



Installation and Configuration Guide

v. 3.0 (rev.2)

08 May 2006

Copyright © Members of the EGEE Collaboration. 2004.

See <http://eu-egee.org/partners> for details on the copyright holders.

EGEE (“Enabling Grids for EscienceE in Europe”) is a project funded by the European Union. For more information on the project, its partners and contributors please see <http://www.eu-egee.org>. You are permitted to copy and distribute verbatim copies of this document containing this copyright notice, but modifying this document is not allowed. You are permitted to copy this document in whole or in part into other documents if you attach the following reference to the copied elements:

“Copyright © 2004. Members of the EGEE Collaboration. <http://www.eu-egee.org>”

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	8
1.1	PURPOSE	8
1.2	TERMINOLOGY	8
2	GLITE DEPLOYMENT	9
2.1	SERVICES AND COMPONENTS	9
2.2	STANDARD DEPLOYMENT MODEL	9
3	GLITE PACKAGES AND DOWNLOADS.....	12
4	THE GLITE CONFIGURATION MODEL	13
4.1	GLITE CONFIGURETION	13
4.1.1	YAIM configuration	13
4.1.2	The gLite Configuration Scripts	13
4.2	THE GLITE CONFIGURATION FILES	14
4.2.1	Configuration Parameters Scope	14
4.2.2	The Local Service Configuration Files.....	14
4.2.3	The Global Configuration File.....	15
4.2.4	The VO List File	17
4.2.5	The Site Configuration File	17
4.2.6	Internal Configuration	18
4.2.7	User environment	19
4.2.8	Default Environment Variables	19
4.2.9	Configuration Overrides.....	19
5	GLITE SECURITY UTILITIES.....	21
5.1	OVERVIEW.....	21
5.1.1	CA Certificates.....	21
5.1.2	edg-mkgridmap.....	21
5.1.3	edg-utils-system and edg-fetch-crl	21
5.2	INSTALLATION PRE-REQUISITES	22
5.3	SECURITY UTILITIES INSTALLATION	22
5.4	SECURITY UTILITIES CONFIGURATION	22
6	INFORMATION AND MONITORING SYSTEM (R-GMA)	25
6.1	SERVICE OVERVIEW.....	25
6.1.1	R-GMA Server	25
6.1.2	R-GMA Clients.....	25
6.1.3	R-GMA deployment modules	26
6.1.4	R-GMA Deployment strategy.....	27
6.2	R-GMA SERVER DEPLOYMENT MODULE.....	28
6.3	R-GMA CLIENT DEPLOYMENT MODULE	28
6.4	R-GMA SERVICETOOL DEPLOYMENT MODULE	28
6.4.1	Service overview.....	28
6.4.2	Installation Pre-requisites	29
6.4.3	R-GMA servicetool installation	29
6.4.4	R-GMA Servicetool Configuration	30
6.4.5	Configuration Walk-Through.....	36
6.5	R-GMA GADGETIN (GIN) DEPLOYMENT MODULE	38
6.5.1	Service Overview.....	38
6.5.2	Installation Pre-requisites	38
6.5.3	R-GMA GadgetIN installation	38
6.5.4	R-GMA GadgetIN Configuration.....	39
6.5.5	Configuration Walk-Through.....	40

7	SERVICE DISCOVERY (SD)	42
7.1	SERVICE OVERVIEW.....	42
7.2	INSTALLATION PRE-REQUISITES	42
7.2.1	Java JRE/JDK.....	42
7.3	SERVICE DISCOVERY INSTALLATION.....	42
7.4	SERVICE DISCOVERY CONFIGURATION	43
7.4.1	Configuration Walk-Through.....	47
8	VOMS SERVER AND ADMINISTRATION TOOLS	49
8.1	SERVICE OVERVIEW.....	49
8.2	INSTALLATION PRE-REQUISITES	49
8.2.1	Security Settings.....	Error! Bookmark not defined.
8.2.2	Java JRE/JDK.....	Error! Bookmark not defined.
8.2.3	Oracle	Error! Bookmark not defined.
8.2.4	MySQL.....	Error! Bookmark not defined.
8.3	VOMS SERVER INSTALLATION VIA APT	ERROR! BOOKMARK NOT DEFINED.
8.4	VOMS SERVER CONFIGURATION	ERROR! BOOKMARK NOT DEFINED.
8.4.1	Configuration Walk-Through.....	Error! Bookmark not defined.
8.5	VOMS SERVER CONFIGURATION OPTIONS	ERROR! BOOKMARK NOT DEFINED.
8.5.1	Configuring/Updating all VOs	Error! Bookmark not defined.
8.5.2	Starting all VOs.....	Error! Bookmark not defined.
8.5.3	Stopping all VOs.....	Error! Bookmark not defined.
8.5.4	Configuring/Adding/Updating a single VO.....	Error! Bookmark not defined.
8.5.5	Removing a single VO (keeping the database).....	Error! Bookmark not defined.
8.5.6	Removing a single VO (dropping the database)	Error! Bookmark not defined.
8.5.7	Starting a single VO.....	Error! Bookmark not defined.
8.5.8	Stopping a single VO.....	Error! Bookmark not defined.
8.5.9	Verifying your configuration.....	Error! Bookmark not defined.
8.5.10	Checking the status of the VOMS server	Error! Bookmark not defined.
9	LOGGING AND BOOKKEEPING SERVER	50
9.1	SERVICE OVERVIEW.....	50
9.2	INSTALLATION PRE-REQUISITES	50
9.2.1	Security Settings.....	50
9.3	JAVA JRE/JDK.....	51
9.4	LOGGING AND BOOKKEEPING SERVER INSTALLATION.....	51
9.5	LOGGING AND BOOKEEPING SERVER CONFIGURATION.....	51
9.6	LOGGING AND BOOKKEEPING CONFIGURATION WALKTHROUGH.....	54
9.7	MANAGING THE LB SERVICES.....	55
9.8	STARTING THE LB SERVICES AT BOOT	56
9.9	PUBLISHING LB SERVICES TO R-GMA.....	56
10	WORKLOAD MANAGER	57
10.1	SERVICE OVERVIEW.....	57
10.2	INSTALLATION PRE-REQUISITES	57
10.2.1	Security Settings.....	57
10.2.2	Java JRE/JDK.....	57
10.2.3	WNS and the Information Systems	57
10.2.4	Apache httpd and mod_ssl.....	58
10.3	WORKLOAD MANAGER SYSTEM INSTALLATION	58
10.4	WORKLOAD MANAGEMENT SYSTEM CONFIGURATION.....	58
10.5	WORKLOAD MANAGEMENT SYSTEM CONFIGURATION WALKTHROUGH	64
10.6	MANAGING THE WMS SERVICES	66
10.7	STARTING THE WMS SERVICES AT BOOT.....	67
10.8	PUBLISHING WMS SERVICES TO R-GMA.....	67
11	THE TORQUE RESOURCE MANAGER	68

11.1	SERVICE OVERVIEW.....	68
11.1.1	<i>TORQUE Server Overview</i>	68
11.1.2	<i>TORQUE Client Overview</i>	68
11.2	INSTALLATION PRE-REQUISITES	68
11.3	TORQUE SERVER.....	68
11.3.1	<i>TORQUE Server Installation</i>	68
11.3.2	<i>TORQUE Server Service Configuration</i>	69
11.3.3	<i>TORQUE Server Configuration Walkthrough</i>	75
11.3.4	<i>Managing the TORQUE Server Service</i>	76
11.3.5	<i>Publishing Torque Services to R-GMA</i>	76
11.4	TORQUE CLIENT	77
11.4.1	<i>TORQUE Client Installation</i>	77
11.4.2	<i>TORQUE Client Configuration</i>	77
11.4.3	<i>TORQUE Client Configuration Walkthrough</i>	78
11.4.4	<i>Managing the TORQUE Client</i>	79
12	COMPUTING ELEMENT	80
12.1	SERVICE OVERVIEW.....	80
12.2	INSTALLATION PRE-REQUISITES	80
12.2.1	<i>Security Settings</i>	80
12.2.2	<i>Java JRE/JDK</i>	81
12.2.3	<i>Resource Management System</i>	81
12.3	COMPUTING ELEMENT SERVICE INSTALLATION	81
12.4	COMPUTING ELEMENT SERVICE CONFIGURATION	82
12.5	COMPUTING ELEMENT CONFIGURATION WALKTHROUGH.....	87
12.6	MANAGING THE CE SERVICES	88
12.7	STARTING THE CE SERVICES AT BOOT	88
12.7.1	<i>Publishing CE Services to R-GMA</i>	89
12.8	WORKSPACE SERVICE TECH-PREVIEW	89
13	DGAS	90
13.1	SERVICE OVERVIEW.....	90
13.1.1	<i>DGAS Server Overview</i>	90
13.1.2	<i>DGAS Client Overview</i>	90
13.2	INSTALLATION PRE-REQUISITES	91
13.3	DGAS SERVER.....	91
13.3.1	<i>DGAS Server Installation</i>	91
13.3.2	<i>DGAS Server Service Configuration</i>	92
13.3.3	<i>DGAS Server Configuration Walkthrough</i>	97
13.3.4	<i>Managing the DGAS Server Service</i>	97
13.4	DGAS CLIENT.....	99
13.4.1	<i>DGAS Client Installation</i>	99
13.4.2	<i>DGAS Client Configuration</i>	99
13.4.3	<i>DGAS Client Configuration Walkthrough</i>	105
13.4.4	<i>Managing the DGAS Client</i>	105
14	WORKER NODE.....	106
14.1	SERVICE OVERVIEW.....	106
14.2	INSTALLATION PRE-REQUISITES	106
14.2.1	<i>Security Settings</i>	106
14.2.2	<i>Java JDK/JRE</i>	106
14.2.3	<i>Resource Management System</i>	106
14.3	WORKER NODE INSTALLATION.....	106
14.4	WORKER NODE CONFIGURATION.....	107
15	DATA CATALOGS (FIREMAN).....	110
15.1	SERVICE OVERVIEW.....	110

15.2	INSTALLATION PRE-REQUISITES	110
15.2.1	<i>Security Settings</i>	110
15.2.2	<i>Java JDK</i>	110
15.2.3	<i>Oracle InstantClient</i>	110
15.3	SINGLE CATALOG INSTALLATION	111
15.4	SINGLE CATALOG CONFIGURATION	111
15.5	SINGLE CATALOG CONFIGURATION WALKTHROUGH.....	117
15.6	PUBLISHING CATALOG SERVICES TO R-GMA	117
16	HYDRA	118
16.1	SERVICE OVERVIEW.....	118
16.2	INSTALLATION PRE-REQUISITES	118
16.2.1	<i>Security Settings</i>	118
16.2.2	<i>Java JDK</i>	118
16.3	HYDRA INSTALLATION.....	118
16.4	HYDRA CONFIGURATION.....	118
16.5	HYDRA CONFIGURATION WALKTHROUGH	122
16.6	STARTING THE HYDRA SERVICES AT BOOT.....	123
16.7	PUBLISHING HYDRA SERVICES TO R-GMA.....	123
17	GLITE I/O	124
17.1	GLITE I/O SERVER	124
17.1.1	<i>Service Overview</i>	124
17.1.2	<i>Installation pre-requisites</i>	124
17.1.3	<i>gLite I/O Server installation</i>	124
17.1.4	<i>gLite I/O Server Configuration</i>	125
17.1.5	<i>gLite I/O Server Configuration Walkthrough</i>	135
17.2	STARTING THE I/O SERVER AT BOOT.....	136
17.3	PUBLISHING I/O SERVER SERVICES TO R-GMA	136
17.4	CLIENT	136
17.4.1	<i>Service Overview</i>	136
17.4.2	<i>Installation pre-requisites</i>	137
17.4.3	<i>gLite I/O Client installation</i>	137
17.4.4	<i>gLite I/O Client Configuration</i>	137
18	AMGA SERVER	139
18.1	SERVICE OVERVIEW.....	139
18.2	INSTALLATION PRE-REQUISITES	139
18.2.1	<i>Security Settings</i>	139
18.2.2	<i>Java JDK/JRE</i>	139
18.2.3	<i>Database backend ODBC drivers</i>	139
18.2.4	<i>Database backend configuration</i>	139
18.3	AMGA SERVER INSTALLATION.....	140
18.4	AMGA SERVER CONFIGURATION.....	140
19	AMGA CLIENT	144
19.1	SERVICE OVERVIEW.....	144
19.2	INSTALLATION PRE-REQUISITES	144
19.2.1	<i>Security Settings</i>	144
19.3	AMGA CLIENT INSTALLATION	144
19.4	AMGA CLIENT CONFIGURATION	144
20	USER INTERFACE	148
20.1	SERVICE OVERVIEW.....	148
20.2	INSTALLATION PRE-REQUISITES	148
20.2.1	<i>Security Settings</i>	148
20.2.2	<i>Java JRE/JDK</i>	148

20.3	UI INSTALLATION.....	148
20.4	UI CONFIGURATION.....	149
20.5	CONFIGURATION FOR THE UI USERS	152
20.6	NOTE.....	153
21	THE GLITE FUNCTIONAL TEST SUITES.....	154
21.1	OVERVIEW.....	154
21.2	I/O TEST SUITE	154
21.2.1	<i>Test suite description</i>	154
21.2.2	<i>Installation Pre-requisites</i>	154
21.2.3	<i>Installation</i>	154
21.2.4	<i>Configuration</i>	154
21.2.5	<i>Execution</i>	154
21.2.6	<i>Test results</i>	155
21.3	CATALOG TEST SUITE.....	155
21.3.1	<i>Test suite description</i>	155
21.3.2	<i>Installation Pre-requisites</i>	155
21.3.3	<i>Installation</i>	155
21.3.4	<i>Configuration</i>	156
21.3.5	<i>Execution</i>	156
21.3.6	<i>Test results</i>	157
21.4	WMS TEST SUITE	157
21.4.1	<i>Test suite description</i>	157
21.4.2	<i>Installation Pre-requisites</i>	157
21.4.3	<i>Installation</i>	157
21.4.4	<i>Configuration</i>	158
21.4.5	<i>Execution</i>	158
21.4.6	<i>Test results</i>	158
21.5	WMS VALIDATION TEST SUITE	159
21.5.1	<i>Test suite description</i>	159
21.5.2	<i>Installation Pre-requisites</i>	159
21.5.3	<i>Installation</i>	159
21.5.4	<i>Configuration</i>	159
21.5.5	<i>Execution</i>	159
21.5.6	<i>Test results</i>	160
21.6	R-GMA TEST SUITE.....	160
21.6.1	<i>Test suite description</i>	160
21.6.2	<i>Installation Pre-requisites</i>	161
21.6.3	<i>Installation</i>	161
21.6.4	<i>Configuration</i>	161
21.6.5	<i>Execution</i>	161
21.6.6	<i>Test results</i>	161
22	SERVICE CONFIGURATION FILE EXAMPLE.....	162
23	SITE CONFIGURATION FILE EXAMPLE.....	167

1 INTRODUCTION

1.1 PURPOSE

From gLite-3.0 on the EGEE middleware contains components that are either configured primary by YAIM scripts, or by YAIM scripts that wrap gLite configuration scripts. In addition there are components that can be configured only via gLite configuration tools. Of course all services could be setup manually.

