gLite

v. 1.1 (rev. 2)

Installation Guide

30 April 2005

The information contained in this document represents the views of EGEE as of the date they are published. EGEE does not guarantee that any information contained herein is errorfree, or up to date.

EGEE MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, BY PUBLISHING THIS DOCUMENT.

TABLE OF CONTENT

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
FTS	File Transfer Service
LB	Logging and Bookkeping
SC	Single Catalog
UI	User Interface
VOMS	Virtual Organization Management Service

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 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.

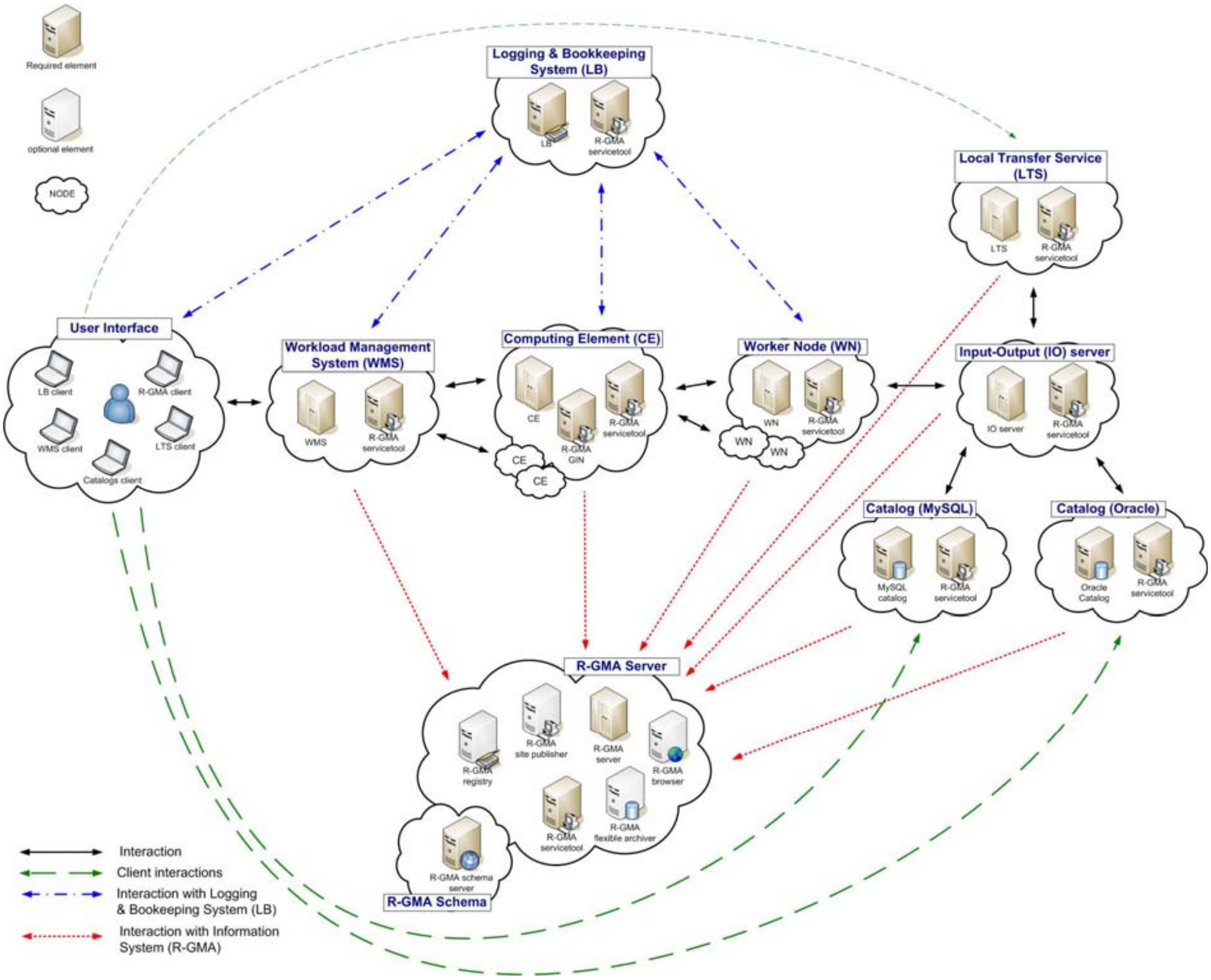

Figure 1: gLite Service Deployment Scenario

- Information and Monitoring Services
  - o 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 1.1.

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.

In the rest of this guide, installation instructions for the individual modules are presented. The order of chapters represents the suggested installation order for setting up a gLite grid.

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/APT.asp).

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
User-defined Parameters		
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 overriden by values specified in the local configuration files. Leave this parameter empty or remove it to use local configuration only.
Advanced Parameters		
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_08	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/grid-mapfile	Location of the grid mapfile
host.gridmap.dir	/etc/grid-security/gridmapdir	The location of the account lease information for dynamic allocation
System Parameters		
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 specify 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 file according to specific criteria and point different services to different files. For example is possible to put all parameters required to configure the Worker Nodes in one file and all parameters for the servers in a separate files, or have a separate file for each node and so on.

Several configuration files can also be managed as a single file by using the XML inclusion mechanism. Using this standard mechanism, it is possible to include by reference one or more files in a master file and point the gLite services configuration scripts to the master file. In order to use this mechanism, the <siteconfig> tag in the master file must be qualified with the XInclude namespace as follows:

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

The individual files can then be included using the tag:

<xi:include href="glite-xxx.cfg.xml" />

where the value of the *href* attribute is a file path relative to the location of the master file. The content of the referenced file is included "as-is" in the master document when it is downloaded from the web server. The gLite service gets a single XML file where all the <*xi:include*> tags are replaced with the content of the referenced files.

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 service come with their 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 any gLite configuration scripts is run, it creates or modifies a general configuration file called ***glite_setenv.sh*** (and ***glite_setenv.csh***) 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 logins 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	/tpm/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. This module is presented first, since it is used by almost all other modules. However, it is not normally installed manually by itself, but automatically as part of the other modules.

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

User-defined Parameters		
cron.mailto		E-mail address to which the stderr of the installed cron jobs is sent
Advanced Parameters		
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 *host.certificate.file* and *host.key.file*
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)
System Parameters		

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. INFORMATION AND MONITORING SYSTEM (R-GMA)

6.1. 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.

The gLite R-GMA Server is normally the first module installed as part of a gLite grid, since all services require it to publish service information.

6.1.1. R-GMA Server

The R-GMA Server 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.

6.1.2. R-GMA Clients

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 (or Service Publisher)***
  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.

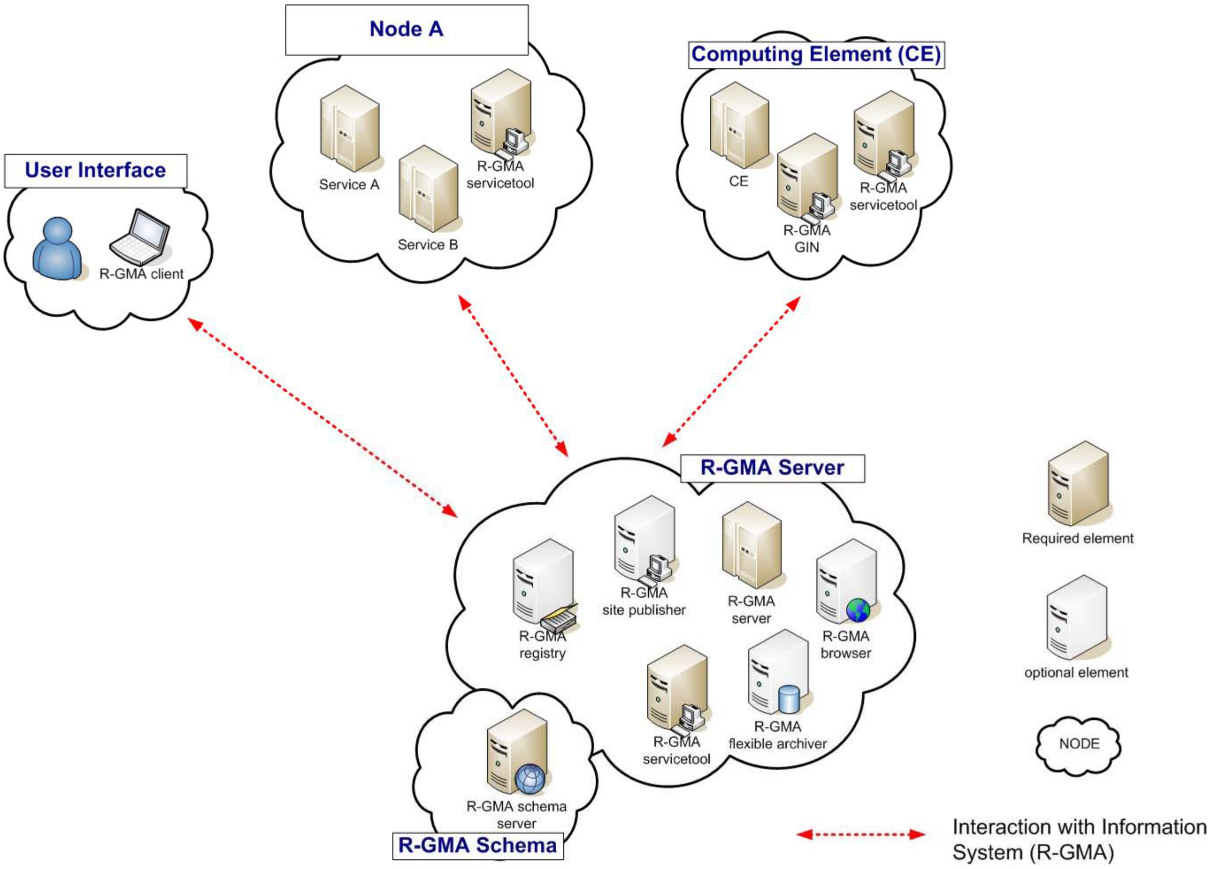

Figure 2 R-GMA components

6.2. 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 3 gives a list of R-GMA deployment modules, their content and/or the list of gLite deployment modules that install/use them.

