename		The prefix of the set of pool accounts to be created for each VO. Existing pool accounts with this prefix are not recreated
pool.account.group		The group name of the pool accounts to be used for each VO. For some batch systems like LSF, this group may need a specific gid. The gid can be set using the <i>pool.lsfgid</i> parameter in the LSF configuration section
pool.account.number		The number of pool accounts to

		create for each VO. Each account will be created with a username of the form prefixXXX where prefix is the value of the <i>pool.account.basename</i> parameter. If matching pool accounts already exist, they are not recreated. The range of values for this parameter is from 1 to 999
cemon.wms.host		The hostname of the WMS server(s) that receives notifications from this CE
cemon.wms.port		The port number on which the WMS server(s) receiving notifications from this CE is listening
cemon.lrms		The type of Local Resource Management System. It can be 'lsf' or 'pbs'. If this parameter is absent or empty, the default type is 'pbs'
cemon.cetype		The type of Computing Element. It can be 'condorc' or 'gram'. If this parameter is absent or empty, the default type is 'condorc'
cemon.cluster		The cluster entry point host name. Normally this is the CE host itself
cemon.cluster-batch-system-bin-path		The path of the lrms commands. For example: '/usr/pbs/bin' or '/usr/local/lsf/bin'. This value is also used to set the PBS_BIN_PATH or LSF_BIN_PATH variables depending on the value of the 'cemon.lrms' parameter
cemon.cesebinds		The CE-SE bindings for this CE node. There are three possible format: configfile 'queue[queue]' se 'queue[queue]'se se_entry point A '.' character for the queue list means all queues. Example: '.' EGEE::SE::Castor /tmp
cemon.queues		A list of queues defined on this CE node. Examples are: long, short, infinite, etc.
pool.lsfgid		The gid of the groups to be used for the pool accounts on some LSF installations, on per each pool account group. This parameter is an array of values containing one value for each VO served by this CE node. The list must match the corresponding lists in the VOMS configuration section. If this is not



		required by your local LSF system remove this parameter or leave the values empty
condor.wms.user		The username of the condor user under which the Condor daemons run on the WMS nodes that this CE serves
lb.user		The account name of the user that runs the local logger daemon. If the user doesn't exist it is created. In the current version, the host certificate and key are used as service certificate and key and are copied in this user's home in the directory specified by the global parameter 'user.certificate.path' in the glite-global.cfg.xml file
iptables.chain		The name of the chain to be used for configuring the local firewall. If the chain doesn't exist, it is created and the rules are assigned to this chain. If the chain exists, the rules are appended to the existing chain
<b>Advanced Parameters</b>		
glite.installer.verbose	True	Enable verbose output
glite.installer.checkcerts	True	Enable check of host certificates
PBS_SPOOL_DIR	/usr/spool/PBS	The PBS spool directory
LSF_CONF_PATH	/etc	The directory where the LSF configuration file is located
globus.osversion	<empty>	The kernel id string identifying the system installed on this node. For example: '2.4.21-20.ELsmp'. This parameter is normally automatically detected, but it can be set here
globus.hostdn	<empty>	The host distinguished name (DN) of this node. This is normally automatically read from the server host certificate. However it can be set here. For example: 'C=ORG, O=DOMAIN, OU=GRID, CN=host/server.domain.org'
condor.version	6.7.6	The version of the installed Condor-C libraries
condor.user	condor	The username of the condor user under which the Condor daemons must run
condor.releasedir	/opt/condor-6.7.6	The location of the Condor package. This path is internally simlinked to /opt/condor-c. This is currently needed by the Condor-C software

CONDOR_CONFIG	<code>\${condor.release.dir}/etc/condor_config</code>	Environment variable pointing to the Condor configuration file
condor.scheddinterval	10	How often should the schedd send an update to the central manager?
condor.localdir	<code>/var/local/condor</code>	Where is the local condor directory for each host? This is where the local config file(s), logs and spool/execute directories are located
condor.blahgahp	<code>\$(GLITE_LOCATION)/bin/blahpd</code>	The path of the gLite blahp daemon
condor.daemonlist	MASTER, SCHEDD	The Condor daemons to configure and monitor
condor.blahpollinterval	120	How often should blahp poll for new jobs?
gatekeeper.port	2119	The gatekeeper listen port
lcg.providers.location	<code>/opt/lcg</code>	The location where the LCG providers are installed.
<b>System Parameters</b>		

**Table 3: CE Configuration Parameters**

3. Configure the R-GMA servicetool. For this you have to configure the servicetool itself as well as configure the sub-services of CE for the publishing via the R-GMA servicetool:

- a. *R-GMA servicetool configuration:*

Copy the R-GMA servicetool configuration file template

`$GLITE_LOCATION/etc/config/templates/glite-rgma-servicetool.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 15 shows a list of the parameters that can be set. More details can be found in section 13.2.4.6.

- b. *Service Configuration for the R-GMA servicetool:*

Modify the R-GMA servicetool related configuration values that are located in the CE configuration file

`glite-ce.cfg.xml`

that was mentioned before. In this file, you will find for each service that should be published via the R-GMA servicetool one instance of a set of parameters that are grouped by the tag

`<instance name="xxxx" service="rgma-servicetool">`

Where xxxx is the name of corresponding subservice. Table 16 on page 85 in the section 13.2.4 about the R-GMA servicetool shows the general list of parameters for each service for the publishing via the R-GMA servicetool.

For CE the following sub-services are published via the R-GMA servicetool and need to be updated accordingly:

- i. Local Logger
- ii. Gatekeeper
- iii. CE Monitor

Again, you find the necessary steps described in section 13.2.4.6.

**Note:** Step 1,2 and 3 can also be performed by means of the remote site configuration file or a combination of local and remote configuration files

4. Install the VOMS server(s) host certificates in the directory `/etc/grid-security/vomsdir`
5. Create the CE static information file `ce-static.ldif` in the directory that you specified in the `cemon.static` parameter
6. As root run the CE configuration file `/opt/glite/etc/config/scripts/glite-ce-config.py`

## 6.5. COMPUTING ELEMENT CONFIGURATION WALKTHROUGH

The CE configuration script performs the following steps:

1. Set the following environment variables if not already set using the values set in the global and CE configuration files:

```
GLITE_LOCATION      [=/opt/glite if not set anywhere]
GLOBUS_LOCATION     [=/opt/globus if not set anywhere]
CONDOR_CONFIG       [=/opt/condor if not set anywhere]
```

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.csf.xml`:

```
GLITE_LOCATION_VAR  [default is /var/glite]
GLITE_LOCATION_LOG  [default is /var/log/glite]
GLITE_LOCATION_TMP  [default is /tmp/glite]
```

3. Load the CE configuration file `$GLITE_LOCATION/etc/config/glite-ce.cfg.xml`
4. Create the number of pool accounts specified the service configuration file using the specified base name, group and GID (if present; this is required by some LSF installation). If the group doesn't exist, it is created. If any of the pool accounts already exists, they are not recreated. All accounts are then configured by modifying their `.bash_profile` and `.bashrc` files to source the `/etc/glite/profile.d/glite_setenv.sh` script created by this configuration process
5. Create the `glite-gatekeeper` configuration file `$GLITE_LOCATION/etc/gatekeeper.conf` by adding all required entries. If the file already exists, a backup copy is created by appending the extension `.1`
6. Create the `jobmanager-fork` file `$GLITE_LOCATION/etc/grid-services/jobmanager-fork` by adding all required entries. If the file already exists, a backup copy is created by appending the extension `'.1'`. Create a link to this file as `$GLITE_LOCATION/etc/grid-services/jobmanager`

7. Create the Globus job manager configuration file `$GLITE_LOCATION/etc/globus-job-manager.conf` by adding all necessary entries. If the file already exists, a backup copy is created by appending the extension `.1`
8. Create the `$GLITE_LOCATION_TMP` dir and set permissions. Similarly, create the `$GLITE_LOCATION_TMP/gram_job_state` dir and set corresponding permissions
9. Create the `$GLITE_LOCATION/etc/lcas/lcas.db` file by adding the necessary entries. If the file already exists, a backup copy is created by appending the extension `‘.1’`
10. Create an empty banned users `$GLITE_LOCATION/etc/lcas/ban_users.db` file if it doesn't exist
11. Update the `/etc/grid-security/grid-mapfile` adding additional LCMAPS VOMS pool accounts entries (if required the optional step of running the `glite-mkgridmap` script can be run to fill the grid-mapfile with user DN mappings from configured VOMS or LDAP servers)
12. Create the group map file `/etc/grid-security/groupmapfile` and add the required LCMAPS VOMS pool account entries
13. Create the LCMAPS DB file `$GLITE_LOCATION/etc/lcmaps/lcmaps.db` by adding all required entries
14. Create the `/etc/grid-security/vomsdir` directory
15. Create the `$GLITE_LOCATION/etc/vomses` file with the VO information set in the configuration file
16. Run GPT post-installation and Globus configuration scripts
17. Create the `/opt/condor-c` link to the Condor package and customize the Condor-C configuration file by adding the required BLAHP entries
18. Configure the Local Logger daemons by creating/verifying the account used to run them and making a copy of the host certificate and key to this user home directory in `.certs` (the location can be configured using the global parameter `user.certificate.path`)
19. Install and configure the CE Monitor and CE Plugin by installing the Monitor war in the local Tomcat installation, creating necessary links to the CE Plugin jars and creating the `$GLITE_LOCATION/etc/glite-ce-ce-plugin/glite-ce-parameters.cfg` file and the predefined subscriptions file `$CATALINA_HOME/webapps/ce-monitor/subscriptions/predifinedSubscriptionList.xml`. If the files already exists, backup copies are created with the extension `‘.1’`
20. Unless the `--noiptables` option is used when running the script or the `iptables.chain` parameter is not defined, the iptables entries described in the security prerequisites sections are configured.
21. The R-GMA servicetool is configured and the services are added to the R-GMA servicetool.
22. The Gatekeeper, the LB Logger client, Tomcat the CE Monitor service are started

## 6.6. MANAGING THE CE SERVICES

The CE configuration script can be run with the following command-line parameters to manage the services:

<code>glite-ce-config.py --start</code>	Starts all CE services (or restart them if they are already running)
<code>glite-ce-config.py --stop</code>	Stops all CE services

glite-ce-config.py --status	Verifies the status of all services. The exit code is 0 if all services are running, 1 in all other cases
-----------------------------	---

### 6.7. STARTING THE CE SERVICES AT BOOT

When the CE configuration script is run, it installs the *gLite* script in the */etc/inet.d* directory and activates it to be run at boot. The *gLite* script runs the *glite-ce-config.py --start* command and makes sure that all necessary services are started in the correct order.

### 6.8. WORKSPACE SERVICE TECH-PREVIEW

This release of the gLite Computing Element module contains a tech-preview of the Workspace Service developed in collaboration with the Globus GT4 team. This service allows a more dynamic usage of the pool accounts with the possibility of leasing an account and releasing it when it's not needed anymore.

To use this service, an alternative configuration script has been provided:

```
/opt/glite/etc/config/scripts/glite-ce-wss-config.py
```

It requires Ant to be properly installed and configured on the server.

No specific usage instructions are provided for the time being. More information about the Workspace Service and its usage can be found at the bottom of the following page from point 8 onwards (the installation and configuration part is done by the glite-ce module):

[http://www.nikhef.nl/grid/lcaslcmaps/install\\_wss\\_lcmaps\\_on\\_lxb2022](http://www.nikhef.nl/grid/lcaslcmaps/install_wss_lcmaps_on_lxb2022)

## 7. WORKLOAD MANAGER

### 7.1. SERVICE OVERVIEW

The Workload Management System (WMS) comprises a set of grid middleware components responsible for the distribution and management of tasks across grid resources, in such a way that applications are conveniently, efficiently and effectively executed.

The core component of the Workload Management System is the Workload Manager (WM), whose purpose is to accept and satisfy requests for job management coming from its clients. For a computation job there are two main types of request: submission and cancellation.

In particular the meaning of the submission request is to pass the responsibility of the job to the WM. The WM will then pass the job to an appropriate Computing Element for execution, taking into account the requirements and the preferences expressed in the job description. The decision of which resource should be used is the outcome of a matchmaking process between submission requests and available resources.

### 7.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 7.2.1. Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called *glite-security-utils* (gLite Security Utilities) is installed and configured automatically when installing and configuring the WMS (refer to Chapter 5 for more information about the Security Utilities module). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl*, *glite-mkgridmap* and *mkgridmap.py* scripts and sets up cron jobs that periodically check for updated revocation lists and grid-mapfile entries
2. Customize the *mkgridmap* configuration file `$GLITE_LOCATION/etc/glite-mkgridmap.conf` by adding the required VOMS server group. The information in this file is used to run the *glite-mkgridmap* script during the Security Utilities configuration to produce the `/etc/grid-security/grid-mapfile`
3. Install the server host certificate `hostcert.pem` and key `hostkey.pem` in `/etc/grid-security`

### 7.3. WORKLOAD MANAGER SYSTEM INSTALLATION

1. Download from the gLite web site the latest version of the WMS installation script *glite-wms\_install.sh*. It is recommended to download the script in a clean directory
2. Make the script executable (`chmod u+x glite-wms_installer.sh`) and execute it
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory `glite-wms` next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.

4. If the installation is performed successfully, the following components are installed:

gLite	in /opt/glite
Condor	in /opt/condor-x.y.z (where x.y.z is the current condor version)
Globus	in /opt/globus

5. The gLite wms configuration script is installed in `$GLITE_LOCATION/etc/config/scripts/glite-wms-config.py`. A template configuration file is installed in `$GLITE_LOCATION/etc/config/templates/glite-wms.cfg.xml`
6. The gLite WMS installs the R-GMA servicetool to publish its information to the information system R-GMA. The details of the installation of the R-GMA servicetool are described in section 13.2.4.5.

#### 7.4. WORKLOAD MANAGEMENT SYSTEM CONFIGURATION

1. Copy the global configuration file template `$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml` to `$GLITE_LOCATION/etc/config` open it and modify the parameters if required (Table 1)
2. Copy the configuration file template from `$GLITE_LOCATION/etc/config/templates/glite-wms.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-wms.cfg.xml`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. You can find a list of parameters in Table 4.

Parameter	Default value	Description
<b>User-defined Parameters</b>		
glite.user.name		Name of the user account used to run the gLite services on this WMS node
glite.user.group		Group of the user specified in the 'glite.user.name' parameter. This group must be different from the pool account group specified by the parameter 'pool.account.group'.
pool.account.basename		The prefix of the set of pool account to be created. Existing pool accounts with this prefix are not recreated
pool.account.group		The group name of the pool accounts to be used. This group must be different from the WMS

		service account group specified by the parameter 'glite.user.group'.
pool.account.number		The number of pool accounts to create. Each account will be created with a username of the form prefixXXX where prefix is the value of the pool.account.basename parameter. If matching pool accounts already exist, they are not recreated. The range of values for this parameter is 1-999
wms.cemon.port		The port number on which this WMS server is listening for notifications from CEs when working in pull mode. Leave this parameter empty or comment it out if you don't want to activate pull mode for this WMS node. Example: 5120
wms.cemon.endpoints		The endpoint(s) of the CE(s) that this WMS node should query when working in push mode. Leave this parameter empty or comment it out if you don't want to activate push mode for this WMS node. Example: 'http://lxb0001.cern.ch:8080/ce-monitor/services/CEMonitor'
information.index.host		Host name of the Information Index node. Leave this parameter empty or comment it out if you don't want to use a BD-II for this WMS node
cron.mailto		E-mail address for sending cron job notifications
condor.condoradmin		E-mail address of the condor administrator
<b>Advanced Parameters</b>		
glite.installer.verbose	true	Sets the verbosity of the configuration script output
glite.installer.checkcerts	true	Switch on/off the checking of the existence of the host certificate files
GSIWUFTPPORT	2811	Port where the globus ftp server is listening
GSIWUFTPDLOG	\${GLITE_LOCATION_LOG}/gsiwuftp.d.log	Location of the globus ftp server log file



GLOBUS_FLAVOR_NAME	gcc32dbg	The Globus libraries flavour to be used
condor.scheddinterval	10	Condor scheduling interval
condor.releasedir	/opt/condor-6.7.6	Condor installation directory
CONDOR_CONFIG	\${condor.releasedir}/etc/condor_config	Condor global configuration file
condor.blahpollinterval	10	How often should blah poll for new jobs?
information.index.port	2170	Port number of the Information Index
information,index.base_dn	mds-vo-name=local, o=grid	Base DN of the information index LDAP server
wms.config.file	\${GLITE_LOCATION}/etc/glite_wms.conf	Location of the wms configuration file
<b>System Parameters</b>		
condor.localdir	/var/local/condor	Condor local directory
condor.daemonlist	MASTER, SCHEDD, COLLECTOR, NEGOTIATOR	List of the condor daemons to start. This must be a comma-separated list of services as it would appear in the Condor configuration file

**Table 4: WMS Configuration Parameters**

3. Configure the R-GMA servicetool. For this you have to configure the servicetool itself as well as configure the sub-services of WMS for the publishing via the R-GMA servicetool:

- a. *R-GMA servicetool configuration:*

Copy the R-GMA servicetool configuration file template

`$GLITE_LOCATION/etc/config/templates/glite-rgma-servicetool.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 15 shows a list of the parameters that can be set. More details can be found in section 13.2.4.6.

- b. *Service Configuration for the R-GMA servicetool:*

Modify the R-GMA servicetool related configuration values that are located in the WMS configuration file

`glite-wms.cfg.xml`

that was mentioned before. In this file, you will find for each service that should be published via the R-GMA servicetool one instance of a set of parameters that are grouped by the tag

```
<instance name="xxxx" service="rgma-servicetool">
```

Where xxxx is the name of corresponding subservice. Table 16 on page 85 in the section 13.2.4 about the R-GMA servicetool shows the general list of parameters for each service for the publishing via the R-GMA servicetool. For WMS the following sub-services are published via the R-GMA servicetool and need to be updated accordingly:

- i. Local Logger
- ii. Proxy Renewal Service
- iii. Log Monitor Service
- iv. Job Controller Service
- v. Network Server
- vi. Workload Manager

Again, you find the necessary steps described in section 13.2.4.6.

*Note: Step 1,2 and 3 can also be performed by means of the remote site configuration file or a combination of local and remote configuration files*

4. Enter the VOMS server information for the VOs you want to use in the *glite-mkgridmap.conf* file in */opt/glite/etc*.
5. As root run the WMS configuration file */opt/glite/etc/config/scripts/glite-wms-config.py*
6. The WMS Service is now ready.

## 7.5. WORKLOAD MANAGEMENT SYSTEM CONFIGURATION WALKTHROUGH

The WMS configuration script performs the following steps:

1. Set the following environment variables if not already set using the values defined in the global and wms configuration files:

```
GLITE_LOCATION=/opt/glite
GLOBUS_LOCATION=/opt/globus
CONDOR_CONFIG=/opt/condor
```

2. Read the following environment variables if set in the environment or in the global gLite configuration file *\$GLITE\_LOCATION/etc/config/glite-global.cfg.xml*:

```
GLITE_LOCATION_VAR
GLITE_LOCATION_LOG
GLITE_LOCATION_TMP
```

3. Load the WMS configuration file *\$GLITE\_LOCATION/etc/config/glite-wms.cfg.xml*
4. Set the following additional environment variables needed internally by the services (this requirement should disappear in the future):

```
CONDORG_INSTALL_PATH=< condor.releasedir >
X509_CERT_DIR=< ca.certificates.dir >
GRID_FUNCTIONS=/etc/grid-functions
PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/externals/bin:$GLOBUS_LOCA
```

```

CONDOR/bin:$CONDOR_INSTALL_PATH/bin:$PATH
LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib:$GLITE_LOCATION/externals/myproxy-1.14/lib:$GLOBUS_LOCATION/lib:$LD_LIBRARY_PATH
GLITE_WMS_LOCATION=$GLITE_LOCATION
GLITE_WMS_LOCATION_VAR=$GLITE_LOCATION_VAR
GLITE_WMS_USER=< glite.user.name >
GLITE_WMS_TMP=/tmp/glitewms
GLITE_WMS_PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/sbin
GLITE_WMS_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLOBUS_LOCATION/lib
GLITE_HOST_KEY=< host.key.file >
GLITE_HOST_CERT=< host.certificate.file >
GLITE_HOST_CERT=/home/< glite.user.name >/hostcert.pem
GLITE_HOST_KEY=/home/< glite.user.name >/hostkey.pem
GLITE_USER=< glite.user.name >

```

5. Create the <glite.user.name> account needed to run the service and set the environment by modifying its .bash\_profile and .bashrc scripts to source the /etc/glite/profile.d/glite\_setenv.sh file created by the WNS configuration script
6. Create the number of pool accounts specified the service configuration file using the specified account name. If the group doesn't exist, it is created. If any of the pool accounts already exists it is not recreated, but it's configured by modifying its .bash\_profile and .bashrc scripts to source the /etc/glite/profile.d/glite\_setenv.sh file created by the WNS configuration script
7. Configure Condor by modifying the Condor configuration file and setting appropriate directory permissions for the <glite.user.name> account used to run the services
8. Configure the Local Logger daemons by creating/verifying the account used to run them and making a copy of the host certificate and key to this user home directory in .certs (the location can be configured using the global parameter *user.certificate.path*)
9. Run the Condor initialization script condorc-initialize as <glite.user.name>
10. Installs the purgeStorage, check-daemons and ns-proxy cron jobs in /etc/cron.d and touches the cron daemon spool directory to activate them
11. Start the following services: GridFTP Server, Workload Manager, Network Server, Job Controller, Log Monitor, Proxy Renewal, Local Logger

## 7.6. MANAGING THE WMS SERVICES

The CE configuration script can be run with the following command-line parameters to manage the services:

glite-ce-config.py --start	Starts all CE services (or restart them if they are already running)
glite-ce-config.py --stop	Stops all CE services
glite-ce-config.py --status	Verifies the status of all services. The exit code is 0 if all services are running, 1 in all other cases
glite-ce-config.py --startservice=xxx	Starts the WMS xxx subservice. xxx can be one of the following: condor = the Condor master and daemons

	<p>ftpd = the Grid FTP daemon  lm = the gLite WMS Logger Monitor daemon  wm = the gLite WMS Workload Manager daemon  ns = the gLite WMS Network Server daemon  jc = the gLite WMS Job Controller daemon  pr = the gLite WMS Proxy Renewal daemon  lb = the gLite WMS Logging &amp; Bookkeeping client</p>
<p>glite-ce-config.py --stopservice=xxx</p>	<p>Stops the WMS xxx subservice. xxx can be one of the following:  condor = the Condor master and daemons  ftpd = the Grid FTP daemon  lm = the gLite WMS Logger Monitor daemon  wm = the gLite WMS Workload Manager daemon  ns = the gLite WMS Network Server daemon  jc = the gLite WMS Job Controller daemon  pr = the gLite WMS Proxy Renewal daemon  lb = the gLite WMS Logging &amp; Bookkeeping client</p>

### 7.7. STARTING THE CE SERVICES AT BOOT

When the WMS configuration script is run, it installs the *gLite* script in the */etc/inet.d* directory and activates it to be run at boot. The *gLite* script runs the *glite-ce-config.py --start* command and makes sure that all necessary services are started in the correct order.