This document now describes how to install and configure middleware components for which either only a gLite configuration tool exists, or the tool has been wrapped for the gLite-3.0 inside YAIM. For these two categories this document will provide the necessary documentation needed for customized setup.

Some of these components are currently not part of the production release, but are supported on an as is basis.

1.2 TERMINOLOGY

Glossary

CE	Computing Element
LB	Logging and Bookkeeping
R-GMA	Relational Grid Monitoring Architecture
SC	Single Catalog
SD	Service Discovery
UI	User Interface
VOMS	Virtual Organization Membership Service
WMS	Workload Management System
WN	Worker Node

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 (in alphabetical order):

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

2.2 STANDARD DEPLOYMENT MODEL

Figure 1 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 (Logical File Names:Grid User Id:Storage URL) mapping of local files
- The Local Transfer Service and the File Transfer Agents provides file transfer/file placement service (FTS/FPS) and it is used for moving files.

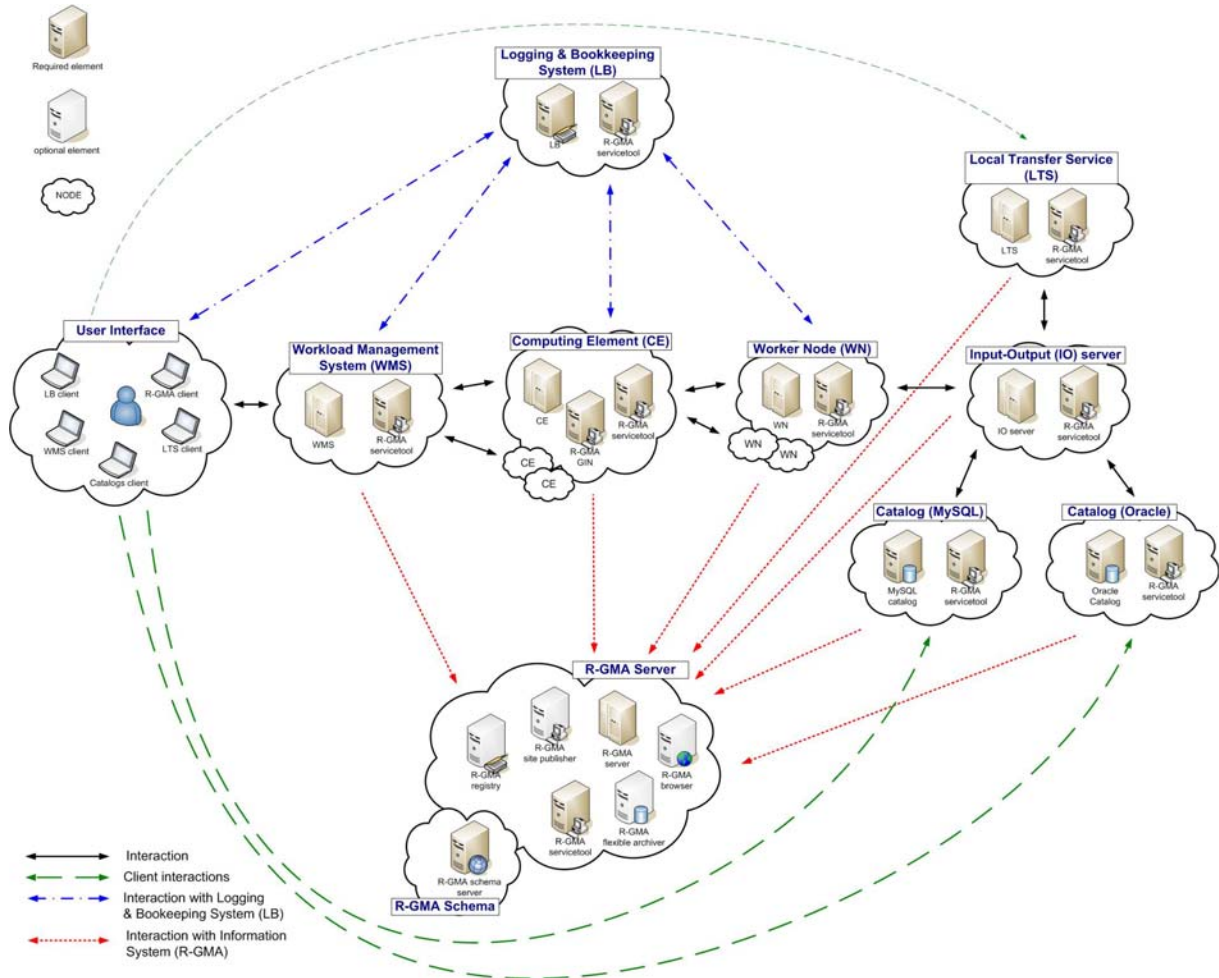


Figure 1: gLite Service Deployment Scenario

- Information and Monitoring Services
 - The R-GMA Server accepts R-GMA 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 1.2.

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 distributed by default using the APT and YUM package managers. More details on the apt/yum 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. 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 or glite-<service>-x.y.z-r.noarch.rpm	These meta RPMs contain the configuration files and scripts required by a particular service, such as ce, wms or rgma. There exist also meta packages for some composite services (eg: WMSLB).

In addition, a mechanism to load remote configuration files from URLs is provided. Refer to the Site Configuration section later in this chapter (4.2.5).

4.1 GLITE CONFIGURETION

4.1.1 YAIM configuration

Some of the services can be configured using the YAIM tool. To get detailed information on the configuration by YAIM refer please to:

<http://grid-deployment.web.cern.ch/grid-deployment/documentation/LCG2-Manual-Install>

4.1.2 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. The default setting is

`$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` or `glite-<service>` RPM.

All scripts have a number of command line switches to perform different actions. The usage instructions can be printed on screen with the command:

`glite-<service>-config.py --help`

The configuration steps for all services and clients, except the User Interface, are executed by running the command:

```
glite-<service>-config.py --configure
```

The services and daemons are started and stopped with:

```
glite-<service>-config.py --start
```

```
glite-<service>-config.py --stop
```

The status of the services and daemons can be verified with:

```
glite-<service>-config.py --status
```

The status switch causes a few status lines to be printed on screen and return 0 if all services are running and 1 if at least one service is not running.

Individual scripts may have additional options.

The User Interface script does not have a --configure switch. Running the command

```
glite-ui-config.py
```

by itself configures the user interface and its various clients and tools.

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		
<code>site.config.url</code>		The URL of the Site Configuration file for this node. The values defined in the Site Configuration file are applied first and are be overridden by values specified in the local configuration files. Leave this parameter empty or remove it to use local configuration only.
Advanced Parameters		
<code>GLITE_LOCATION</code>	<code>/opt/glite</code>	
<code>GLITE_LOCATION_VAR</code>	<code>/var/glite</code>	
<code>GLITE_LOCATION_LOG</code>	<code>/var/log/glite</code>	
<code>GLITE_LOCATION_TMP</code>	<code>/tmp/glite</code>	
<code>GLOBUS_LOCATION</code>	<code>/opt/globus</code>	Environment variable pointing to the Globus package.
<code>EDG_LOCATION</code> [New in gLite 3.0]	<code>/opt/edg</code>	Environment variable pointing to the location of EDG specific software.
<code>GPT_LOCATION</code>	<code>/opt/gpt</code>	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/gridmapfile	Location of the grid mapfile
host.gridmap.dir	/etc/grid-security/gridmapdir	The location of the account lease information for dynamic allocation
host.groupmapfile	/etc/grid-security/groupmapfile	Location of the groupmapfile
host.groupmap.dir	/etc/grid-security/groupmapdir	The location of the group lease information for dynamic allocation
X509_VOMS_DIR	/etc/grid-security/vomsdir	The directory when VOMS Server certificates are stored. [Example=/etc/grid-security/vomsdir][Type='string']
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
modify.user.env	True	If this parameter is set to true, the user environment files are modified to source the glite_setenv.sh script. Otherwise no modification is done. Possible values are true or false. Default is true
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 VO List File

gLite 1.5 introduced a new method for configuring VOs. VO-specific parameters are encapsulated in a new <vo> tag and all VOs can be listed in a single file used by all modules on a node or all nodes in the same site configuration structure (see the following paragraph 4.3.4 for more information about using site configuration).

The usage of the new VO configuration method is explained in details in the VO Configuration Guide document that can be found at:

http://glite.web.cern.ch/glite/packages/R1.5/R20051130/doc/VO_Configuration_Guide.doc

4.2.5 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-global.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 a comma-separated list of host names of the target nodes where the service must be deployed, for example:

```
<node name="host1.domain.com, host2.domain.com, ..., hostN.domain.com">
...
</node>
```

where hostX.domain.com must be the output of the command `hostname -f` on the target 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 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 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 overrides 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" xpointer="//siteconfig" />
```

where the value of the `href` attribute is a file path relative to the location of the master file or a fully qualified URL pointing the file. The `glite-xxx.cfg.xml` file must have the document root:

```
<siteconfig>
```

All children of the `<siteconfig>` root in the referenced file are 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.

4.2.6 Internal Configuration

The configuration scripts and files described above represent the common configuration interfaces of all gLite services. However, since the gLite middleware is a combination of various old and new services, not all services can natively use the common configuration

model. Many services come with their 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.

4.2.7 User environment

When any gLite configuration script 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 logs in various ways (interactive, non-interactive or script).

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

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

4.2.8 Default Environment Variables

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

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

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

4.2.9 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 a number of utilities and 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 gLite (not LCG) modules. However, it is not normally installed manually by itself, but automatically as part of the other modules.

5.1.1 CA Certificates

In contrary to the previous gLite releases (1.x) the CA Certificate are not installed together with the gLite security utilities and a new metapackage (lcg-CA) should be installed manually in order to install the CA certificates. See section 5.3.

5.1.2 edg-mkgridmap

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

/opt/edg/sbin

and

\$GLITE_LOCATION/etc

The script is run automatically for all services that need it by setting the install.mkgridmap.cron parameter to true in the service configuration file. It can also be run manually of course.

The Security Utilities module configuration script also installs a crontab file in /etc/cron.d that executes the wrapper mkgridmap.py script every night 4 hours by default. The wrapper script calls the edg-mkgridmap script and performs some additional check. The installation of this cron job and the execution of the mkgridmap.py script during the configuration are optional and can be enabled using the provided 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 edg-utils-system and edg-fetch-crl

The edg-utils-system replaces the fetch-crl rpm, but contains a revised script used to update the CA Certificate Revocation Lists compatible with LCG (edg-fetch-crl). This script is installed in:

/opt/edg/sbin

The Security Utilities module configuration script installs a crontab file in /etc/cron.d that executes the glite-fetch-crl every six hours. In addition, a random delay can be added to the scheduled time to help preventing peak loads on the CEs web servers. The CRLs are

installed in the same directory as the CA certificates, `/etc/grid-security/certificates`. The output and error messages are sent to the log file `/var/log/glite/glite-fetch-crl-cron.log`.

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 normally not installed by itself, but as part of another module. However, in case the functionality provided by this module is required separately from the other gLite modules, it is possible to install it as follows:

1. Installation via APT

Install APT if not yet installed following the instructions at

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

and install the gLite Security Utility and CA certificates by executing

```
apt-get install glite-security-utils-config lcg-CA
```

or

```
yum install glite-security-utils-config lcg-CA
```

[New in 3.0] CA Certificates should be installed manually

2. Installation via gLite installer scripts

Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

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

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

The `fetch.crl` and `mkgridmap` cron jobs are installed in `/etc/cron.d` (depending on the selected options).

The security utils configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/glite-security-utils-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.

5.4 SECURITY UTILITIES CONFIGURATION

As the module is normally not installed manually by itself, but automatically as part of the other modules, you will only need to do steps 1 to 3. Step 4 and 5 are only required if you have installed the module standalone yourself – otherwise these steps are executed automatically by the module that uses the security utils module.

1. Change to the configuration directory:

```
cd /opt/glite/etc/config
```

2. Copy the configuration file templates from the templates directory
`cp templates/* .`
3. Customize the configuration files by replacing the 'changeme' value in all user-defined parameters with the proper value:
 - The file `glite-global.cfg.xml` contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.
 - The file `glite-security-utils.cfg.xml` contains the security utils related configuration values. Table 2 It shows the list of parameters that can be set.

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. Change to the script directory:
`cd /opt/glite/etc/config/scripts`
5. Configure the security utils by executing the security utils configuration script:
`./glite-rgma-server-config.py`

Parameter	Default value	Description
User-defined Parameters		
<code>cron.mailto</code>		E-mail address to which the stderr of the installed cron jobs is sent
Advanced Parameters		
<code>glite.installer.verbose</code>	True	Produce verbose output when running the script
<code>glite.installer.checkcerts</code>	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 <code>host.certificate.file</code> and <code>host.key.file</code>
<code>fetch-crl.cron.tab</code>	00 */6 * * *	The cron tab to use for the fetch-crl cron job.
<code>install.fetch-crl.cron</code>	True	Install the glite-fetch-crl cron job. Possible values are 'true' (install the cron job) or 'false' (do not install the cron job)
<code>fetch-crl.script</code>	<code>\${EDG_LOCATION}/s bin/edg-fetch-crl</code>	The full path of the fetch crl script script.
<code>Fetch-crl.cron.random.delay</code>	True	This property can be set to true to introduce a delay between 1 and 30 minutes

		(modulo 60) to the minutes part of the value of fetch-crl.cron.tab. The delay is randomly generated everytime the configuration script is run and then added to the cron tab. This delay helps preventing peak loads on the CA web servers in case too many nodes use the same schedule
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)
mkgridmap.cron.tab	15 */4 * * *	The cron tab to use for the mkgridmap cron job
mkgridmap.script	/opt/edg/sbin/edg-mkgridmap	The full path of the mkgridmap script.
mkgridmap.conf	\${GLITE_LOCATION}/etc/glite-mkgridmap.conf	The full path of the mkgridmap config file

System Parameters

Table 2: Security Utilities Configuration Parameters

6 INFORMATION AND MONITORING SYSTEM (R-GMA)

The information system is used to store and publish information about the different parts of your grid (services, sites etc.) and to query this information by interested users and services via the service discovery. The installation and configuration of the gLite information system R-GMA is described in this chapter together with the installation of its specific information publisher and consumers. The installation of the service discovery (that can be used with different information systems) is described in Chapter 7.

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 functionality of the R-GMA system can be logically split in a server part (which in turn consists of several parts) and several clients:

6.1.1 R-GMA Server

The R-GMA server is the server part of the R-GMA infrastructure that is used by the different producers and consumers. 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 component that is used for the schema 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.

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.

6.1.2 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 that is coming from the R-GMA site-publisher, servicetool and GIN constantly available. By default the GLUE tables 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.

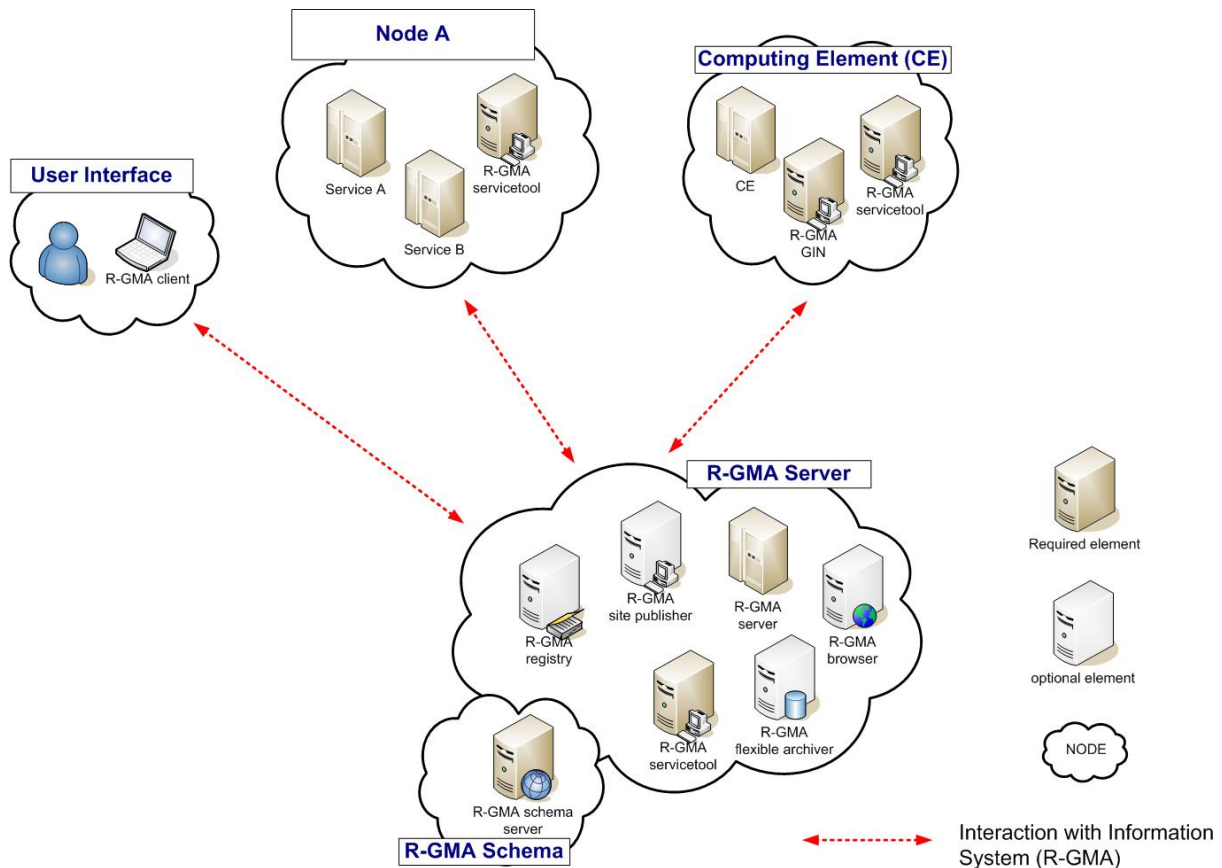


Figure 2 R-GMA components

6.1.3 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 1

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

<i>Deployment module</i>	<i>Contains</i>	<i>Used / included by</i>
R-GMA server	R-GMA server R-GMA registry server R-GMA schema server R-GMA browser R-GMA site publisher R-GMA archiver R-GMA servicetool	
R-GMA client	RGMA client APIs	Service Discovery (SD) (Chapter 7) Worker Node (WN) (Chapter 13) User Interface (UI) (Chapter 18)
R-GMA servicetool	R-GMA servicetool	R-GMA server VOMS Server (<i>Chapter 8</i>) Logging & Bookkeeping (<i>Chapter 9</i>) Workload Management System (<i>Chapter 10</i>) Torque Server (<i>Chapter 11</i>) Computing Element (<i>Chapter 12</i>) Data Catalog (<i>Chapter 15</i>) File Transfer Service (<i>Chapter Error! Reference source not found.</i>) File Transfer Agents (<i>Chapter Error! Reference source not found.</i>) Hydra (<i>Chapter 16</i>) I/O-Server (<i>Chapter 17</i>)
R-GMA GIN	R-GMA GadgetIN	Computing Element (<i>Chapter 12</i>)

Table 3: R-GMA deployment modules

6.1.4 R-GMA Deployment strategy

In order to use the information system R-GMA, you need to first setup the R-GMA server infrastructure and then setup the necessary clients that publish the information to the Information system as well as query the information system.

To do this, 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. It is recommended to have at least two registries per site.
- There must be one site publisher per site.
- You can choose to enable/disable the archiver. If you want to have your data archived after the expiration time you will need a data archiver. It is recommended to have at least two archivers per site.

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 R-GMA SERVER DEPLOYMENT MODULE

In gLite R3.0 the R-GMA server is natively installed and configured by YAIM

6.3 R-GMA CLIENT DEPLOYMENT MODULE

In gLite R3.0 the R-GMA client is natively installed and configured by YAIM.

6.4 R-GMA SERVICETOOL DEPLOYMENT MODULE

6.4.1 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 and configured automatically with other modules and doesn't need to be installed or configured independently.

You can publish both gLite and non-gLite services to R-GMA. If you publish gLite services, the R-GMA servicetool is installed together with the corresponding service. If you want to publish a non-gLite service, you have to install the R-GMA servicetool deployment module separately it yourself.

The published service information contains several information about the service according to the GLUE standard like service name, service type or status.

6.4.2 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.

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

6.4.2.2 Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA servicetool. This release requires v. 1.4.2 revision 08. 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.

6.4.3 R-GMA servicetool installation

If you install the R-GMA servicetool as part of another deployment module (e.g. the single catalog), the R-GMA servicetool is installed automatically and you can continue with the configuration description in the next section. Otherwise, the R-GMA servicetool can be installed in the following ways:

a) Installation via APT

Install APT if not yet installed following the instructions at

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

and install the gLite R-GMA servicetool by executing

```
apt-get install glite-rgma-servicetool-config
```

or

```
yum install glite-rgma-servicetool-config
```

b) Installation via gLite installer scripts

Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

This will install the following deployment modules:

- R-GMA servicetool
- Security utils (see chapter 5 for details)

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
```

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.

6.4.4 R-GMA Servicetool Configuration

6.4.4.1 Servicetool configuration when installing other gLite services

If you install the R-GMA servicetool as part of another deployment module (e.g. the single catalog), the R-GMA servicetool is configured automatically together with the other deployment module. In this case you only need to do provide the necessary configuration information. The actual configuration is done via the other gLite deployment module.

1. Change to the configuration directory:

```
cd /opt/glite/etc/config
```

2. Copy the configuration file templates from the templates directory:

```
cp templates/* .
```

3. Customize the configuration files by replacing the 'changeme' value in all user defined parameters with the proper value:

- The file *glite-global.cfg.xml* contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.
- The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. **Error! Reference source not found.** shows the configuration values that can be set.
- The file *glite-rgma-servicetool.cfg.xml* contains the R-GMA client specific configuration values. Table 4 shows the configuration values that can be set.
- The file *glite-security-utils.cfg.xml* contains the security utils specific configuration values. Refer to Table 2 for the list of parameters and chapter 5 for the description of the security utils.

Parameter	Default value	Description
User-defined Parameters		
rgma.servicetool.siteId	\${HOSTNAME}	Unique Id of 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] [Type: 'string']

			This parameter obsoletes the parameter: rgma.servicetool.sitename
Advanced Parameters			
glite.installer.verbose	True		Enable verbose output. [Type: 'boolean'] <i>Example : true</i>
rgma.servicetool.activate	True		Turn on/off servicetool for the node. [Type: 'boolean'] <i>Example : true</i>
rgma.servicetool.enable	True		Enable this service to be published to R-GMA. This parameter can be also specified separately per servicetool instance in your service configuration file. The value defined here is the fallback value if no value is defined in the individual servicetool instance. [Example: 'true'] [Type: 'boolean']
rgma.servicetool.name	<empty string>		Human-readable name for the service. Need not be globally unique. If value is empty/not specified, the serviceld is taken as the service name. This parameter can be also specified separately per servicetool instance in your service configuration file. The value defined here is the fallback value if no value is defined in the individual servicetool instance. [Example: Testservice to interface to my application] [Type: 'String']
rgma.servicetool.url_wSDL	description	not available	URL of a WSDL document for the service. Put 'not available' if no wsdl url is available. This parameter can be also specified separately per servicetool instance in your service configuration file. The value defined here is the fallback value if no value is defined in the individual servicetool instance. <i>Example:</i> <i>http://example.rl.uk/service?WSDL</i> [Type: 'string']
rgma.servicetool.url_semantics		not available	URL of a document containing a detailed description of the service and how it should be used. Put 'not available' if not url is available. This parameter can be also specified separately per servicetool instance in your service configuration file. The value defined here is the fallback

rgma.servicetool.vo	<p>value if no value is defined in the individual servicetool instance. <i>Example:</i> http://example.rl.ac.uk/service/semantics.html [Type: 'string']</p> <p>List of VOs that this service is considered part of. This parameter can be also specified separately per servicetool instance in your service configuration file. Optional parameter - you can specify one or several or it can be left empty or be removed. The value defined here is the fallback value if no value is defined in the individual servicetool instance. <i>Example: EGEE</i> [Type: 'string']</p>
rgma.servicetool.associatedService	<p>List of service names that this service is associated with. This parameter can be also specified separately per servicetool instance in your service configuration file. The value defined here is the fallback value if no value is defined in the individual servicetool instance. Optional parameter - you can specify one or several or it can be left empty or be removed. <i>Example:</i> YOURhostname_YOURvname_YOURservicetype [Type: 'string']"></p>
rgma.servicetool.param	<p>List of extra parameters for the service to be published. The structure for each entry is key=value. This parameter can be also specified separately per servicetool instance in your service configuration file. The value defined here is the fallback value if no value is defined in the individual servicetool instance. Optional parameter - you can specify one or several or it can be left empty or be removed. <i>Example: yourkey=yourvalue</i> [Type: 'string']</p>

System Parameters

Table 4: R-GMA servicetool configuration parameters

If the *rgma.servicetool.activate* parameter is set to false, the servicetool daemon is not started and no service publishing occurs. This can be used on gLite nodes in case the R-GMA Server is not used.

It is also possible to prevent individual services from being published by setting the `rgma.servicetool.enable` parameter to false in the service instance.

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.

6.4.4.2 Standalone servicetool configuration when for non gLite services

If you install the R-GMA servicetool as part of another deployment module (e.g. the single catalog), the R-GMA servicetool is configured automatically together with the other deployment module. In this case you only need to do provide the necessary configuration information. The actual configuration is done via the other gLite deployment module.

1. Change to the configuration directory:

```
cd /opt/glite/etc/config
```

2. Copy the configuration file templates from the templates directory:

```
cp templates/* .
```

3. Customize the configuration files by replacing the 'changeme' value in all user defined parameters with the proper value:

- The file `glite-global.cfg.xml` contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.
- The file `glite-rgma-common.cfg.xml` contains the common R-GMA configuration values. **Error! Reference source not found.** shows the configuration values that can be set.
- The file `glite-rgma-servicetool.cfg.xml` contains the R-GMA client specific configuration values. Table 4 shows the configuration values that can be set.
- The file `glite-security-utils.cfg.xml` contains the security utils specific configuration values. Refer to Table 2 for the list of parameters and chapter 5 for the description of the security utils.
- The file `glite-rgma-servicetool-externalServices.cfg.xml` contains a template for the configuration of a service to be published via the rgma-servicetool. Table 5 contains the set of parameters that can be set for each service. Customize the configuration files by replacing the 'changeme' value in all user defined parameters with the proper value. If you want to publish more than one non-gLite service, create additional servicetools instance for each service to be published and modify them accordingly. The instance names must be unique.

Parameter	Default value	Description
Mandatory parameters		
<code>rgma.servicetool.enable</code>	True	Publish this service via the R-GMA servicetool. If this variable set to false the other values below are not taken into account. <i>Example: true</i>
<code>rgma.servicetool.service_type</code>		The type of the service: <ul style="list-style-type: none"> • Unique string in reversed domain name structure. • For all gLite software the structure is

rgma.servicetool.name

org.glite.<subsystem>.<component>
where

- <subsystem> is the name of the subsystem
 - <component> is the name of the individual component
- For all external software corresponding prefixes can be chosen (e.g. following their package domain names).

Example:

org.glite.data.FiremanCatalog

The name of the service:

- Globally unique string including hostname and VO name (if available).
- For all gLite software the structure is

<hostname>_<VOname>_<service-type>
where

- <hostname> is the fully qualified DNS hostname (e.g. lxb1212.cern.ch)
- <VO-name> is the name of the VO the service is serving (only specified if VO specific service)
- <service-type> is the string used for the 'Service Type' above.

Examples:

lxb1212.cern.ch_org.glite.data.io-server
or

lxb1270.cern.ch_EGEE_org.glite.rgma.RgmaServer

rgma.servicetool.
service_version

The version of the service in the form 'major.minor.patch'. For the moment we recommend to use the version of the deployment scripts.