6.2.1. R-GMA Deployment strategy

Deployment module	***Contains***	***Used / included by***
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) File Transfer Service (Oracle) 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 3: R-GMA deployment modules

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 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 at least one registry server per grid, but there can be several.
- 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).

6.2.2. R-GMA Server deployment module

6.2.2.1. 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 6.1.1) as well as the R-GMA clients – R-GMA servicetool, site publisher and data archiver (see section 6.1.2):

- 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*.*

6.2.2.2. 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.

6.2.2.3. 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 14 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*

6.2.2.4. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA Server. This release requires v. 1.4.2 revision 08. The JDK/JRE version to be used is a parameter in the gLite global 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.

6.2.2.5. 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:

gLite	in /opt/glite ($GLITE_LOCATION)
gLite-essentials- java	in $GLITE_LOCATION/externals/share
MySQL-server	in /usr
MySQL-client	in /usr
Tomcat	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.

6.2.2.6. 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 4 shows the list of parameters that can be set.

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

Table 4: R-GMA common configuration parameters

3. 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 5 shows the list of parameters that can be set.

Parameter	Default value	Description
User-defined Parameters		
rgma.server. mysql_root_password		MySQL root password. *Example: verySecret*
rgma.server. run_schema_service		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. *Example: yes*
rgma.server. run_registry_service		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'. *Example: yes*
rgma.server. run_browser		Run an R-GMA browser (yes|no). Running a browser is optional but useful. *Example: yes*
rgma.server. run_archiver		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). *Example: yes*
rgma.server.run_site_publisher		Run the R-GMA site-publisher (yes|no). Running the site-publisher publishes your site to R-GMA. *Example: yes*
rgma.site-publisher.contact.system_administrator		Contact email address of the site system administrator. *Example: systemAdministrator@mysite.com*

rgma.site-publisher.contact.user_support		Contact email address of the user support. *Example: userSupport@mysite.com*
rgma.site-publisher.contact.site_security		Contact email address of the site security responsible. *Example: security@mysite.com*
rgma.site-publisher.location.latitude		Latitude of your site. Please go to 'http://www.multimap.com/' to find the correct value for your site. *Example: 46.2341*
rgma.site-publisher.location.longitude		Longitude of your site. Please go to 'http://www.multimap.com/' to find the correct value for your site. *Example: 6.0447*
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output. *Example : true*
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. *Example: 1000*
rgma.site-publisher.site-name	${HOSTNAME}	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). It normally defaults to the DNS name of the R-GMA Server running the Site Publisher service *Example: lxb1420.cern.ch*
rgma.server.archiver_configuration_file	${GLITE_LOCATION}/etc/rgma-flexible-archiver/glue-config.props	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') *Example: '/my/path/my_config_file.props'*
System Parameters		
rgma.server. webapp.path	R-GMA	Path under which R-GMA server should be deployed. *Example: R-GMA*
rgma.server. war.name	R-GMA.war	Name of war file for R-GMA server. *Example: R-GMA.war*

Table 5: 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.

6.2.2.7. 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*:

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

2. Sets 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]
CATALINA_HOME	to the location specified in the global configuration file [default is /var/lib/tomcat5/]
JAVA_HOME	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.

6.2.3. R-GMA Client deployment module

6.2.3.1. 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.

6.2.3.2. 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.

6.2.3.3. 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 14 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).

6.2.3.4. 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.

6.2.3.5. 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:

gLite	in /opt/glite ($GLITE_LOCATION)
gLite-essentials- java	in $GLITE_LOCATION/externals/share
gLite-essentials- cpp	in $GLITE_LOCATION/externals/
swig-runtime	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.

6.2.3.6. 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).
2. 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 4 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 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.*

set.proxy.path	false	If this parameter is true, the configuration script sets the GRID_PROXY_FILE and X509_USER_PROXY environment variables to the default value /tmp/x509up_u`id -u`. The parameter is set to false by default, since these environment variables are normally handled by other modules (like the gLite User Interface and the CE job wrapper on the Worker Nodes) and setting them here may create conflicts. It may be however necessary to let the R-GMA client set the variables for stand-alone use Example: false [type: 'boolean']

4. Table 6 shows a list of parameters that can be set.

Parameter	Default value	Description
User-defined Parameters		
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
System Parameters		
set.proxy.path	false	If this parameter is true, the configuration script sets the

		GRID_PROXY_FILE and X509_USER_PROXY environment variables to the default value /tmp/x509up_u`id -u`. The parameter is set to false by default, since these environment variables are normally handled by other modules (like the gLite User Interface and the CE job wrapper on the Worker Nodes) and setting them here may create conflicts. It may be however necessary to let the R-GMA client set the variables for stand-alone use Example: false [type: 'boolean']

Table 6: 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:

5. Run the R-GMA Client configuration file

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

6. The R-GMA Client is ready.

6.2.3.7. R-GMA Client Configuration Walkthrough

The R-GMA Client 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]

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]
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.

6.2.4. R-GMA servicetool deployment module

6.2.4.1. 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.

6.2.4.2. 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.

6.2.4.3. 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 14 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).

6.2.4.4. 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.

6.2.4.5. 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.

6.2.4.6. 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).
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 4 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 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 7 shows a list of the parameters that can be set.

Parameter	Default value	Description
User-defined Parameters		
rgma.servicetool.sitename		DNS name of the site for the published services (in general the hostname). *Example: lxb2029.cern.ch*
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output. *Example : true*
rgma.servicetool.activate	true	Turn on/off servicetool for the node. Example: true] [Type: 'boolean']
System Parameters		

Table 7: R-GMA servicetool configuration parameters

4. 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 8 shows the list of parameters for each service that you have to set accordingly. If the *rgma.servicetool.activate* parameter is set to false, then the servicetool daemon is not started and no service publishing occurs. This can be used on gLite nodes in case R-GMA Server is not used. It is possible to prevent individual services from being published by setting the *rgma.servicetool.enable* parameter to false in the service instance.
5. 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
User-defined Parameters		
rgma.servicetool.enable	true	Publish this service via the R-GMA servicetool. If this varaiable set to false the other values below are not taken into account. *Example: true*
rgma.servicetool.name		Name of the service. This should be globally unique. *Example: host.name.service_name*

rgma.servicetool. url_endpoint		URL to contact the service at. This should be unique for each service. *Example: http://your.host.name:port/serviceGroup/ServiceName*
rgma.servicetool. service_type		The service type. This should be uniquely defined for each service type. Currently two methods of type naming are recommended: • The targetNamespace from the WSDL document followed by a space and then the service name • A URL owned by the body or individual who defines the service type *Example: Name of service type*
rgma.servicetool. service_version		Service version in the form 'major.minor.patch' *Example: 1.2.3*
rgma.servicetool. publish_interval		How often to publish the service details (like endpoint, version etc). (Unit: seconds) *Example: 600*
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' *Example: /opt/glite/bin/myService/serviceStatus*
rgma.servicetool. status_interval		How often check and publish service status. (Unit: seconds) *Example: 60*
rgma.servicetool.url_wsdl		URL of a WSDL document for the service (leave blank if the service has no WSDL). *Example: http://your.host.name/service/wsdl.html*
rgma.servicetool. url_semantics		URL of a document containing a detailed description of the service and how it should be used. *Example:* *http://your.host.name/service/semantics.html*
Advanced Parameters		
System Parameters		

Table 8: R-GMA servicetool configuration parameters for a service to be published via the R-GMA servicetool