### 7.8. PUBLISHING WMS SERVICES TO R-GMA

The WMS services are published to R-GMA using the R-GMA Service Tool service. The Service Tool service is automatically installed and configured when installing and configuring the WMS module. The WMS configuration file contains a separate configuration section (an *<instance/>*) for each WMS sub-service. The required values must be filled in the configuration file before running the configuration script.

For more details about the R-GMA Service Tool service refer to section 13.2.4 later in this guide.

## 8. LOGGING AND BOOKKEEPING SERVER

### 8.1. SERVICE OVERVIEW

The Logging and Bookkeeping service (LB) tracks jobs in terms of events (important points of job life, e.g. submission, finding a matching CE, starting execution etc.) gathered from various WMS components as well as CEs (all those have to be instrumented with LB calls).

The events are passed to a physically close component of the LB infrastructure (locallogger) in order to avoid network problems. This component stores them in a local disk file and takes over the responsibility to deliver them further.

The destination of an event is one of Bookkeeping Servers (assigned statically to a job upon its submission). The server processes the incoming events to give a higher level view on the job states (e.g. Submitted, Running, Done) which also contain various recorded attributes (e.g. JDL, destination CE name, job exit code, etc.).

Retrieval of both job states and raw events is available via legacy (EDG) and WS querying interfaces.

Besides querying for the job state actively, the user may also register for receiving notifications on particular job state changes (e.g. when a job terminates). The notifications are delivered using an appropriate infrastructure. Within the EDG WMS, upon creation each job is assigned a unique, virtually non-recyclable job identifier (JobId) in an URL form.

The server part of the URL designates the bookkeeping server which gathers and provides information on the job for its whole life.

LB tracks jobs in terms of events (e.g. Transfer from a WMS component to another one, Run and Done when the jobs starts and stops execution). Each event type carries its specific attributes. The entire architecture is specialized for this purpose and is job-centric: any event is assigned to a unique Grid job. The events are gathered from various WMS components by the LB producer library, and passed on to the locallogger daemon, running physically close to avoid any sort of network problems.

The locallogger's task is storing the accepted event in a local disk file. Once it's done, confirmation is sent back and the logging library call returns, reporting success.

Consequently, logging calls have local, virtually non-blocking semantics. Further on, event delivery is managed by the interlogger daemon. It takes the events from the locallogger (or the disk files on crash recovery), and repeatedly tries to deliver them to the destination bookkeeping server (known from the JobId) until it succeeds finally.

### 8.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 8.2.1. Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` can be installed by downloading and running from

the gLite web site (<http://www.glite.org>) the script *glite-security-utils\_installer.sh* (Chapter 5). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl* script and sets up a crontab that periodically check for updated revocation lists

2. Install the server host certificate *hostcert.pem* and key *hostkey.pem* in */etc/grid-security*

### 8.3. LOGGING AND BOOKKEEPING SERVER INSTALLATION

1. Download from the gLite web site the latest version of the LB installation script *glite-lb\_install.sh*. It is recommended to download the script in a clean directory
2. Make the script executable (*chmod u+x glite-lb\_installer.sh*) and execute it or execute it with *sh glite-lb\_install.sh*
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory *glite-lb* next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

gLite	in /opt/glite
Globus	in /opt/globus

5. The gLite LB configuration script is installed in *\$GLITE\_LOCATION/etc/config/scripts/glite-lb-config.py*. A template configuration file is installed in *\$GLITE\_LOCATION/etc/config/templates/glite-lb.cfg.xml*
6. The gLite LB installs the R-GMA servicetool to publish its information to the information system R-GMA. The details of the installation of the R-GMA servicetool are described in section 13.2.4.5.

### 8.4. LOGGING AND BOOKKEEPING SERVER CONFIGURATION

1. Copy the global configuration file template *\$GLITE\_LOCATION/etc/config/template/glite-global.cfg.xml* to *\$GLITE\_LOCATION/etc/config* open it and modify the parameters if required (Table 1).
2. Copy the configuration file template from *\$GLITE\_LOCATION/etc/config/templates/glite-lb.cfg.xml* to *\$GLITE\_LOCATION/etc/config/glite-lb.cfg.xml* and modify the parameters values as necessary (Table 5). Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The list of parameters can be found in Table 5.

Parameter	Default value	Description
<b>User-defined Parameters</b>		
glite.user.name		The account used to run the LB daemons
glite.user.group		Group of the user specified in the 'glite.user.name' parameter. Leave it empty or comment it out to use the same as 'glite.user.name'
<b>Advanced Parameters</b>		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
lb.database.name	lbserver20	The mySQL database name to create for storing LB data. In this version it must be set to the given value.
lb.database.username	lbserver	The username to be used to access the local mySQL server. Now it must be set to the default value
lb.index.list	owner location destination	Definitions of indices on all the currently supported indexed system attributes
<b>System Parameters</b>		

**Table 5: LB Configuration Parameters**

3. Configure the R-GMA servicetool. For this you have to configure the servicetool itself as well as configure the sub-services of LB for the publishing via the R-GMA servicetool:

- c. *R-GMA servicetool configuration:*

Copy the R-GMA servicetool configuration file template

`$GLITE_LOCATION/etc/config/templates/glite-rgma-servicetool.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 15 shows a list of the parameters that can be set. More details can be found in section 13.2.4.6.

- d. *Service Configuration for the R-GMA servicetool:*

Modify the R-GMA servicetool related configuration values that are located in the LB configuration file

`glite-lb.cfg.xml`

that was mentioned before. In this file, you will find for each service that should be published via the R-GMA servicetool one instance of a set of parameters that are grouped by the tag

```
<instance name="xxxx" service="rgma-servicetool">
```

Where xxxx is the name of corresponding subservice. Table 16 on page 85 in the section 13.2.4 about the R-GMA servicetool shows the general list of parameters for each service for the publishing via the R-GMA servicetool. For LB the following sub-services are published via the R-GMA servicetool and need to be updated accordingly:

- i. Log Server

Again, you find the necessary steps described in section 13.2.4.6.

*Note: Step 1,2 and 3 can also be performed by means of the remote site configuration file or a combination of local and remote configuration files*

4. As root run the LB configuration file `$GLITE_LOCATION/etc/config/scripts/glite-lb-config.py`
5. The LB Service is now ready.

## 8.5. LOGGING AND BOOKKEEPING CONFIGURATION WALKTHROUGH

The LB configuration script performs the following steps:

1. Set the following environment variables if not already set using the values defined in the global and lb configuration files:

```
GLITE_LOCATION      [default is /opt/glite]
GLOBUS_LOCATION    [default is /opt/globus]
```

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.cfg.xml`:

```
GLITE_LOCATION_VAR
GLITE_LOCATION_LOG
GLITE_LOCATION_TMP
```

3. Load the LB configuration file `$GLITE_LOCATION/etc/config/glite-lb.cfg.xml`
4. Set the following additional environment variables needed internally by the services (this requirement should disappear in the future):

```
PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/externals/bin:$GLOBUS_LOCATION/bin:$PATH
LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib:$GLOBUS_LOCATION/lib:$LD_LIBRARY_PATH
GLITE_HOST_CERT=/home/$GLITE_USER/hostcert.pem
GLITE_HOST_KEY=/home/$GLITE_USER/hostkey.pem
GLITE_CERT_DIR=< ca.certificate.dir >
```

5. Create or verify the `$GLITE_USER` account and configure it by modifying its `.bash_profile` and `.bashrc` scripts to source the `/etc/glite/profile.d/glite_setenv.sh` file created by the LB configuration script



6. Create and configure the MySQL database table required to store event information if it doesn't exist
7. Create the required symlinks to the MySQL sock file
8. Start MySQL and the LB Server daemons

## 9. WORKER NODE

### 9.1. SERVICE OVERVIEW

The gLite Standard Worker Node is a set of clients required to run jobs sent by the gLite Computing Element via the Local Resource Management System. It currently includes the gLite I/O Client, the Logging and Bookkeeping Client, the R-GMA Client and the WMS Checkpointing library.

### 9.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 9.2.1. Security Settings

Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` can be installed by downloading and running from the gLite web site (<http://www.glite.org>) the script `glite-security-utils_installer.sh` (Chapter 5). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the `glite-fetch-crl` script and sets up a crontab that periodically check for updated revocation lists

#### 9.2.2. Java JDK/JRE

The Java JRE or JDK are required to run the R-GMA Client in the Worker Node. This release requires v. 1.4.2 (revision 04 or greater). The JDK version to be used is a configuration parameter in the `glite-global-cfg.xml` file. Please change it according to your version and location (see also sections 4.2.3 and 6.4 for more details).

#### 9.2.3. Resource Management System

The Resource Management System client must be installed on the WN before installing and configuring the WN module. This release of the CE module supports PBS, Torque and LSF.

### 9.3. WORKER NODE INSTALLATION

1. Download from the gLite web site the latest version of the WN installation script `glite-wn_install.sh`. It is recommended to download the script in a clean directory
2. Make the script executable (`chmod u+x glite-wn_installer.sh`) and execute it or execute it with `sh glite-wn_install.sh`
3. Run the script as root. The requirement to run it as root is only temporary. The WN clients will be configured to be installed and run as non-privileged user before the final release. All the required RPMS are downloaded from the gLite software repository in the directory `glite-wn` next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

gLite I/O Client                      in `/opt/glite`

gLite LB Client	in /opt/glite
glite R-GMA Client	in /opt/glite
gLite WMS Checkpointing	in /opt/glite
Globus	in /opt/globus

- The gLite WN configuration scripts are installed in `$GLITE_LOCATION/etc/config/scripts/glite-wn-config.py`. A template configuration file is installed in `$GLITE_LOCATION/etc/config/templates/glite-wn.cfg.xml`. Since the WN is a collection of clients, the individual configuration scripts and files are also installed and they must be run. Please refer to the appropriate Chapters in this guide to install and configure the clients. This chapter only deals with WN general issues.

#### 9.4. WORKER NODE CONFIGURATION

- Copy the global configuration file template `$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml` to `$GLITE_LOCATION/etc/config`, open it and modify the parameters if required (Table 1)
- Copy the configuration file template from `$GLITE_LOCATION/etc/config/templates/glite-wn.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-wn.cfg.xml` and modify the parameters values as necessary (Table 6)
- Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The following parameters can be set:

Parameter	Default value	Description
<b>User-defined Parameters</b>		
voms.voname		The names of the VOs that this WN node can serve
pool.account.basename		The prefix of the set of pool account to be created. Existing pool accounts with this prefix are not recreated
pool.account.group		The group name of the pool accounts to be used
pool.account.number		The number of pool accounts to create. Each account will be created with a username of the form prefixXXX where prefix is the value of the pool.account.basename parameter. If matching pool accounts already exist, they are not recreated. The range of values for this parameter is 1-999
data.services		Information used for creation of the

		services.xml (ServiceDiscovery replacement) file. This file is used by the Data CLI tools. The format is: name;URL;serviceType where name is the unique name of the service. This is used in the command line if special services need to be addressed. URL is the service endpoint and serviceType is the java class defining the type of the service.
<b>Advanced Parameters</b>		
glite.installer.verbose	true	Enable verbose output
<b>System Parameters</b>		
wn.serviceList	glite-io-client, glite-rgma-client	The gLite services, clients or applications that compose this worker node. This parameters takes a comma-separated list of service names

**Table 6: WN Configuration Parameters**

4. Run the WN configuration file `$GLITE_LOCATION/etc/config/scripts/glite-wn-config.py`
5. The WN is now ready.

## 9.5. WORKER NODE CONFIGURATION WALKTHROUGH

The WN configuration script performs the following steps:

1. Set the following environment variables if not already set using the values defined in the global and lb configuration files:

```
GLITE_LOCATION          [default is /opt/glite]
```

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.cfg.xml`:

```
GLITE_LOCATION_VAR
GLITE_LOCATION_LOG
GLITE_LOCATION_TMP
```

3. Load the WN configuration file `$GLITE_LOCATION/etc/config/glite-wn.cfg.xml`
4. Set the following additional environment variables needed internally by the services (this requirement should disappear in the future):

```
PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/externals/bin:$PATH
LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib:$LD
```

## `_LIBRARY_PATH`

5. Create the number of pool accounts specified the service configuration file using the specified account name. If the group doesn't exist, it is created. If any of the pool accounts already exists it is not recreated, but it's configured by modifying its `.bash_profile` and `.bashrc` scripts to source the `/etc/glite/profile.d/glite_setenv.sh` file created by the WNS configuration script
6. Run each registered client configuration script

## 10. GLITE I/O

### 10.1. GLITE I/O SERVER

#### 10.1.1. Service Overview

GLite I/O server consists basically on the server of the AliEn aiod project, modified to support GSI authentication, authorization and name resolution plug-ins, together with other small features and bug fixes.

It includes plug-ins to access remote files using the dcap or the rfio client library.

It can interact with the FiReMan Catalog, the Replica Metadata Catalog and Replica Location Service, with the File and Replica Catalogs or with the Alien file catalog.

#### 10.1.2. Installation pre-requisites

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

##### 10.1.2.1. Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` (gLite Security Utilities) can be installed by downloading and running from the gLite web site (<http://www.glite.org>) the script `glite-security-utils_installer.sh` (Chapter 5). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the `glite-fetch-crl`, `glite-mkgridmap` and `mkgridmap.py` scripts and sets up cron jobs that periodically check for updated revocation lists and `grid-mapfile` entries
2. Customize the `mkgridmap` configuration file `$GLITE_LOCATION/etc/glite-mkgridmap.conf` by adding the required VOMS server groups. The information in this file is used to run the `glite-mkgridmap` script during the Security Utilities configuration to produce the `/etc/grid-security/grid-mapfile`
3. Install the server host certificate `hostcert.pem` and key `hostkey.pem` in `/etc/grid-security`

##### 10.1.2.2. Castor SRM

With some configuration of the Castor SRM, it is necessary to register the host DN of the gLite I/O Server in the Castor SRM server `gridmap-file`.

#### 10.1.3. gLite I/O Server installation

1. Download from the gLite web site the latest version of the gLite I/O server installation script `glite-io-server_installer.sh`. It is recommended to download the script in a clean directory
2. Make the script executable (`chmod u+x glite-io-server_installer.sh`) and execute it or execute it with `sh glite-io-server_installer.sh`
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory `glite-io-server` next to the installation script and the

installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.

4. If the installation is performed successfully, the following components are installed:

gLite I/O Server in /opt/glite  
 Globus in /opt/globus

5. The gLite I/O server configuration script is installed in

`$GLITE_LOCATION/etc/config/scripts/glite-io-server-config.py.`

A template configuration file is installed in

`$GLITE_LOCATION/etc/config/templates/glite-io-server.cfg.xml`

6. The gLite I/O server installs the R-GMA servicetool to publish its information to the information system R-GMA. The details of the installation of the R-GMA servicetool are described in section 13.2.4.5.

#### 10.1.4. gLite I/O Server Configuration

1. Copy the global configuration file template

`$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

open it and modify the parameters if required (Table 1)

2. Copy the configuration file template from

`$GLITE_LOCATION/etc/config/templates/ glite-io-server.cfg.xml`

to

`$GLITE_LOCATION/etc/config/ glite-io-server.cfg.xml`

and modify the parameter values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 7 gives an overview of the parameters that can be set. The R-GMA servicetool related parameters can be found in Table 15.

The parameters in the file can be divided into two categories:

- a. *Common parameters (first part of Table 7)*

These are the configuration parameters that are independent of the VO. Change all *changeme* values to the corresponding values.

Also you will find a section for the R-GMA servicetool to publish information about the rfiid. Adapt also these configuration values accordingly. You can find more information on the values and the R-GMA servicetool in section 13.2.4 and 13.2.4.6.

- b. *VO dependant gLite I/O Server parameters (second part of Table 7)*

A separate gLite I/O server instance can be installed for each VO that this server must support. The configuration file contains the list of parameters for

each VO, grouped by the tag

```

<instance name="changme" service="io-server">
...
</instance>

```

At least one VO instance must be defined. If you want to support multiple VOs, create a separate instance for each VO by copyi/paste the <instance> section in this file.

Next, change the name of each VO instance from 'changeme' to the VO name and adapt the parameters of each instance accordingly.

Also, there is an <instance> section for the R-GMA servicetool to publish the I/O server to R-GMA. For each VO instance that you create above, you have to create a <instance> of the R-GMA servicetool accordingly by copy paste the corresponding R-GMA <instance> section. Next, change the instance name from 'changeme' for each instance to the real VO name and adapt the parameters

```

rgma.servicetool.name
rgma.servicetool.url_endpoint
rgma.servicetool.status_script

```

for each VO instance accordingly. You can find more information on the values and the R-GMA servicetool in section 13.2.4 and 13.2.4.6.

<i>Common parameters</i>		
<i>All parameters defined in this table are common to all instances.</i>		
Parameter	Default value	Description
<b>User-defined Parameters</b>		
<i>I/O Daemon initialization parameters</i>		
init.username		The username of the user running the I/O Daemon. If using a astor with a castor SRM, in some configurations this user must be a valid user on the Castor server. If the user doesn't exist on this I/O Server, it will be created. The uid specified in the 'init.uid' parameters may be used.
init.groupname		The groupname of the user running the I/O Daemon. If using a Castor SRM, in some configurations this group must be a valid user on the Castor server. If the group doesn't exist I/O Server, it will be created. The gid specified in the 'init.gid' parameters may be used.
init.uid		The userid of the user running the I/O Daemon. If using a Castor



		SRM, in some configurations the same uid of the Castor user specified in the 'init.username' parameter must be set. Leave this parameter empty or comment it out to use a system assigned uid.
init.gid		The gid of the user running the I/O Daemon. If using a Castor SRM, in some configurations the same gid of the Castor group specified in the 'init.groupname' parameter must be set. Leave this parameter empty or comment it out to use a system assigned gid.
<b>Advanced Parameters</b>		
<i>General gLite initialization parameters</i>		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check for host certificate
<i>SSL Configuration parameters</i>		
service.certificates.type	host	This parameter is used to specify if service or host certificates should be used for the services. If this value is 'host', the existing host certificates are copied to the service user home in the directory specified by the 'user.certificate.path' parameter; the 'service.certificate.file' and 'service.key.file' parameters are ignored. If the value is 'service' the service certificates must exist in the location specified by the 'service.certificate.file' and 'service.key.file' parameters
service.certificate.file	<empty>	The service certificate (private key) file location.
<i>I/O Daemon parameters</i>		
io-daemon.MaxTransfers	20	The maximum number of concurrent transfers
io-resolve-common.SePort	8443	The port of the remote file operation server
io-resolve-common.SeProtocol	rfio	The protocol to be used to contact the remote file operation server. Currently the supported values are: * rfio: use the remote file io (rfio) protocol to access remotely the file * file: use normal posix operations

		to access a local file (useful only for testing purposes)
io-resolve-common.RootPathRule	abs_dir	<p>The rule to be applied to define the path for creating new files. Allowed values are:</p> <ul style="list-style-type: none"> <li>* abs_dir: The file name will be created by appending the file name to the path specified by RootPath configuration parameter</li> <li>* user_home_dir: the file name will be created by appending the file name to a path specified by the RootPath configuration parameter, a directory with the user name first letter and then the complete user name. [Note: Since at the moment the user name that is retrieved is the distinguished name, using that option is not suggested]</li> </ul>
io-authz-fas.FileOwner	<empty>	When checking the credentials, perform an additional check on that name to verify it was the user's name. Default value is an empty string, that means that this additional test is not performed
io-authz-fas.FileGroup	<empty>	When checking the credentials, perform an additional check on that name to verify it was one of the user's groups. Default value is an empty string, that means that this additional test is not performed
io-resolve-fireman.OverwriteOwnership	false	Overwrite the ownership of the file when creating it. If set to true, the newly created file will have as owner the values set by the FileOwner and FileGroup configuration parameters.
io-resolve-fireman.FileOwner	<empty>	The name of the group that will own any newly created file. This parameter is meaningful only if OverwriteOwnership is set to true. In case this parameter is not set, the Replica Catalog default will apply. Default value is an empty string.
io-resolve-fireman.FileGroup	<empty>	The name of the group of any newly created file. This parameters is meaningful only if OverwriteOwnership is set to true.