Example: 1.2.3

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

Optional parameters

rgma.servicetool. url_endpoint		<p>URI to contact the service at. This is a service specific string. If no URL is available a string 'not available' should be set.</p> <p><i>Example:</i> http://myService/homepage</p>
rgma.servicetool. publish_interval	3600	<p>How often to publish the service details (like endpoint, version etc). in seconds. <i>Example: 3600</i></p>
rgma.servicetool. status_interval	30	<p>How often check and publish service status (running/not running) in seconds. <i>Example: 30</i></p>
rgma.servicetool.url_wsdl		<p>URL of a WSDL document for the service. This is a service specific string. If no URL is available a string 'not available' should be set.</p> <p><i>Example:</i> https://{\$HOSTNAME}:8443/EGEE/gLite-data-catalog-service-meta/services/MetadataCatalog?wsdl</p>
rgma.servicetool. url_semantics		<p>URL of a document containing a detailed description of the service and how it should be used. This is a service specific string. If no URL is available a string 'not available' should be set.</p> <p><i>Example:</i> http://egee-jra1-dm.web.cern.ch/egee-jra1-dm/</p>
rgma.servicetool.vo		<p>List of VOs that this service is considered part of. Optional parameter - you can specify one or several or it can be left empty or be removed.</p> <p><i>Example: EGEE</i> [Type: 'string']</p>
rgma.servicetool. associatedService		<p>List of service names that this service is associated with. Optional parameter - you can specify one or several or it can be left empty or be removed. <i>Example:</i> <i>YOURhostname_ YOURvname_ YO URservicetype</i> [Type: 'string']"></p>
rgma.servicetool.param		<p>List of extra parameters for the service to be published. The structure for each entry is key=value. Optional parameter - you can specify one or several or it can be left empty or be removed.</p>

Example: yourkey=yourvalue
 [Type: 'string']

Table 5: R-GMA servicetool configuration parameters for a service to be published via the R-GMA servicetool

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. Change to the script directory
`cd /opt/glite/etc/config/scripts/`
5. Add the service specific configuration values (for each service) to the R-GMA servicetool. To do this, run the R-GMA servicetool configuration script
`./glite-rgma-servicetool-config.py --addExternalServices`
 All services configured in the external services file *glite-rgma-servicetool-externalServices.cfg.xml* be published
6. Configure the R-GMA servicetool by running the R-GMA servicetool configuration script with the option `--configure`
`./glite-rgma-servicetool-config.py --configure`
7. Start the R-GMA servicetool:
`./glite-rgma-servicetool-config.py --start`
 Check if any error message is displayed and if necessary fix the parameters values and restart the script.
8. Verify that the installation is successful by running
`./glite-rgma-servicetool-config.py --status`

The R-GMA Servicetool is completely configured.

6.4.4.3 Checking which services the R-GMA servicetool is publishing

If you want to see which services will be/are published by the rgma-servicetool, you can run the rgma-servicetool configuration script with the option `-c`:

```
./glite-rgma-servicetool-config.py -c
```

This will print – besides the general settings of R-GMA – also the list of information that will be published by the rgma-servicetool.

6.4.4.4 Removing published services

A new option has been added to the configuration script. You can now also remove published services from the local servicetool cache:

```
./glite-rgma-server-config.py --removeService=serviceName
```

This command stops servicetool from publishing the service, but it doesn't remove the service publication from the R-GMA Server. The service will stop appearing in R-GMA when the expiration period is reached. The configuration files must also be modified to remove the unwanted service, otherwise it would be reinstalled next time the script is run. If you want to stop publishing a service temporarily is preferable to set its `rgma.servicetool.enable` parameter to false in the service configuration file.

6.4.5 Configuration Walk-Through

After installing the gLite R-GMA servicetool module as described in this chapter, proceed as follows.

Step 1: Install the Java run-time libraries (obtained from the Sun Java web site):

```
rpm -ivh j2re-1_4_2_08-linux-i586.rpm
```

Step 2: Change to the configuration directory:

```
cd /opt/glite/etc/config
```

Step 3: Copy the configuration templates from the templates directory:

```
cp templates/* .
```

Step 4: Customize the configuration files by replacing the changeme values with appropriate parameters according to the following table.

File name: glite-rgma-servicetool.cfg.xml

rgma.servicetool.siteId	<the unique id of the site in which your service is running (see the chapter Error! Reference source not found. about the R-GMA server and site-publisher)>
-------------------------	--

File name: glite-rgma-common.cfg.xml

rgma.server.hostname	<your R-GMA Server>
rgma.schema.hostname	<your R-GMA Schema Server>
rgma.registry.hostname	<your R-GMA Registry Server>

File name: glite-security-utils.cfg.xml

cron.mailto	<your own address>
-------------	--------------------

The following steps are only necessary if you have installed the R-GMA servicetool standalone and not as part of another module (e.g. the WN or UI) that uses the R-GMA client. Otherwise, these steps are handled by the configuration of the other module that uses the R-GMA servicetool.

Step 5: Change to the scripts directory:

```
cd /opt/glite/etc/config/scripts
```

Step 6: Execute the glite-rgma-servicetool-config.py script:

```
./glite-rgma-servicetool-config --configure
```

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

The gLite RGMA servicetool configuration was successfully completed

6.5 R-GMA GADGETIN (GIN) DEPLOYMENT MODULE

6.5.1 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.

6.5.2 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.

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

6.5.2.2 Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA GadgetIN. This release requires v. 1.4.2 revision 08. 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.

6.5.3 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 R-GMA GadgetIN can be installed in the following ways:

a) Installation via APT

Install APT if not yet installed following the instructions at

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

and install the gLite R-GMA GadgetIN by executing

```
apt-get install glite-rgma-gin-config
```

or

```
yum install glite-rgma-gin-config
```

b) Installation via gLite installer scripts

Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

This will install the following deployment modules:

- R-GMA GIN
- Security utils (see chapter 5 for details)

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
```

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.

6.5.4 R-GMA GadgetIN Configuration

If you install the R-GMA GIN as part of another deployment module (e.g. the CE), the R-GMA GIN is configured automatically together with the other deployment module. In this case you only need to do steps 1 to 3 before executing the configuration script of the other deployment module.

1. Change to the configuration directory:

```
cd /opt/glite/etc/config
```

2. Copy the configuration file templates from the templates directory:

```
cp templates/* .
```

3. Customize the configuration files by replacing the 'changeme' value in all user defined parameters with the proper value:

- The file *glite-global.cfg.xml* contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.
- The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. **Error! Reference source not found.** shows the configuration values that can be set.
- The file *glite-rgma-gin.cfg.xml* contains the R-GMA client specific configuration values. Table 6 shows the configuration values that can be set.
- The file *glite-security-utils.cfg.xml* contains the security utils specific configuration values. Refer to Table 2 for the list of parameters and chapter 5 for the description of the security utils.

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

The following steps are only necessary if you have installed GIN separately and not as part of another gLite deployment module. Otherwise the other deployment module will take care about these steps:

4. Change to the script directory:

```
cd /opt/glite/etc/config/scripts
```

5. Configure the R-GMA GIN by executing the R-GMA GIN configuration script:

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

Running the configuration script will automatically configure the security utils as well so there is no need to run the configuration script of the security utils in addition. Check if any error message is displayed and if necessary fix the parameters values and restart the script. If the configuration is successful you should see at the end the message:

The gLite R-GMA GIN was successfully configured.

1. Start the R-GMA GIN:

```
./glite-rgma-gin-config.py --start
```

Check if any error message is displayed and if necessary fix the parameters values and restart the script.

2. Verify that the installation is successful by running

```
./glite-rgma-gin-config.py --status
```

The R-GMA GIN is completely configured and running.

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 <i>Example: no</i>
rgma.gin.run_fmon_provider		Run fmon backend (yes no). This is used by LCG for gridice. <i>Example: no</i>
rgma.gin.run_ce_provider		Run ce backend (yes no). <i>Example: yes</i>
Advanced Parameters		
glite.installer.verbose	True	Enable verbose output. <i>Example : true</i>
System Parameters		

Table 6: R-GMA GadgetIN configuration parameters

6.5.5 Configuration Walk-Through

After installing the gLite R-GMA GIN module as described in this chapter, proceed as follows.

Step 1: Install the Java run-time libraries (obtained from the Sun Java web site):

```
rpm -ivh j2re-1_4_2_08-linux-i586.rpm
```

Step 2: Change to the configuration directory:

```
cd /opt/glite/etc/config
```

Step 3: Copy the configuration templates from the templates directory:

```
cp templates/* .
```

Step 4: Customize the configuration files by replacing the changeme values with appropriate parameters according to the following table.

File name: glite-rgma-gin.cfg.xml

rgma.gin.run_generic_info_provider	<yes if you want to run generic information provider (gip) backend (within LCG this comes with the ce and se), no otherwise>
rgma.gin.run_fmon_provider	<yes if you want to run fmon backend (this is used by LCG for gridice.), no otherwise>
rgma.gin.run_ce_provider	<yes if you want to run the ce backend, no otherwise>

File name: glite-rgma-common.cfg.xml

rgma.server.hostname	<your R-GMA Server>
rgma.schema.hostname	<your R-GMA Schema Server>
rgma.registry.hostname	<your R-GMA Registry Server>

File name: glite-security-utils.cfg.xml

cron.mailto	<your own address>
-------------	--------------------

The following steps are only necessary if you have installed the R-GMA GIN standalone and not as part of another module (e.g. the CE) that uses the R-GMA GIN. Otherwise, these steps are handled by the configuration of the other module that uses the R-GMA GIN.

Step 5: Change to the scripts directory:

```
cd /opt/glite/etc/config/scripts
```

Step 6: Execute the glite-rgma-gin-config.py script:

```
./glite-rgma-gin-config --configure
```

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

The gLite RGMA gin service configuration was successfully completed

7 SERVICE DISCOVERY (SD)

7.1 SERVICE OVERVIEW

The Service Discovery module is the counterpart to the information system. It allows the different gLite modules to discover the endpoint of other gLite modules they are interested in. The Service Discovery module can use several information systems

- R-GMA (see chapter 6),
- Berkeley Database Information Index (BDII)
- Files that contain the necessary information

or any combination of these systems to discover the corresponding services.

The gLite Service Discovery module is installed together with the gLite modules that are using Service Discovery – you do not need to install it separately.

The following modules presently use Service Discovery:

- User Interface (UI) (Chapter 18)
- Worker Node (WN) (Chapter 13)
- Data Transfer Agents (Chapter 18)

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 Java JRE/JDK

The Java JRE or JDK are required to run the Java API of the Service Discovery. 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.

7.3 SERVICE DISCOVERY INSTALLATION

Normally the Service Discovery is automatically installed as part of another deployment module (e.g. the User Interface) and you can continue with the configuration description in the next section.

If you want to use the service discovery based on the R-GMA information system, you will also have to install in addition the R-GMA client yourself (see chapter 6.3 for details) as this module is not installed together with the service discovery by default and the service discovery uses the R-GMA client to obtain the information from the R-GMA server.

If you want to install the service discovery standalone, the installation steps are:

a) Installation via APT

Install APT if not yet installed following the instructions at

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

and install the gLite service discovery by executing

```
apt-get install glite-service-discovery-config
```

or

```
yum install glite-service-discovery-config
```

b) *Installation via gLite installer scripts*

Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

This will install the following deployment modules:

- Service discovery

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

```
gLite in /opt/glite ($GLITE_LOCATION)
```

The gLite service discovery configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/serviceDiscovery.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.

7.4 SERVICE DISCOVERY CONFIGURATION

The service discovery is configured automatically together with the other deployment module that it was downloaded with and that uses Service Discovery. You will only need to adapt the configuration:

1. Change to the configuration directory:


```
cd /opt/glite/etc/config
```
2. Copy the configuration file templates from the templates directory


```
cp templates/* .
```
3. Customize the configuration files by replacing the 'changeme' value in all user-defined parameters with the proper value:
 - The file *glite-global.cfg.xml* contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.
 - The file *glite-service-discovery.cfg.xml* contains the Service Discovery specific configuration values. Table 7 shows the configuration values that can be set.
 - The file *glite-service-discovery-file-based-example.cfg.xml* is not used/loaded by the configuration file. It contains for the file based service discovery the full set of parameters that can be configured for each service entry as an example. You can use this file as a reference to copy paste entries in the individual file based service discovery entries. Normally all necessary entries exist already in the corresponding configuration files. Table 8 shows the corresponding list of configuration parameters that can be set.

Parameter	Default value	Description
User-defined parameters		
service-discovery.type		Service discovery implementation to be used.

Possible values are:

- *file*
use (static) file based service discovery
- *rgma*
use (dynamic) R-GMA based service discovery
- *bdii*
use (dynamic) BDII based service discovery

Several implementations can be specified that will be tried/used in the specified order.

[Type: string]

Example: file

service-discovery.site

Site name to be used to find a service nearby. This parameter must match the specified site name of the services that have to be discovered. Leave the parameter empty if you don't want to specify a site.

[Type: 'string']

Example: cern.ch

service-discovery.vo

Default VO to be used to find a friendly VO. Leave the parameter empty if you don't want to specify a default VO.

[Type: 'string']

Example: EGEE

Configuration for BDII based service discovery:

If you don't use BDII leave the parameter empty or remove it

service-
discovery.bdii.provider

Host and port of the BDII service for service discovery. Leave empty or remove parameter if you do not use BDII as information provider.

[Type: 'string']

Example:

lxb1386.cern.ch:2170

Advanced Parameters

glite.installer.verbose True

Enable verbose output.

[Type: 'boolean']

Example : true

System Parameters

Table 7: Service Discovery common configuration parameters

Parameter	Default value	Description
User-defined parameters		
service-discovery.file. service_name		The globally unique name of the service. The convention is 'service_host'_'vo_name'_'service_type'. [Type: 'string'] <i>Example:</i> <i>my.hostname.com_myVO_or_g.glite.FiremanCatalog</i>
service-discovery.file. url_endpoint		URL endpoint of the service. [Type: 'string'] <i>Example:</i> <i>http://my.hostname.com:8443/myVO/glite-data-catalog-service-fr/services/FiremanCatalog</i>
service-discovery.file. service_version		Service version in the form 'major.minor.patch' of the used service. [Type: 'string'] <i>Example:</i> 1.2.3
service-discovery.file. wsdl		URL for WSDL of the service. This parameter is optional. Remove it or leave it empty if you don't want to specify a sitename. <i>Example:</i> http://myhost:8443/myService/wsdl [Type:'string']
service-discovery.file. administration		URL for administration of the service. This parameter is optional. Remove it or leave it empty if you don't want to specify a sitename. <i>Example:</i> http://myhost:8443/myService/administration [Type: 'string']
service-discovery.file. site		Site name for this service. This parameter is optional. Remove it or leave it empty if you don't want to specify a sitename. <i>Example:</i> <i>host.site.org</i> [Type: 'string']
		List of supported vo for this service. You can specify zero,

Advanced Parameters

service-discovery.file.
service_type

one or several vo's. This parameter is optional. Remove it or leave it empty if you don't want to specify any vo.

Example: EGEE

[Type: 'string']"

service-discovery.file.
param

The service type of the used service. This must match the type used to publish the corresponding service. (see 'rgma.servicetool.service_type' for the corresponding service for R-GMA as information source)

[Type: 'string']

Example:

org.glite.FiremanCatalog

List of extra parameters for the service. You can specify zero, one or several entries. The structure for each entry is key=value. This parameter is optional. Remove it or leave it empty if you don't want to specify any extra parameter.

Example: param=value

[Type: 'string']

service-discovery.file.
associatedService

List of associated services. You can specify zero, one or several entries. This parameter is optional. Remove it or leave it empty if you don't want to specify any associated services.

Example:

MyAssociatedService

[Type: 'string']

System Parameters

Table 8: Service Discovery configuration parameters for file based information service

4. If you want to use *file based service discovery*, you will also need to configure the service file entries:

You will find the necessary configuration parameters in the configuration file of the service (e.g. for the File Transfer Client in the file `glite-file-transfer-service-client.cfg.xml`) that is using service discovery as separate `<instance service="service-discovery.file">` parameter lists. You will have to modify for each of these 'instance parameter list' the parameters. Table 8 shows the list of parameters for each service that has to be discovered via file based service discovery that you have to set accordingly.

5. If you want to use *R-GMA based service discovery*, you will also need to install and configure the R-GMA client. For the installation of the R-GMA client see chapter **Error! Reference source not found.**. In addition to the above mentioned configuration files, you will need to configure the R-GMA client by copying and configuring the R-GMA client onfiguration files *glite-rgma-common.cfg.xml* and *glite-rgma-client.cfg.xml* (see chapter **Error! Reference source not found.** for details).

Note: Step 1, 2, 3, 4 and 5 can also be performed by means of the remote site configuration file or a combination of local and remote configuration files

You do not need to run the configuration script as this is done automatically by the configuration script of the deployment module that uses service discovery.

7.4.1 Configuration Walk-Through

Normally this configuration script doesn't need to be run manually, since it is run by the service configuration scripts using service discovery.

If a manual configuration is required, the following steps can be followed. After installing the gLite service discovery module as described in this chapter, proceed as follows.

Step 1: If you want to use service discovery based on the information published in the R-GMA server, install the R-GMA client (see chapter **Error! Reference source not found.**)

Step 2: Change to the configuration directory:

```
cd /opt/glite/etc/config
```

Step 3: Copy the configuration templates from the templates directory:

```
cp templates/* .
```

Step 4: Customize the configuration files by replacing the changeme values with appropriate parameters according to the following table.

File name: **glite-service-discovery.cfg.cfg.xml**

service-discovery.type	<Decide where the information of the services are stored. If they are stored in R-GMA, specify rgma, if they are stored in BDII specify bdii, otherwise specify file. You can also specify a combination of file, rgma and bdii as separate array values>
service-discovery.site	<the site name of the service if you want to find only services on a specified site. Leave the parameter empty if you don't want to specify a site>
service-discovery.vo	<specify a vo if you have a default VO to be used. Leave the parameter empty if you don't want to specify a default VO>

The following file only exists/has to be modified if you want to use R-GMA based service

discovery

File name: glite-rgma-common.cfg.xml

rgma.server.hostname	<your R-GMA Server>
rgma.schema.hostname	<your R-GMA Schema Server>
rgma.registry.hostname	<your R-GMA Registry Server>

Step 5: Change to the scripts directory:

```
cd /opt/glite/etc/config/scripts
```

Step 6: Run the configuration script of the service that is using service discovery

```
./glite-XXX -config --configure
```

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

The gLite xxx service configuration was successfully completed

8 VOMS SERVER AND ADMINISTRATION TOOLS

8.1 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 occurring batch operations, or simply to serve as an alternative to the full blown web interface. It is also useful for bootstrapping the service.

The VOMS server can use MySQL or ORACLE as a backend.

8.2 INSTALLATION AND CONFIGURATION

This chapter is maintained in a separate document now. To find the latest up-to-date installation and configuration guide, please go to EDMS:

<https://edms.cern.ch/document/818502/>

9 LOGGING AND BOOKKEEPING SERVER

gLite Logging and Bookkeeping service it should be installed and configured in conjunction with gLite Workload Management System. gLite LB is installable and configurable together with gLite WMS using YAIM's installation and configuration target WMSLB.

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.eugridpma.org>).
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 the Sun Java web site and install it if you have not yet installed it.

9.4 LOGGING AND BOOKKEEPING SERVER INSTALLATION

1. Install APT if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite LB by executing

```
apt-get install glite-LB
```

or

```
yum install glite-LB
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

2. 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.
3. If the installation is performed successfully, the following components are installed:

gLite	in /opt/glite
Globus	in /opt/globus
MySQL	in /usr/bin/mysql

4. 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`
5. 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 6.4.

9.5 LOGGING AND BOOKEEPING SERVER CONFIGURATION

1. Copy the global configuration file templates

```

$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml
$GLITE_LOCATION/etc/config/template/glite-security-utils.cfg.xml
$GLITE_LOCATION/etc/config/template/glite-rgma-common.cfg.xml
  
```

to

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

open it and modify the parameters if required (see sections 4.2.3 and 5 and 6.4).

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 9). 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 9.

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 or comment it out to use the same as 'glite.user.name'
mysql.root.password		The mysql root password
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
rgma.servicetool.activate	true	Turn on/off servicetool for the node. [Example: true] [Type: 'boolean']
set.mysql.root.password	false	If this parameter is true, then the root password of the mysql database is set to the value specified in mysql.root.password if it not yet set. This parameter has

mysql.max_allowed_packet	17	no effect if the database root password is already set. It can be used to ease automated installation and configuration of the service, if mysql is not managed in some other way. [Example: false][Type: boolean]
This parameter allows to set the max_allowed_packet parameter in the mysql configuration file /etc/my.cnf. The default recommended value for the LB server is 17MB. [Example: 17][Type: Integer][Unit: MB]		

System Parameters

lb.index.list	owner location destination	Definitions of indices on all the currently supported indexed system attributes
---------------	----------------------------------	---

Table 9: LB Configuration Parameters

All servicetools parameters have been removed in the gLite 1.5, since the servicetool instances used to publish services are automatically handled by the configuration script. The instances can still be defined as in previous versions if the automatic values have to be overridden.

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:
 - 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 4 shows a list of the parameters that can be set. More details can be found in section 6.4.

- b. *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 5 in the section 6.4 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 6.4.

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

After installing the gLite LB module as described in this chapter, proceed as follows.

Step 1: Install the Java run-time libraries (obtained from the Sun Java web site):

```
rpm -ivh j2re-1_4_2_08-linux-i586.rpm
```

Step 2: Change to the configuration directory:

```
cd /opt/glite/etc/config
```

Step 3: Copy the configuration templates from the templates directory:

```
cp templates/* .
```

Step 4: Customize the configuration files by replacing the changeme values with appropriate parameters according to the following table.

File name: `glite-lb.cfg.xml`

<code>glite.user.name</code>	<define your own, must be the same as in the WMS module if installed on the same host>
<code>glite.user.group</code>	<define your own, must be the same as in the WMS module if installed on the same host>
<code>mysql.root.password</code>	<define your own, must be the same as in the WMS module if installed on the same host>

File name: glite-global.cfg.xml

site.config.url	<empty>
-----------------	---------

File name: glite-rgma-common.cfg.xml

rgma.server.hostname	<your R-GMA Server>
rgma.schema.hostname	<your R-GMA Schema Server>
rgma.registry.hostname	<your R-GMA Registry Server>

File name: glite-rgma-servicetool.cfg.xml

rgma.servicetool.sitename	<your site name as registered in R-GMA>
---------------------------	---

File name: glite-security-utils.cfg.xml

cron.mailto	<your own address>
-------------	--------------------

Step 5: Change to the scripts directory and execute the glite-lb-config.py script

```
./glite-lb-config.py --configure
```

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

The gLite Logging and bookkeeping Server configuration was successfully completed

Step 6: Start the LB services

```
./glite-lb-config.py --start
```

Check if any error message is displayed and if necessary take any corrective action as reported. If the operation is successful you should see at the end the message:

The gLite Logging and bookkeeping Server was successfully started

Step 7: Verify that the LB service have been correctly published by connecting to your R-GMA Browser with your Internet browser

```
https://<your R-GMA browser>:8443/R-GMA
```

You should see your LB service registered in the Services list

9.7 MANAGING THE LB SERVICES

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

<code>glite-lb-config.py --configure</code>	Configures all LB services
<code>glite-lb-config.py --start</code>	Starts all LB services (or restart them if they are already running)
<code>glite-lb-config.py --stop</code>	Stops all LB services
<code>glite-lb-config.py --status</code>	Verifies the status of all services. The exit code is 0 if all services are running, 1 in all other cases

9.8 STARTING THE LB SERVICES AT BOOT

When the LB 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-lb-config.py --start* command and makes sure that all necessary services are started in the correct order.

9.9 PUBLISHING LB SERVICES TO R-GMA

The LB 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 LB module. The instance are automatically created and configured by the LB configuration script, but the values can be overridden by defining the instance manually as in previous versions.

For more details about the R-GMA Service Tool service refer to section 6.4 in this guide.

10 WORKLOAD MANAGER

gLite Workload Management System it should be installed and configured in conjunction with gLite Logging and Bookkeeping service. gLite WMS is installable and configurable together with gLite LB using YAIM's installation and configuration target WMSLB.

10.1 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.

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.

10.2.1 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.eugridpma.org>). The security module gLite Security Utilities is installed and configured automatically when installing and configuring the WMS (refer to Chapter 5 for more information about the Security Utilities module). The module installs 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. Install the server host certificate `hostcert.pem` and key `hostkey.pem` in `/etc/grid-security`

10.2.2 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 the Sun Java web site and install it if you have not yet installed it.

10.2.3 WNS and the Information Systems

The workload Management System currently uses both R-GMA and BD-II as Information Systems. The WMS RGMA Purchaser, introduced in gLite 1.4, allows extracting information about CEs and CE-SE Bindings from R-GMA, where they are automatically published by the R-GMA CE Information Provider. Alternatively this information can be extracted by the GRIS

Purchaser from BD-II, where it can be published automatically using GIP on the CE. SE information can at this time only be extracted from BD-II. In order to submit jobs with data input conditions, either R-GMA and BD-II or BD-II alone are required. BD-II is a well known component of existing GRID middleware (e.g. LCG). Please, consult LCG guides for documentation on how to install and configure the BD-II. Other modes of operation for the information flow (synchronous and asynchronous pull mode), do not strictly require the usage of either R-GMA or BD-II, since both WMS and CE can be configured with static information about the respective endpoints. If WMS is used in push mode, all the CE information has to be filled in according to the current used Glue Schema inside it. For this reason the current deployment module foresees the insertion of the BD-II contact hostname, port and base DN as optional parameters.

10.2.4 Apache httpd and mod_ssl

The Apache httpd service and the mod_ssl module must be preinstalled on the WMS host before installing the glite-wms-config RPM. The httpd and mod_ssl RPMS are not currently distributed in the gLite APT cache and they must be taken from the operating system distribution.

10.3 WORKLOAD MANAGER SYSTEM INSTALLATION

1. Install APT if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite WMS by executing

```
apt-get install glite-WMS
```

or

```
yum install glite-WMS
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

2. 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

3. 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`
4. 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 6.4.3.

10.4 WORKLOAD MANAGEMENT SYSTEM CONFIGURATION

1. Copy the global configuration file templates

```
$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml
$GLITE_LOCATION/etc/config/template/glite-security-utils.cfg.xml
$GLITE_LOCATION/etc/config/template/glite-rgma-common.cfg.xml
```

to

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

open them and modify the parameters as required (Table 1 and Chapters 5 and 6)

2. Copy the WMS 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 10.

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'.
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'
lb.server		Host name and port of the Logging and Bookkeeping Server to be used by the Workload Manager Proxy. The port is normally 9000. If LB is installed on this node together with WMS, you can leave this parameter empty or comment it out. Example: lxb0001.cern.ch:9000
mysql.root.password		The mysql root password
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
gpbox.hostname		Hostname of the GPBox server that manages policies for this WMS. Leave this parameter empty or comment it out to disable policy management. [Example: gpbox.cern.ch][Type: string]
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
rgma.servicetool.activate	true	Turn on/off R-GMA Service Publishing for the WMS services. [Example: true] [Type: 'boolean']
account.discovery	false	Automatically discover pool accounts using pool account base names. If this parameter is set to true, the script will look for accounts starting with one of the base names set in the

		pool.account.basename parameter and followed by a valid numeral. No attempt to create additional accounts is done, but the discovered accounts will be configured
wms.config.file	`\${GLITE_LOCATION}/etc/glite_wms.conf	Location of the wms configuration file
lb.locallogger	`\${HOSTNAME}:9002	Host name and port of the local Logging and Bookkeeping logger to be used by the Workload Manager Proxy. This is normally running on the WMS server itself. Example: lxb0001.cern.ch:9000
set.mysql.root.password	false	If this parameter is true, then the root password of the mysql database is set to the value specified in mysql.root.password if it not yet set. This parameter has no effect if the database root password is already set. It can be used to ease automated installation and configuration of the service, if mysql is not managed in some other way
mysql.max_allowed_packet	17M	This parameters allows to set the max_allowed_packet parameter in the mysql configuration file /etc/my.cf. The default recommended value is 17MB.
		[New in gLite 3.0]
GSIWUFTPPORT	2811	Port where the globus ftp server is listening
GSIWUFTPDLOG	`\${GLITE_LOCATION_LOG}/gsiwuftp.log	Location of the globus ftp server log file
enable.purchasing.from.rgma	true	Enable the R-GMA purchaser. If this parameter is set to false the other parameters are ignored Example: true
rgma.query.timeout	30	Time out value in seconds of a purchase request. Example: 30
rgma.consumer.ttl	300	Time to live in seconds of the R-GMA consumer. Example: 300
rgma.consumer.life.cycle	30	Life cycle in seconds of the R-GMA consumer. Example: 30
ism.rgma.purchasing.rate	120	ISM purchasing rate in seconds.

e		Example: 120
wmproxy.MinPerusalTimeInterval	10	Integer representing the minimum number of seconds between two subsequent savings of job files for perusal. If this parameter is not specified a default value is 10 secs is used. [Example: 10][Type: integer][Unit: seconds]
gpbox.port.number	6699	Port number of the GPBox server that manages policies for this WMS. [Example: 6699][Type: integer]
condor.scheddinterval	10	Condor scheduling interval
condor.releasedir	/opt/condor-6.7.10	Condor installation directory
condor.CLASSAD_LIFE_TIME	60	How often should the collector check for machines that don't have ClassAds from the condor_master and send email about it?
condor.NEGOTIATOR_UPDATE_INTERVAL	20	condor_negotiator update interval
condor.MASTER_UPDATE_INTERVAL	20	condor_master update interval
condor.UPDATE_INTERVAL	20	Default update interval
condor.NEGOTIATOR_INTERVAL	30	The time interval, in seconds, at which the negotiator daemon updates the status of jobs
condor.HIGHPORT	50000	Specifies a higher limit of given port numbers for Condor to use
condor.LOWPORT	1500	Specifies a lower limit of given port numbers for Condor to use
CONDOR_CONFIG	\${condor.releasedir}/etc/condor_config	Condor global configuration file
condor.ENABLE_GRID_MONITOR	true	Enables the grid monitor. It must be set to true if this WMS node submits to LCG CEs. Valid values are true or false. [Example: true][Type: Boolean]
condor.blahpollinterval	10	How often should blah poll for new jobs?
information.index.port	2170	Port number of the Information Index

Information.index.base_dn	mds-vo-name=local, o=gris	Base DN of the information index LDAP server
disable.gris.purchasing	true	If this parameter is set to true, the WMS will not try to poll all CEs listed in the BD-II information service to validate them [Example: true] [Type: boolean]
GLOBUS_FLAVOR_NAME	gcc32dbg	The Globus libraries flavour to be used
System Parameters		
wms.si.service.type	org.glite.SEIndex	Service type of the gLite SEIndex service. Used in locating replicas with Fireman catalogs
wms.dli.service.type	data-location-interface	Service type of the LFC DLI service. Used in locating replicas with LCG catalogs
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 10: WMS Configuration Parameters

From gLite release 1.5 the VO-specific parameters have been moved to the separate vo-list-cfg.xml file. The WMS instances are created automatically by iterating on all defined VOs. For more information about using the new VO configuration model refer to the VO Configuration Guide and to Chapter 4 of this Installation Guide. Also all R-GMA Servicetool instances have been removed from the configuration file, since the instances are now created and configured automatically by the configuration script. The instances can still be configured manually as in previous versions if the automatic values have to be overridden.

3. Configure the R-GMA servicetool configure the servicetool configuration file using the template in

`$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 4 shows a list of the parameters that can be set. More details can be found in section 6.4

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 WMS configuration file `/opt/glite/etc/config/scripts/glite-wms-config.py`
5. The WMS Service is now ready.

10.5 WORKLOAD MANAGEMENT SYSTEM CONFIGURATION WALKTHROUGH

After installing the gLite WMS module as described in this chapter, proceed as follows.

Step 1: Install the Java run-time libraries (obtained from the Sun Java web site):