6.2.4.7. 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
   ```

6.2.5. R-GMA GadgetIN (GIN) deployment module

6.2.5.1. 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.

6.2.5.2. 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.

6.2.5.3. 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 14 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).

6.2.5.4. 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.

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

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.

6.2.5.6. 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 4 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 9 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
User-defined Parameters		
rgma.gin.run_generic_info_provider		Run generic information provider (gip) backend (yes|no). Within LCG this comes with the ce and se *Example: no*
rgma.gin.run_fmon_provider		Run fmon backend (yes|no). This is used by LCG for gridice. *Example: no*
rgma.gin.run_ce_provider		Run ce backend (yes|no).
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output. *Example : true*
System Parameters		

Table 9: R-GMA GadgetIN configuration parameters

6.2.5.7. 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.

7. VOMS SERVER AND ADMINISTRATION TOOLS

7.1. SERVICE OVERVIEW

VOMS serves as a central repository for user authorization information, providing support for sorting users into a general group hierarchy, keeping track of their roles, etc. Its functionality may be compared to that of a Kerberos KDC server. The VOMS Admin service is a web application providing tools for administering member databases for VOMS, the Virtual Organization Membership Service.

VOMS Admin provides an intuitive web user interface for daily administration tasks and a SOAP interface for remote clients. (The entire functionality of the VOMS Admin service is accessible via the SOAP interface.) The Admin package includes a simple command-line SOAP client that is useful for automating frequently occuring batch operations, or simply to serve as an alternative to the fullblown web interface. It is also useful for bootstrapping the service.

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* 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*

7.2.2. 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.

7.3. 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 Admnistration 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*

7.4. 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 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.* 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
VO Instances parameters		
voms.vo.name		Name of the VO associated with the VOMS instance
voms.port.number		Port number of the VOMS instance
vo.admin.e-mail		E-mail address of the VO admin
vo.ca.URI		URI from where the CRls are downloaded
User-defined Parameters		
voms.mysql.passwd		Password (in clear) of the root user of the MySQL server used for VOMS databases
Advanced Parameters		
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

System Parameters

Table 10: 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.

7.5. 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:$GLOBUS_LOCA
       TION/bin:$PATH
   LD_LIBRARY_PATH=$GLITE_LOCATION/lib:$GLITE_LOCATION/externals/lib
                   $GLOBUS_LOCATION/lib:$LD_LIBRARY_PATH
   ```

7.6. VOMS ADMINISTRATORS REGISTRATION

After the installation and configuration of the VOMS Server and Admin Tools, it is necessary to register at least one administrator for each registered VO running the following command on the VOMS server:

```
$GLITE_LOCATION/bin/vomsadmin --vo <VO name> create-user <certificate.pem> assign-role VO VO-Admin <certificate.pem>
```

where *<VO name>* is the name of the registered VO for which to register the administrator and *<certificate.pem>* is the path to the public certificate of the administrator. For more information, please refer to the VOMS Administrative Tools guide on the gLite web site.

8. WORKLOAD MANAGER

8.1. 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.

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* (gLite Security Utilities) is installed and configured automatically when installing and configuring the WMS (refer to Chapter 14 for more information about the Security Utilites 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

8.2.2. Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA Servicetool service. 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.

8.3. 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.x (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 6.2.4.5.

8.4. 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 11.

Parameter	Default value	Description
User-defined Parameters		
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’.

voms.voname		The names of the VOs that this WMS node can serve (array parameter)
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. 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 Example: 15000
voms.vomscertsubj		The subject of the host certificate of the VOMS server for each VO. Example: /C=ORG/O=DOMAIN/OU=GRID/CN=host.domain.org
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
Advanced Parameters		
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}/gsiwuftpd.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 fil
condor.blahpollinterval	10	How often should blahp 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
System Parameters		

condor.localdir	/var/local/condor	Condor local directory
condor.daemonlist	MASTER, SCHEDD, COLLECTOR, NEGOTIATOR	List of the condor daemons to start. This must a comma-separated list of services as it would appear in the Condor configuration file

Table 11: 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 7 shows a list of the parameters that can be set. More details can be found in section 6.2.4.6.

   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 8 on page 39 in the section 6.2.4 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 6.2.4.6.

***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.

8.5. 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:$GLOBUS_LOCATION/bin:$CONDORG_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

8.6. MANAGING THE WMS SERVICES

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

glite-wms-config.py --start	Starts all WMS services (or restart them if they are already running)
glite-wms-config.py --stop	Stops all WMS services
glite-wms-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-wms-config.py --startservice=xxx	Starts 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 & Bookkeeping client

glite-wms-config.py --stopservice=xxx	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 & Bookkeeping client

8.7. STARTING THE WMS 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.

8.8. PUBLISHING WMS SERVICES TO R-GMA

The WMS services are published to R-GMA using the R-GMA Servicetool service. The Servicetool 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 6.2.4 in this guide.

9. LOGGING AND BOOKKEEPING SERVER

9.1. 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.

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

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 14). 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*

9.3. JAVA JRE/JDK

The Java JRE or JDK are required to run the R-GMA Servicetool service. 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.

9.4. 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 6.2.4.5.

9.5. LOGGING AND BOOKEEPING 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 12). 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 12.

Parameter	Default value	Description
User-defined Parameters		
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 of comment it out to use the same as 'glite.user.name'
Advanced Parameters		
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
System Parameters		

Table 12: 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 7 shows a list of the parameters that can be set. More details can be found in section 6.2.4.6.

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 8 on page 39 in the section 6.2.4 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 6.2.4.6.

***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.

9.6. 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:$GLOBUS_LOCA
           TION/bin:$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

10. THE TORQUE RESOURCE MANAGER

10.1. 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*.

10.1.1. 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.

10.1.2. 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).

10.2. 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.

10.3. TORQUE SERVER

10.3.1. 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 6.2.4.5.

10.3.2. TORQUE Server Service Configuration

11. 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 13)
12. 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 13. The R-GMA servicetool related parameters can be found in Table 7.

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

    a. *Common parameters* (first part of Table 13)

       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 13)

       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 13)

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.

Common parameters		
Parameter	Default value	Description
User-defined Parameters		
torque-server.force		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.
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output.
torque-server.name	${HOSTNAME}	Name of the machine where the job server is running, it usually corresponds to the

		Computing Element: Example: ${HOSTNAME}.
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	Secifies 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. Ths parameter specifies the global poll interval for all resource managers.
maui.log.filename	${GLITE_LOCATION_LOG}/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]"
System Parameters		

Worker node instances		
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].

Queue instances		
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 enqueuing 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 13: 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 7 shows a list of the parameters that can be set. More details can be found in section 6.2.4.6.
3. *Service Configuration for the R-GMA servicetool:*

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

   glite-torque-server.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 8 on page 39 in the section 6.2.4 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 6.2.4.6.

***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.

12.1.1. TORQUE Server Configuration Walkthrough

The Torque Server configuration script performs the following steps:

1. Load the Torque Server configuration file $GLITE_LOCATION/etc/config/glite-torque-server.cfg.xml
2. Add the torque and maui ports to /etc/services.
3. Create the /var/spool/pbs/server_name file containing the torque server hostname.
4. Create the list with the torque clients under /var/spool/pbs/server_priv/nodes.
5. Create the pbs_server configuration.
6. Start the pbs_server.
7. Look for changes in the pbs_server configuration since the last time the Torque Server was configured.
8. Establish the server configuration performing the necessary updates.
9. 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).
10. Execute the defined queues configuration
11. 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.
12. 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.
13. Run the /opt/edg/sbin/edg-pbs-shostsequiv script.
14. Look for duplicated key entries in /etc/ssh/ssh_known_hosts.
15. 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).

16. 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.
17. Run /opt/edg/sbin/edg-pbs-knownhosts to add the keys to /etc/ssh/ssh_known_hosts.
18. 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.
19. Restart the sshd daemon to take the changes into account.
20. Restart the pbs_server.
21. Create the maui configuration file in /var/spool/maui/maui.cfg.
22. Start the maui service.
23. Configure the servicetool to register the torque services defined in the configuration file.

12.1.2. Managing the TORQUE Server Service

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

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

12.2. TORQUE CLIENT

12.2.1. 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.*

12.2.2. 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)
2. 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
User-defined Parameters		
torque-server.name		Name of the machine where the job server is running, it usually corresponds to the Computing Element: Example: ${HOSTNAME}.
se.name		Storage Element name (if necessary).
Advanced Parameters		
Glite.installer.verbose	True	Enable verbose output.
mom-server.logevent	255	Sets the mask that determines which event types are logged by pbs_mom.
mom-server.loglevel	4	Specifies the verbosity of logging with higher numbers specifying more verbose logging. Values may range between 0 and 7
System Parameters		

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

12.2.3. TORQUE Client Configuration Walkthrough

The Torque Client configuration script performs the following steps:

1. Load the Torque Client configuration file *$GLITE_LOCATION/etc/config/glite-torque-client.cfg.xml*
2. Create the /var/spool/pbs/server_name file containing the torque server hostname.
3. Add the torque and maui ports to /etc/services.

4. 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.
5. Look for duplicated key entries in /etc/ssh/ssh_known_hosts.
6. 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.

12.2.4. Managing the TORQUE Client

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

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

13. COMPUTING ELEMENT

13.1. 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 5 for more details about R-GMA and the R-GMA servicetool.

13.2. 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.

13.2.1. 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 14 for more information about the Security Utilites module). The module contains the latest version of the CA certificates plus a number of certificate utilities and security utilities. In particular this module installs the *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). Please note that 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
-I <Chain_Name> 11 -m state --state NEW -m udp -p udp --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.

13.2.2. 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 Java 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 13.4 for more details).

13.2.3. 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. A gLite deployment module for installing Torque and Maui as RMS ar provided, please refer to chapter 10 for more information.