		In case this parameter is not set, the Replica Catalog default will apply. Default value is an empty string.
io-resolve-fr.OverwriteOwnership	false	Overwrite the ownership of the file when creating it. If set to true, the newly created file will have as owner the values set by the FileOwner and FileGroup configuration parameters. Default value is false.
io-resolve-fr.FileOwner		The name of the user that will own any newly created file. This parameter is meaningful only if OverwriteOwnership is set to true. In case this parameter is not set, the Replica Catalog default will apply. Default value is an empty string.
io-resolve-fr.FileGroup		The name of the group of any newly created file. This parameter is meaningful only if OverwriteOwnership is set to true. In case this parameter is not set, the Replica Catalog default will apply. Default value is an empty string.
<b>System Parameters</b>		
<i>I/O Daemon parameters</i>		
io-daemon.EnablePerfMonitor	false	Enable the Performance Monitor. If set to true, a process will be spawned to monitor the performance of the server and create some of the statistics.
io-daemon.PerfMonitorPort	9998	The Performance Monitor port
io-daemon.CacheDir	<empty>	The directory where cached files should be stored
io-daemon.CacheDirSize	0	The maximum size of the directory where cached files should be stored
io-daemon.PreloadCacheSize	5000000	The size of the preloaded cache
io-daemon.CacheLevel	0	The gLite I/O Cache Level
io-	false	Resynchronize the cache when the

daemon.ResyncCache		daemon starts
io-daemon.TransferLimit	100000000	The maximum bitrate expressed in b/s that should be used
io-daemon.CacheCleanupThreshold	90	When a cache clean up is performed, the cache will be clean up to that value. It should be intended as percentage, i.e. a value of 70 means that after a cleanup, the cache will be filled up to 70% of its maximum size
io-daemon.CacheCleanupLimit	90	Represent the limit that, when reached, triggers a cache clean up. It should be intended in percentage, i.e. a value of 90 means that when the 90% of cache is filled, the cached will be cleaned up up to the value specified by the CacheCleanupThreshold configuration parameter
io-daemon.RedirectionList	<empty>	The redirection list that should be used in the Cross-Link Cache Architecture
io-resolve-common.DisableDelegation	true	Don't use client's delegated credentials to contact the Web Services
io-authz-catalogs.DisableDelegation	true	Don't use client's delegated credentials to contact the RMC Service
io-authz-fas.DisableDelegation	true	Don't use client's delegated credentials to contact the FAS service
io-resolve-fr.DisableDelegation	true	Don't use client's delegated credentials to contact the RMC Service

*VO dependant gLite I/O Server instances*

*A separate gLite I/O Server instance can be installed for each VO that this server must support. The values in this table ('<instance>' section in the configuration file) are specific to that instance. At least one instance must be defined. Create additional instance sections for each additional VO you want to support on this node.*

Parameter	Default value	Description
<b>User-defined Parameters</b>		
vo.name		The name of the VO served by this instance.

io-daemon.Port		<p>The port to be used to contact the server. Please note that this port is only used for authentication and session establishment messages. When the real data transfer will be performed using a QUANTA parallel TCP stream a pool of sockets are opened on the server side binding a tuple of available ports from 50000 to 51000.</p> <p>This port should not be higher than 9999 and different I/O server instances should not run on contiguous ports (for example set one to 9999 and another one to 9998)</p>
init.CatalogType		<p>The type of catalog to use:</p> <ul style="list-style-type: none"> <li>- 'catalogs' (EDG Replica Location Service and Replica Metadata Catalog),</li> <li>- 'fireman' (gLite Fireman Catalog),</li> <li>- 'fr' (File and Replica Catalog)</li> </ul> <p>The parameters not used by the chosen catalog type can be removed or left empty</p>
<p><i>io-resolve-common parameters are required by all types of catalogues</i></p>		
io-resolve-common.SrmEndPoint		<p>The endpoint of the SRM Server. If that value starts with httpg://, the GSI authentication will be used (using the CGSI GSOAP plugin), otherwise no authentication is requested.</p> <p><i>Example:</i>  <a href="httpg://gridftp05.cern.ch:8443/srm/managerV1">httpg://gridftp05.cern.ch:8443/srm/managerV1</a></p>
io-resolve-common.SeHostname		<p>The name of the Storage Element where the files are staged. It's the hostname of the remote file operation server.</p> <p><i>Example:</i> <a href="http://gridftp05.cern.ch">gridftp05.cern.ch</a></p>
io-resolve-common.RootPath		<p>The path that should be prefixed to the filename when creating new files.</p> <p><i>Example:</i>  <a href="#">/castor/cern.ch/user/g/glite/VO-NAME/SE/</a></p>
<p><i>EDG RLS/RM parameters</i>  <i>The parameters are only required when using the EDG catalogs.</i></p>		

<i>Leave them empty or comment them if not used.</i>		
io-Authz-catalogs.RmcEndPoint		The endpoint of the RMC catalog. If that value starts with http:// the GSI authentication will be used (using the CGI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. This is also the value of the 'io-resolve-catalogs.RmcEndpoint' parameter. <i>Example: <a href="https://lxb2028:8443/VO-NAME/edg-replica-metadata-catalog/services/edg-replica-metadata-catalog">https://lxb2028:8443/VO-NAME/edg-replica-metadata-catalog/services/edg-replica-metadata-catalog</a></i>
io-resolve-catalogs.RIsEndpoint		The endpoint of the RIs catalog. If that value starts with http:// the GSI authentication will be used (using the CGI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. <i>Example: <a href="https://lxb2028:8443/VO-NAME/edg-local-replica-catalog/services/edg-local-replica-catalog">https://lxb2028:8443/VO-NAME/edg-local-replica-catalog/services/edg-local-replica-catalog</a></i>
<i>Parameters required by the Fireman and FR catalogs.</i>		
io-Authz-fas.FasEndPoint		The endpoint of the Fas catalog. If that value starts with http:// the GSI authentication will be used (using the CGI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. <i>Examples: <a href="http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FAS">http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FAS</a> (for FR) <a href="http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FiremanCatalog">http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FiremanCatalog</a> (for Fireman)</i>
<i>Fireman parameters</i>		
io-resolve-fireman.FiremanEndpoi		The endpoint of the FiReMan catalog. If that value starts with

nt		<p>http:// the GSI authentication will be used (using the CGI SOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGI SOAP plugin in SSL compatible mode), otherwise no authentication is requested.</p> <p><i>Example:</i>  <a href="http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FiremanCatalog">http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FiremanCatalog</a></p>
<b>FR parameters</b>		
io-resolve-fr.ReplicaEndPoint		<p>The endpoint of the Replica catalog. If that value starts with http:// the GSI authentication will be used (using the CGI SOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGI SOAP plugin in SSL compatible mode), otherwise no authentication is requested.</p> <p><i>Example:</i>  <a href="http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/ReplicaCatalog">http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/ReplicaCatalog</a></p>
io-resolve-fr.FileEndPoint		<p>The endpoint of the File catalog. If that value starts with http:// the GSI authentication will be used (using the CGI SOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGI SOAP plugin in SSL compatible mode), otherwise no authentication is requested. If that value is not set, the File Catalogs will not be contacted and the io-resolve-fr plug-in will managed only GUIDs.</p> <p><i>Example:</i>  <a href="http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FileCatalog">http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FileCatalog</a></p>
<b>Advanced Parameters</b>		
<i>Logging parameters</i>		
log.Priority	DEBUG	<p>The log4cpp log level. Possible values are: DEBUG, INFO, WARNING, ERROR, CRITICAL, ALERT, FATAL</p>

log.FileName	\$GLITE_LOCATION/LOG/glite-io-server.log	The location of the log file
--------------	--	------------------------------

**Table 7: gLite I/O Server Configuration Parameters**

3. Configure the R-GMA servicetool:

Copy the R-GMA servicetool configuration file template

`$GLITE_LOCATION/etc/config/templates/glite-rgma-servicetool.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 15 shows a list of the parameters that can be set. More details can be found in section 13.2.4.6.

**Note:** Step 1,2 and 3 can also be performed by means of the remote site configuration file or a combination of local and remote configuration files

4. As root run the gLite I/O server configuration file

`$GLITE_LOCATION/etc/config/scripts/glite-io-server-config.py`

5. The gLite I/O server is now ready.

### 10.1.5. gLite I/O Server Configuration Walkthrough

The gLite I/O server configuration script performs the following steps:

1. Set the following environment variables if not already set using the values defined in the global and gLite I/O server configuration files:

```
GLITE_LOCATION      [default is /opt/glite]
GLOBUS_LOCATION     [default is /opt/globus]
```

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.cfg.xml`:

```
GLITE_LOCATION_VAR
GLITE_LOCATION_LOG
GLITE_LOCATION_TMP
```

3. Load the WN configuration file `$GLITE_LOCATION/etc/config/glite-io-server.cfg.xml`
4. Set the following additional environment variables needed internally by the services (this requirement should disappear in the future):

```
PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/externals/bin:$PATH
LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib:$LD
```



## `_LIBRARY_PATH`

5. Create or verify the `$GLITE_USER` account and configure it by modifying its `.bash_profile` and `.bashrc` scripts to source the `/etc/glite/profile.d/glite_setenv.sh` file created by the configuration script
6. Copy the host certificates to the user account and link the `gridmap` file to this user account.
7. Configure the R-GMA servicetool and the service instances to publish via R-GMA.
8. Start the services

## 10.2. CLIENT

### 10.2.1. Service Overview

The gLite I/O Client provides some APIs (both posix and not) for accessing remote files using `glite-io`. It consists basically on a C wrapper of the `AlienIOclient` class provided by the `org.glite.data.io-base` module.

### 10.2.2. Installation pre-requisites

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 10.2.2.1. Security Settings

Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` can be installed by downloading and running from the gLite web site (<http://www.glite.org>) the script `glite-security-utils_installer.sh` (Chapter 5). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the `glite-fetch-crl` script and sets up a crontab that periodically check for updated revocation lists

### 10.2.3. gLite I/O Client installation

1. Download from the gLite web site the latest version of the gLite I/O server installation script `glite-io-client_installer.sh`. It is recommended to download the script in a clean directory
2. Make the script executable (`chmod u+x glite-io-client_installer.sh`) and execute it or execute it with `sh glite-io-client_installer.sh`
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory `glite-io-server` next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

gLite in `/opt/glite`  
Globus in `/opt/globus`

- The gLite I/O client configuration script is installed in `$GLITE_LOCATION/etc/config/scripts/glite-io-client-config.py`. A template configuration file is installed in `$GLITE_LOCATION/etc/config/templates/glite-io-client.cfg.xml`

#### 10.2.4. gLite I/O Client Configuration

- Copy the global configuration file template `$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml` to `$GLITE_LOCATION/etc/config`, open it and modify the parameters if required (Table 1)
- Copy the configuration file template `$GLITE_LOCATION/etc/config/templates/glite-io-client.cfg.xml` to `$GLITE_LOCATION/etc/config/` and modify the parameter values as necessary (Table 6)
- Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The following parameters can be set:

Parameter	Default value	Description
<b>User-defined Parameters</b>		
io-client.Server	changeme	The hostname where the gLite I/O Server is running
<b>Advanced Parameters</b>		
glite.installer.verbose	true	Enable configuration script verbose output
io-client.ServerPort	9999	The port that the gLite I/O Server is listening for
io-client.EncryptName	true	Enable encryption of the file name when sending a remote open request
io-client.EncryptData	false	Enable encryption of the data block send and received
log.Priority	DEBUG	The log4cpp log level. Possible values are: 'DEBUG', 'INFO', 'WARNING', 'ERROR', 'CRITICAL', 'ALERT', 'FATAL'
log.FileName	<code>\$GLITE_LOCATION_LOG/glite-io-client.log</code>	The location of the log file
<b>System Parameters</b>		
io-client.CacheLevel	7	The AliEn aiod Cache Level value
io-client.NumberOfStreams	1	Number of QUANTA tcp parallel streams

**Table 8: gLite I/O Client configuration parameters**



4. As root run the gLite I/O client configuration file  
`$GLITE_LOCATION/etc/config/scripts/glite-io-client-config.py`
5. The gLite I/O client is now ready.

## 11. LOCAL TRANSFER SERVICE

### 11.1. SERVICE OVERVIEW

The data movement services of gLite are responsible to securely transfer files between Grid sites. The transfer is performed always between two gLite Storage Elements having the same transfer protocol available to them (usually gsiftp). The gLite Local Transfer Service is composed of two separate services, the File Transfer Service and the File Placement Service, and a number of transfer agents.

The File Transfer Service is responsible for the actual transfer of the file between the SEs. It takes the source and destination names as arguments and performs the transfer. The FTS is managed by the site administrator, i.e. there is usually only one such service serving all VOs. The File Placement Service performs the catalog registration in addition to the copy. It makes sure that the catalog is only updated if the copy through the FTS was successful. The user will see this as a single atomic operation. The FPS is instantiated per VO. If a single node must support multiple VOs, then multiple instances of the FPS can be installed and configured.

The Data Transfer Agents perform data validation and scheduling operation. There are currently three agents, the Checker, the Fetcher and the Data Integrity Validator. They are instantiated per VO.

### 11.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 11.2.1. Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` can be installed by downloading and running from the gLite web site (<http://www.glite.org>) the script `glite-security-utils_installer.sh` (Chapter 5). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the `glite-fetch-crl` script and sets up a crontab that periodically check for updated revocation lists
2. Install the server host certificate `hostcert.pem` and key `hostkey.pem` in `/etc/grid-security`

#### 11.2.2. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA Server. This release requires v. 1.4.2 (revision 04 or greater). The JDK/JRE version to be used is a parameter in the configuration file. Please change it according to your version and location.

Due to license reasons, we cannot redistribute Java. Please download it from <http://java.sun.com/> and install it if you have not yet installed it.

### 11.3. LOCAL TRANSFER SERVICE INSTALLATION

1. Download from the gLite web site the latest version of the LTS installation script *glite-data-local-transfer-service\_install.sh*. It is recommended to download the script in a clean directory
2. Make the script executable (*chmod u+x glite-data-local-transfer-service\_installer.sh*) and execute it or execute it with *sh glite-data-local-transfer-service\_install.sh*
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory *glite-data-local-transfer-service* next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

gLite	in /opt/glite
Tomcat	in /var/lib/tomcat5
MySQL	in /usr/bin/mysql

5. The gLite LTS configuration script is installed in *\$GLITE\_LOCATION/etc/config/scripts/glite-data-local-transfer-service-config.py*. A template configuration file is installed in *\$GLITE\_LOCATION/etc/config/templates/glite-data-local-transfer-service.cfg.xml*

#### 11.4. LOCAL TRANSFER SERVICE CONFIGURATION

1. Copy the global configuration file template *\$GLITE\_LOCATION/etc/config/template/glite-global.cfg.xml* to *\$GLITE\_LOCATION/etc/config*, open it and modify the parameters if required (Table 1)
2. Copy the configuration file template from *\$GLITE\_LOCATION/etc/config/templates/glite-data-local-transfer-service.cfg.xml* to *\$GLITE\_LOCATION/etc/config/glite-data-local-transfer-service.cfg.xml* and modify the parameters values as necessary (Table 9)
3. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The following parameters can be set:

Parameter	Default value	Description
<b>User-defined Parameters</b>		
<i>transfer-fps.vo.name</i>		Name of the VO for a given instance
<i>transfer-fps.db.name</i>		Database name for File Placement Service. Ex. 'transfer'
<i>transfer-fps.db.user</i>		Name of the database user owning the fps database
<i>transfer-fps.db.password</i>		Password for accessing the fps database
<i>transfer-fps.catalog.url</i>		URL of the catalog to connect to.

		Ex.: <a href="http://lxb1427.cern.ch:8080/glite-data-catalog-service-fr-mysql/services/FiremanCatalog">http://lxb1427.cern.ch:8080/glite-data-catalog-service-fr-mysql/services/FiremanCatalog</a>
transfer-agent.user.name		User which will own the transfer-agent in the crontab
<b>Advanced Parameters</b>		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
glite-data-transfer-fps.DBRESOURCENAME	transfer	
transfer-fps.catalog.update.interval	10	Interval between checks for files in Done state by the CatalogUpdater thread (in miliseconds)
transfer-fps.catalog.query.attribute	<empty>	If this attribute is not set for the service, every client can see every job. If this attribute is set for the service, then only clients exposing this attribute can see all the jobs. Other clients can only see jobs which correspond to their DN
transfer-fps.catalog.submit.attribute	<empty>	If this attribute is not set for the service, every client can submit new jobs. If this attribute is set for the service, only clients exposing this attribute are able to submit new jobs
transfer-fps.catalog.cancel.attribute	<empty>	If this attribute is not set for the service, any client can cancel any job. If this attribute is set for the service, clients not exposing this attribute can only cancel their jobs (the ones with the same DN), while clients exposing this attribute are allowed to cancel any job
transfer-agent-vo.Quota	70	The percentage of the max concurrent transfers that the VO is allowed to submit. For example, if the Transfer Service is able to process 1000 request at the same time, a quota of 60 means that the VO can run simoutaneously up to 600 (1000 * 0.6) Transfers.
transfer-agent-fts-urlcopy.MaxTransfers	10	The maximum number of transfers the Glite UrlCopy can process simultaneously. Those transfers

		are intended per VO.
transfer-agent-fts-urlcopy.Streams	1	The number of parallel streams that should be used during the transfer.
glite-data-transfer-fps.CATALOG_DISABLE	false	Disable catalog lookup
transfer-agent.check.frequency	2	Delay (in minutes) between two transfer-agent Check actions
transfer-agent.fetch.frequency	3	Delay (in minutes) between two transfer-agent Check actions
<b>System Parameters</b>		
transfer-fps.db.driver-class	org.gjt.mm.mysql.Driver	JDBC driver classname
transfer-fps.name	glite-data-transfer-fps	FPS name
transfer-fps.docBase	\${GLITE_LOCATION}/share/java/glite-data-transfer-fps.war	Location of the transfer-fps.war file
transfer-fps.db.url	jdbc:mysql://localhost:3306/\${transfer-fps.db.name}	Used DB URL
transfer-agent-dao-mysql.HostName	localhost	Host where the MySQL database is running
transfer-agent-dao-mysql.DBName	\${transfer-fps.db.name}	Must match the transfer-fps.db.name
transfer-agent-dao-mysql.User	\${transfer-fps.db.user}	Must match the transfer-fps.db.user
transfer-agent-dao-mysql.Password	\${transfer-fps.db.password}	Must match the transfer-fps.db.password
transfer-agent-vo.Name	\${transfer-fps.vo.name}	Must match the transfer-fps.vo.name
transfer-agent-dao-mysql.SocketName	/var/log/mysql/mysql.sock	MySQL sock file
transfer-fps.CHANNEL_NAME		In the current release should not be changed
transfer-fps.CHANNEL_DOMAIN_A		In the current release should not be changed
transfer-fps.CHANNEL_DOMAIN_B		In the current release should not be changed
transfer-fps.CHANNEL_CONTAINER		In the current release should not be

CT		changed
transfer-fps.CHANNEL_BAND WIDTH		In the current release should not be changed

**Table 9: Local Transfer Service Configuration Parameters**

4. As root run the LTS configuration file `$GLITE_LOCATION/etc/config/scripts/glite-data-local-transfer-service-config.py`
5. The Local Transfer Service is now ready.