```
rpm -ivh j2re-1_4_2_08-linux-i586.rpm
```

Step 2: Change to the configuration directory:

```
cd /opt/glite/etc/config
```

Step 3: Copy the configuration templates from the templates directory:

```
cp templates/* .
```

Step 4: Customize the configuration files by replacing the changeme values with appropriate parameters according to the following table.

File name: `glite-wms.cfg.xml`

<code>glite.user.name</code>	<specify an account name>
<code>glite.user.group</code>	<specify a group name or leave empty to use the same as account name>
<code>voms.voname</code>	<your VOMS server hostname>
<code>voms.vomspport</code>	15000
<code>voms.vomscertsbj</code>	<your VOMS server certificate subject>
<code>pool.account.basename</code>	<define your own>
<code>pool.account.group</code>	<define your own>
<code>pool.account.number</code>	<as many as you like>
<code>wms.cemon.port</code>	5120
<code>information.index.host</code>	<your BD-II server host name or empty if it is not used>
<code>wms.cemon.endpoint</code>	<enter a list of CE hostnames that you want to send jobs requests to (push mode) or set just one empty value or comment the parameter to disable push mode>
<code>lb.server</code>	<enter the hostname of your LB server or localhost if LB is running on the same host>
<code>mysql.root.password</code>	<define your own, must be the same as in the LB module if

	installed on the same host>
cron.mailto	<your email address>
gpbox.hostname	<the hostname of the gpbox managing policies for this WMS node if it is used. Leave blank if GPBox is not used>
condor.condoradmin	<your email address>

File name: glite-global.cfg.xml

site.config.url	<empty>
-----------------	---------

File name: glite-rgma-common.cfg.xml

rgma.server.hostname	<your R-GMA Server>
rgma.schema.hostname	<your R-GMA Schema Server>
rgma.registry.hostname	<your R-GMA Registry Server>

File name: glite-rgma-servicetool.cfg.xml

rgma.servicetool.sitename	<your site name as registered in R-GMA>
---------------------------	---

File name: glite-security-utils.cfg.xml

cron.mailto	<your own address>
install.mkgridmap.cron	True

Step 5: Change to the scripts directory:

```
cd /opt/glite/etc/config/scripts
```

Step 6: Execute the glite-wms-config.py script:

```
./glite-wms-config --configure
```

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

The gLite WMS Service configuration was successfully completed

Step 7: Start the WMS services:

```
./glite-wms-config --start
```

Check if any error message is displayed and if necessary take any corrective action as reported and restart the script. If the operation is successful you should see at the end the message:

The gLite WMS Service was successfully started

Step 8: Verify that the WMS services have been correctly published by connecting to your R-GMA Browser with your Internet browser

https://<your R-GMA browser>:8443/R-GMA

You should see your WMS services registered in the Services list

10.6 MANAGING THE WMS SERVICES

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

<code>glite-wms-config.py --configure</code>	Configures all WMS services
<code>glite-wms-config.py --start</code>	Starts all WMS services (or restart them if they are already running)
<code>glite-wms-config.py --stop</code>	Stops all WMS services
<code>glite-wms-config.py --status</code>	Verifies the status of all services. The exit code is 0 if all services are running, 1 in all other cases
<code>glite-wms-config.py --startservice=xxx</code>	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 wmp = the WMPProxy Server service
<code>glite-wms-config.py --stopservice=xxx</code>	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 wmp = the WMPProxy Server service

10.7 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-wms-config.py --start* command and makes sure that all necessary services are started in the correct order.

10.8 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 instance are automatically created and configured by the WMS configuration script, but the values can be overridden by defining the instance manually as in previous versions.

For more details about the R-GMA Service Tool service refer to section 6.4 in this guide.

11 THE TORQUE RESOURCE MANAGER

gLite implementation of the Torque resource manager configuration. This configuration is accessible through YAIM target "TORQUE_server". It should be noted that YAIM contains function for configuration of the Torque server (by using function: `config_torque_server`). Both methods are completely independent

11.1 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*.

11.1.1 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.

The Torque Server can be configured to run the BLAHP log parser daemon. This daemon will be responsible to provide the logs to BLAHP. By default this option is activated.

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.

11.1.2 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).

11.2 INSTALLATION PRE-REQUISITES

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

11.3 TORQUE SERVER

11.3.1 TORQUE Server Installation

1. Install APT if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite Torque Server by executing

```
apt-get install glite-torque-server-config
```

or

```
yum install glite-torque-server-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

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

gLite in /opt/glite (\$GLITE_LOCATION)

torque in /var/spool/pbs

3. 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`
4. 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 6.4.

11.3.2 TORQUE Server Service Configuration

1. Copy the global configuration file template

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

to

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

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

2. Copy the VO configuration file template

```
$GLITE_LOCATION/etc/config/vo-list.cfg.xml
```

to

```
$GLITE_LOCATION/etc/vo-list.cfg.xml
```

open it and add the VOs instances required and their parameters.

3. 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 11. The R-GMA servicetool related parameters can be found in **Table 4**.

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

- a. *Common parameters* (first part of Table 11)

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

- b. *Torque client / Worker node specific values* (second part of Table 11)

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 11)

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		<p>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.</p> <p>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.</p>
Advanced Parameters		
glite.installer.verbose	True	Enable verbose output.
use.log.parser	True	This option must be set to true to run the BLAHP log parser daemon in the port specified by the pbs.log.parser.port variable. Valid values for this parameter are true or false
PBS_SPOOL_DIR	/var/spool/pbs	The PBS spool directory
torque-server.name	\${HOSTNAME}	Name of the machine where the job server is running, it usually corresponds to the Computing Element: Example: \${HOSTNAME}.
pbs.log.parser.port	33332	This is the port where the log parser is listening for log requests. [Example: 33332] [Type: integer]
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.acl-host.list		List of hosts which may request services from this server. This list contains the network name of the hosts. Local requests, i.e. from the

			server host itself, are always accepted even if the host is not included in the list. Format: [+ -] hostname.domain[,...]; default value: all hosts
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 users, other than job owner, are allowed to query the status of or select the job.
torque-server.scheduler.interaction	600		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

maui.server.port	40559	between jobs. Port on which the Maui server will listen for client connections, by default 40559.
maui.server.mode	NORMAL	Specifies how Maui interacts with the outside world. Possible values NORMAL, TEST AND SIMULATION.
maui.defer.time	00:01:00	Specifies amount of time a job will be held in the deferred state before being released back to the Idle job queue. Format [[DD:]HH:]MM:]SS
maui.rm.poll.interval	00:00:10	Maui will refresh its resource manager information every 10 seconds. This parameter specifies the global poll interval for all resource managers.
maui.log.filename	`\${GLITE_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 virtual processors of the machine. Example: 1,2 , [Type: string].
torque-wn.attribute	Glite	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. [Type: string]
queue.type	Execution	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	True	Defines if the queue will or will not accept new jobs. When false the queue is disabled and will not accept jobs.
queue.started	True	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	False	Attribute which when true directs the server to use the queue group access control list acl_groups.
queue.acl.groups		List which allows or denies enqueueing of jobs owned by members of the listed groups. The groups in the list are groups on the server host, not submitting hosts. Syntax: '[+ -]group_name[,...]' Example: +iteam,+egee,-test authorizes the test group users to submit jobs to this queue..

Table 11: TORQUE Server Configuration Parameters

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:

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 1 shows a list of the parameters that can be set. More details can be found in section 4.2.3.

For Torque-server the following sub-services are published via the R-GMA servicetool:

- ii. Torque PBS server
- iii. Torque maui

Again, you find the necessary steps described in section 6.4.

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

As root run the Torque Server Configuration script (with the `--configure` option in order to configure the service) `/opt/glite/etc/config/scripts/glite-torque-server-config.py --configure`.

Once the services have been properly configured (no service will be running) it will be necessary to start them all. To do so, follow the next step.

As root start the Torque Server services by running the configuration script with the `--start` option.

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

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

11.3.3 TORQUE Server Configuration Walkthrough

The Torque Server configuration script performs the following steps:

1. Load the Torque Server configuration file `$GLITE_LOCATION/etc/config/glite-torque-server.cfg.xml` and the servicetool configuration file `$GLITE_LOCATION/etc/config/glite-rgma-servicetool.cfg.xml`
2. Stop the services that are running
3. Add the torque and maui ports to `/etc/services`.
4. Create the `/var/spool/pbs/server_name` file containing the torque server hostname.
5. Create the list with the torque clients under `/var/spool/pbs/server_priv/nodes`.
6. Create the `pbs_server` configuration.
7. Start the `pbs_server`.
8. Look for changes in the `pbs_server` configuration since the last time the Torque Server was configured.
9. Establish the server configuration performing the necessary updates.

10. 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).
11. Execute the defined queues configuration
12. 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 be included in the `/etc/ssh/shosts` file to allow `HostbasedAuthentication`.
13. 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.
14. Run the `/opt/edg/sbin/edg-pbs-shostsequiv` script.
15. Look for duplicated key entries in `/etc/ssh/ssh_known_hosts`.
16. 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).
17. 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.
18. Run `/opt/edg/sbin/edg-pbs-knownhosts` to add the keys to `/etc/ssh/ssh_known_hosts`.
19. 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`.
20. Restart the `sshd` daemon to take the changes into account.
21. Stop the `pbs_server`.
22. Create the maui configuration file in `/var/spool/maui/maui.cfg`.
23. Create the `servicetool` instances and configure the `servicetool` to register them.

11.3.4 Managing the TORQUE Server Service

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

<code>glite-torque-server-config.py</code> <code>-configure</code>	Configures all TORQUE SERVER services (<code>pbs_server</code> , <code>maui</code> , <code>BLAHP</code> log server and <code>servicetool</code>)
<code>glite-torque-server-config.py</code> <code>-start</code>	Starts all TORQUE CLIENT services (or restart them if they are already running, <code>pbs_mom</code>)
<code>glite-torque-server-config.py</code> <code>-stop</code>	Stops all TORQUE SERVER services (<code>pbs_server</code> , <code>maui</code> and <code>servicetool</code>)
<code>glite-torque-server-config.py</code> <code>-status</code>	Checks the status of the TORQUE SERVER services

11.3.5 Publishing Torque Services to R-GMA

The torque 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 Torque module. The instance are automatically created and configured by the Torque configuration script, but the values can be overridden by defining the instance manually as in previous versions.

For more details about the R-GMA Service Tool service refer to section 6.4 in this guide.

11.4 TORQUE CLIENT

11.4.1 TORQUE Client Installation

1. Install APT if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite Torque Client by executing

```
apt-get install glite-torque-client-config
```

or

```
yum install glite-torque-client-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

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

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

3. 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`.

11.4.2 TORQUE Client Configuration

1. Copy the global configuration file template `$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml` to `$GLITE_LOCATION/etc/config`, open it and modify the parameters if required (see Table 12)

2. Copy the VO configuration file template

```
$GLITE_LOCATION/etc/config/vo-list.cfg.xml
to
$GLITE_LOCATION/etc/vo-list.cfg.xml
```

open it and add the VOs instances required and their parameters

3. 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: <code>\${HOSTNAME}</code> .
Advanced Parameters		
glite.installer.verbose	True	Enable verbose output.
mpi.copy.enable	False	When using MPI it may be necessary to copy information between worker nodes. This variable activates HostBasedAuthentication if set to True. Possible values: True and False
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		

Table 12: TORQUE Client Configuration Parameters

- As root run the Torque Client Configuration file with the `--configure` option `/opt/glite/etc/config/scripts/glite-torque-client-config.py --configure`.
Once the services have been properly configured (no service will be running) it will be necessary to start them all. To do so, follow the next step.
- As root start the Torque Client services by running the Torque Client Configuration File:
`/opt/glite/etc/config/scripts/glite-torque-client-config.py --start`

11.4.3 TORQUE Client Configuration Walkthrough

The Torque Client configuration script performs the following steps:

- Load the Torque Client configuration file `$GLITE_LOCATION/etc/config/glite-torque-client.cfg.xml`

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.

11.4.4 Managing the TORQUE Client

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

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

12 COMPUTING ELEMENT

Installation and configuration of glite-CE is accessible through YAIM by using the installation and configuration target "glite-CE"

12.1 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.

12.2 INSTALLATION PRE-REQUISITES

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

12.2.1 Security Settings

1. Install one or more Certificate Authorities certificates in `/etc/grid-security/certificates`. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (<http://www.gridpma.org/>). A special security module called `glite-security-utils` (gLite Security Utilities) is installed and configured automatically when installing and configuring the CE (refer to Chapter 13 for more information about the Security Utilities 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.

12.2.2 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.

12.2.3 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 are provided, please refer to chapter 11 for more information.

12.3 COMPUTING ELEMENT SERVICE INSTALLATION

1. Install APT if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite Computer Element by executing

```
apt-get install glite-CE
```

or

```
yum install glite-CE
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

2. 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)

3. 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`
4. 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 6.4.