13.3. 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 /opt/glite ($GLITE_LOCATION)
Condor	in /opt/condor-x.y.x (where x.y.z is the current Condor version)
Globus	in /opt/globus ($GLOBUS_LOCATION)
Tomcat	in /var/lib/tomcat5 (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 6.2.4.5.

13.4. COMPUTING ELEMENT SERVICE CONFIGURATION

2. 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).
3. Copy the following configuration file templates

   $GLITE_LOCATION/etc/config/templates/glite-ce.cfg.xml
   $GLITE_LOCATION/etc/config/templates/glite-security-utilities.cfg.xml
   $GLITE_LOCATION/etc/config/templates/glite-rgma-common.cfg.xml
   $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.* The following parameters can be set (please refer to the Security Utilities and R-GMA chapters for a description of the parameters used by those modules):

Parameter	Default value	Description
User-defined Parameters		
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 reponsible 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.vomscertsubj		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 *pool.lsfgid* 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 *pool.account.basename* 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 Managment 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. The format is: '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
Advanced Parameters		
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 mormally 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	${condor.releasedir}/etc/condor_config	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	/var/local/condor	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	${GLITE_LOCATION}/bin/blahpd	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	/opt/lcg	The location where the LCG providers are installed.
System Parameters		

Table 14: CE Configuration Parameters

4. 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 7 shows a list of the parameters that can be set. More details can be found in section 6.2.4.6.
    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 8 on page 39 in the section 6.2.4 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 6.2.4.6.

***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*

5. Install the VOMS server(s) host certificates in the directory /etc/grid-security/vomsdir
6. As root run the CE configuration file */opt/glite/etc/config/scripts/glite-ce-config.py*

13.5. 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 *$GLITE_LOCATION/etc/vomses* file with the VO information set in the configuration file
16. Run GPT post-installation and Globus configuration scripts
17. Create the /opt/condor-c link to the Condor package and customize the Condor-C configuration file by adding the required BLAHP entries
18. Configure the Local Logger daemons by creating/verifying the account used to run them and making a copy of the host certificate and key to this user home directory in .certs (the location can be configured using the global parameter *user.certificate.path*)
19. Install and configure the CE Monitor and CE Plugin by installing the Monitor war in the local Tomcat installation, creating necessary links to the CE Plugin jars and create the predefined subscriptions file *$CATALINA_HOME/webapps/ce-monitor/subscriptions/predifinedSubscriptionList.xml*. If the files already exists, backup copies are created with the extension '.1'
20. Configure the information providers by creating the *$GLITE_LOCATION/etc/glite-ce-ce-plugin/lcg-info-generic.conf* file (used by the LCG Info Providers and R-GMA Gin service). Please note that the script configures a set of default parameters. If a different configuration is needed the file *$GLITE_LOCATION/etc/glite-ce-ce-plugin/lcg-info-generic.conf* should be edited by hand by the system administrator and the following command should be run:

    ```
    /opt/lcg/sbin/lcg-info-generic-config /opt/glite/etc/glite-ce-ce-plugin/lcg-info-generic.conf
    ```

21. 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.
22. The R-GMA servicetool is configured and the services are added to the R-GMA servicetool.
23. The Gatekeeper, the LB Logger client, Tomcat the CE Monitor and the R-GMA services are started

13.6. MANAGING THE CE 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

13.7. 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.

13.8. 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

14. WORKER NODE

14.1. 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 Bookeeping Client, the R-GMA Client and the WMS Checkpointing library. The gLite Torque Client module can be installed together with the WN module if Torque is used as a batch system.

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

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 14). 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

14.2.2. 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 Java 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 13.4 for more details).

14.2.3. 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.

14.3. 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

5. 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.

14.4. WORKER NODE 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 *$GLITE_LOCATION/etc/config/templates/glite-wn.cfg.xml* to *$GLITE_LOCATION/etc/config/glite-wn.cfg.xml* and modify the parameters values as necessary (Table 15)
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
User-defined Parameters		
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 nqme of the service. This is used in the command line if special services need to be adressed. URL is the service endpoint and serviceType is the java class defining the type of the service.
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
System Parameters		
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 15: WN Configuration Parameters

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

14.5. 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

15. DATA CATALOGS (FIREMAN)

15.1. 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:

gilte-data-single-catalog	➔	MySQL version
gilte-data-single-catalog-oracle	➔	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.

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. 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 14). 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*
3. Install the VOMS Server(s) host certificate in the directory */etc/grid-security/vomsdir*. This is necessary to extract the VOMS information from the VOMS proxies.

15.2.2. 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 Java 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.

15.2.3. 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

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

15.3. SINGLE CATALOG INSTALLATION

1. Download from the gLite web site the latest version of the SC installation script *glite-data-single-catalog_install.sh (*or *glite-data-single-catalog-oracle_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
Tomcat 5	in /var/lib/tomcat5

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*

15.4. 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 16 to Table 19)
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
User-defined Instance Parameters per each VO		
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 acessing the catalog database

Table 16: Single Catalog for MySQL Configuration Parameters for each VO instance

Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
allow.unsecure.port	false	Enable using the unsecure port 8080. It can be true or false. Example: false
db.force.create	false	If the catalog mysql database has already been created on this node, running the configuration script will drop and recreate it if this parameter is set to true. If the parameter is set to false the database will be created only if it doesn't exist. The default value is false [Type: boolean]
catalog-service-fr-mysql.DOCBASE	${GLITE_LOCATION}/share/java/glite-data-catalog-service-fr-mysql.war	Location of the glite-data-catalog-service-fr-mysql.war file
catalog-service-fr-mysql.DBDRIVERCLASS	org.gjt.mm.mysql.Driver	JDBC driver classname
catalog-service-fr-mysql.MODULE.NAME	glite-data-catalog-service-fr-mysql	Catalog service module name
catalog-service-fr-mysql.MESSAGINGON	false	If 'true', then a connection to the specified messaging system is attempted and messages will be

		produced.
catalog-service-fr-mysql.MESSAGINGJNDIHOST		The host of the JNDI server that contains the messaging system connetion factories and topic/queue objects.
catalog-service-fr-mysql.MESSAGINGJNDIPORT		The port of the JNDI server that contains the messaging system connetion factories and topic/queue objects.
catalog-service-fr-mysql.MESSAGINGJMSNAME		The JNDI name of the 'local' messaging server to connect to.
catalog-service-fr-mysql.MESSAGINGTOPIC		The JNDI name of the topic that the messages should be produced on.
System Parameters		
catalog-service-fr-mysql.DBURL	jdbc:mysql://localhost:3306/${catalog-service-fr-mysql.DBNAME}	URL of the database

Table 17: Single Catalog for MySQL Common Configuration Parameters

Parameter	Default value	Description
User-defined Parameters		
catalog-service-fr.VONAME		Name of the Virtual Organisation which is served by the catalog instance
catalog-service-fr.DBNAME		Database name used for a catalog service
catalog-service-fr.DBUSER		Database user name owning the catalog database
catalog-service-fr.DBPASSWORD		Password for acessing the catalog database
catalog-service-fr.DBHOST		Hostname of the Oracle server ex: lxfs5502.cern.ch
Advanced Parameters		
catalog-service-fr.DBURL	jdbc:oracle:thin:@${catalog-service-fr.DBHOST}:1521:${catalog-service-fr.DBNAME}	URL of the database

Table 18: Single Catalog for Oracle Configuration Parameters for each VO instance

Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
allow.unsecure.port	false	Enable using the unsecure port 8080. It can be true or false. Example: false
catalog-service-fr.MESSAGINGON	false	If 'true', then a connection to the specified messaging system is attempted and messages will be produced.
catalog-service-fr.MESSAGINGJNDIHOST		The host of the JNDI server that contains the messaging system connetion factories and topic/queue objects.
catalog-service-fr.MESSAGINGJNDIPORT		The port of the JNDI server that contains the messaging system connetion factories and topic/queue objects.
catalog-service-fr.MESSAGINGJMSNAME		The JNDI name of the 'local' messaging server to connect to.
catalog-service-fr.MESSAGINGTOPIC		The JNDI name of the topic that the messages should be produced on.
System Parameters		
catalog-service-fr.DOCBASE	${GLITE_LOCATION}/share/java/glite-data-catalog-service-fr.war	Location of the glite-data-catalog-service-fr-mysql.war file
catalog-service-fr.DBDRIVERCLASS	oracle.jdbc.driver.OracleDriver	JDBC driver classname
catalog-service-fr.MODULE.NAME	glite-data-catalog-service-fr	Catalog service module name
catalog-service-fr.oracle-jdbc.classpath	${CATALINA_HOME}/common/lib	Path to the Oracle JDBC drivers

Table 19: Single Catalog for Oracle Common 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.