### 11.5. LOCAL TRANSFER SERVICE CONFIGURATION WALKTHROUGH

The Local Transfer Service configuration script performs the following steps:

1. Set the following environment variables if not already set using the values defined in the global and service configuration files:

```
GLITE_LOCATION          [default is /opt/glite]
```

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.cfg.xml`:

```
GLITE_LOCATION_VAR
GLITE_LOCATION_LOG
GLITE_LOCATION_TMP
CATALINA_HOME
```

3. Load the global and the Local Transfer Service configuration files `$GLITE_LOCATION/etc/config/glite-data-local-transfer-service.cfg.xml`
4. Set the following additional environment variables needed internally by the services (this requirement should disappear in the future):

```
PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/externals/bin:$PATH
LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib:
                $LD_LIBRARY_PATH
```

5. Create the FPS MySQL database and give access to the specified user with the given password
6. Installs and configures the File Placement Service instance(s) by loading the web application Tomcat context file and loading the schema and the static channel info data in MySQL
7. Installs the gLite Data Management base libraries in the Tomcat lib directory
8. Installs and configures the data transfer agents instance(s) by creating the service configuration files and installing the agent cron jobs and touché the cron daemon pool directory to activate the jobs



## 12. SINGLE CATALOG

### 12.1. SERVICE OVERVIEW

On the Grid, the user identifies files using Logical File Names (LFN).

The LFN is the key by which the users refer to their data. Each file may have several replicas, i.e. managed copies. The management in this case is the responsibility of the File and Replica Catalog.

The replicas are identified by Site URLs (SURLs). Each replica has its own SURL, specifying implicitly which Storage Element needs to be contacted to extract the data. The SURL is a valid URL that can be used as an argument in an SRM interface (see section [\*]). Usually, users are not directly exposed to SURLs, but only to the logical namespace defined by LFNs. The Grid Catalogs provide mappings needed for the services to actually locate the files. To the user the illusion of a single file system is given.

Currently gLite provides two different modules for installing the catalog on MySQL or on Oracle. The names of the modules are:

<code>glite-data-single-catalog</code>	→	MySQL version
<code>glite-data-single-catalog-oracle</code>	→	Oracle version

In what follows the installation instructions are given for a generic single catalog version. Whenever the steps or requirements differ for MySQL and Oracle it will be noted. Replace `glite-data-single-catalog` with `glite-data-single-catalog-oracle` to use the Oracle version.

### 12.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 12.2.1. Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` can be installed by downloading and running from the gLite web site (<http://www.glite.org>) the script `glite-security-utils_installer.sh` (Chapter 5). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the `glite-fetch-crl` script and sets up a crontab that periodically check for updated revocation lists
2. Install the server host certificate `hostcert.pem` and key `hostkey.pem` in `/etc/grid-security`

#### 12.2.2. Java JDK

The Java JDK is required to run the Single Catalog Server. This release requires v. 1.4.2 (revision 04 or greater). The JDK version to be used is a parameter in the configuration file. Please change it according to your version and location.

Due to license reasons, we cannot redistribute Java. Please download it from <http://java.sun.com/> and install it if you have not yet installed it.

### 12.2.3. Oracle JDBC Drivers

The Oracle version requires the JDBC drivers (ocrs12.jar, ojdbc14.jar, orai18n.jar) to be installed on the server before running the installation script. These packages cannot be redistributed and are subject to export restrictions. Please download them from the Oracle web site

[http://www.oracle.com/technology/software/tech/java/sqlj\\_jdbc/htdocs/jdbc101040.html](http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc101040.html)

and install them in `${CATALINA_HOME}/common/lib`.

### 12.3. SINGLE CATALOG INSTALLATION

1. Download from the gLite web site the latest version of the SC installation script *glite-data-single-catalog\_install.sh*. It is recommended to download the script in a clean directory
2. Make the script executable (*chmod u+x glite-data-single-catalog\_installer.sh*) and execute it or execute it with *sh glite-data-single-catalog\_install.sh*
3. Run the script as root. All the required RPMs are downloaded from the gLite software repository in the directory *glite-data-local-transfer-service* next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

gLite            in /opt/glite

5. The gLite SC configuration script is installed in *\${GLITE\_LOCATION}/etc/config/scripts/glite-data-single-catalog-config.py*. A template configuration file is installed in *\${GLITE\_LOCATION}/etc/config/templates/glite-data-single-catalog.cfg.xml*

### 12.4. SINGLE CATALOG CONFIGURATION

1. Copy the global configuration file template *\${GLITE\_LOCATION}/etc/config/template/glite-global.cfg.xml* to *\${GLITE\_LOCATION}/etc/config*, open it and modify the parameters if required (Table 1)
2. Copy the configuration file template from *\${GLITE\_LOCATION}/etc/config/templates/glite-data-single-catalog.cfg.xml* to *\${GLITE\_LOCATION}/etc/config/glite-data-single-catalog.cfg.xml* and modify the parameters values as necessary (Table 10)
3. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The following parameters can be set:

Parameter	Default value	Description
-----------	---------------	-------------

<b>User-defined Parameters</b>		
catalog-service-fr-mysql.VONAME		Name of the Virtual Organisation which is served by the catalog instance
catalog-service-fr-mysql.DBNAME		Database name used for a catalog service
catalog-service-fr-mysql.DBUSER		Database user name owning the catalog database
catalog-service-fr-mysql.DBPASSWORD		Password for accessing the catalog database
<b>Advanced Parameters</b>		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
<b>System Parameters</b>		
catalog-service-fr-mysql.DBURL	Mysql: jdbc:mysql://localhost:3306/\${catalog-service-fr-mysql.DBNAME}	URL of the database

**Table 10: Single Catalog Configuration Parameters**

4. As root run the Single Catalog configuration file  
`$GLITE_LOCATION/etc/config/scripts/glite-data-single-catalog-config.py`
5. The Single Catalog is now ready.

## 12.5. SINGLE CATALOG CONFIGURATION WALKTHROUGH

The Single Catalog configuration script performs the following steps:

1. Set the following environment variables if not already set using the values defined in the global and lb configuration files:

`GLITE_LOCATION`                      [default is /opt/glite]

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.cfg.xml`:

`GLITE_LOCATION_VAR`  
`GLITE_LOCATION_LOG`  
`GLITE_LOCATION_TMP`

3. Load the Local Transfer Service configuration file  
*\$GLITE\_LOCATION/etc/config/glite-data-single-catalog.cfg.xml*
4. Set the following additional environment variables needed internally by the services  
(this requirement should disappear in the future):

```
PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/externals/bin:$GLOBUS_LOCA  
TION/bin:$PATH  
LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib:  
$LD_LIBRARY_PATH  
GLITE_HOST_CERT=/home/$GLITE_USER/hostcert.pem  
GLITE_HOST_KEY=/home/$GLITE_USER/hostkey.pem  
GLITE_CERT_DIR=< ca.certificate.dir >
```

## 13. INFORMATION AND MONITORING SYSTEM (R-GMA)

### 13.1. SERVICE OVERVIEW

The R-GMA (Relational Grid Monitoring Architecture) is the Information and Monitoring Service of gLite. It is based on the Grid Monitoring Architecture (GMA) from the Grid Global Forum (GGF), which is a simple Consumer-Producer model that models the information infrastructure of a Grid as a set of *consumers* (that request information), *producers* (that provide information) and a central *registry* which mediates the communication between producers and consumers. R-GMA offers a global view of the information as if each Virtual Organisation had one large relational database.

Producers contact the registry to announce their intention to publish data, and consumers contact the registry to identify producers, which can provide the data they require. The data itself passes directly from the producer to the consumer: it does not pass through the registry.

R-GMA adds a standard query language (a subset of SQL) to the GMA model, so consumers issue SQL queries and receive tuples (database rows) published by producers, in reply. R-GMA also ensures that all tuples carry a *time-stamp*, so that monitoring systems, which require time-sequenced data, are inherently supported.

#### 13.1.1. R-GMA SERVER

The R-GMA server part is divided into four components:

- **R-GMA server**  
The server component of the information system. There can be one or several servers per grid site.
- **R-GMA schema server**  
The server for the schema used in the information system. There is one common global schema server for the grid.
- **R-GMA registry server**  
The registry server for the grid. There can be one or several registry servers per grid.
- **R-GMA browser**  
Browser to browse the information of the information system via a web browser.

#### 13.1.2. R-GMA CLIENT

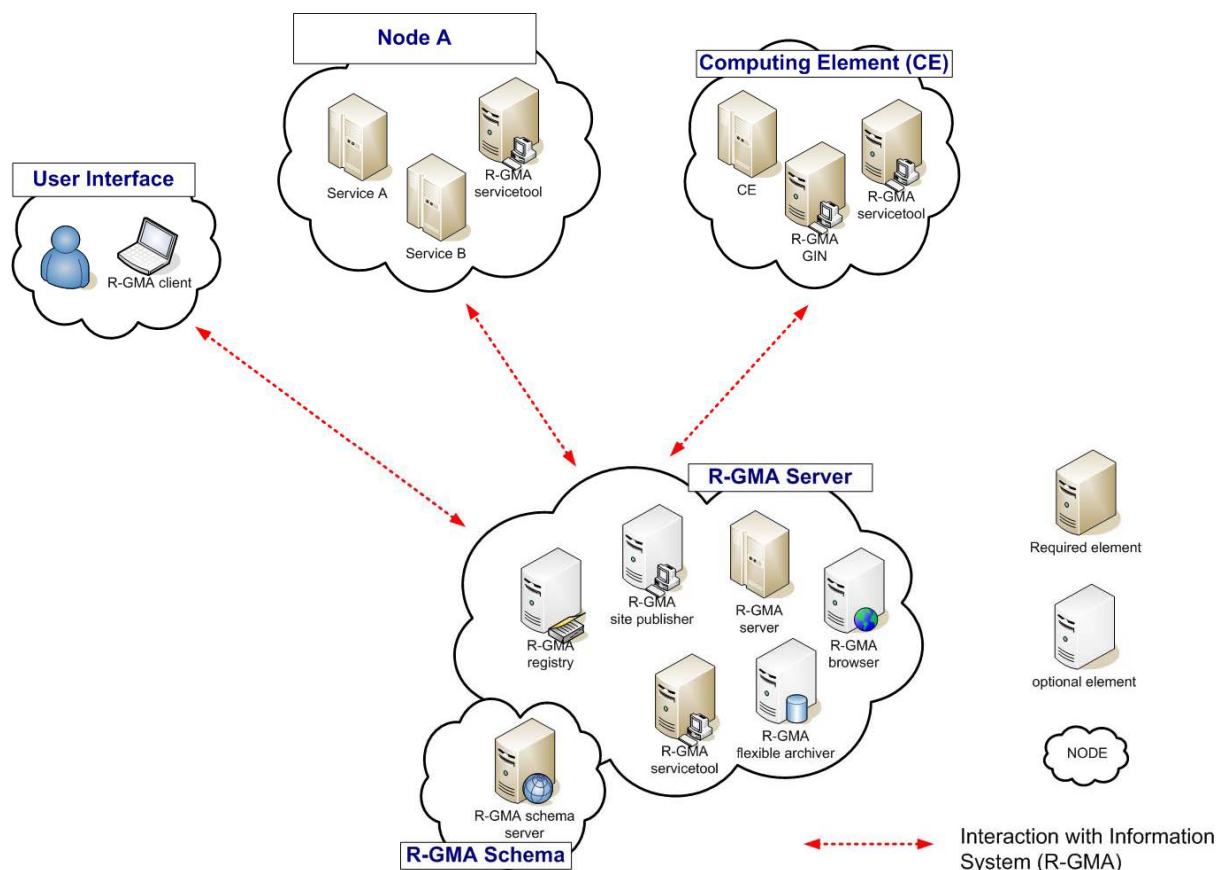
The client part of R-GMA contains the producer and consumers of information. There is one generic client and a set of four specialized clients to deal with a certain type of information:

- **Generic client**  
A generic set of APIs for different languages and command line interfaces that are installed on the User interface.
- **R-GMA servicetool**  
Client to publish the existence and status of (a) service(s). The servicetool is used by each service that wants to publish its existence and status to R-GMA.
- **R-GMA site publisher**  
Client to publish the existence of a site. Each site must have one R-GMA site publisher.
- **R-GMA GadgetIN (GIN)**  
Client to extract information from MDS and to republish it to R-GMA. The R-GMA GadgetIN is used by the Computing Element (CE) to publish its information.

- **R-GMA data archiver (flexible archiver)**

Client to make the data from the R-GMA site-publisher, servicetool and GIN constantly available. By default the glue and service tables are archived, however this can be configured.

Figure 2 gives an overview of the R-GMA architecture and the distribution of the different R-GMA components.



**Figure 2 R-GMA components**

## 13.2. R-GMA DEPLOYMENT MODULES

In order to facilitate the installation of the information system R-GMA, the different components of the server and clients have been combined into one R-GMA server deployment module and several client sub-deployment modules that are automatically installed together with the corresponding gLite deployment modules that use them. Table 11 gives a list of R-GMA deployment modules, their content and/or the list of gLite deployment modules that install/use them.

### 13.2.1. R-GMA Deployment strategy

In order to use the information system R-GMA, you first have to install the R-GMA server on one node. If you want, you can install further R-GMA servers on other nodes. The following

<b>Deployment module</b>	<b>Contains</b>	<b>Used / included by</b>
R-GMA server	R-GMA server R-GMA registry server R-GMA schema server R-GMA browser R-GMA site publisher R-GMA data archiver R-GMA servicetool	
R-GMA client	RGMA client APIs	User Interface (UI) Worker Node (WN)
R-GMA servicetool	R-GMA servicetool	Computing Element (CE) Data Local Transfer Service Data Single Catalog (MySQL) Data Single Catalog (Oracle) I/O-Server Logging & Bookkeeping (LB) R-GMA server Torque Server VOMS Server Workload Management System (WMS)
R-GMA GIN	R-GMA GadgetIN	Computing Element (CE)

**Table 11: R-GMA deployment modules**

rules have to be taken into account when installing a single or multiple servers and enabling/disabling the different options of the server(s):

- There must be exactly one schema server for your grid.
- There must be one or several registry server per grid.
- There must be one site publisher per site.
- You can choose to enable/disable the data archiver.

Next, you can install the different services, e.g. the Computing Element. All necessary R-GMA components needed by a service are automatically downloaded and installed together with the service. You will only need to configure the corresponding parts of R-GMA by modifying the corresponding configuration files accordingly.

There is one common R-GMA configuration file (*glite-rgma-common.cfg.xml*) that is used by all R-GMA components to handle common R-GMA settings and that is shipped with each R-GMA component. In addition, each R-GMA component comes with its own configuration file (see the following sections for details).

## 13.2.2. R-GMA Server deployment module

### 13.2.2.1. R-GMA Server deployment module overview

The R-GMA server is the central server of the R-GMA service infrastructure. It contains the four R-GMA server parts – server, schema, registry and browser (see section 13.1.1) as well as the R-GMA clients – R-GMA servicetool, site publisher and data archiver (see section 13.1.2):

- The *R-GMA server* is always turned on to enable the server capabilities.
- For the *R-GMA schema*, the user can decide to turn it on or off. Remember there must be exactly one central schema server for the grid.
- For the *R-GMA registry*, the user can decide to turn it on or off – there can be one or more registries per grid.
- For the *R-GMA browser*, the user can decide to turn it on or off – it is recommended to turn it on to give the user the possibility to browse the information of the information system via a web browser.
- The *R-GMA servicetool* is always turned on to publish the server services.
- For the *R-GMA site publisher*, the user can decide to turn it on or off. There must be one site publisher enabled per site. If you run multiple R-GMA server on one site, turn on the site publisher for one server and off for the others.
- For the *R-GMA data archiver (flexible archiver)*, the user can decide to turn it on or off.

### 13.2.2.2. Installation Pre-requisites

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

### 13.2.2.3. Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called *glite-security-utils* (gLite Security Utilities) is installed and configured automatically when installing and configuring the R-GMA Server (refer to Chapter 5 for more information about the Security Utilities module). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl*, *glite-mkgridmap* and *mkgridmap.py* scripts and sets up cron jobs that periodically check for updated revocation lists and grid-mapfile entries if required).
2. Install the server host certificate *hostcert.pem* and key *hostkey.pem* in `/etc/grid-security`

### 13.2.2.4. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA Server. This release requires v. 1.4.2 (revision 04 or greater). The JDK/JRE version to be used is a parameter in the configuration file. Please change it according to your version and location.

Due to license reasons, we cannot redistribute Java. Please download it from <http://java.sun.com/> and install it if you have not yet installed it.



### 13.2.2.5. R-GMA Server Installation

1. Download the latest version of the R-GMA server installation script *glite-rgma-server\_installer.sh* from the gLite web site. It is recommended to download the script in a clean directory, e.g. gLite.
2. Make the script executable

```
chmod u+x glite-rgma-server_installer.sh
```

and execute it or execute it with

```
sh glite-rgma-server_installer.sh
```

3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory *glite-rgma-server* next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

<i>gLite</i>	in /opt/glite (\$GLITE_LOCATION)
<i>gLite-essentials- java</i>	in \$GLITE_LOCATION/externals/share
<i>MySQL-server</i>	in /usr
<i>MySQL-client</i>	in /usr
<i>MySQL-shared-compat</i>	in /usr
<i>Tomcat</i>	in /var/lib/tomcat5

5. The gLite R-GMA server configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/glite-rgma-server-config.py.
```

All the necessary template configuration files are installed into

```
$GLITE_LOCATION/etc/config/templates/
```

The next section will guide you through the different files and necessary steps for the configuration.

### 13.2.2.6. R-GMA Server Configuration

1. Copy the global configuration file template

```
$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml
```

to

```
$GLITE_LOCATION/etc/config,
```

open it and modify the parameters if required. For details about the parameters, please see Table 1.

2. Copy the R-GMA common configuration file template