12.4 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`
`$GLITE_LOCATION/etc/config/templates/glite-dgas-client.cfg.xml` to `$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, R-GMA and DGAS chapters for a description of the parameters used by those modules):

Parameter	Default value	Description
User-defined Parameters		
<code>cemon.wms.host</code>		The hostname of the WMS server(s) that receives notifications from this CE
<code>cemon.wms.port</code>		The port number on which the WMS server(s) receiving notifications from this CE is listening
<code>cemon.wms.host.subject</code>		Array of the host certificate subjects of the WMS server(s) that are allowed to query the CE Monitor service on this CE
<code>cemon.lrms</code>		The type of Local Resource Management System. It can be <code>pbs</code> , <code>lsf</code> or <code>condor</code> . The value <code>pbs</code> is also used for <code>torque</code> . If this parameter is absent or empty, the default type is <code>pbs</code>
<code>cemon.lrms.version</code>		The version of Local Resource

		Management System
cemon.cetype		<p>he type of Computing Element. It can be blah, condor or gram.</p> <p>If this parameter is absent or empty, the default type is blah.</p>
cemon.cluster		<p>The cluster entry point host name. Normally this is the CE host itself</p>
cemon.cluster-batch-system-bin-path		<p>The path of the lrms commands. For example: '/usr/pbs/bin' or '/usr/local/lfs/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</p>
cemon.cesebinds		<p>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</p>
cemon.queues		<p>A list of queues defined on this CE node. Examples are: long, short, infinite, etc.</p>
use.log.parser		<p>Set this option to true to use a separate log parser. Valid values for this parameter are true or false. [Example: false] [Type: boolean]</p>
log.parser.address		<p>The IP address of the remote LRMS server running the log parser daemon. Leave this parameter empty or comment it out if the LRMS is running on this CE server or if the log parser is not used.</p>
lb.user		<p>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</p>
iptables.chain		<p>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</p>
Advanced Parameters		
glite.installer.verbose	True	Enable verbose output

glite.installer.checkcerts	True	Enable check of host certificates
rgma.servicetool.activate	true	Turn on/off R-GMA Service Publishing for the CE services. [Example: true] [Type: 'boolean']
account.discovery	false	Automatically discover pool accounts using pool account base names.
dgas.client.enabled	true	This variable allows configuring the dgas client in the CE. It can be true or false. [Example: true][Type: boolean]
notifications.condition	GlueCEStateWaitingJobs<3	"An expression using Glue schema objects that is evaluated to instruct CE Monitor how to notify the WMS servers of its availability. If the expression evaluates to true, availability notifications are sent and the CE is added to the WMS ISM cache. If the expression evaluates to false, expiration notifications are sent and the CE is removed from the WMS ISM cache.
create.sgm.account	true	If this parameter is set to true, the sgm accounts are created using values from the VO configuration file. [Example: true][Type: boolean]
custom.runtime.environment		The entries specified in this array parameter are added to the CE info provider file as additional GlueHostApplicationSoftwareRunTimeEnvironment entries. [Example: MY_APP_1_0_0] [Type: 'string']
PBS_SPOOL_DIR	/var/spool/PBS	The PBS spool directory
LSF_CONF_PATH	/etc	The directory where the LSF configuration file is located
pbs.log.parser.port	33332	The port where the log parser is listening for log request on the PBS server.
lsf.log.parser.port	33333	The port where the log parser is listening for log request on the LSF server.
globus.osversion	<empty>	The kernel id string identifying the system installed on this node. For example: '2.4.21-20.ELsmp'. This parameter is normally automatically detected, but it can be set here
globus.hostdn	<empty>	The host distinguished name (DN) of this node. This is normally

		automatically read from the server host certificate. However it can be set here. For example: 'C=ORG, O=DOMAIN, OU=GRID, CN=host/server.domain.org'
condor.version	6.7.10	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-6.7.10	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
rgma.gin.run_ce_provider	yes	Run the CE backend for R-GMA Gin
lcg.providers.location	/opt/lcg	The location where the LCG providers are installed.
ce.gridftp.enable	false	Enable startup of the gridftp server.
[new in gLite 3.0]		
System Parameters		
ce-monitor.DOCBASE	\${GLITE_LOCATION}/share/webapps/ce-monitor.war	Location of the ce-monitor.war file.

Table 13: CE Configuration Parameters

From the gLite release 1.5 the VO-specific parameters have been moved to the separate vo-list-cfg.xml file. The CE instances are created automatically by iterating on all defined VOs. For more information about using the new VO configuration model refer to the VO Configuration Guide and to Chapter 4 of this Installation Guide. Also all R-GMA Servicetool instances have been removed from the configuration file, since the instances are now created and configured automatically by the configuration

script. The instances can still be configured manually as in previous versions if the automatic values have to be overridden.

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 4 shows a list of the parameters that can be set. More details can be found in section 6.4:

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 5 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 6.4.

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 with the `–configure` option:

```
/opt/glite/etc/config/scripts/glite-ce-config.py –configure
```

Once the services have been properly configured (no service will be running) it will be necessary to start them all. To do so, follow the next step.
7. As root start the CE services by running the CE configuration file with the `–start` option:

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

12.5 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

12.6 MANAGING THE CE SERVICES

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

<code>glite-ce-config.py --configure</code>	Configure the CE services
<code>glite-ce-config.py --start</code>	Starts all CE services (or restart them if they are already running)
<code>glite-ce-config.py --stop</code>	Stops all CE services
<code>glite-ce-config.py --status</code>	Verifies the status of all services. The exit code is 0 if all services are running, 1 in all other cases

12.7 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.

12.7.1 Publishing CE Services to R-GMA

The CE 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 CE module. The instance are automatically created and configured by the CE configuration script, but the values can be overridden by defining the instance manually as in previous versions.

For more details about the R-GMA Service Tool service refer to section 6.4 in this guide.

12.8 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

13 DGAS

13.1 SERVICE OVERVIEW

The DataGrid Accounting System (DGAS) software aims to be a full featured distributed Grid accounting toolkit. Since it is conceived and designed to be completely grid oriented, it is fully distributed without having a central repository of accounting information. It instead relies upon a network of independent accounting servers used to keep the accounting/transaction records of groups of GridUsers and GridResources.

DGAS can be used to account classic Computational Usage Records like CPU Time, memory usage and so on. It can also be used as an Economic Accounting system, treating information about the cost of the jobs executed by each GridUser on the single GridResources. This feature can be exploited for example by a Grid Service Provider that wants to charge its users for the provided service. The economic accounting can also be used to implement the so called Economic Brokering of the grid resources (selection of execution sites and services based on economic principles in order to improve the balancing of the workload).

13.1.1 DGAS Server Overview

This deployment module contains and configures the Price Authority (PA) service the Home Location Register (HLR) service and the High Availability Daemon (HAD).

The **Price Authority (PA)** is a key component of the DGAS toolkit, providing the features necessary for Economic Accounting. In a few words, a PA server is an entity that assigns the prices to the subset of grid resources within its administrative domain. The prices that are kept in a historic price database can be assigned manually or using different dynamic pricing algorithms. The price of a resource is used to compute the cost for a job. The given cost can then be charged to the user that submitted the job.

The **Home Location Register (HLR)** service is the part of DGAS that is responsible for keeping the accounting information for both grid users and grid resources. It receives the accounting information, the so called Usage Records from the grid resources, and stores them for later retrieval. These usage records are the basis for the job cost computation¹, the phase in which the HLR computes the cost for a given job. The job cost can then be debited to the grid user and credited to the grid resource, thus implementing an economic accounting for the the grid activities of the single users. Information can be gathered from the HLR service on a per user, per resource, per job basis.

Since DGAS treats important information, it has to provide a high availability. The High Availability Daemon (HAD) is responsible for continuously monitoring the status of a service. In case of failure it restarts the daemon, thus avoiding long down periods due to service failures.

13.1.2 DGAS Client Overview

The dgas client deployment module is responsible for configuring the dgas related services that will run in the Computing Element, that is, the gianduia service, the cePushd and the ATM one.

It is important to notice that the DGAS client also needs to be installed, though not configured, in the Worker Node.

The dgas client deployment module is responsible for configuring the following services: gianduia, cePushD, ceServerd and the HAD daemon.

The **Gianduia** service daemons are installed on a Computing Element (or a generic grid resource) in order to collect the usage records of the executed user jobs and send them to the DGAS HLR service for accounting.

The **CEPushD** daemon uses the files created by Gianduia (or by another service that creates compatible files) and uses the information available in the file to initiate the transmissions of the usage records to the User HLR service, thus initiating the accounting procedures for the jobs.

The files created by Gianduia are treated in a queue and asynchronously processed. When a job's usage record is successfully sent to the User HLR, the corresponding file is removed from the queue and deleted such that it doesn't pollute the CE file system. If a job's usage record can't be correctly transmitted, the process will be retried for a tunable amount of times, after which it will be marked as unprocessable. In this case the related information is not deleted such that it is still available to the CE site manager.

The **ceServerd** is a light weight daemon running together with Gianduia and collecting information transmitted from the Worker Node (WN) on which the job is actually running. The ceServerd is contacted by an equally light weight client that is run by the job's JobWrapper on the WN.

The **dgas-ce-getAcctLogd** is a daemon used when a site installes the LRMS master on a node different than the CE. Since Usage records are composed from information coming from both CE and LRMS master log files, this daemon can be used to send to the CE the accounting logs needed by gianduia.

The **HAD** daemon will behave in the same way as it is doing in the dgas server, that is, monitoring the status of a service and restarting it in case it dies.

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.

13.3 DGAS SERVER

13.3.1 DGAS Server Installation

1. Install APT, if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite DGAs Server by executing

apt-get install glite-dgas-server-config

or

yum install glite-dgas-server-config

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

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

gLite in /opt/glite (\$GLITE_LOCATION)

3. The gLite dgas-server configuration script is installed in \$GLITE_LOCATION/etc/config/scripts/glite-dgas-server-config.py. A template configuration file is installed in \$GLITE_LOCATION/etc/config/templates/glite-dgas-server.cfg.xml
4. The gLite dgas-server installs the security utils rpms as well as the security utils config RPM.
5. The gLite dgas-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 6.4.

13.3.2 DGAS Server Service Configuration

1. Copy the global configuration file template
\$GLITE_LOCATION/etc/config/template/glite-global.cfg.xml
to
\$GLITE_LOCATION/etc/config
open it and modify the parameters if required
2. Copy the VO configuration file template
\$GLITE_LOCATION/etc/config/vo-list.cfg.xml
to
\$GLITE_LOCATION/etc/vo-list.cfg.xml
open it and add the VOs instances required and their parameters.
2. Copy the configuration file template from
\$GLITE_LOCATION/etc/config/templates/glite-dgas-server.cfg.xml
to

`$GLITE_LOCATION/etc/config/glite-dgas-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*. Note that in the current version only the PA service is published in servicetool.

As this deployment module can be used to configure both the PA and the HLR services, two parameters (`pa-server.enabled` and `hlr-server.enabled`) have been included. These parameters allow the user to select the service to be installed, that is, the PA service, the HLR service or both services.

<i>Configuration parameters</i>		
User-defined Parameters		
Parameter	Default Value	Description
<code>mysql.root.password</code>		Password (clear) of the root user of the MySQL server used for the database creation. A password has to be provided.
<code>pa-server.db.server</code>		The database server to store the PA database [Example: localhost]
<code>pa-server.db.user</code>		The database user to access the PA database, [Example: pauser]
<code>pa-server.db.password</code>		The password of the database user to access the PA database, [Example: papassword]
<code>hlr-server.db.server</code>		The database server to store the HLR database [Example: localhost]
<code>hlr-server.db.user</code>		The database user to access the PA database, [Example: hlruser]
<code>hlr-server.db.password</code>		The password of the database user to access the PA database, [Example: hlrpassword]
<code>hlr-server.user</code>		User used to run the HLR daemons. Example [root]
Advanced Parameters		
<code>glite.installer.verbose</code>	<code>true</code>	Enable verbose output. [Example: 'true'] [Type: 'boolean']

pa-server.enabled	yes	Select this option if you want to configure the pa server. Format: true, false
pa-server.db.name	pa	Specifies the database that keeps the history of the resource's prices.
pa-server.port	56567	The port on which the PA server should listen
pa-server.logfile	\${GLITE_LOCATION_LOG}/pad.log	Default PA log file
pa-server.lockfile	\${GLITE_LOCATION_VAR}/pa.lock	default PA lock file
pa-server.had.lockfile	\${GLITE_LOCATION_VAR}/pa-had.lock	Lock file for the had daemon
pa-server.contact	@CONTACT	X509 certificate subject used to authenticate the PA server
pa.pricing.ttl	3600	the minimum time-to-live (validity period) for resource prices. When a price quotation is requested by a client a new price will be computed only if this period is expired, otherwise the current valid price will be returned. The value specified here is to be understood as a default value. The PA administrator may set different TTLs for the single CEs, using the command line user interface
hlr-server.enabled		Select this option if you want to configure the hlr server. Format: true, false
hlr-server.db.name	hlr	Specifies the database that will store accounting information (accounts and usage records)
hlr-server.db.tmp.name	hlr_tmp	Specifies the database that will contain temporary information (usage records that have still to be processed)
hlr-server.port	56568	The port on which the HLR server should listen
hlr-server.logfile	\${GLITE_LOCATION_LOG}/hlrd.log	Default HLR log file
hlr-server.lockfile	\${GLITE_LOCATION_VAR}/hlr.lock	default HLR lock file
hlr-server.dgas.var	\${GLITE_LOCATION_VAR}	

	R}/dgas	
hlr-server.had.lockfile	\${GLITE_LOCATION_VA R}/hlr-had.lock	Lock file for the had daemon
hlr-server.thread.number	5	Maximum number of contemporary threads of the HLR server
hlr-server.proxyfile	/tmp/hostProxyFile	Specifies where to store the host proxy file
hlr-transaction-manager.logfile	\${GLITE_LOCATION_LO G}/hlr_qmgrd.log	Default log file for the transaction manager daemon
hlr-transaction-manager.lockfile	\${GLITE_LOCATION_VA R}/hlr_qmgr.lock	Default lock file for the transacton manager daemon
hlr-transaction-manager.expiration.period	600	Expiration period (in seconds) for a transaction in the usage record queue (database for temporary storage of information that still has to be processed).After this time the priority of the transaction is lowered
hlr-transaction-manager.queue.depth	10	Number of levels in the queue for unprocessed usage records. Transactions enter the queue with priority 0 and are increased when the system can't process it (the priority is raised after the expiration period defined in hlr-transaction-manager.expiration.period
hlr-transaction-manager.processed	20	Maximum number of transactions processed at each iteration of the process.The higher the number of transactions, the higher the resource consumption of the process
hlr-transaction-manager.interval	30	Interval between two iterations of transaction processing. The lower the interval, the higher the resource consumption of the process
rgma.servicetool.activate	true	Turn on/off servicetool for the node.[Example: true] [Type: 'boolean']
System Parameters		
pa.pricing.schema	manual	The pricing scheme that will be adopted for the determination of resource

		prices. If set to manual the PA administrator can use the local command line user interface to set fixed prices for each CE of the site. If set to dynamic the price will be determined dynamically according to some of the resource's static and dynamic attributes (such as number of jobs in the queue). The algorithm that is used is specified by the parameter pa_price_dll_name
pa.pricing.dll.name	libglite_dgas_paPriceAl gMan.so	specifies which shared library contains the algorithm used for pricing

Table 14: DGAS Server configuration parameters

From gLite release 1.5 the VO-specific parameters have been moved to the separate vo-list-cfg.xml file. The DGAS instances are created automatically by iterating on all defined VOs. For more information about using the new VO configuration model refer to the VO Configuration Guide and to Chapter 4 of this Installation Guide. Also all R-GMA Servicetool instances have been removed from the configuration file, since the instances are now created and configured automatically by the configuration script. The instances can still be configured manually as in previous versions if the automatic values have to be overridden.