15.5. 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 Single Catalog 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 >
   ```

15.6. PUBLISHING CATALOG SERVICES TO R-GMA

The Fireman Catalog 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 Catalog modules. The Catalogs configuration file contains a separate configuration section (an <*instance/*>) for each Catalog 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 the RGMA section in this guide.

16. FILE TRANSFER SERVICE ORACLE

16.1. 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.

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.

16.2.1. 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*
3. Install the VOMS Server(s) host certificate in the directory */etc/grid-security/vomsdir*. This is necessary to extract the VOMS information from the VOMS proxies.

16.2.2. 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.

16.2.3. Oracle InstantClient

The Oracle Instant Client is required to run the File Transfer Service. Due to license reasons, we cannto redistribute it. Version 10.1.0.3-1 can be downloaded from the Oracle Web Site (http://www.oracle.com/technology/software/tech/oci/instantclient/htdocs/linuxsoft.html).

16.2.4. Oracle Database Configuration

Before installing the File Transfer Service module, it is necessary to create users in Oracle and assign specific privileges. To create a new user with the necessary privileges, do the following as DBA:

```
create user <DBUSER> identified by '<DBPASSWORD>';
grant resource to <DBUSER>;
grant create session to <DBUSER>;
grant create synonym to <DBUSER>;
grant connect to <DBUSER>;
grant create any procedure to <DBUSER>;
grant create any sequence to <DBUSER>;
grant create trigger to <DBUSER>;
grant create type to <DBUSER>;
```

You may otionally grant debugging privileges:

```
grant debug any procedure to <DBUSER>;
grant debug connect session to <DBUSER>;
```

16.3. FILE TRANSFER SERVICE INSTALLATION

1. Download from the gLite web site the latest version of the FTS installation script *glite-file-transfer-service-oracle_install.sh*. It is recommended to download the script in a clean directory
2. Make the script executable (*chmod u+x glite-file-transfer-service-oracle_installer.sh*) and execute it or execute it with *sh glite-data-file-transfer-service-oracle_install.sh*
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-file-transfer-service-oracle 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. Download and install the Oracle Instant Client Basic version 10.1.0.3-1 (oracle-instantclient-basic-10.1.0.3-1.i386.rpm) and Oracle Instant Client SQL-Plus (oracle-instantclient-sqlplus-10.1.0.3-1.i386.rpm). To do so, connect to the Oracle Web Site.
6. The gLite FTS configuration script is installed in *$GLITE_LOCATION/etc/config/scripts/glite-file-transfer-service-config-oracle.py*. A template configuration file is installed in *$GLITE_LOCATION/etc/config/templates/glite-file-transfer-service-oracle.cfg.xml*

16.4. FILE TRANSFER SERVICE ORACLE 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 *$GLITE_LOCATION/etc/config/templates/glite-file-transfer-service-oracle.cfg.xml* to *$GLITE_LOCATION/etc/config/glite-file-transfer-service-oracle.cfg.xml* and modify the parameters values as necessary (**Error! Reference source not found.**)
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:

Per instance Parameters

Parameter	Default value	Description
User-defined Parameters		
data-transfer-fts.DBINDEXNAME		Name of the VO for a given instance
data-transfer-fts.DBUSER		Name of the database user owning the transfer database
data-transfer-fts.DBPASSWORD		Password for accessing the transfer database
data-transfer-fts.DBHOST		Hostname of the transfer database
data-transfer-fts.DBNAME		Database name of the transfer database
data-transfer-fts.DBINSTANCE		Instance name of the transfer database (depending on the version and configuration of the Oracle database, this may be the same as data-transfer-fts.DBNAME or it may differ)
data-transfer-fts.DBINDEXNAME		Tablespace name for the index in the transfer database
Advanced Parameters		
data-transfer-fts.DBURL	jdbc:oracle:thin:@${data-transfer-fts.DBHOST}:1521:${data-transfer-fts.DBINSTANCE}	
data-transfer-fts.SECURITY_ENABLED	true	If set to 'false', no authorization will be made at all,regardless of the

		attribute settings below and regardless of whether a secure connector is used or not. Setting to 'true' will requires the use of a secure connector and the use of an appropriately authorized certificate.
data-transfer-fts.QUERY_ATTRIBUTE	none	Normally a user is only permitted to list their own jobs.If a user's certificate contains this VOMS attribute, they are additionally permitted to list any job in the service
data-transfer-fts.QUERY_MAPFILE		Normally a user is only permitted to list their own jobs.If a client's certificate subject name is listed in this file, they are additionally permitted to list any job on the service
data-transfer-fts.SUBMIT_ATTRIBUTE	none	If this attribute is set for the service, a client may submit jobs to the service
data-transfer-fts.SUBMIT_MAPFILE	${GLITE_LOCATION}/etc/glite-data-transfer-submit-mapfile	If a client's certificate subject name is listed in this file, a client may submit jobs to the service
data-transfer-fts.CANCEL_ATTRIBUTE	none	Normally a user is only permitted to cancel their own jobs. If a user's certificate contains this VOMS attribute, they are additionally permitted to cancel any job in the service
data-transfer-fts.CANCEL_MAPFILE	${GLITE_LOCATION}/etc/glite-data-transfer-cancel-mapfile	Normally a user is only permitted to cancel their own jobs. If a client's certificate subject name is listed in this file, they are additionally permitted to cancel any job on the service
data-transfer-fts.MANAGER_ATTRIBUTE	none	If a user's certificate contains this VOMS attribute, they are additionally permitted to do any operation upon the service including manage channels
data-transfer-fts.MANAGER_MAPFILE	${GLITE_LOCATION}/etc/glite-data-transfer-manager-mapfile	If a client's certificate subject name is listed in this file, they are additionally permitted to do any operation upon the service including manage channels.
transfer-agent.log.Priority	WARN	WARN, DEBUG, INFO

transfer-agent.log.Filename	${GLITE_LOCATION_LOG}/glite-transfer-agent-${data-transfer-fts.VONAME}.log	The location of the log file
transfer-agent-vo.Quota	70	The percentage of the concurrent transfers that the VO is allowed to submit. for example, a value of 70 means that the VO can run have up to 70% of MaxTransfers at the same time
transfer-agent-vo.DisableDelegation	False	Disable the Delegation. If that parameter is set to true, the transfer will be performed using the service credentails, otherwise it will use the proxy certificate downloaded from MyProxy
transfer-agent-fsm.EnableHold	true	When this paramter is set to true, a transfer will be moved in case a (consecutive) failure(s) to the Hold state, waiting for manual intervention, otherwise it will go in TransferFailed
transfer-agent-myproxy.Server		The host name of the MyProxy Server. If that parameter is not set or is empty, the myproxy default will apply
transfer-agent-myproxy.ProxyLifetime	86400	The lifetime in seconds of the proxy certificates that will be created
transfer-agent-myproxy.Repository	/tmp	The location where the certificates retrieved from the MyProxy Service will be stored. That location must already exist
transfer-agent-myproxy.MinValidityTime	3600	"The minimum validity time (in seconds) an already existent certificate should have before submitting a new job. In case the certificate couldn't satisfy that requirement, a new certificate will be retrieved from the MyProxy Service
transfer-agent-actions.MaxRetries	3	The maximum number of retries that should be attempted before moving the file to Hold or Failed
transfer-agent-actions.ResubmitDelay	600	The delay in second before a Waiting transfer is resubmitted

transfer-agent-fts-urlcopy.MaxTransfers	10	The maximum number of transfers that can process simultaneously for each channel
transfer-agent-fts-urlcopy.Streams	1	The number of parallel streams that should be used during the transfer
transfer-agent-fts-urlcopy.LogLevel	WARN	The Log Level for the Glite URL Copy Transfer. Allowed values are: DEBUG, INFO, WARN, and ERROR
transfer-agent-scheduler.MaxFailures	0	The number of consecutive failures before an Action is considered disabled for DisableTime seconds. If that value is set to zero, tactions will never be disabled and the parameter DisableTime is ignored
transfer-agent-scheduler.DisableTime	300	The number of seconds an action stays disabled
transfer-agent-scheduler.Allocate_Interval	10	The time interval (in seconds) used to schedule the Action Allocate. If that value is not set, the Action will not be scheduled
transfer-agent-scheduler.Check_Interval	10	The time interval (in seconds) used to schedule the Action Check. If that value is not set, the Action will not be scheduled
transfer-agent-scheduler.Cancel_Interval	10	The time interval (in seconds) used to schedule the Action Cancel. If that value is not set, the Action will not be scheduled"
transfer-agent-scheduler.Fetch_Interval	10	The time interval (in seconds) used to schedule the Action Fetch. If that value is not set, the Action will not be scheduled
transfer-agent-scheduler.BasicRetry_Interval	10	The time interval (in seconds) used to schedule the Action BasicRetry. If that value is not set, the Action will not be scheduled
transfer-agent-scheduler.DataIntegrity_Interval	3600	"The time interval (in seconds) used to schedule the Action DataIntegrity. If that value is not set, the Action will not be scheduled
data-transfer-fts.DOCBASE	${GLITE_LOCATION}/share/java/glite-data-transfer-fts.war	"Location of the FTS WAR file

data-transfer-fts.DBDRIVERCLASS	oracle.jdbc.driver.OracleDriver	Java class name of the JDBC driver
transfer-agent-vo.Name	${data-transfer-fts.VONAME}	The name of the VO which the File Placement Service Queue belongs to -- should be the same as the transfer-fts VO name
transfer-agent-myproxy.Port	0	The port of the MyProxy Server. If that parameter is not set or is 0, the myproxy default will applies
transfer-agent-dao-oracle.ConnectString	${data-transfer-fts.DBHOST}:1521/${data-transfer-fts.DBNAME}	The Oracle ConnectString identifying the DB
transfer-agent-dao-oracle.User	${data-transfer-fts.DBUSER}	Must match with transfer-fts.DBUSER
transfer-agent-dao-oracle.Password	${data-transfer-fts.DBPASSWORD}	Password for accessing the transfer database. Must match with transfer-fts.DBPASSWORD
transfer-agent-dao-oracle.StatementCacheSize	0	The Size of the statement Cache.0 means that the caching is disabled. Note: since some memory leaks has been observed, it better for the moment to keep the cache disabled
transfer-agent-dao-oracle.ConnectionCheckInterval	60	Specify the time interval, in seconds, to periodically check if the connection is alive. If 0 is specified,the connection is checked on every utilization

Table 20: File Transfer Service Oracle Configuration Parameters (per instance)

Global Parameters

Parameter	Default value	Description
User-defined Parameters		
init.username		"The username of the user running the agents daemons. Example: gproduct
init.groupname		The groupname of the user running the agents daemons. Example: gm
init.uid		"The userid of the user running the agents daemons.Example: 13022
init.gid		The gid of the user running the agent daemons. Example: 2739

Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
allow.unsecure.port	false	Enable using the unsecure port 8080. It can be true or false. Example: false
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		The service certificate (public key) file location
service.key.file		The service certificate (private key) file location
System Parameters		
data-transfer-fts.oracle-instantclient.location	/usr/lib/oracle/10.1.0.3/client/	Location of the Oracle Instantclient installation

4. Put in the three mapfiles (submit, cancel and manager) the DNs of the users allowed to perform the specified operations
5. As root run the FTS configuration file *$GLITE_LOCATION/etc/config/scripts/glite-file-transfer-service-oracle-config.py*
6. The File Transfer Service Oracle is now ready.

16.5. FILE TRANSFER SERVICE ORACLE CONFIGURATION WALKTHROUGH

The File 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.

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 File Transfer Service Oracle configuration files *$GLITE_LOCATION/etc/config/glite-file-transfer-service-oracle.cfg.xml*

4. Set the following additional environment variables needed internally by the services.

5. Copy the Oracle jar file in the Tomcat location

6. Configure Tomcat.

7. Check the existence of the Oracle JDBC drivers.

8. Install the security utils.

9. Create the user/group accounts and set the right permissions.

10. Copy the host certificates to the user account.

11. Configure the File Transfer Service instances

    a. Checking environment variables (CATALINA_HOME)
    b. Creates the contex xml file
    c. Checks if the schemas already exist
    d. Uploads the Oracle Data Base Schemas (the common transfer schema and the transfer-agents schema)

12. Configure the Transfer Agents instances

    a. Create the transfer agents configuration files
    b. Run the configuration tool

16.6. PUBLISHING FILE TRANSFER SERVICES TO R-GMA

The FTS 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 FTS module. The FTS configuration file contains a separate configuration section (an <*instance/*>) for each FTS 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 the RGMA section in this guide.

17. METADATA CATALOG

17.1. SERVICE OVERVIEW

Metadata is in general a notion of 'data about data'. There are many aspects of metadata, like descriptive metadata, provenance metadata, historical metadata, security metadata, etc. Whatever is its nature, metadata is associated with items, named to be unique within the catalog.

The gLite Metadata Catalog makes no assumption on what each of these items represents (a file, a job on the grid ...). To each of these items a user may associate two sets of information:

1. Groups of key/value pairs (attributes), defined within schemas;
2. Permissions, just like in the gLite Fireman catalog, expressed via BasicPermissions and ACLs.

The functionality offered allows the user to manage the schemas, set and get values of attributes, perform queries using metadata values and manage the access permissions on each individual item.

17.2. 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.

17.2.1. 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 14). 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*

17.2.2. Java JDK

The Java JRE/JDK is required to run the Metadata Catalog Server. This release requires v. 1.4.2 (revision 04 or greater). The Java 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.

17.3. METADATA CATALOG INSTALLATION

1. Download from the gLite web site the latest version of the MC installation script *glite-data-metadata-catalog_install.sh*. It is recommended to download the script in a clean directory

2. Make the script executable (chmod u+x glite-data-metadata-catalog_installer.sh) and execute it or execute it with sh glite-data-metadata-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 ($GLITE_LOCATION)
MySQL-server	in	/usr
MySQL-client	in	/usr
Tomcat	in	/var/lib/tomcat5

5. The gLite MC configuration script is installed in

   $GLITE_LOCATION/etc/config/scripts/glite-data-metadata-catalog-config.py.

   A template configuration file is installed in

   $GLITE_LOCATION/etc/config/templates/glite-data-metadata-catalog.cfg.xml

17.4. METADATA 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 templates from
   $GLITE_LOCATION/etc/config/templates/glite-data-metadata-catalog.cfg.xml
   $GLITE_LOCATION/etc/config/templates/glite-security-utilities.cfg.xml
   $GLITE_LOCATION/etc/config/templates/glite-rgma-common.cfg.xml
   $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.*
3.

   There are three parts in the configuration file:
   a. Global metadata catalog configuration parameters
   b. VO specific instances of configuration parameters
   c. VO specific instances of configuration parameters for the R-GMA servicetool.

   First modify the common metadata catalog configuration parameters that are not VO specific. Table 21 shows a list of the global metadata catalog configuration variables that can be set:

Parameter	Default value	Description
User-defined Parameters		
data.metadata-catalog.		MySQL root password.

mysql_root_password		
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
System Parameters		
data.metadata-catalog.DBDRIVERCLASS	org.gjt.mm.mysql.Driver	JDBC driver classname
data.metadata-catalog.DBRESOURCENAME	meta	Name of the JNDI objetcs that is holding the DB connection object.
data.metadata-catalog.DOCBASE	${GLITE_LOCATION}/share/java/glite-data-catalog-service-meta.war	Location of the glite-data-catalog-service-fr-mysql.war file.
data.metadata-catalog.ATTRIBUTE_HELPER_CLASS	org.glite.data.catalog.service.meta.helpers.attribute.MySQLAttributeHelper	Name of the class (including the package name) implementing the logic for operations on attributes (MetadataBase - getAttributes, setAttributes, etc.)
data.metadata-catalog.CATALOG_HELPER_CLASS	org.glite.data.catalog.service.meta.helpers.catalog.MySQLCatalogHelper	Name of the class (including the package name) implementing the logic for operations on entries (MetadataCatalog - createEntry and removeEntry)
data.metadata-catalog.SCHEMA_HELPER_CLASS	org.glite.data.catalog.service.meta.helpers.schema.MySQLSchemaHelper	Name of the class (including the package name) implementing the logic for operations on schemas (MetadataSchema-createSchema, dropSchema, etc.)
data.metadata-catalog.AUTHORIZATION_HELPER_CLASS	org.glite.data.catalog.service.meta.helpers.authz.MySQLAuthorizationHelper	Name of the class (including the package name) implementing the logic for authorization (acess control) on entries in the catalog (FASBase - setPermission, getPermission, etc... plus the internal policy for creation of new entries and schemas).
data.metadata-catalog.schema-file	${GLITE_LOCATION}/etc/glite-data-catalog-service-meta/schema/mysql/mysql-schema.sql	Location of metadata catalog schema file

Table 21: Common Metadata Catalog Configuration Parameters

Next, configure the VO specific metadata catalog configuration parameters: In the configuration file you find a set of parameters for an instance called ‘changeme’ gouped by the tag

<instance name=”changeme”>

Create one set of parameters for each VO you want the metadata catalog support (by copying the corresponding <instance> enclosed parameters and by changing the instance name for each of these instances to the corresponding VO name.
Next adapt the parameters inside each instance accordingly. All the values with a token value of ‘changeme’ must be changed. Table 22: VO specific instance Metadata Catalog Configuration Parameters shows a list of variables that can be set:

Parameter	Default value	Description
User-defined Parameters		
data.metadata-catalog.VO		Name of the Virtual Organisation which is served by the catalog instance.
data.metadata-catalog.DBNAME		Name of Database used for the catalog service.
data.metadata-catalog.DBUSER		Database user name to access the catalog database.
data.metadata-catalog.DBPASSWORD		Password of database user specified in 'data.metadata-catalog.DBUSER'.
Advanced Parameters		
System Parameters		
Data.metadata-catalog.DBURL	jdbc:mysql://${HOSTNAME}:3306/${data.metadata-catalog.DBNAME}	URL of the database
Data.metadata-catalog.PATH	/${data.metadata-catalog.VO}/glite-data-catalog-service-meta	Path to the web application

Table 22: VO specific instance Metadata Catalog Configuration Parameters

4. The next point will discuss the configuration of the R-GMA and the R-GMA related configuration parameters. Please refer to the Security Utilities chapter for a description of the parameters used by this module.

3. Configure the R-GMA servicetool:
5. For this you have to configure the servicetool itself as well as configure the sub-services of the Metadata catalog for the publishing via the R-GMA servicetool:
   a. *R-GMA servicetool configuration:*

      Modify the common configuration parameters of R-GMA that can be found in the file

      glite-rgma-common.cfg.xml

      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

4 shows a list of the parameters that can be set. More details can be found in section 6.2.4.6.

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

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

glite-data-metadata-catalog.cfg.xml

that was mentioned before.
In this file, you will find one instance of a set of the rgma servicetool parameters for one VO that are grouped by the tag