```
$GLITE_LOCATION/etc/config/templates/glite-rgma-common.cfg.xml
```

to

```
$GLITE_LOCATION/etc/config
```

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 12 shows the list of parameters that can be set.

Parameter	Default value	Description
<b>User-defined parameters</b>		
rgma.server.hostname		Hostname of the R-GMA server. <i>Example: lxb1420.cern.ch</i>
rgma.schema.hostname		Host name of the R-GMA schema service. <i>Example: lxb1420.cern.ch</i> (See also configuration parameter 'rgma.server.run_schema_service' in the R-GMA server configuration file in case you install a server)
rgma.registry.hostname		Host name of the R-GMA registry service. You must specify at least one and you can specify several if you want to use several registries. <i>Example: lxb1420.cern.ch</i> (See also configuration parameter 'rgma.server.run_registry_service' in the R-GMA server configuration file in case you install a server).
<b>Advanced Parameters</b>		
<b>System Parameters</b>		
rgma.user.name	rgma	Name of the user account used to run the R-GMA gLite services. <i>Example: rgma</i>
rgma.user.group	rgma	Group of the user specified in the parameter 'rgma.user.name'. <i>Example: rgma</i>

**Table 12: R-GMA common configuration parameters**

- Copy the R-GMA server configuration file template

`$GLITE_LOCATION/etc/config/templates/glite-rgma-server.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 13 shows the list of parameters that can be set.

Parameter	Default value	Description
<b>User-defined Parameters</b>		
rgma.server.mysql_root_password		MySQL root password. <i>Example: verySecret</i>
rgma.server.run_schema_service	no	Run a schema server by yourself (yes no). If you want to run it on your machine set 'yes' and set the parameter 'rgma.schema.hostname' to the hostname of your machine otherwise set it to 'no' and set the 'rgma.schema.hostname' to the host name of the schema server you want to use. <i>Example: yes</i>
rgma.server.run_registry_service	no	Run a registry server by yourself (yes no). If you want to run it on your machine set 'yes' and add your hostname to the parameter list 'rgma.registry.hostname' otherwise set it to 'no'. <i>Example: yes</i>
rgma.server.run_browser	yes	Run an R-GMA browser (yes no). Running a browser is optional but useful. <i>Example: yes</i>
rgma.server.run_archiver	no	Run the R-GMA data archiver (yes no). Running an archiver makes the data from the site-publisher, servicetool and GadgetIN constantly available. If you turn on this option, by default the glue and service tables are archived. To change the archiving behaviour, you have to create/change the archiver configuration file and point the parameter 'rgma.server.archiver_configuration_file' to this location (see below). <i>Example: yes</i>
rgma.server.run_site_publisher	yes	Run the R-GMA site-publisher (yes no). Running the site-publisher publishes your site to R-GMA. <i>Example: yes</i>
rgma.site-publisher.site-name		Hostname of the site. It has to be a DNS entry owned by the site and does not have to be shared with another site (i.e it uniquely identifies the site). <i>Example: lxb1420.cern.ch</i>
rgma.site-publisher.contact.system_administrator		Contact email address of the site system administrator. <i>Example:</i>

		<i>systemAdministrator@mysite.com</i>
rgma.site-publisher.contact.user_support		Contact email address of the user support. <i>Example: userSupport@mysite.com</i>
rgma.site-publisher.contact.site_security		Contact email address of the site security responsible. <i>Example: security@mysite.com</i>
rgma.site-publisher.location.latitude		Latitude of your site. Please go to <a href="http://www.multimap.com/">'http://www.multimap.com/'</a> to find the correct value for your site. <i>Example: 46.2341</i>
rgma.site-publisher.location.longitude		Longitude of your site. Please go to <a href="http://www.multimap.com/">'http://www.multimap.com/'</a> to find the correct value for your site. <i>Example: 6.0447</i>
<b>Advanced Parameters</b>		
glite.installer.verbose	true	Enable verbose output. <i>Example : true</i>
rgma.server.httpconnector.maxthread	1000	Maximum number of threads that are created for the tomcat http connector to process requests. This, in turn specifies the maximum number of concurrent requests that the connector can handle. <i>Example: 1000</i>
rgma.server.archiver_configuration_file	<code>\${GLITE_LOCATION}/etc/rgma-flexible-archiver/glue-config.props</code>	Configuration file to be used to setup the flexible-archiver database to select which tables are supposed to be backuped. By default, the glue and service tables are selected. (See also parameter 'rgma.server.run_archiver') <i>Example: '/my/path/my_config_file.props'</i>
<b>System Parameters</b>		
rgma.server.webapp.path	R-GMA	Path under which R-GMA server should be deployed. <i>Example: R-GMA</i>
rgma.server.war.name	R-GMA.war	Name of war file for R-GMA server. <i>Example: R-GMA.war</i>

**Table 13: R-GMA server Configuration Parameters**

4. As root run the R-GMA server configuration file  
`$GLITE_LOCATION/etc/config/scripts/glite-rgma-server-config.py`
5. The R-GMA Server is completely configured and running.

### 13.2.2.7. R-GMA Server Configuration Walkthrough

The R-GMA configuration script performs the following steps:

1. Reads the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.csf.xml`:

<code>GLITE_LOCATION_VAR</code>	[default is <code>/var/glite</code> ]
<code>GLITE_LOCATION_LOG</code>	[default is <code>/var/log/glite</code> ]
<code>GLITE_LOCATION_TMP</code>	[default is <code>/tmp/glite</code> ]

2. Sets the following environment variables if not already set using the values set in the global and R-GMA configuration files:

<code>GLITE_LOCATION</code>	[ <code>=/opt/glite</code> if not set anywhere]
<code>CATALINA_HOME</code>	to the location specified in the global configuration file [default is <code>/var/lib/tomcat5/</code> ]
<code>JAVA_HOME</code>	to the location specified in the global configuration file

3. Configures the gLite Security Utilities module
4. Checks the directory structure of `$GLITE_LOCATION`.
5. Load the R-GMA server configuration file  
`$GLITE_LOCATION/etc/config/glite-rgma-server.cfg.xml`  
and the R-GMA common configuration file  
`$GLITE_LOCATION/etc/config/glite-rgma-common.cfg.xml`  
and checks the configuration values.
6. Prepares the tomcat environment by:
  - a. setting the `CATALINA_OPTS` for the maximum java heap size '`-Xmx`' to half the memory size of your machine.
  - b. setting the maximum number of threads for the http connector using the configuration value.
  - c. deploying the R-GMA server application by creating the corresponding context file in `$CATALINA_HOME/conf/Catalina/localhost/XXX.xml` where `XXX` is the deploy path name of the R-GMA server specified in the configuration file (the default is R-GMA).
7. Configures the general R-GMA setup by running the R-GMA setup script  
`$GLITE_LOCATION/share/rgma/scripts/rgma-setup.py`  
using the configuration values from the configuration file for server, schema and registry hostname.
8. Exports the environment variable `RGMA_HOME` to `$GLITE_LOCATION`
9. Starts the MySQL server.
10. Sets the MySQL root password using the configuration value. If the password is already set, the script verifies if the present password and the one specified by the configuration file are the same.
11. Configures the R-GMA server by running the R-GMA server setup script  
`$GLITE_LOCATION/share/rgma/scripts/rgma-server-setup.py`  
using the option to run/not run a schema, registry and browser from the configuration file.
12. Fills the MySQL DB with the R-GMA configuration.
13. Configures the R-GMA server security by updating the file

`$GLITE_LOCATION/etc/rgma-server/ServletAuthentication.props`

with the location of the keys and cert files for tomcat.

14. If the site publisher or data archiver (flexible-archiver) are turned on in the configuration, the R-GMA client security is configured:
  - a. The hostkey and certificates are copied to the `.cert` subdirectory of the R-GMA user home directory.
  - b. The security configuration file for the client `'$GLITE_LOCATION/etc/rgma/ClientAuthentication.props'` is updated with the location of the cert and key files.
15. If the site publisher is turned on in the configuration, the site publisher will be configured:
  - a. The configuration file `'$GLITE_LOCATION/etc/rgma-publish-site/site.props'` is updated with the site name and the corresponding contact addresses.
16. If the data archiver (flexible-archiver) is turned on in the configuration, the flexible archiver is configured:
  - a. The configuration file for the archiver properties, specified in the configuration parameter `'rgma.server.archiver_configuration_file'` is copied to `'$GLITE_LOCATION/etc/rgma-flexible-archiver/flexy.props'`.
  - b. The flexible-archiver database is set up via `'$GLITE_LOCATION/bin/rgma-flexible-archiver-db-setup \ $GLITE_LOCATION/etc/rgma-flexible-archiver/flexy.props'`.
17. Starts the MySQL server.
18. Starts the tomcat server and gives time to go up to full speed before continuing
19. Starts the data archiver if enabled.
20. Starts the site publisher if enabled.

### 13.2.3. R-GMA Client deployment module

#### 13.2.3.1. Service Overview

The R-GMA Client module is a set of client API in C, C++, Java and Python to access the information and monitoring functionality of the R-GMA system. The client is automatically installed as part of the User Interface and Worker Node.

#### 13.2.3.2. Installation Pre-requisites

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 13.2.3.3. Security Settings

Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` (gLite Security Utilities) is installed and configured automatically when installing and configuring the R-GMA Client (refer to Chapter 5 for more information about the Security Utilities module). The module contains the latest version of the CA certificates

plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl*, *glite-mkgridmap* and *mkgridmap.py* scripts and sets up cron jobs that periodically check for updated revocation lists and grid-mapfile entries if required).

#### 13.2.3.4. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA client java API. This release requires v. 1.4.2 (revision 04 or greater). The JDK/JRE version to be used is a parameter in the configuration file. Please change it according to your version and location.

Due to license reasons, we cannot redistribute Java. Please download it from <http://java.sun.com/> and install it if you have not yet installed it.

#### 13.2.3.5. R-GMA Client Installation

If you install the client as part of another deployment module (e.g. the UI), the R-GMA client is installed automatically and you can continue with the configuration description in the next section. Otherwise, the installation steps are:

1. Download the latest version of the R-GMA Client installation script *glite-rgma-client\_installer.sh* from the gLite web site. It is recommended to download the script in a clean directory
2. Make the script executable

```
chmod u+x glite-rgma-client_install.sh
```

and execute it or execute it with

```
sh glite-rgma-client_install.sh
```

3. All the required RPMS are downloaded from the gLite software repository in the directory *glite-rgma-client* next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

<i>gLite</i>	in /opt/glite (\$GLITE_LOCATION)
<i>gLite-essentials- java</i>	in \$GLITE_LOCATION/externals/share
<i>gLite-essentials- cpp</i>	in \$GLITE_LOCATION/externals/
<i>swig-runtime</i>	in \$GLITE_LOCATION/externals/

5. The gLite R-GMA configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/glite-rgma-client-config.py.
```

All the necessary template configuration files are installed into

```
$GLITE_LOCATION/etc/config/templates/
```

The next section will guide you through the different files and necessary steps for the configuration.

#### 13.2.3.6. R-GMA Client Configuration

1. Copy the global configuration file template

```
$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml
```

to

```
$GLITE_LOCATION/etc/config,
```

open it and modify the parameters if required (Table 1).

- Copy the R-GMA common configuration file template

`$GLITE_LOCATION/etc/config/templates/glite-rgma-common.cfg.xml`

to

`$GLITE_LOCATION/etc/config/glite-rgma-common.cfg.xml`

open it and modify the parameters values as necessary (see Table 12 for details). Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*.

- Copy the R-GMA client configuration file template from

`$GLITE_LOCATION/etc/config/templates/glite-rgma-client.cfg.xml`

to

`$GLITE_LOCATION/etc/config/glite-rgma-client.cfg.xml`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 14 shows a list of parameters that can be set.

Parameter	Default value	Description
<b>User-defined Parameters</b>		
<b>Advanced Parameters</b>		
glite.installer.verbose	True	Enable verbose output
<b>System Parameters</b>		

**Table 14: R-GMA Client Configuration Parameters**

If you use the R-GMA client as a sub-deployment module that is downloaded and used by another deployment module, the configuration script is run automatically by the configuration script of the other deployment module and you can skip the following steps. Otherwise:

- Run the R-GMA Client configuration file

`$GLITE_LOCATION/etc/config/scripts/glite-rgma-client-config.py`

- The R-GMA Client is ready.

### 13.2.3.7. R-GMA Client Configuration Walkthrough

The R-GMA Client configuration script performs the following steps:

- Set the following environment variables if not already set using the values set in the global and CE configuration files:

GLITE\_LOCATION                      [=/opt/glite if not set anywhere]

- Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.csf.xml`:

GLITE\_LOCATION\_VAR                  [default is /var/glite]

GLITE\_LOCATION\_LOG                  [default is /var/log/glite]

GLITE\_LOCATION\_TMP                  [default is /tmp/glite]

JAVA\_HOME                              [=/usr/java/j2sdk1.4.2\_04]



3. Load the R-GMA Client configuration file  
`$GLITE_LOCATION/etc/config/glite-rgma-client.cfg.xml`  
and the R-GMA common configuration file  
`$GLITE_LOCATION/etc/config/glite-rgma-common.cfg.xml`  
and checks the configuration values.
4. Checks the `$GLITE_LOCATION` directory structure
5. Checks the Java installation
6. Checks the configuration values
7. Configures the gLite Security Utilities module
8. Runs the R-GMA Client internal set up script:  

```
python $GLITE_LOCATION/share/rgma/scripts/rgma-setup.py --server <server_hostname>  
--schema <schema_hostname > --registry < registry_hostname >
```
9. Configures the R-GMA client security by updating the  
`$GLITE_LOCATION/etc/rgm/ClientAuthentication.props`  
file.

#### 13.2.4. R-GMA servicetool deployment module

##### 13.2.4.1. Service overview

The R-GMA servicetool is an R-GMA client tool to publish information about the services it knows about and their current status. The tool is divided into three parts:

A daemon monitors regularly configuration files containing information about the services a site has installed. At regular intervals, this information is published to the ServiceTable. Each service specifies a script that needs to be run to obtain status information. The scripts are run by the daemon at the specified frequency and the results are inserted into the ServiceStatus table.

The second part of the tool is a command line program that modifies the configuration files to add delete and modify services. It does not communicate with the daemon directly but the next time the daemon scans the configuration file the changes will be published.

The third part of the tool is a command line program to query the service tables for status information.

This service is normally installed automatically with other modules and doesn't need to be installed independently.

##### 13.2.4.2. Installation Pre-requisites

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

##### 13.2.4.3. Security Settings

Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` (gLite Security Utilities) is installed and configured automatically when

installing and configuring the R-GMA Servicetool (refer to Chapter 5 for more information about the Security Utilities module). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl*, *glite-mkgridmap* and *mkgridmap.py* scripts and sets up cron jobs that periodically check for updated revocation lists and grid-mapfile entries if required).

#### 13.2.4.4. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA servicetool. This release requires v. 1.4.2 (revision 04 or greater). The JDK/JRE version to be used is a parameter in the configuration file. Please change it according to your version and location.

Due to license reasons, we cannot redistribute Java. Please download it from <http://java.sun.com/> and install it if you have not yet installed it.

#### 13.2.4.5. R-GMA servicetool installation

If you install the servicetool as part of another deployment module (e.g. the Computing element), the R-GMA servicetool is installed automatically and you can continue with the configuration description in the next section. Otherwise, the installation steps are:

1. Download the latest version of the R-GMA Servicetool installation script *glite-rgma-servicetool\_installer.sh* from the gLite web site. It is recommended to download the script in a clean directory.
2. Make the script executable

```
chmod u+x glite-rgma-servicetool_installer.sh
```

and execute it or execute it with

```
sh glite-rgma-sevicetool_installer.sh
```

3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory *glite-rgma-servicetool* next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

```
gLite                in /opt/glite ($GLITE_LOCATION)
gLite-essentials java in $GLITE_LOCATION/externals/share
```

5. The gLite R-GMA servicetool configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/glite-rgma-servicetool-config.py.
```

All the necessary template configuration files are installed into

```
$GLITE_LOCATION/etc/config/templates/
```

The next section will guide you through the different files and necessary steps for the configuration.

#### 13.2.4.6. R-GMA Servicetool Configuration

1. Copy the global configuration file template

```
$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml
```

to

`$GLITE_LOCATION/etc/config`

open it and modify the parameters if required (Table 1).

- Copy the R-GMA common configuration file template from

`$GLITE_LOCATION/etc/config/templates/glite-rgma-common.cfg.xml`

to

`$GLITE_LOCATION/etc/config/glite-rgma-common.cfg.xml`

open it and modify the parameters values as necessary (see Table 12 for details).

Some parameters have default values; others must be changed by the user. All

parameters that must be changed have a token value of *changeme*.

- Copy the R-GMA servicetool configuration file template

`$GLITE_LOCATION/etc/config/templates/glite-rgma-servicetool.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 15 shows a list of the parameters that can be set.

Parameter	Default value	Description
<b>User-defined Parameters</b>		
<code>rgma.servicetool.sitename</code>		DNS name of the site for the published services (in general the hostname). <i>Example: lxb2029.cern.ch</i>
<b>Advanced Parameters</b>		
<code>glite.installer.verbose</code>	True	Enable verbose output. <i>Example : true</i>
<b>System Parameters</b>		

**Table 15: R-GMA servicetool configuration parameters**

- Next you have to configure the information for each service that you want to be published by the R-GMA servicetool. The configuration parameters for the services are inside the configuration file of the service as separate “instance” parameter lists. In order to configure the service and the servicetool you have to modify for each of these ‘instance parameter list’ the parameters. Table 16 shows the list of parameters for each service that you have to set accordingly.
- You do not need to run the configuration script as this is done automatically by the configuration script of the deployment module that contains the corresponding services.

Parameter	Default value	Description
<b>User-defined Parameters</b>		
rgma.servicetool.enable	true	Publish this service via the R-GMA servicetool. If this variable set to false the other values below are not taken into account. <i>Example: true</i>
rgma.servicetool.name		Name of the service. This should be globally unique. <i>Example: host.name.service_name</i>
rgma.servicetool.url_endpoint		URL to contact the service at. This should be unique for each service. <i>Example:</i> <a href="http://your.host.name:port/serviceGroup/ServiceName">http://your.host.name:port/serviceGroup/ServiceName</a>
rgma.servicetool.service_type		The service type. This should be uniquely defined for each service type. Currently two methods of type naming are recommended: <ul style="list-style-type: none"> <li>• The targetNamespace from the WSDL document followed by a space and then the service name</li> <li>• A URL owned by the body or individual who defines the service type</li> </ul> <i>Example: Name of service type</i>
rgma.servicetool.service_version		Service version in the form 'major.minor.patch' <i>Example: 1.2.3</i>
rgma.servicetool.publish_interval		How often to publish the service details (like endpoint, version etc). (Unit: seconds) <i>Example: 600</i>
rgma.servicetool.status_script		Script to run to determine the service status. This script should return an exit code of 0 to indicate the service is OK, other values should indicate an error. The first line of the standard output should be a brief message describing the service status (e.g. 'Accepting connections') <i>Example:</i> <a href="#">/opt/glite/bin/myService/serviceStatus</a>
rgma.servicetool.status_interval		How often check and publish service status. (Unit: seconds) <i>Example: 60</i>
rgma.servicetool.url_wSDL		URL of a WSDL document for the service (leave blank if the service has no WSDL). <i>Example:</i> <a href="http://your.host.name/service/wSDL.ht">http://your.host.name/service/wSDL.ht</a>

		<a href="#"><i>ml</i></a>
rgma.servicetool. url_semantics		URL of a document containing a detailed description of the service and how it should be used. <i>Example:</i> <i>http://your.host.name/service/semantics.html</i>
<b>Advanced Parameters</b>		
<b>System Parameters</b>		

**Table 16: R-GMA servicetool configuration parameters for a service to be published via the R-GMA servicetool**

### 13.2.4.7. R-GMA servicetool configuration walkthrough

The R-GMA configuration script performs the following steps:

1. Set the following environment variables if not already set using the values set in the global and R-GMA configuration files:

```

GLITE_LOCATION      [=/opt/glite if not set anywhere]
JAVA_HOME           to the location specified in the global configuration file
  
```

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.csf.xml`:

```

GLITE_LOCATION_VAR  [default is /var/glite]
GLITE_LOCATION_LOG  [default is /var/log/glite]
GLITE_LOCATION_TMP  [default is /tmp/glite]
  
```

3. Checks the directory structure of `$GLITE_LOCATION`.
4. Configures the R-GMA servicetool by creating the servicetool configuration file at `$GLITE_LOCATION/etc/rgma-servicetool/rgma-servicetool.conf` that specifies the sitename
5. Takes the set of parameters for the R-GMA servicetool from each instance in the service configuration files and for each of these instances, a configuration file at `$GLITE_LOCATION/etc/rgma-servicetool/services/XXXX.service` is created, where XXXX is the name of the service.
6. Starts the R-GMA servicetool via `/etc/init.d/rgma-servicetool start`

## 13.2.5. R-GMA GadgetIN (GIN) deployment module

### 13.2.5.1. Service Overview

The R-GMA GadgetIN (GIN) is an R-GMA client to extract information from MDS and to republish it to R-GMA. The R-GMA GadgetIN is installed and used by the Computing Element (CE) to publish its information and does not need to be installed independently.

### 13.2.5.2. Installation Pre-requisites

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

### 13.2.5.3. Security Settings

Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called *glite-security-utils* (gLite Security Utilities) is installed and configured automatically when installing and configuring the R-GMA Servicetool (refer to Chapter 5 for more information about the Security Utilities module). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl*, *glite-mkgridmap* and *mkgridmap.py* scripts and sets up cron jobs that periodically check for updated revocation lists and grid-mapfile entries if required).

### 13.2.5.4. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA GadgetIN. This release requires v. 1.4.2 (revision 04 or greater). The JDK/JRE version to be used is a parameter in the configuration file. Please change it according to your version and location.

Due to license reasons, we cannot redistribute Java. Please download it from <http://java.sun.com/> and install it if you have not yet installed it.

### 13.2.5.5. R-GMA GadgetIN installation

If you install the R-GMA GadgetIN as part of another deployment module (e.g. the Computing element), the R-GMA GadgetIN is installed automatically and you can continue with the configuration description in the next section. Otherwise, the installation steps are:

1. Download the latest version of the R-GMA GadgetIN installation script *glite-rgma-gin\_installer.sh* from the gLite web site. It is recommended to download the script in a clean directory.
2. Make the script executable
 

```
chmod u+x glite-rgma-gin_installer.sh
```

 and execute it or execute it with
 

```
sh glite-rgma-gin_installer.sh
```
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory `glite-rgma-gin` next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:
 

gLite	in <code>/opt/glite (\$GLITE_LOCATION)</code>
gLite-essentials java	in <code>\$GLITE_LOCATION/externals/share</code>
5. The gLite R-GMA gin configuration script is installed in
 

```
$GLITE_LOCATION/etc/config/scripts/glite-rgma-gin-config.py.
```

 All the necessary template configuration files are installed into

`$GLITE_LOCATION/etc/config/templates/`

The next section will guide you through the different files and necessary steps for the configuration.

### 13.2.5.6. R-GMA GadgetIN Configuration

1. Copy the global configuration file template

`$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml`

to

`$GLITE_LOCATION/etc/config,`

open it and modify the parameters if required (Table 1).

2. Copy the R-GMA common configuration file template from

`$GLITE_LOCATION/etc/config/templates/glite-rgma-common.cfg.xml`

to

`$GLITE_LOCATION/etc/config/glite-rgma-common.cfg.xml`

open it and modify the parameters values as necessary (see Table 12 for details).

Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*.

3. Copy the GadgetIN configuration file template

`$GLITE_LOCATION/etc/config/templates/glite-rgma-gin.cfg.xml`

to

`$GLITE_LOCATION/etc/config`

and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 17 shows a list of parameters that need to be set.

If you use the R-GMA GadgetIN as a sub-deployment module that is downloaded and used by another deployment module (e.g. the CE), the configuration script is run automatically by the configuration script of the other deployment module and you can skip step 3. Otherwise:

4. Run the R-GMA GadgetIN configuration file

`$GLITE_LOCATION/etc/config/scripts/glite-rgma-client-config.py`

5. The R-GMA GadgetIN is ready.

Parameter	Default value	Description
<b>User-defined Parameters</b>		
<code>rgma.gin.run_generic_info_provider</code>		Run generic information provider (gip) backend (yes no). Within LCG this comes with the ce and se <i>Example: no</i>
<code>rgma.gin.run_fmon_provider</code>		Run fmon backend (yes no). This is used by LCG for gridice. <i>Example: no</i>

rgma.gin.run_ce_provider		Run ce backend (yes no).
<b>Advanced Parameters</b>		
glite.installer.verbose	True	Enable verbose output. <i>Example : true</i>
<b>System Parameters</b>		

**Table 17: R-GMA GadgetIN configuration parameters**

### 13.2.5.7. R-GMA GadgetIN Configuration Walkthrough

The R-GMA GadgetIN configuration script performs the following steps:

1. Set the following environment variables if not already set using the values set in the global and R-GMA configuration files:

```

GLITE_LOCATION           [=/opt/glite if not set anywhere]
JAVA_HOME                to the location specified in the global
configuration file

```

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.csf.xml`:

```

GLITE_LOCATION_VAR       [default is /var/glite]
GLITE_LOCATION_LOG       [default is /var/log/glite]
GLITE_LOCATION_TMP       [default is /tmp/glite]

```

3. Configures the gLite Security Utilities module
4. Checks the directory structure of `$GLITE_LOCATION`.
5. Verify the Java installation.
6. Configures the general R-GMA setup by running the R-GMA setup script `$GLITE_LOCATION/share/rgma/scripts/rgma-setup.py` using the configuration values from the configuration file for server, schema and registry hostname.
7. Configures the R-GMA client security by updating the `$GLITE_LOCATION/etc/rgm/ClientAuthentication.props` file.
8. Configures the R-GMA GadgetIN via `$GLITE_LOCATION/bin/rgma-gin-config --gip=<yes|no> --fmon=<yes|no> --glite-ce=<yes|no>` with the yes/no option depending on the configuration settings.



## 14. VOMS SERVER AND ADMINISTRATION TOOLS

### 14.1. SERVICE OVERVIEW

[To Be Added]

### 14.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 14.2.1. Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` can be installed by downloading and running from the gLite web site (<http://www.glite.org>) the script `glite-security-utils_installer.sh` (Chapter 5). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the `glite-fetch-crl` script and sets up a crontab that periodically check for updated revocation lists
2. Install the server host certificate `hostcert.pem` and key `hostkey.pem` in `/etc/grid-security`

#### 14.2.2. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA Server. This release requires v. 1.4.2 (revision 04 or greater). The JDK/JRE version to be used is a parameter in the configuration file. Please change it according to your version and location.

Due to license reasons, we cannot redistribute Java. Please download it from <http://java.sun.com/> and install it if you have not yet installed it.

### 14.3. VOMS SERVER INSTALLATION

1. Download from the gLite web site the latest version of the VOMS Server installation script `glite-voms-server_install.sh`. It is recommended to download the script in a clean directory
2. Make the script executable (`chmod u+x glite-voms-server_installer.sh`) and execute it
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory `glite-voms-server` next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

gLite	in /opt/glite
Tomcat	in /var/lib/tomcat5

5. The gLite VOMS Server and VOMS Administration configuration script is installed in `$GLITE_LOCATION/etc/config/scripts/glite-voms-server-config.py`. A template configuration file is installed in `$GLITE_LOCATION/etc/config/templates/glite-voms-server.cfg.xml`

#### 14.4. VOMS SERVER CONFIGURATION

1. Copy the global configuration file template `$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml` to `$GLITE_LOCATION/etc/config`, open it and modify the parameters if required (Table 1)
2. Copy the configuration file template from `$GLITE_LOCATION/etc/config/templates/glite-voms-server.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-voms-service.cfg.xml` and modify the parameters values as necessary (Table 18)
3. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. 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 `<instance/>` tag. A default instance is already defined and can be directly configured. Additional instances can be added by simply copying and pasting the `<instance/>` section, assigning a name and changing the parameters values as desired. The following parameters can be set:

Parameter	Default value	Description
<b>User-defined Parameters</b>		
voms.vo.name		Name of the VO associated with the VOMS instance
voms.port.number		Port number of the VOMS instance
voms.code		VOMS code. In the multi-instance scenario this parameter MUST have a unique value for each VOMS instance
vo.admin.dn		Certificate DN of the VO admin
vo.admin.ca		CA of the VO admin
vo.admin.cn		Common name of the VO admin
vo.admin.e-mail		E-mail address of the VO admin
vo.ca.URI		URI from where the CRIs are downloaded
voms.mysql.passwd		Password (in clear) of the root user of the MySQL server used for VOMS databases
voms.db.user		Name of the VOMS database owner. In multi-instance scenario this value CAN be redefined in each 'instance' section
voms.db.passwd		Password (clear) of the VOMS database owner. In multi-instance scenario this value CAN be redefined in the 'instance' section

<b>Advanced Parameters</b>		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
voms-admin.install	true	Install the VOMS Admin interface on this server
<b>System Parameters</b>		

**Table 18: VOMS Configuration Parameters**

4. As root run the VOMS Server configuration file  
`$GLITE_LOCATION/etc/config/scripts/glite-voms-server-config.py`
5. The VOMS Server is now ready.

#### 14.5. VOMS SERVER CONFIGURATION WALKTHROUGH

The VOMS Server configuration script performs the following steps:

1. Set the following environment variables if not already set using the values defined in the global and lb configuration files:

```
GLITE_LOCATION          [default is /opt/glite]
CATALINA_HOME           [default is /var/lib/tomcat5]
```

2. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.cfg.xml`:

```
GLITE_LOCATION_VAR
GLITE_LOCATION_LOG
GLITE_LOCATION_TMP
```

3. Load the VOMS Server configuration file `$GLITE_LOCATION/etc/config/glite-voms-server.cfg.xml`
4. Set the following additional environment variables needed internally by the services (this requirement should disappear in the future):

```
PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/externals/bin:$GLOBUS_LOCATION/bin:$PATH
LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib:$GLOBUS_LOCATION/lib:$LD_LIBRARY_PATH
```

## 15. USER INTERFACE

### 15.1. SERVICE OVERVIEW

The gLite user Interface is a suite of clients and APIs that users and applications can use to access the gLite services. The gLite User Interface includes the following components:

- Data Catalog command-line clients and APIs
- Data Transfer command-line clients and APIs
- gLite I/O Client and APIs
- R-GMA Client and APIs
- VOMS command-line tools
- Workload Management System clients and APIs
- Logging and bookkeeping clients and APIs

### 15.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

#### 15.2.1. Security Settings

1. <http://www.gridpma.org/>A security module called *glite-security-utils* is installed and configured automatically by <http://www.glite.org/> by the UI installer. The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs (for the root install) the *glite-fetch-crl* script and sets up a crontab that periodically check for updated revocation lists. In case of the non-privileged user installation the CRL update is left to the decision of the user and adding of it into the user's crontab is the manual step to do.

#### 15.2.2. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA Server. This release requires v. 1.4.2 (revision 04 or greater). The JDK/JRE version to be used is a parameter in the configuration file. Please change it according to your version and location.

Due to license reasons, we cannot redistribute Java. Please download it from <http://java.sun.com/> and install it if you have not yet installed it.

### 15.3. UI INSTALLATION

The gLite User Interface can be installed as root or as non-privileged user. The installation procedure is virtually identical. The root installation installs by default the UI RPMS in the standard location `/opt/glite` (the location of the gLite RPMS can be changed by means of the *prefix* command line switch).

The non-privileged user installation does not differ from the root one. The user installation is still based on the services provided by the rpm program (dependency checking, package removal and upgrade), but uses a copy of the system RPM database created in user space and used for the local user installation. This approach allows performing a non-privileged user installation and still keeping the advantages of using a package manager.

Location of the gLite UI installed by the non-privileged user is by default set to ``pwd`/glite_ui` (glite\_ui directory in the current working directory).

The destination directory of both root and user installations can be modified by using of the `--basedir <path>` option of the ui installer script, where the `<path>` MUST be an absolute path.

The installation steps are the same in both the root and no-root installation cases:

1. Download from the gLite web site the latest version of the UI installation script *glite-ui\_install.sh*. It is recommended to download the script in a clean directory
2. Make the script executable (*chmod u+x glite-ui\_installer.sh*) and execute it. If needed, pass the `--basedir <path>` option to specify the target installation directory.
3. Run the script as root or as normal user. All the required RPMS are downloaded from the gLite software repository in the directory *glite-ui* next to the installation script and the installation procedure is started. If some RPM is already installed, they are upgraded if necessary. Check the screen output for errors or warnings. This step can fail in case if some of the OS RPMs are missing. These RPMs MUST be installed manually from the OS distribution CD, or by apt/yum tools
4. If the installation is performed successfully, the following components are installed:
  - a) root installation:
    - gLite in /opt/glite (= GLITE\_LOCATION)
    - Globus in /opt/globus (= GLOBUS\_LOCATION)
    - GPT in /opt/gpt (= GPT\_LOCATION)
  - b) user installation:
 

gLite, Globus and GPT are installed in the tree from ``pwd`/glite_ui` by removing the /opt/[glite, globus, gpt] part. The GLITE\_LOCATION, GLOBUS\_LOCATION and GPT\_LOCATION variables are set to the ``pwd`/glite_ui` value.
5. The gLite UI configuration script is installed in `$GLITE_LOCATION/etc/config/scripts/glite-voms-server-config.py`. A template configuration file is installed in `$GLITE_LOCATION/etc/config/templates/glite-ui.cfg.xml`

#### 15.4. UI CONFIGURATION

1. Copy the global configuration file template `$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml` to `$GLITE_LOCATION/etc/config`, open it and modify the parameters if required (Table 1)
2. Copy the configuration file templates from `$GLITE_LOCATION/etc/config/templates/glite-ui.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-ui-service.cfg.xml`  
`$GLITE_LOCATION/etc/config/templates/glite-io-client.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-io-client.cfg.xml`  
`$GLITE_LOCATION/etc/config/templates/glite-rgma-client.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-rgma-client.cfg.xml`  
`$GLITE_LOCATION/etc/config/templates/glite-rgma-common.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-rgma-common.cfg.xml`  
`$GLITE_LOCATION/etc/config/templates/glite-security-utils.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-security-utils.cfg.xml`

and modify the parameter values as necessary (Table 19). For glite-io-client, glite-rgma-client, glite-rgma-common and glite-security-utils configuration files refer please the corresponding cahpters of this guide. Alternatively, a site configuration file can be used (refer to section 4.2.4 for more information)

- Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The configuration file contains one or more <set> sections, one per each VO that the UI must be configured for. It also contains a global <parameters> section  
The following parameters can be set:

For the <set/> section:

Parameter	Default value	Description
<b>User-defined Parameters</b>		
name		Name of set
ui.VirtualOrganisation		Name of the VO corresponding to this set
ui.NSAddresses		Array of the WMS Network Servers for a given VO
ui.LBAddresses		Array of Logging and Bookkeeping servers corresponding to each NS server
ui.MyProxyServer		MyProxy server to use
ui.voms.server		VOMS server name for given VO
ui.voms.port		"VOMS server port number
ui.voms.cert.subject		DN of the VOMS server's certificate
py-ui.requirements		Requirements for job matchmaking

Parameter	Default value	Description
<b>User-defined Parameters</b>		
py-ui.DefaultVo		Default VO to connect
<b>Advanced Parameters</b>		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	false	Switch on/off the checking of the existence of the host certificate files
py-ui.rank	- other.GlueCEStateEstimatedResponseTime	
py-ui.RetryCount	3	
py-ui.OutputStorage	"/tmp"	
py-ui.ListenerStorage	"/tmp"	
py-ui.LoggingTimeout	10	

py-ui.LoggingSyncTimeout	10	
py-ui.NSLoggerLevel	1	
py-ui.DefaultStatusLevel	1	
py-ui.DefaultLogInfoLevel	1	
<b>System Parameters</b>		

**Table 19: UI Configuration Parameters**

4. Run the UI configuration file `$GLITE_LOCATION/etc/config/scripts/glite-ui-config.py`
5. The gLite user Interface is now ready.

### 15.5. UI CONFIGURATION WALKTHROUGH

The UI configuration script performs the following steps:

1. To avoid the unnecessary editing of the global configuration file, the installer tries to modify the `GLITE_LOCATION*` `GLOBUS_LOCATION` variables to point to the currently installed UI instance. All modifications are reported in the configuration script output.
2. Set the following environment variables if not already set using the values defined in the global and ui configuration files:

```
GLITE_LOCATION          [default is /opt/glite or `pwd`/glite_ui]
GLOBUS_LOCATION        [default is /opt/globus or `pwd`/glite_ui]
```

3. Read the following environment variables if set in the environment or in the global gLite configuration file `$GLITE_LOCATION/etc/config/glite-global.cfg.xml`:

```
GLITE_LOCATION_VAR
GLITE_LOCATION_LOG
GLITE_LOCATION_TMP
```

4. Load the UI configuration file `$GLITE_LOCATION/etc/config/glite-ui.cfg.xml` or a site configuration file from the specified URL
5. Set the following additional environment variables:

```
PATH=$GLITE_LOCATION/bin:$GLITE_LOCATION/externals/bin:$GLOBUS_LOCATION/bin:$GPT_LOCATION/bin:$PATH
LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib:$GLOBUS_LOCATION/lib:$GPT_LOCATION/lib:$LD_LIBRARY_PATH
```

6. Saves the necessary configuration variables into the `/etc/glite/profile.d/` directory for the root installation and to the `$HOME/.glite` directory for the user installation. For the user installation the script also modifies the `.bashrc` and `.cshrc` files to automatically source these files during the login.

## 15.6. CONFIGURATION FOR THE UI USERS

To get the environment configured correctly, each gLite UI user MUST before the first using of the glite UI to run the `$GLITE_LOCATION/etc/config/scripts/glite-ui-config.py` configuration script. The value of the `GLITE_LOCATION` variable MUST be previously communicated by the administrator of the UI installation. In this case the script creates the copy of the `$GLITE_LOCATION/etc/vomses` file in the `$HOME/.vomses` file (required by the VOMS client) and sets up the automatic sourcing of the UI instance parameters.

## 15.7. NOTE

To assure the correct functionality of the gLite UI after the execution of the `glite-ui-config.py` script, it is necessary either:

- 1) to source the corresponding file with UI environment parameters in `/etc/glite/profile.d/` or `~<ui_manager>/.glite` directory
- 2) log off and log in back. The file with UI environment variables will be sourced automatically.



## 16. TORQUE

### 16.1. SERVICE OVERVIEW

TORQUE (Tera-scale Open-source Resource and QUEue manager) is a resource manager providing control over batch jobs and distributed compute nodes. It is a community effort based on the original *PBS* project and has incorporated significant advances in the areas of scalability and fault tolerance.

The torque system is composed by a *pbs\_server* which provides the basic batch services such as receiving/creating a batch job or protecting the job against system crashes. The *pbs\_mom* (second service) places the job into execution when it receives a copy of the job from a Server. The *mom\_server* creates a new session as identical to a user login session as if possible. It also has the responsibility for returning the job's output to the user when directed to do so by the *pbs\_server*. The job *scheduler*, is another daemon which contains the site's policy controlling which job is run and where and when it is run. The scheduler appears as a batch Manager to the server. The scheduler being used by the torque module is *maui*.

#### 16.1.1. TORQUE Server Overview

This deployment module contains and configures the *pbs\_server* (server configuration, queues creation, etc ...) and *maui* services. It is also responsible for registering both services into RGMA via the *servicetool* deployment module.

The *sshd* configuration required for the torque clients to copy their output back to the torque server is also carried out in this module.

A Torque Server (the Computing Element node) could easily work as a Torque Client (the Worker Node) by including and configuring the *pbs\_mom* service. By design the Torque Server deployment module does not include the RPMS and configuration necessary to make it work as a Torque Client. The only additional task to make a Torque Server be also a Torque Client is the installation and configuration of the Torque Client deployment module.

#### 16.1.2. TORQUE Client Overview

This deployment module configures the *pbs\_mom* service aimed at being installed in the worker nodes. It's also responsible for the *ssh* configuration to allow copying the job output back to the Torque Server (Computing Element).

### 16.2. INSTALLATION PRE-REQUISITES

These installation instructions are based on the RPMS distribution of gLite. It is also assumed that the target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux or CentOS. Whenever a package needed by gLite is not distributed as part of gLite itself, it is assumed it can be found in the list of RPMS of the original OS distribution.

### 16.3. TORQUE SERVER

#### 16.3.1. TORQUE Server Installation

1. Download from the gLite web site the latest version of the Torque Server installation script *glite-torque-server\_installer.sh*. It is recommended to download the script in a clean directory
2. Make the script executable (*chmod u+x glite-torque-server\_install.sh*).

3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory `glite-torque-server` next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

```

gLite  in /opt/glite ($GLITE_LOCATION)
torque in /var/spool/pbs
  
```

5. The gLite torque-server configuration script is installed in `$GLITE_LOCATION/etc/config/scripts/glite-torque-server-config.py`. A template configuration file is installed in `$GLITE_LOCATION/etc/config/templates/glite-torque-server.cfg.xml`
6. The gLite torque-server installs the R-GMA servicetool to publish its information to the information system R-GMA. The details of the installation of the R-GMA servicetool are described in section 13.2.4.5.

### 16.3.2. TORQUE Server Service Configuration

Copy the global configuration file template

```
$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml
```

to

```
$GLITE_LOCATION/etc/config
```

open it and modify the parameters if required (see Table 20)

Copy the configuration file template from

```
$GLITE_LOCATION/etc/config/templates/glite-torque-server.cfg.xml
```

to

```
$GLITE_LOCATION/etc/config/glite-torque-server.cfg.xml
```

and modify the parameters values as necessary. Some parameters have default values, others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The parameters that can be set can be found in Table 20. The R-GMA servicetool related parameters can be found in Table 15.

The parameters in the file can be divided into two categories:

- a. *Common parameters* (first part of Table 20)

These are the configuration parameters that are independent of the worker node and ques instances. Change all *changeme* values to the corresponding values.

- b. *Torque client / Worker node specific values* (second part of Table 20)

For every torque client (Worker Node) to be configured in the Torque Server the configuration file contains the list of parameters grouped by the tag

```

<instance name="changeme" service="wn-torque">
....
</instance>
  
```

At least one worker node instance must be defined. If you want to use multiple clients, create a separate instance for each client by copying/pasting the <instance> section in this file.

Next, change the name of each client instance from 'changeme' to the client name and adapt the parameters of each instance accordingly.

c. *Queues* (third part of Table 20)

For every queue to be created in the Torque Server the configuration file contains the list of parameters grouped by the tag

```
<instance name="xxxx " service="pbs-queue">
...
</instance>
```

where xxxx is the name of the queue. Adapt the parameters of each instance accordingly. If you want to configure more queues please add a separate instance by copying/pasting the <instance> section in this file for each queue.

By default, the configuration file defines three queues (short, long and infinite) with different values and with acl\_groups disabled. It's up to the users to customize their queues depending on their requirements.

<i>Common parameters</i>		
Parameter	Default value	Description
<b>User-defined Parameters</b>		
torque-server.name		Name of the machine where the job server is running, it usually corresponds to the Computing Element: Example: <code>\${HOSTNAME}</code> .
torque-server.force	False	This parameter specifies the behaviour of the pbs_server setting parameters and queue creation. In case it is True it will take the whole control of the queue creation/deletion. That means that if it's specified a queue in the config file and latter removed from the configuration file it will also be removed in the pbs_server configuration, on the contrary, no queue removal will be performed.
se.name		Storage Element name (if necessary).

<b>Advanced Parameters</b>		
glite.installer.verbose	True	Enable verbose output.
torque-server.scheduling	True	When the attribute scheduling is set to true, the server will call the job scheduler, if false the job scheduler is not called. The value of scheduling may be specified on the pbs_server command line with the -a option.
torque-server.acl-host.enable	False	Enables the server host access control list. Values True,False.
torque-server.default.queue	short	The queue which is the target queue when a request does not specify a queue name, must be set to an existing queue.
torque-server.log.events	511	A bit string which specifies the type of events which are logged, Default value 511 (all events).
torque-server.query.other-jobs	True	The setting of this attribute controls if general suers, other than job owner, are allowd to query the status of or select the job.
torque-server.scheduler.interaction		The time, in seconds, between iterations of attempts by the batch server to schedule jobs.On each iteration, the server examines the available resources and runnable jobs to see if a job can be initiated.This examination also occurs whenever a running batch job terminates or a new job is placed in the queued state in an execution queue.
torque-server.default.node	glite	A node specification to use if there is no other supplied. specification. This attribute is only used by servers where a nodes file exist in the server_priv directory providing a list of nodes to the server. If the nodes file does does a single node.
torque-server.node.pack	False	Controls how multiple processor nodes are allocated to jobs. If this attribute is set to true, jobs

		will be assigned to the multiple processor nodes with the fewest free processors. This packs jobs into the fewest possible nodes leaving multiple processor nodes free for jobs which need many processors on a node. If set to false, jobs will be scattered across nodes reducing conflicts over memory between jobs. If unset, the jobs are packed on nodes in the order that the nodes are declared to the server (in the nodes file) nodes reducing conflicts over memory between jobs.
maui.server.port	40559	Port on which the Maui server will listen for client connections, by default 40559.
maui.server.mode	NORMAL	Specifies how Maui interacts with the outside world. Possible values NORMAL, TEST AND SIMULATION.
maui.defer.time	00:01:00	Specifies amount of time a job will be held in the deferred state before being released back to the Idle job queue. Format [[DD:]HH:]MM:]SS
maui.rm.poll.interval	00:00:10	Maui will refresh its resource manager information every 10 seconds. This parameter specifies the global poll interval for all resource managers.
maui.log.filename	`\${GLITE_LOG_CATALOG}/maui.log	Name of the maui log file
maui.log.max.size	10000000	Maximum allowed size (in bytes) the log file before it will be rolled.
maui.log.level	1	Specifies the verbosity of Maui logging where 9 is the most verbose (NOTE: each logging level is approximately an order of magnitude more verbose than the previous level. Values [0..9])
<b>System Parameters</b>		

<i>Worker node instances</i>		
Torque-wn.name		Worker Node name to be used by the torque server. It can also be the CE itself. Example: lxb1426.cern.ch. [Type: string].
torque-wn.number.processors		Number of processors of the machine. Example: 1,2 , .... [Type: string].
torque-wn.attribute		Attribute that can be used by the server for different purposes (for example to establish a default node. [Type: string].

<i>Queue instances</i>		
queue.name		Queue name
queue.type		Must be set to either Execution or Routing. If a queue is from routing type the jobs will be routed to another server (route_destinations attributed).
queue.resources.max.cpu.time		Maximum amount of CPU time used by all processes in the job. Format: seconds, or [[HH:]MM:]SS.
queue.max.wall.time		Maximum amount of real time during which the job can be in the running state. Format: seconds, or [[HH:]MM:]SS.
queue.enabled		Defines if the queue will or will not accept new jobs. When false the queue is disabled and will not accept jobs.
queue.started		It set to true, jobs in the queue will be processed, either routed by the server if the queue is a routing queue or scheduled by the job scheduler if an execution queue. When False, the queue is considered stopped.
queue.acl.group.enable		Attribute which when true directs the server to use the

		queue group access control list acl_groups.
queue.acl.groups		List which allows or denies enqueueing of jobs owned by members of the listed groups. The groups in the list are groups on the server host, not submitting hosts. Syntax: [+]-]group_name[,...].Example: +test authorizes the test group users to submit jobs to this queue.

**Table 20: TORQUE Server configuration parameters**

1. Configure the R-GMA servicetool. For this you have to configure the servicetool itself as well as configure the sub-services of Torque server for the publishing via the R-GMA servicetool:
2. *R-GMA servicetool configuration:*  
Copy the R-GMA servicetool configuration file template `$GLITE_LOCATION/etc/config/templates/glite-rgma-servicetool.cfg.xml` to `$GLITE_LOCATION/etc/config` and modify the parameters values as necessary. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme*. Table 15 shows a list of the parameters that can be set. More details can be found in section 13.2.4.6.
3. *Service Configuration for the R-GMA servicetool:*  
Modify the R-GMA servicetool related configuration values that are located in the `glite-torque-server.cfg.xml` file that was mentioned before. In this file, you will find for each service that should be published via the R-GMA servicetool one instance of a set of parameters that are grouped by the tag `<instance name="xxxx" service="rgma-servicetool">` Where `xxxx` is the name of corresponding subservice. Table 16 on page 85 in the section 13.2.4 about the R-GMA servicetool shows the general list of parameters for each service for the publishing via the R-GMA servicetool. For Torque-server the following sub-services are published via the R-GMA servicetool and need to be updated accordingly:
  - ii. Torque PBS server
  - iii. Torque maui
 Again, you find the necessary steps described in section 13.2.4.6.

*Note: Step 1,2 and 3 can also be performed by means of the remote site configuration file or a combination of local and remote configuration files*

4. As root run the Torque Server Configuration file `/opt/gLite/etc/config/scripts/gLite-torque-server-config.py`.

Once reached this point the Torque Server Service is ready and the Torque Clients have to be properly installed and configured.

### 16.3.3. TORQUE Server Configuration Walkthrough

The Torque Server configuration script performs the following steps:

5. Load the Torque Server configuration file `$GLITE_LOCATION/etc/config/gLite-torque-server.cfg.xml`
6. Add the torque and maui ports to `/etc/services`.
7. Create the `/var/spool/pbs/server_name` file containing the torque server hostname.
8. Create the list with the torque clients under `/var/spool/pbs/server_priv/nodes`.
9. Create the `pbs_server` configuration.
10. Start the `pbs_server`.
11. Look for changes in the `pbs_server` configuration since the last time the Torque Server was configured.
12. Establish the server configuration performing the necessary updates.
13. Create the queues configuration. It will check if any new queue has been defined in the configuration file, if any queue has been removed and depending on the value of the value **torque-server.force** it will behave in a different way (see `torque-server.force` parameter description).
14. Execute the defined queues configuration
15. Create the `/opt/edg/etc/edg-pbs-shostsequiv.conf` file used by the script `edg-pbs-shostsequiv`. This file includes the list of nodes that will included in the `/etc/ssh/shosts` file to allow HostbasedAuthentication.
16. Create the `edg-pbs-shostsequiv` script. This file contains a crontab entry to call periodically the `/opt/edg/sbin/edg-pbs-shostsequiv` script. This file is then added to the `/etc/cron.d/` directory.
17. Run the `/opt/edg/sbin/edg-pbs-shostsequiv` script.
18. Look for duplicated key entries in `/etc/ssh/ssh_known_hosts`.
19. Create the configuration file `/opt/edg/etc/edg-pbs-knownhosts.conf`. This file contains the nodes which keys will be added to the `/etc/ssh/ssh_known_hosts` file apart from the torque client nodes (which are taken directly from the torque server via the `pbsnodes -a` command).
20. Create the `edg-pbs-knownhosts` script. This script contains a crontab entry to call periodically the `/opt/edg/sbin/edg-pbs-knownhosts` script. This file is then added to the `/etc/cron.d/` directory.
21. Run `/opt/edg/sbin/edg-pbs-knownhosts` to add the keys to `/etc/ssh/ssh_known_hosts`.



22. Create the required sshd configuration (modifying the `/etc/ssh/sshd_config` file) to allow the torque clients (Worker Nodes) copying their output directly to the Torque Server via HostBasedAuthentication.
23. Restart the sshd daemon to take the changes into account.
24. Restart the pbs\_server.
25. Create the maui configuration file in `/var/spool/maui/maui.cfg`.
26. Start the maui service.
27. Configure the servicetool to register the torque services defined in the configuration file.

#### 16.3.4. Managing the TORQUE Server Service

The TORQUE SERVER configuration script can be run with the following command-line parameters to manage the services:

<code>glite-torque-server-config.py --start</code>	Starts all TORQUE SERVER services or restart them if they are already running (pbs_server, maui and servicetool)
<code>glite-torque-server-config.py --stop</code>	Stops all TORQUE SERVER services (pbs_server, maui and servicetool)

### 16.4. TORQUE CLIENT

#### 16.4.1. TORQUE Client Installation

1. Download from the gLite web site the latest version of the torque-server installation script `glite-torque-client_installer.sh`. It is recommended to download the script in a clean directory.
2. Make the script executable (`chmod u+x glite-torque-client_install.sh`).
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory `glite-torque-client` next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. If the installation is performed successfully, the following components are installed:

```
gLite           in /opt/glite ($GLITE_LOCATION)
Torque client   in /var/spool/pbs
```

5. The gLite torque-client configuration script is installed in `$GLITE_LOCATION/etc/config/scripts/glite-torque-client-config.py`. A template configuration file is installed in `$GLITE_LOCATION/etc/config/templates/glite-torque-client.cfg.xml`.

#### 16.4.2. TORQUE Client Configuration

1. Copy the global configuration file template `$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml` to

`$GLITE_LOCATION/etc/config`, open it and modify the parameters if required (see Table)

- Copy the configuration file template from `$GLITE_LOCATION/etc/config/templates/glite-client-server.cfg.xml` to `$GLITE_LOCATION/etc/config/glite-torque-client.cfg.xml` and modify the parameters values as necessary. Some parameters have default values, others must be changed by the user. All parameters that must be changed have a token value of *changeme*. The following parameters can be set:

**Note:** Step 1 and 2 can also be performed by means of the remote site configuration file or a combination of local and remote configuration files

Parameter	Default value	Description
<b>User-defined Parameters</b>		
torque-server.name		Name of the machine where the job server is running, it usually corresponds to the Computing Element: Example: <code>\${HOSTNAME}</code> .
se.name		Storage Element name (if necessary).
<b>Advanced Parameters</b>		
Glite.installer.verbose	True	Enable verbose output.
mom-server.logevent	255	Sets the mask that determines which event types are logged by pbs_mom.
<b>System Parameters</b>		

- As root run the Torque Client Configuration file `/opt/glite/etc/config/scripts/glite-torque-client-config.py`.

### 16.4.3. TORQUE Client Configuration Walkthrough

The Torque Client configuration script performs the following steps:

- Load the Torque Client configuration file `$GLITE_LOCATION/etc/config/glite-torque-client.cfg.xml`
- Create the `/var/spool/pbs/server_name` file containing the torque server hostname.
- Add the torque and maui ports to `/etc/services`.
- Create the required ssh configuration (modifying the `/etc/ssh/ssh_config` file) to allow the torque client (Worker Nodes) used HostbasedAuthentication in order to copy its output back to the Torque Server.
- Look for duplicated key entries in `/etc/ssh/ssh_known_hosts`.
- Create the configuration file `/opt/edg/etc/edg-pbs-knownhosts.conf`. This file contains the nodes which keys will be added to the `/etc/ssh/ssh_known_hosts` file apart from the torque client nodes (which are taken directly from the torque server via the `pbsnodes -a` command).

7. Create the `edg-pbs-knownhosts` script. This script contains a crontab entry to call periodically the `/opt/edg/sbin/edg-pbs-knownhosts` script. This file is then added to the `/etc/cron.d/` directory.
8. Create the `pbs_mom` configuration file under `/var/spool/pbs/mom_priv/config`.
9. Start the `pbs_mom` service.

#### 16.4.4. Managing the TORQUE Client

The TORQUE CLIENT configuration script can be run with the following command-line parameters to manage the services:

<code>glite-torque-client-config.py</code> <code>-start</code>	Starts all TORQUE CLIENT services (or restart them if they are already running, <code>pbs_mom</code> )
<code>glite-torque-client-config.py</code> <code>-stop</code>	Stops all TORQUE CLIENT services ( <code>pbs_mom</code> )

## 17. THE GLITE FUNCTIONAL TEST SUITES

### 17.1. OVERVIEW

There are three test suites described in this section, gLite I/O, Catalog and WMS.

The I/O test suite covers basic gLite I/O functionality (open file, create a file, read a file, write to a file, get info associated with a handle, close a file), some regression tests and cycles of glite-put and glite-get of several files.

The Catalogs test suite covers the creation and removal of directories, list entries in a directory, and the creation of entries in a directory through single and bulk operations.

The WMS test suite contains 9 tests:

- **1- DNS:** Check that a reverse DNS lookup works for all of the supplied host names.
- **2 and 3 - US\_script/ US\_jdl:** a Job Storm where the user can provide either the JDL file or the Script to send and run on the WNs (not both). In our case, each job runs a script to locate the BrokerInfo file.
- **4 - CEGate:** it is a Globus gatekeeper test, checking that globus-job-run, a globus-job-submit, and a globusrun with two-phase commit all work.
- **5 – 6 CECycle:** Checks one by one that the site specific CEs are operational
- **7 - RB\_val:** storm of different kind of jobs.
- **8 and 9 - CalStormR3, CalStormR0:** Job storm where each job sleeps for a given period of time (5 minutes). 10 streams of 100 jobs each are submitted in a parallel in a stream.
- **10 JS\_sleep :** This submits a simple job which echoes the [OK] string. It does this from multiple jobs running on WNs, i.e. a job storm.

### 17.2. I/O TEST SUITE

#### 17.2.1. Installation Pre-requisites

The gLite I/O test suite depends on the glite-data-io-client RPM, so it is recommended to install and execute the I/O tests from a UI machine. The I/O test suite depends on CppUnit too, that should also be installed in the machine. It can be downloaded from the gLite external dependencies web page.

#### 17.2.2. Installation

This test suite is installed using the glite-testsuites-data-io-server-1.0.5 rpm that can be obtained from the gLite web site using wget plus the URL of the rpm. The installation of the rpm will deploy the tests under \$GLITE\_LOCATION/test/glite-io-server directory.

#### 17.2.3. Configuration

Before running the test suite, check the following points:

- The user account that runs the tests must have these environment variables set:  
GLITE\_LOCATION (usually under /opt/glite)  
GLOBUS\_LOCATION (usually under /opt/globus)  
LD\_LIBRARY\_PATH (including: \$GLITE\_LOCATION/lib:\$GLOBUS\_LOCATION/lib)  
PATH (including: \$GLITE\_LOCATION/bin:\$GLOBUS\_LOCATION/bin)
- The user distinguish name that runs the tests must be included in the '/etc/grid-security/grid-mapfile' file of the gLite I/O server machine. This should be already the case if the configuration of your io-client is pointing to a valid io-server.
- Also, the user must have a voms-proxy before running the tests, typing: voms-proxy-init – voms your\_vo\_name

Note: if all the tests that you try to run fail, check if the problem is in the configuration of your io-client, io-server or catalog. If all is correctly configured, you should be able to put a file in a SE using the glite-put command.

#### 17.2.4. Execution

You can run the tests from the command line or using QMTest:

a) From the command line, you can execute 12 different binaries that are located at `$GLITE_LOCATION/test/glite-io-server/bin`, so you can run them executing: `$GLITE_LOCATION/test/glite-io-server/bin/gLite-io-****`

These tests check the basic I/O functionality: open a remote file, create a remote file, read a file, write to a file, set a file read/write pointer, get information about the file associated with the given handle and close a file. There are also 5 regression tests that check some of the bugs reported in Savannah.

Apart from those tests, you can also run a Perl test that is located at `$GLITE_LOCATION/test/glite-io-server/scripts/run_gliteIO_test.pl` to do cycles of glite-put and glite-get of several files. As an example, to do a glite-put and glite-get of 1000 files of a maximum size of 1MB in 1000 cycles (only one file per cycle), you should type: `$GLITE_LOCATION/test/glite-io-server/scripts/run_gliteIO_test.pl -l /tmp -c 1 -f 1M -n 1 -s 1000M -o your_vo_name`

Where `-l` specifies the log directory, `-c` the number of cycles to run, `-f` the maximal file size, `-n` the number of files to be transferred in a cycle, and `-s` the maximal total file size.

b) Using QMTest:

- Requirements:
  - If you don't have QMTest installed in your machine, you can download it from <http://www.codesourcery.com/qmtest/download.html>
  - Make sure that 'qmtest' is in your PATH.
- Set the environment variable:
 

```
QMTEST_CLASS_PATH=$GLITE_LOCATION/test/config/qmtest/test-classes/
```
- Start QMTest through the GUI typing:
 

```
qmtest -D $GLITE_LOCATION/test/config/qmtest gui --address "<your hostname>" --port <port number> --no-browser &
```
- Open a browser pointing to `http://<your hostname>:<port number>/test/dir` to access to the QMTest GUI. Once there, click on "glite-io", and then on "r\_1\_0\_0" to see all the tests. Select any test or a whole test suite and run it clicking on "Run this test" or "Run this Suite" from the pull-down menu. You may want to modify any of the parameters of each test. You can do this modifying the .qmt files that are located in the directory `$GLITE_LOCATION/test/config/qmtest/glite-io.qms/r_1_0_0.qms/`

#### 17.2.5. Test results

a) From the command line:

The test results can be visualized in stdout or in an XML file generated in the directory where the tests are called tests.xml

b) Using QMTest:

Using QMTest you can see the tests' output clicking on the right side of each test where it says "Details".

## 17.3. CATALOG TEST SUITE

### 17.3.1. Installation Pre-requisites

The gLite Catalog test suite depends on the `glite-data-catalog-interface` and `glite-data-catalog-fireman-api-c` RPMs, so it is recommended to install and execute the tests from a UI machine.

### 17.3.2. Installation

This test suite is installed using the `glite-testsuites-data-catalog-fireman-1.0.5-1` rpm that can be obtained from the gLite web site using `wget` plus the URL of the rpm. The installation of the rpm will deploy the tests under `$GLITE_LOCATION/test/glite-data-catalog-fireman` directory.

### 17.3.3. Configuration

Before running the test suite, check the following points:

- The user account that runs the tests must have these environment variables set:  
`GLITE_LOCATION` (usually under `/opt/glite`)  
`GLOBUS_LOCATION` (usually under `/opt/globus`)  
`LD_LIBRARY_PATH` (including: `$GLITE_LOCATION/lib:$GLOBUS_LOCATION/lib`)  
`PATH` (including: `$GLITE_LOCATION/bin:$GLOBUS_LOCATION/bin`)
- The user must have a `voms-proxy` before running the tests, typing: `voms-proxy-init -voms your_vo_name`

### 17.3.4. Execution

You can run the tests from the command line or using QMTest:

a) From the command line, you can execute the binaries that are located at `$GLITE_LOCATION/test/glite-data-catalog-fireman/bin`

The `gLite-fireman-create-test` creates a number of entries in the catalog in one single operation. This binary accepts the following parameters:

- `-e <the Catalog endpoint>`
- `-n <the number of entries to create>`
- `-p <the path where the entries will be created>`

An example of calling this test may be:

```
$GLITE_LOCATION/test/glite-data-catalog-fireman/bin/gLite-fireman-create-test -e  
"http://lxb2081.cern.ch:8080/egtest/glite-data-catalog-service-fr-  
mysql/services/FiremanCatalog" -n 1000 -p "/TestsDir/02_"
```

On the other hand, the `gLite-fireman-create-bulk-test` creates entries in bulk operations. The parameters accepted are:

- `-l <to enable verbose output>`
- `-e <the Catalog endpoint>`
- `-n <the number of entries to create>`
- `-s <the size of bulk entries>`
- `-p <the path where the entries will be created>`

As an example, we could execute:

```
$GLITE_LOCATION/test/glite-data-catalog-fireman/bin/gLite-fireman-create-bulk-test -l -e  
"http://lxb2081.cern.ch:8080/egtest/glite-data-catalog-service-fr-  
mysql/services/FiremanCatalog" -n 1000 -s 100 -p "/TestsDir/01_"
```

Note: For both tests, it is supposed that the 'TestsDir' directory already exists in the catalog.

b) Using QMTest:

- Requirements:
  - If you don't have QMTest installed in your machine, you can download it from <http://www.codesourcery.com/qmtest/download.html>
  - Make sure that 'qmtest' is in your PATH.
- Set the environment variable:  
QMTEST\_CLASS\_PATH=\$GLITE\_LOCATION/test/config/qmtest/test-classes/
- Start QMTest through the GUI typing:  
qmtest -D \$GLITE\_LOCATION/test/config/qmtest gui --address "<your hostname>" --port <port number> --no-browser &
- Open a browser pointing to <http://<your hostname>:<port number>/test/dir> to access to the QMTest GUI. Once there, click on "glite-fireman", and then on "r\_1\_0\_0" to see all the tests. Select any test or a whole test suite and run it clicking on "Run this test" or "Run this Suite" from the pull-down menu. You should modify first some of the parameters of each test, specially the Catalog endpoint. You can do this modifying the .qmt files that are located in the directory \$GLITE\_LOCATION/test/config/qmtest/glite-fireman.qms/r\_1\_0\_0.qms/

Note that for this release you should execute these tests in a particular order, so for example, in this way, to run the remove directory test you must first have run the create directory test. The recommended order to run these tests in QMTest is this one:

- 1) [glite-fireman-getversion-fireman-mysql](#)
- 2) [glite-fireman-getversion-fireman-oracle](#)
- 3) [glite-fireman-mkdir-fireman-mysql](#)
- 4) [glite-fireman-mkdir-fireman-oracle](#)
- 5) [glite-fireman-rmdir-fireman-mysql](#)
- 6) [glite-fireman-rmdir-fireman-oracle](#)
- 7) [glite-fireman-mkdir-fireman-mysql](#)
- 8) [glite-fireman-mkdir-fireman-oracle](#)
- 9) [arda-create-1000-entries-bulk-100-fireman-oracle](#)
- 10) [arda-create-1000-entries-per-call-fireman-oracle](#)
- 11) [arda-create-1000-entries-per-call](#)
- 12) [glite-fireman-put-lfn-800-chars](#)
- 13) [glite-fireman-readdir-fireman-mysql](#)
- 14) [glite-fireman-readdir-fireman-oracle](#)

This situation will be improved in a next release.

### 17.3.5. Test results

a) From the command line:

The test results can be visualized in stdout.

b) Using QMTest:

Using QMTest you can see the output of each test clicking on "Details".

## 17.4. WMS TEST SUITE

### 17.4.1. Installation Pre-requisites

Need to have access to a gLite UI in order to install the testsuite RPM

### 17.4.2. Installation

This test suite is installed using the glite-testsuites-wms-2.0.1 rpm that can be obtained from the gLite web site (e.g.

[http://glite.web.cern.ch/glite/packages/\\*\\*release\\*\\*/bin/rhel30/i386/RPMS](http://glite.web.cern.ch/glite/packages/**release**/bin/rhel30/i386/RPMS)).

The installation of the rpm will deploy the tests under \$GLITE\_LOCATION/test/glite-wms directory.

### 17.4.3. Configuration

This test suite should be run from the UI.

Before running the test suite, check the following points:

- Export the variable GSI\_PASSWORD to the value of the actual password for your proxy file (required during the creation of the proxy)
 