```
<instance name="Metadata Catalog for VO changeme"
           service="rgma-servicetool">
```

Create one instance (grouped parameters) per VO that your metadata catalog is supporting, replace the 'changeme' in the instance name (see above) by the name of your VO and set the parameter

'vo.name'

also to the name of your VO. The other parameters in the instance have default values and don't need to be changed. Table 8 on page 39 in the section 6.2.4 about the R-GMA servicetool shows the general list of parameters for each instance for the publishing via the R-GMA servicetool. Again, you find the necessary steps described in section 6.2.4.6.

***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 Metadata Catalog configuration file
   $GLITE_LOCATION/etc/config/scripts/glite-data-metadata-catalog-config.py
5. The Metadata Catalog is now ready.

17.5. METADATA CATALOG CONFIGURATION WALKTHROUGH

The Metadata Catalog 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*:

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

2. Sets 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]
CATALINA_HOME	to the location specified in the global configuration file [default is /var/lib/tomcat5/]
JAVA_HOME	to the location specified in the global configuration file

3. Configures the gLite Security Utilities module
4. Verifies the JAVA installation

5. Checks the configuration values
6. Stops MySQL server if it is running
7. Starts mySQL server
8. Sets the MySQL root password
9. Stops Tomcat
10. Configures Tomcat
11. Configures the different VO instances inside Tomcat:
    a. Creates the DB user in MySQL
    b. Configures the context.xml in Tomcat
    c. Installs the web service for the VO
12. Configures the R-GMA servicetool and servicetool instances
13. Stops MySQL server
14. Starts the MySQL server.
15. Starts the tomcat server and gives time to go up to full speed before continuing
16. Starts the R-GMA servicetool.

18. GLITE I/O

18.1. GLITE I/O SERVER

18.1.1. 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.

18.1.2. 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.

18.1.2.1. 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*

18.1.2.2. 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.

18.1.3. 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 6.2.4.5.

18.1.4. 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 23 gives an overview of the parameters that can be set. The R-GMA servicetool related parameters can be found in Table 7.

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

   a. *Common parameters (first part of Table 23)*

      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 rfiod. Adapt also these configuration values accordingly. You can find more information on the values and the R-GMA servicetool in section 6.2.4 and 6.2.4.6.

   b. *VO dependant gLite I/O Server parameters (second part of Table 23)*

      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 6.2.4 and 6.2.4.6.

Common parameters *All parameters defined in this table are common to all instances.*		
Parameter	Default value	Description
User-defined Parameters		
I/O Daemon initialization parameters		
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 empy 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 empy or comment it out to use a system assigned gid.
Advanced Parameters		
General gLite initialization parameters		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check for host certificate
SSL Configuration parameters		
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		The service certificate (public key) file location.
service.key.file		The service certificate (private key) file location.
I/O Daemon parameters		
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	The rule to be applied to define the path for creating new files. Allowed values are: * abs_dir: The file name will be created by appending the file name to the path specified by RootPath configuration parameter * 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]
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
System Parameters		
I/O Daemon parameters		
io-daemon.EnablePerfMonitor	false	Enable the Performace 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 Performace 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-daemon.ResyncCache	false	Resynchronize the cache when the 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
User-defined Parameters		
vo.name		The name of the VO served by this

		instance.
io-daemon.Port		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. This port should not be higher than 9999 and different I/O server instances should not run on contigous ports (for example set one to 9999 and another one to 9998)
init.CatalogType		The type of catalog to use: - 'catalogs' (EDG Replica Location Service and Replica Metadata Catalog), - 'fireman' (gLite Fireman Catalog), - 'fr' (File and Replica Catalog) The parameters not used by the chosen catalog type can be removed or left empty
io-resolve-common parameters are required by all types of catalogues		
io-resolve-common.SrmEndPoint		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. *Example: httpg://gridftp05.cern.ch:8443/srm/managerV1*
io-resolve-common.SeHostname		The name of the Storage Element where the files are staged. It's the hostname of the remote file operation server. *Example: gridftp05.cern.ch*
io-resolve-common.RootPath		The path that should be prefixed to the filename when creating new files. *Example: /castor/cern.ch/user/g/glite/VO-NAME/SE/*
EDG RLS/RM parameters		

The parameters are only required when using the EDG catalogs. *Leave them empty or comment them if not used.*		
io-authz-catalogs.RmcEndPoint		The endpoint of the RMC catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. This is also the value of the 'io-resolve-catalogs.RmcEndpoint' parameter. *Example: https://lxb2028:8443/VO-NAME/edg-replica-metadata-catalog/services/edg-replica-metadata-catalog*
io-resolve-catalogs.RlsEndpoint		The endpoint of the Rls catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. *Example: https://lxb2028:8443/VO-NAME/edg-local-replica-catalog /services/edg-local-replica-catalog*
Parameters required by the Fireman and FR catalogs.		
io-authz-fas.FasEndpoint		The endpoint of the Fas catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. *Examples:* *http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FAS* (for FR) *http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FiremanCatalog (for Fireman)*
Fireman parameters		
io-resolve-		The endpoint of the FiReMan

fireman.FiremanEndpoint		catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. *Example:* *http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FiremanCatalog*
FR parameters		
io-resolve-fr.ReplicaEndPoint		The endpoint of the Replica catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. *Example:* *http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/ReplicaCatalog*
io-resolve-fr.FileEndPoint		The endpoint of the File catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP 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. *Example:* *http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FileCatalog*
Advanced Parameters		
Logging parameters		
log.Priority	DEBUG	The log4cpp log level. Possible values are: DEBUG, INFO, WARNING, ERROR, CRITICAL, ALERT, FATAL

log.FileName	$GLITE_LOCATION_LOG/glite-io-server-${vo.name}-${ init.CatalogType }.log	The location of the log file for this instance

Table 23: 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 7 shows a list of the parameters that can be set. More details can be found in section 6.2.4.6.

***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.

18.1.5. 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

18.2. CLIENT

18.2.1. 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.

18.2.2. 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.

18.2.2.1. 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 14). 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

18.2.3. 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

5. 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*

18.2.4. gLite I/O 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)
2. 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)
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:
4.

VO dependant gLite I/O Client instances *A separate gLite I/O Client instance can be installed for each VO that this client 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 the client to support*		
Parameter	Default value	Description
User-defined Parameters		
vo.name		The name of the VO for this instance.
io-client.ServerPort		The port that the gLite I/O Server is listening at for this VO
log.FileName	$GLITE_LOCATION_LOG/glite-io-client-${vo.name}.log	The location of the log file for this client instance

Parameter	Default value	Description
User-defined Parameters		
io-client.Server	changeme	The hostname where the gLite I/O Server is running
Advanced Parameters		
glite.installer.verbose	true	Enable configuration script verbose output
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'
System Parameters		
io-client.CacheLevel	7	The AliEn aiod Cache Level value
io-client.NumberOfStreams	1	Number of QUANTA tcp parallel streams

Table 24: gLite I/O Client configuration parameters

5. Run the gLite I/O client configuration file $GLITE_LOCATION/etc/config/scripts/glite-io-client-config.py
6. The gLite I/O client is now ready.

19. USER INTERFACE

19.1. 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 Managemenet System clients and APIs
- Logging and bookkeeping clients and APIs

19.2. 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.

19.2.1. Security Settings

1. 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 *fetch-crl* script using the fetch-crl RPM from the EU-GridPMA 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 it into the user's crontab is a manual step to do.

19.2.2. Java JRE/JDK

The Java JRE or JDK are required to run the UI. 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.

19.3. 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 by the user 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 (unless already installed) are installed in the tree from `pwd`/glite_ui by removing the /opt/[glite, globus, gpt] prefix. The GLITE_LOCATION, GLOBUS_LOCATION and GPT_LOCATION variables are set to the `pwd`/glite_ui value. If Globus and GPT are already installed before installing the gLite UI, they are not reinstalled and the existing GLOBUS_LOCATION and GPT_LOCATION can be used

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

19.4. UI CONFIGURATION

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

   from *$GLITE_LOCATION/etc/config/templates/glite-io-client.cfg.xml* to *$GLITE_LOCATION/etc/config/glite-io-client.cfg.xml*

   from *$GLITE_LOCATION/etc/config/templates/glite-rgma-client.cfg.xml* to *$GLITE_LOCATION/etc/config/glite-rgma-client.cfg.xml*

   from *$GLITE_LOCATION/etc/config/templates/glite-rgma-common.cfg.xml* to *$GLITE_LOCATION/etc/config/glite-rgma-common.cfg.xml* from *$GLITE_LOCATION/etc/config/templates/glite-security-utils.cfg.xml* to *$GLITE_LOCATION/etc/config/glite-security-utils.cfg.xml*

and modify the parameter values as necessary (Table 25). 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 including all parameters from all configuration files (refer to section 4.2.4 for more information)

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 configuration file contains one or more <set> sections, one per each VO that the UI must be configured for. It also contains a common <parameters> section used for all VOs.
   The following parameters can be set:

   For the <set/> section:

Parameter	Default value	Description
User-defined Parameters		
name		Name of set
ui.VirtualOrganisation		Name of the VO corresponding to this set
ui.NSAddresses		Array of the WMS Network Servers for this 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 this 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 for this VO

For the common parameters:

Parameter	Default value	Description
User-defined Parameters		
py-ui.DefaultVo		Default VO to connect
Advanced Parameters		
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	
System Parameters		

Table 25: 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.

19.5. 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_LOCATIO

N/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.
7. Runs the gLite I/O Client and gLite R-GMA Client configuration scripts
8. Runs the UI-specific configuration steps

19.6. 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.

19.7. 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 glite_setenv.[sh|csh] file in /etc/glite/profile.d/ or $HOME/.glite directory depending on the type of installation
2) log off and log in back. The file with UI environment variables will be sourced automatically.

20. THE GLITE FUNCTIONAL TEST SUITES

20.1. OVERVIEW

There are four suites described in this section, gLite I/O, Catalog, WMS and R-GMA.

20.2. I/O TEST SUITE

20.2.1. Test suite description

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.

20.2.2. Installation Pre-requisites

The gLite IO test suite depends on glite-data-io-client, so it is recommended to install and execute the IO tests from a UI machine. The IO test suite depends on CppUnit too, that should also be installed in the machine.

20.2.3. Installation

This test suite is installed using glite-testsuites-data-io-server 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/bin directory.

20.2.4. 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
- If you use TestManager to run the tests, you have to modify the following parameters in the configuration file, /opt/glite/test/etc/glite-data-io-server/ioServerTests.xml:
    - arg name="-c glite-io-client-egtest-mysql": where the IO config file pointing to the desired IO server instance should be specified
    - -o egtest-mysql: where the per VO DB instance should be specified

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.

20.2.5. Execution

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

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

These tests check the basic IO 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 '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/bin/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 TestManager:

- Requirements:

  - If you don't have TestManager installed in your machine, you can download the RPM from the gLite web site.

  - Python version 2.2.0 or higher.

- To execute all the tests type:

  python /opt/TestManager-1.3.0/testtools/TestManager.py /opt/glite/test/etc/glite-data-io-server/ioServerTests.xml

  (TestManager.py comes in the TestManager package, and ioServerTests.xml should be under $GLITE_LOCATION/test/etc/glite-data-io-server directory)

20.2.6. 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 TestManager:

Load form your preferred browser the index.html file that has been created under the 'report' directory.

20.3. CATALOG TEST SUITE

20.3.1. Test suite description

The Catalog 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. Additionally it includes file permission tests against the catalog secure interface.

20.3.2. 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.

20.3.3. Installation

This test suite is installed using the glite-testsuites-data-catalog-fireman 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/bin directory.

20.3.4. 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
- If you use TestManager to run the tests, you have to modify the following parameters in the configuration file, /opt/glite/test/etc/glite-data-catalog-fireman/ catalogsTests.xml:
  - <arg name="-s http://lxb2028.cern.ch:8080/EGEE/glite-data-catalog-service-fr/services/FiremanCatalog": where the relevant catalog endpoint should be specified
  - arg name="-c glite-io-client-egtest-mysql: where the IO config file pointing to the desired IO server instance should be specified

20.3.5. Execution

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

a) From the command line, you can execute the binaries that are located at $GLITE_LOCATION/test/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/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/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 TestManager:

- Requirements:
  - If you don't have TestManager installed in your machine, you can download the RPM from the gLite web site.
  - Python version 2.2.0 or higher.
- To execute all the tests type:

  python /opt/TestManager-1.3.0/testtools/TestManager.py /opt/glite/test/etc/glite-data-io-server/catalogsTests.xml

  (TestManager.py comes in the TestManager package, and catalogsTests.xml should be under $GLITE_LOCATION/test/etc/glite-data-catalog-fireman directory)

20.3.6. Test results

a) From the command line:

The test results can be visualized in stdout.

b) Using TestManager:

Check the index.html file that has been created under the 'report' directory.

20.4. WMS TEST SUITE

20.4.1. Test suite description

The WMS test suite contains 10 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 - InputOutput**: It submits a storm of 30 basic jobs involving Input and Output sandboxes
- **8 - RB_val**: storm of different kind of jobs.
- **9 and 10 - 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.
- **11 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.

20.4.2. Installation Pre-requisites

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

20.4.3. 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).

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

20.4.4. 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.reg`). A template of this file is provided at /opt/glite/test/glite-wms/opt/edg/tests/etc/config_tests_conf/regressionTest.reg. You should modify it accordingly to your testbed setup. The CE name should be changed in the –site parameter, and the –forcingVO parameter set to the VO to be used to run the tests.
- Customize the machine names for the specific roles (CE, WMS, WNs, SE ,MyProxy) of the testbed nodes inside the file
  $GLITE_LOCATION /test/glite-wms/opt/edg/tests/etc/test_site-LocalTB.conf.

20.4.5. Execution

Before running the tests, you should be placed in the directory $GLITE_LOCATION /test/glite-wms.

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=/opt/glite/test/glitewms/opt/edg/tests/etc/config_tests_conf
/regressionTest.reg RTest
```