```
Bash: export GSI_PASSWORD=myPerSonalSecreForProxy1243
Tcsh  setenv GSI_PASSWORD myPerSonalSecreForProxy1243
```
- Export the variable REFVO to the name of the reference VO you want to use for the test
 

```
Bash: export REFVO=egtest
Tcsh: setenv REFVO egtest
```
- Define the Regression Test file (`regressionTest.file`)
- Write in a file all the single commands for the tests you want to be executed.
  - You can either call directly the BaseTest command to execute them, or call them individually from the main script, or better call them subsequently from a regression test file. The recommended way of doing it is to execute all tests in one shot.
  - In the directory \$GLITE\_LOCATION /test/wms/ you can find the example file regressionTest.file containing 10 example tests belonging to the standard JRA-1 testing testbed suite (see for example <http://cern.ch/glite-wms-test>)
- Customize the machine names for the specific roles (CE, WMS, WNs, SE, MyProxy) of the testbed nodes inside the file \$GLITE\_LOCATION /test/wms/opt/edg/tests/etc/test\_site-LocalTB.conf

### 17.4.4. Execution

Run the set of tests by launching the MainScript (located at \$GLITE\_LOCATION /test/glite-wms/opt/edg/bin/MainScript) with the following options:

```
opt/edg/bin/MainScript --forcingVO=egtest --verbose
--regFile=regressionTest.file RTest
```

To keep the log in a file you can also do :

```
opt/edg/bin/MainScript --forcingVO=egtest
--verbose --regFile=regressionTest.file RTest | tee MyLogFile
```



#### 17.4.5. Test results

The output of the test suite is written under `/tmp/<username>` in a file specified by the suite itself.

The name of the actual `index.html` and the tarzipped file with all required HTML for all tests is stated at the end of the test execution in the standard output.

For example the suite shows the following 2 lines at the end of its execution:

```
HTML      in:          /tmp/reale/050401-003320_LocalTB/index.html
TarBall   in: lxb1409.cern.ch /tmp/reale/050401-003320_LocalTB/tarex.tgz
```

Normally this needs to be put in the doc root of your Web Server, and to be unzipped and untared there.

The log file of the execution should normally be copied to the “annex” subdir of the directory structure you get by unzipping and untaring the `tarex.tgz`, and be renamed there as “MainLog”.

The HTML output allows for the monitor of the test execution, examination of the test log files, contains a detailed description of each test performed and displays the time required for the execution of the test itself.



## 18. APPENDIX A: SERVICE CONFIGURATION FILE EXAMPLE

This is an example of local service configuration file for a Computing Element node using PBS as batch system.

```
<!-- Default configuration parameters for the gLite CE Service -->
<config>
  <parameters>

    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
    <!-- User-defined parameters - Please change them -->
    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

    <!-- VOs configuration
         These parameters are matching arrays of values containing one value
         for each VO served by this CE node -->

    <voms.voname
      description="The names of the VOs that this CE node can serve">

      <value>EGEE</value>

    </voms.voname>

    <voms.vomsnode
      description="The full hostname of the VOMS server responsible for each VO.
      Even if the same server is responsible for more than one VO, there must
      be exactly one entry for each VO listed in the 'voms.voname' parameter.
      For example: 'host.domain.org'">

      <value>lxb000.cern.ch</value>

    </voms.vomsnode>

    <voms.vomsport
      description="The port on the VOMS server listening for request for each VO
      This is used in the vomses configuration file
      For example: '15000'">

      <value>17001</value>

    </voms.vomsport>

    <voms.voms-certsub
      description="The subject of the host certificate of the VOMS
      server for each VO. For example: '/C=ORG/O=DOMAIN/OU=GRID/CN=host.domain.org'">

      <value>/C=CH/O=CERN/OU=GRID/CN=lxb000.cern.ch</value>

    </voms.voms-certsub>

    <!-- Pool accounts configuration
         The following parameters must be set for both LSF and PBS/Torque systems
         The pool accounts are created and configured by default if these parameters
         are defined. You can remove these parameters to skip pool accounts configuration,
         however it is better to configure the parameters and let the script verify
         the correctness of the installation.
         These parameters are matching arrays of values containing one value
         for each VO served by this CE node. The list must match
         the corresponding lists in the VO configuration section -->

    <pool.account.basename
      description="The prefix of the set of pool accounts to be created for each VO.
      Existing pool accounts with this prefix are not recreated">

      <value>egee</value>

    </pool.account.basename>

    <pool.account.group
      description="The group name of the pool accounts to be used for each VO.
      For some batch systems like LSF, this group may need a specific gid. The gid can
      be
      set using the pool.lsfgid parameter in the LSF configuration section">

      <value>egee-gr</value>

    </pool.account.group>
```

```

    <pool.account.number
      description="The number of pool accounts to create for each VO. Each account
already      will be created with a username of the form prefixXXX where prefix
              is the value of the pool.account.basename parameter. If matching pool accounts
              exist, they are not recreated.
              The range of values for this parameter is from 1 to 999">
              <value>40</value>
    </pool.account.number>

    <!-- CE Monitor configuration
      These parameters are required to configure the CE Plugin for the
readme.txt    CE Monitor web service. More information about the following
              parameters can be found in $GLITE_LOCATION/share/doc/gLite-ce-ce-plugin/ce-info-
              or in the CE chapter of the gLite User Manual -->

    <cemon.wms.host
CE"          description="The hostname of the WMS server that receives notifications from this
              value="lxb0001.cern.ch"/>

    <cemon.wms.port
this CE     description="The port number on which the WMS server receiving notifications from
              is listening"
              value="8500"/>

    <cemon.lrms
description="The type of Local Resource Management System. It can be 'lsf' or 'pbs'
              If this parameter is absent or empty, the default type is 'pbs'"
              value="pbs"/>

    <cemon.cetype
description="The type of Computing Element. It can be 'condorc' or 'gram'
              If this parameter is absent or empty, the default type is 'condorc'"
              value="condorc"/>

    <cemon.cluster
itself"     description="The cluster entry point host name. Normally this is the CE host
              value="lxb0002.cern.ch"/>

    <cemon.static
description="The name of the configuration file containing static information"
              value="`${GLITE_LOCATION}/etc/gLite-ce-ce-plugin/ce-static.ldif"/>

    <cemon.cluster-batch-system-bin-path
'/usr/local/lsf/bin'
description="The path of the lrms commands. For example: '/usr/pbs/bin' or
depending on the value
              This value is also used to set the PBS_BIN_PATH or LSF_BIN_PATH variables
              of the 'cemon.lrms' parameter"
              value="/usr/pbs/bin"/>

    <cemon.cesebinds
description="The CE-SE bindings for this CE node. There are three possible format:
configfile
'queue[|queue]' se
'queue[|queue]'se se entry point
A . character for the queue list means all queues
Example: '.' EGEE::SE::Castor /tmp">
              <value>'.' EGEE::SE::Castor /tmp </value>
    </cemon.cesebinds>

    <cemon.queues
description="A space-separated list of the queues defined on this CE node
Example: blah-pbs-egge-high"
              value=" blah-pbs-egge-high "/>

    <!--
    <!-- LSF configuration
      The following parameters are specific to LSF. They may have to be set
depending on your local LSF configuration.
      If LSF is not used, remove this section -->

    <pool.lsfgid

```

```

installations,
one value
the values empty">
description="The gid of the groups to be used for the pool accounts on some LSF
on per each pool account group. This parameter is an array of values containing
for each VO served by this CE node. The list must match
the corresponding lists in the VOMS configuration section
If this is not required by your local LSF system remove this parameter or leave
the values empty">
<value>changeme</value>
</pool.lsfgid>
-->
<!-- Condor configuration -->
<condor.wms.user
description="The username of the condor user under which
the Condor daemons run on the WMS nodes that this CE serves"
value="wmsegee"/>
<!-- Logging and Bookkeeping -->
<lb.user
description="The account name of the user that runs the local logger daemon
If the user doesn't exist it is created. In the current version, the
host certificate and key are used as service certificate and key and are
copied in this user's home in the directory specified by the global
parameter 'user.certificate.path' in the glite-global.cfg.xml file"
value="lbegee"/>
<!-- Firewall configuration -->
<iptables.chain
description="The name of the chain to be used for configuring the local firewall.
If the chain doesn't exist, it is created and the rules are assigned to this
chain.
If the chain exists, the rules are appended to the existing chain"
value="EGEE-DEFAULT-INPUT"/>
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
<!-- Advanced parameters - Change them if you know what you're doing -->
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
<!-- gLite configuration -->
<glite.installer.verbose
description="Enable verbose output"
value="true"/>
<glite.installer.checkcerts
description="Enable check of host certificates"
value="true"/>
<!-- PBS configuration
The following parameters are specific to PBS. They may have to be set
depending on your local PBS configuration.
If PBS is not used, remove this section -->
<PBS_SPOOL_DIR
description="The PBS spool directory"
value="/usr/spool/PBS"/>
<!-- LSF configuration
The following parameters are specific to LSF. They may have to be set
depending on your local LSF configuration.
If LSF is not used, remove this section -->
<LSF_CONF_PATH
description="The directory where the LSF configuration file is located"
value="/etc"/>
<!-- Globus configuration -->
<globus.osversion
description="The kernel id string identifying the system installed on this node.
For example: '2.4.21-20.ELsmp'. This parameter is normally automatically detected,
but it can be set here"
value=""/>
<globus.hostdn
description="The host distinguished name (DN) of this node. This is normally
automatically
read from the server host certificate. However it can be set here. For example:
'C=ORG, O=DOMAIN, OU=GRID, CN=host/server.domain.org'"
value=""/>

```

```

<!-- Condor configuration -->
<condor.version
  description="The version of the installed Condor-C libraries"
  value="6.7.3"/>

<condor.user
  description="The username of the condor user under which
  the Condor daemons must run"
  value="condor"/>

<condor.releasedir
  description="The location of the Condor package. This path is internally simlinked
  to /opt/condor-c. This is currently needed by the Condor-C software"
  value="/opt/condor-6.7.3"/>

<CONDOR_CONFIG
  description="Environment variable pointing to the Condor
  configuration file"
  value="{condor.releasedir}/etc/condor_config"/>

<condor.scheddinterval
  description="How often should the schedd send an update to the central manager?"
  value="10"/>

<condor.localdir
  description="Where is the local condor directory for each host?
  This is where the local config file(s), logs and
  spool/execute directories are located"
  value="/var/local/condor"/>

<condor.blahgahp
  description="The path of the gLite blahp daemon"
  value="$GLITE_LOCATION/bin/blahpd"/>

<condor.daemonlist
  description="The Condor daemons to configure and monitor"
  value="MASTER, SCHEDD"/>

<condor.blahpollinterval
  description="How often should blahp poll for new jobs?"
  value="120"/>

<gatekeeper.port
  description="The gatekeeper listen port"
  value="2119"/>

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
<!-- System parameters - You should leave these alone -->
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

</parameters>
</config>

```



## 19. APPENDIX B: SITE CONFIGURATION FILE EXAMPLE

This is an example of site configuration file for the same CE node as in Appendix A. In order to propagate the full configuration from the central configuration server, the configuration file in Appendix A can be simply replaced with the following single line:

```
<config/>
```

Alternatively, any parameter left in local service file and properly defined in the case of user-defined parameters will override the values set in the site configuration file. The following file also contains a default parameters section with the parameters required by the gLite Security Utilities module. This default section is inherited by all nodes.

```
<!-- Default configuration parameters for the gLite CE Service -->
<siteconfig>
  <parameters>

    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
    <!-- User-defined parameters - Please change them -->
    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

    <cron.mailto
      description="E-mail address for sending cron job notifications"
      value="egee-admin@cern.ch"/>

    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
    <!-- Advanced parameters - Change them if you know what you're doing -->
    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

    <!-- Installer configuration -->
    <glite.installer.verbose
      description="Enable verbose output"
      value="true"/>

    <install.fetch-crl.cron
      description="Install the glite-fetch-crl cron job. Possible values are
      'true' (install the cron job) or 'false' (do not install the cron job)"
      value="true"/>

    <install.mkgridmap.cron
      description="Install the glite-mkgridmap cron job and run it once.
      Possible values are 'true' (install the cron job) or 'false' (do
      not install the cron job)"
      value="false"/>

    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
    <!-- System parameters - You should leave these alone -->
    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

  </parameters>

  <node name="lxb0002.cern.ch">
    <parameters>

      <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
      <!-- User-defined parameters - Please change them -->
      <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

      <!-- VOs configuration
      These parameters are matching arrays of values containing one value
      for each VO served by this CE node -->

      <voms.voname
        description="The names of the VOs that this CE node can serve">

        <value>EGEE</value>

      </voms.voname>

      <voms.vomsnode
        description="The full hostname of the VOMS server responsible for each VO.
        Even if the same server is responsible for more than one VO, there must
        be exactly one entry for each VO listed in the 'voms.voname' parameter.
        For example: 'host.domain.org'">
```



```

        <value>lxb0000.cern.ch</value>

</voms.vomsnode>

<voms.vomsport
  description="The port on the VOMS server listening for request for each VO
  This is used in the vomses configuration file
  For example: '170001'">

  <value>15001</value>

</voms.vomsport>

<voms.vomscertssubj
  description="The subject of the host certificate of the VOMS
  server for each VO. For example: '/C=ORG/O=DOMAIN/OU=GRID/CN=host.domain.org'">

  <value>/C=CH/O=CERN/OU=GRID/CN=lxb0000.cern.ch </value>

</voms.vomscertssubj>

<!-- Pool accounts configuration
  The following parameters must be set for both LSF and PBS/Torque systems
  The pool accounts are created and configured by default if these parameters
  are defined. You can remove these parameters to skip pool accounts configuration,
  however it is better to configure the parameters and let the script verify
  the correctness of the installation.
  These parameters are matching arrays of values containing one value
  for each VO served by this CE node. The list must match
  the corresponding lists in the VO configuration section -->

<pool.account.basename
  description="The prefix of the set of pool accounts to be created for each VO.
  Existing pool accounts with this prefix are not recreated">

  <value>egee</value>

</pool.account.basename>

<pool.account.group
  description="The group name of the pool accounts to be used for each VO.
  For some batch systems like LSF, this group may need a specific gid. The gid can
  be
  set using the pool.lsfgid parameter in the LSF configuration section">

  <value>egee</value>

</pool.account.group>

<pool.account.number
  description="The number of pool accounts to create for each VO. Each account
  will be created with a username of the form prefixXXX where prefix
  already
  is the value of the pool.account.basename parameter. If matching pool accounts
  exist, they are not recreated.
  The range of values for this parameter is from 1 to 999">

  <value>40</value>

</pool.account.number>

<!-- CE Monitor configuration
  These parameters are required to configure the CE Plugin for the
  CE Monitor web service. More information about the following
  parameters can be found in $GLITE_LOCATION/share/doc/glite-ce-ce-plugin/ce-info-
  readme.txt
  or in the CE chapter of the gLite User Manual -->

<cemon.wms.host
  description="The hostname of the WMS server that receives notifications from this
  CE"
  value="lxb0001.cern.ch"/>

<cemon.wms.port
  description="The port number on which the WMS server receiving notifications from
  this CE
  is listening"
  value="8500"/>

<cemon.lrms
  description="The type of Local Resource Management System. It can be 'lsf' or 'pbs'

```

```

        If this parameter is absent or empty, the default type is 'pbs'"
        value="pbs"/>

<cemon.cetype
  description="The type of Computing Element. It can be 'condorc' or 'gram'
  If this parameter is absent or empty, the default type is 'condorc'"
  value="condorc"/>

<cemon.cluster
  description="The cluster entry point host name. Normally this is the CE host
  itself"
  value="lxb0002.cern.ch"/>

<cemon.static
  description="The name of the configuration file containing static information"
  value="{GLITE_LOCATION}/etc/glite-ce-ce-plugin/ce-static.ldif"/>

<cemon.cluster-batch-system-bin-path
  description="The path of the lrms commands. For example: '/usr/pbs/bin' or
  '/usr/local/lsf/bin'
  This value is also used to set the PBS_BIN_PATH or LSF_BIN_PATH variables
  depending on the value
  of the 'cemon.lrms' parameter"
  value="/usr/pbs/bin"/>

<cemon.cesebinds
  description="The CE-SE bindings for this CE node. There are three possible format:
  configfile
  'queue[|queue]' se
  'queue[|queue]'se se entry point
  A . character for the queue list means all queues
  Example: '.' EGEE::SE::Castor /tmp">

  <value>'.' EGEE::SE::Castor /tmp</value>

</cemon.cesebinds>

<cemon.queues
  description="A space-separated list of the queues defined on this CE node
  Example: blah-pbs-egee-high"
  value="blah-pbs-egee-high"/>

<!-- LSF configuration
  The following parameters are specific to LSF. They may have to be set
  depending on your local LSF configuration.
  If LSF is not used, remove this section -->

<!--
  <pool.lsfgid
  description="The gid of the groups to be used for the pool accounts on some LSF
  installations,
  on per each pool account group. This parameter is an array of values containing
  one value
  for each VO served by this CE node. The list must match
  the corresponding lists in the VOMS configuration section

  If this is not required by your local LSF system remove this parameter or leave
  the values empty">

  <value></value>

</pool.lsfgid>

-->

<!-- Condor configuration -->
<condor.wms.user
  description="The username of the condor user under which
  the Condor daemons run on the WMS nodes that this CE serves"
  value="wmsegee"/>

<!-- Logging and Bookkeeping -->
<lb.user
  description="The account name of the user that runs the local logger daemon
  If the user doesn't exist it is created. In the current version, the
  host certificate and key are used as service certificate and key and are
  copied in this user's home in the directory specified by the global
  parameter 'user.certificate.path' in the glite-global.cfg.xml file"
  value="lbegee"/>

<!-- Firewall configuration -->
<iptables.chain
  description="The name of the chain to be used for configuring the local firewall.

```

chain.

If the chain doesn't exist, it is created and the rules are assigned to this

If the chain exists, the rules are appended to the existing chain"

```
value="EGEE-DEFAULT-INPUT"/>
```

```

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
<!-- Advanced parameters - Change them if you know what you're doing -->
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

<!-- gLite configuration -->
<glite.installer.verbose
  description="Enable verbose output"
  value="true"/>

<glite.installer.checkcerts
  description="Enable check of host certificates"
  value="true"/>

<!-- PBS configuration
  The following parameters are specific to PBS. They may have to be set
  depending on your local PBS configuration.
  If PBS is not used, remove this section -->

<PBS_SPOOL_DIR
  description="The PBS spool directory"
  value="/usr/spool/PBS"/>

<!-- LSF configuration
  The following parameters are specific to LSF. They may have to be set
  depending on your local LSF configuration.
  If LSF is not used, remove this section -->

<LSF_CONF_PATH
  description="The directory where the LSF configuration file is located"
  value="/etc"/>

<!-- Globus configuration -->
<globus.osversion
  description="The kernel id string identifying the system installed on this node.
  For example: '2.4.21-20.ELsmp'. This parameter is normally automatically detected,
  but it can be set here"
  value=""/>

<!-- Condor configuration -->
<condor.version
  description="The version of the installed Condor-C libraries"
  value="6.7.3"/>

<condor.user
  description="The username of the condor user under which
  the Condor daemons must run"
  value="condor"/>

<condor.releasedir
  description="The location of the Condor package. This path is internally simlinked
  to /opt/condor-c. This is currently needed by the Condor-C software"
  value="/opt/condor-6.7.3"/>

<CONDOR_CONFIG
  description="Environment variable pointing to the Condor
  configuration file"
  value="{condor.releasedir}/etc/condor_config"/>

<condor.scheddinterval
  description="How often should the schedd send an update to the central manager?"
  value="10"/>

<condor.localdir
  description="Where is the local condor directory for each host?
  This is where the local config file(s), logs and
  spool/execute directories are located"
  value="/var/local/condor"/>

<condor.blahgahp
  description="The path of the gLite blahp daemon"
  value="$GLITE_LOCATION/bin/blahpd"/>

<condor.daemonlist
  description="The Condor daemons to configure and monitor"
  value="MASTER, SCHEDD"/>

<condor.blahpollinterval

```

```
description="How often should blahp poll for new jobs?"
value="10"/>

<gatekeeper.port
description="The gatekeeper listen port"
value="2119"/>

<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
<!-- System parameters - You should leave these alone -->
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->

</parameters>
</node>
</siteconfig>
```