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

```
opt/edg/bin/MainScript --forcingVO=egtest  --verbose
--regFile=/opt/glite/test/glitewms/opt/edg/tests/etc/config_tests_conf
/regressionTest.reg RTest | tee MyLogFile
```

20.4.6. 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.

20.5. R-GMA TEST SUITE

20.5.1. Test suite description

This test suite implements the test plan described at:
https://edms.cern.ch/document/568064

The tests implemented are:

test1: Creates a CONTINUOUS Primary Producer and Consumer locally, inserts one tuple and checks it can be consumed.
test2: Creates a LATEST Primary Producer and Consumer locally, inserts one tuple and checks it can be consumed.
test3: Creates a HISTORY Primary Producer and Consumer locally, inserts one tuple and checks it can be consumed.
test4A: Creates a CONTINUOUS Primary Producer and Consumer locally, inserts 1000 tuples and checks they can be consumed (MEMORY storage).
test4B: Creates a LATEST Primary Producer and Consumer locally, inserts 1000 tuples and checks they can be consumed (DATABASE storage).
test4C: Creates a HISTORY Primary Producer and Consumer locally, inserts 1000 tuples and checks they can be consumed (DATABASE storage).
test5: Submits a job to the Grid to create a HISTORY Primary Producer and insert 1000 tuples. Waits for job to complete, then creates a HISTORY consumer locally to check the tuples can be consumed (DATABASE storage).
test6: As test5, but with 10 jobs each publishing 100 tuples.
test7: Creates a HISTORY Primary Producer locally and inserts 1000 tuples, then submits a job to the Grid to create a HISTORY Consumer to check the tuples can be consumed (DATABASE storage).
test8: As test 7, but with 10 jobs each consuming the 1000 tuples.
test9: (will only do this if time)
test10: Checks retention periods and termination intervals are respected.
test11: (not sure this is possible from a UI as a standard user)
test12: Checks a (configurable) list of tables for reasonable content.

NB. For test4, these are the only three combinations of query type and storage that are supported by the RC1 server code. Tests for the remaining other combinations will be added when the server supports them (RC2?).

20.5.2. Installation Pre-requisites

These tests are designed to be run on a gLite UI machine with the Workload Management System and R-GMA client (C++ API) software installed.

20.5.3. Installation

This test suite is installed using the glite-testsuites-rgma RPM that can be obtained from the gLite web site (e.g. 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/rgma directory.

20.5.4. Configuration

The GLITE_LOCATION environment variable must be defined (so you should source glite_setenv.sh before running these tests). The RGMA_HOME environment variable will default to GLITE_LOCATION if it is not set explicitly.

You must have a valid Grid proxy certificate to run these tests (e.g. by running voms-proxy-init). The X509_USER_PROXY environment variable will default to /tmp/x509up_u${UID} if it is not set explicitly.

You must also have set up the gLite Grid job submission environment, i.e. the commands glite-job-submit, glite-job-status and glite-job-output must work.

There are some user-configurable parameters in "testprops.txt" to allow timings to be adjusted if tests fail due to very slow systems causing timeouts. You should not normally need to change these.

20.5.5. Execution

To run the tests, change to a working directory (e.g. /tmp) and run the script (with no parameters, e.g. /home/.../test1.sh). The script will create a sub-directory named after the test and process id in the current directory and place any working files there. All diagnostics (including test success or failure messages) will be written to standard error. All tests return 0 on success of 1 on error.

20.5.6. Test results

The script will create a sub-directory named after the test and process id in the current directory and place any working files there. All diagnostics (including test success or failure messages) will be written to standard error. All tests return 0 on success of 1 on error.

21. 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>

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

                <!-- 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 reponsible 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.vomscertsubj
                        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.vomscertsubj>

                <!-- 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>egeegr</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
                         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 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 Managment 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>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"/>

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

                <!-- 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 mormally
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"/>

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

        </parameters>
</config>
```

22. 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>

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

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

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

                <!-- 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"/>

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

        </parameters>

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

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

                 <!-- 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 reponsible 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.vomscertsubj
                         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.vomscertsubj>

                 <!-- 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>egeegr</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
                         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 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 Managment 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.
                        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"/>

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

                <!-- 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"/>

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

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