3. Configure the R-GMA servicetool. For this you have to configure the servicetool itself as well as configure the sub-services of dgas server for the publishing via the R-GMA servicetool:
4. 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..

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. As root run the Dgas Server Configuration script (with the `--configure` option in order to configure the service) `/opt/glite/etc/config/scripts/glite-dgas-server-config.py --configure`.
Once the services have been properly configured (no service will be running) it will be necessary to start them all. To do so, follow the next step.
6. As root start the Dgas Server services by running the configuration script with the `--start` option.
`/opt/glite/etc/config/scripts/glite-dgas-server-config.py --start`

Once reached this point the Dgas Server Service is ready to be used.

13.3.3 DGAS Server Configuration Walkthrough

The Dgas Server configuration script performs the following steps:

1. Load the Dgas Server configuration file `$GLITE_LOCATION/etc/config/glite-dgas-server.cfg.xml`
2. Stop the services that are running
3. Check the host certificates
4. Check the host certificates exist and are in the right location.
5. Configure the security-utils.
6. Start mysql.
7. Configure the mysql root password.
8. If the `pa-server.enabled` variable is set to true:
 - Create the `dgas_pa.conf` file
 - Check if the pa database exist, if not, create it.
9. If the `hlr-server.enabled` variable is set to true:
 - Check the host certificate is in the grid-mapfile, if that is not the case, add it.
 - Create the hlr user.
 - Create the `dgas_hlr.conf` configuration file.
 - Check if the hlr server databases exist, if not, create them.
10. Create the servicetool instances and configure the servicetool to register them.
11. Stop mysql

13.3.4 Managing the DGAS Server Service

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

<code>glite-dgas-server-config.py --configure</code>	Configures the DGAS SERVER services (PA and/or HLR) and servicetool.
<code>glite-dgas-server-config.py --start</code>	Starts all DGAS services and servicetool if <code>rgma.servicetool.activate</code> is set to true.

<code>glite-dgas-server-config.py - stop</code>	Stops all DGAS services (PA and/or HLR) and servicetool
<code>glite-dgas-server-config.py - status</code>	Checks the status of the DGAS SERVER services

13.4 DGAS CLIENT

13.4.1 DGAS Client Installation

1. Install APT, if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite DGAS Client by executing

```
apt-get install glite-dgas-client-config
```

or

```
yum install glite-dgas-client-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

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

gLite in /opt/glite (\$GLITE_LOCATION)

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

13.4.2 DGAS Client Configuration

6. 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)
7. Copy the configuration file template from \$GLITE_LOCATION/etc/config/templates/glite-dgas-client.cfg.xml to \$GLITE_LOCATION/etc/config/glite-dgas-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 Parameters		
dgas-client.atmClient.resource. PA.id		Specifies the contact string of the PA where the Computing Element is registered (i.e. the PA that is responsible for

Parameter	Default Value	Description
		setting the CE's price).The PA contact string is formed as: PA host name:port:subject of host cert
dgas-client.atmClient.resource.Bank.id		Specifies the contact string of the site HLR where the Computing Element is registered (i.e. the HLR that manages the CE's account). The HLR contact string is formed as: HLR host name:port:subject of host cert
dgas-client.gianduia.lsfAcctLog Dir		This specifies on LSF systems where to find the accounting logs. The 'gianduia' daemon is able to find the value automatically on most installations. It is therefore necessary to specify it only on non standard installations of LSF .This is usually defined as <code>[[{path_lsf}/mnt/work/{nome cluster}/logdir]</code> . If pbs is used instead of lsf the value should be empty
Advanced parameters		
glite.installer.verbose	True	Enable verbose output.[Example: 'true'] [Type: 'boolean']
dgas-client.atmClient.economic Accounting	No	Used by the site manager to specify if he wants users to be cahrged for resource usage (virtual credits exchanged between User HLR and Resource HLR).Values: no yes
dgas-client.CeServerd.lockFile Name	`\${GLITE_LOCATION_VAR}/dgas_ce_Serverd.lock	Lock File for the daemon
dgas-client.CeServerd.logFile Name	`\${GLITE_LOCATION_VAR}/dgas_ce_Serverd.log	Log File for the daemon
dgas-	`\${GLITE_LOCATION_VAR}	Lock File for the had lock

Parameter	Default Value	Description
client.CeServerd.hadlock FileName	R}/dgas_ce_ServerdHAD. lock	daemon
dgas- client.ce_pushd.gridUser	Dgas	User account used by the pushd for using the user proxy certificates to contact the User HLR server
dgas- client.ce_pushd.dgasURD ir	\${GLITE_LOCATION_VA R}/dgasURBox/	Specifies the spool directory where the daemon searches for the job usage records and user proxies
dgas- client.ce_pushd.dgasErrD ir	\${GLITE_LOCATION_VA R}/dgasURBox/ERR/	Specifies the spool directory where the daemon moves the usage recordd and user proxies that couldn't be processed after a given number of retries.
dgas- client.ce_pushd.qDepth	5	Specifies the depth of the daemon priority queue. Usage records traverse this queue before being moved in dgasErrDir
dgas- client.ce_pushd.qMult	3	Number of times the daemon tries to process the transmission of a usage record before lowering its priority in the queue
dgas- client.ce_pushd.lockFileN ame	\${GLITE_LOCATION_VA R}/dgas_ce_pushd.lock ame	Lock File for the daemon
dgas- client.ce_pushd.mainPollI nterval	10	Time, in seconds, between two usage record processing cycles
dgas- client.ce_pushd.queuePol lInterval	50	Time, in seconds, after which the system processes lower priority usage records in the queue
dgas- client.ce_pushd.forceLoc alFirst	No	Specifies an alternate routing for the usage record forwarding process between CE, User HLR and Resource HLR. If it is set to yes, usage records are signed with the CE's

Parameter	Default Value	Description
		host credentials and sent to the site HLR (Resource HLR) first.If it is set to no usage records are signed with the user's credentials and sent to the User HLR first.
dgas-client.ce_pushd.forceLocalOnly	No	If set to yes, it specifies that the usage records _MUST_ be sent by the daemon to the Resource/Site HLR _ONLY_ . No copies of the usage records are sent to the User HLR. Usage record for jobs executed on this CE will be available to its Resource HLR _ONLY_ . NO economic accounting is possible if this parameter is set to yes
dgas-client.gianduia.chocolateBox	\${GLITE_LOCATION_VAR}/dgasRawBox/	This parameter specifies the spool directory where the gianduia daemon retrieves the usage record skeleton transferred from the JobWrapper of the job
dgas-client.gianduia.garbageCollector	\${GLITE_LOCATION_VAR}/garbageCollector/	This is the location where files are copied by 'gianduia' if severe errors are present
dgas-client.gianduia.lockFileName	\${GLITE_LOCATION_VAR}/dgas_gianduia.lock	This is the file name for the 'gianduia' daemon lock file
dgas-client.gianduia.logFileName	\${GLITE_LOCATION_VAR}/gianduia.log	Default log file name for the gianduia daemon
Dgas-client.gianduia.mainPollInterval	60	Interval between the attempts to process the base usage records (building the full usage record from the skeleton and the information from the LRMS log)
dgas-client.gianduia.queuePollInterval	600	Interval between two cleanups of the UR directory. During cleanup the daemon checks for garbage in the UR

Parameter	Default Value	Description
		directory garbage clean-up interval in seconds
dgas-client.gianduia.pbsAcctLogDir	/var/spool/pbs/server_priv/accounting/	This is the location of the directory where PBS accounting logs are stored
dgas-client.gianduia.keyList	GlueHostBenchmarkSF00,GlueHostBenchmarkSI00	Comma-separated list of parameters that we want gianduia to retrieve from an Idif file specified in the 'IdifDefaultFiles' and 'glueLdifFile' files. The key/value pairs will be appended to the usage record. Example: keyList = GlueHostBenchmarkSF00,GlueHostBenchmarkSI00
dgas-client.gianduia.IdifDefaultFiles	/opt/glite/etc/glite-ce-ce-plugin/out.Idif	Comma-separated list of files to be searched for the keys specified in the 'keyList' parameter. Example: IdifDefaultFiles = /opt/glite/etc/glite-ce-ce-plugin/out.Idif
dgas-client.gianduia.glueLdifFile		File to search for the parameters specified in 'keyList'. It overrides the contents of IdifDefaultFiles. Example: glueLdifFile = '/opt/glite/etc/glite-ce-ce-plugin/out.Idif'
dgas-client.getAcctLogd.aclFile	"\${GLITE_LOCATION}/etc/getAcctLogd.acl	Specifies the file for host ACL. In this file the sys admin must specify the hosts that are allowed to send their logs to the CE (the LRMS master node hostname)
dgas-client.getAcctLogd.listeninPort	56565	The listening port used by the daemon.
dgas-client.getAcctLogd.logFileName	\${GLITE_LOCATION_VAR}/getAcctLogd.log	The log file where the listener logs its activities.
dgas-client.getAcctLogd.lockFileName	\${GLITE_LOCATION_VAR}/getAcctLogd.lock	Lock file name

Parameter	Default Value	Description
dgas-client.getAcctLogd.output File	\${GLITE_LOCATION_VA R}	The file where the listening daemon writes the contents of the file sent by the client.
dgas-client.getAcctLogd.output Dir		The directory where the output file is written. If outputDir is specified, outputFile is ignored and the name of the output file will be the same as the one read by the client, and it shall be put into the dir specified by outputDir. This is useful for instance when more than one file needs to be sent to the CE, wich is for instance the PBS/Torque use case.
System parameters		
dgas-client.ce_pushd.defaultUserHLR		Address of an HLR that can be used as a default User HLR for users who's HLR server is not specified in the job JDL or in the UI conf file. Must be used carefully
dgas-client.gianduia.gianduiotti Box	\${dgas-client.ce_pushd.dgasURDir}	This specifies the directory where 'gianduia' puts the full usage record for the job once it is finished and the LRMS UR is retrieved from the LRMS accounting log.

Table 15: DGAS Server configuration parameters

From the gLite release 1.5 the VO-specific parameters have been moved to the separate vo-list-cfg.xml file. The DGAS instances are created automatically by iterating on all defined VOs. For more information about using the new VO configuration model refer to the VO Configuration Guide and to Chapter 4 of this Installation Guide. Also all R-GMA Servicetool instances have been removed from the configuration file, since the instances are now created and configured automatically by the configuration script. The instances can still be configured manually as in previous versions if the automatic values have to be overridden.

- As root run the DGAS Client Configuration file with the --configure option `/opt/glite/etc/config/scripts/glite-dgas-client-config.py --configure.`

Once the services have been properly configured (no service will be running) it will be necessary to start them all. To do so, follow the next step.

9. As root start the DGAS Client services by running the Dgas Client Configuration File:
`/opt/glite/etc/config/scripts/glite-dgas-client-config.py --start`

13.4.3 DGAS Client Configuration Walkthrough

The Dgas Client configuration script performs the following steps:

10. Load the Dgas Client configuration file `$GLITE_LOCATION/etc/config/glite-dgas-client.cfg.xml`
11. Create the `dgas_atmClient` configuration file.
12. Create the `dgas-client.ce_pushd.gridUser` user
13. Create the `dgas_ce_pushd` configuration file.
14. Create the `dgas_gianduia` configuration file
15. Create the `getAcctLogd` configuration file
16. Create the `getAcctLogd.aciFile` file

13.4.4 Managing the DGAS Client

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

<code>glite-dgas-client-config.py --subservice</code>	This option is mainly used by services calling a sequence of clients to be configured. This option should be used with the <code>--configure</code> option. Example: <code>glite-dgas-client-config.py --subservice --configure</code>
<code>glite-dgas-client-config.py --configure</code>	Configures all DGAS CLIENT services
<code>glite-dgas-client-config.py --start</code>	Starts all DGAS CLIENT services (or restart them if they are already running, <code>pbs_mom</code>)
<code>glite-dgas-client-config.py --stop</code>	Stops all DGAS CLIENT services (<code>pbs_mom</code>)
<code>glite-dgas-client-config.py --status</code>	Checks the status of the DGAS CLIENT services

14 WORKER NODE

From gLite release 3.0 the gLite-WN metapackage contains a merge of the gLite WN and LCG WN. **Since the `glite-wn-config.py` configures only the gLite part of the worker node it is recommended to use YAIM target “WN” to configure the worker node. Any direct usage of the `glite-wn-config.py` script is not recommended and can cause unexpected misfunctionality (complete or partial) of the service.**

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 Bookkeeping 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/>).

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.

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 WN module supports the following Resource Management Systems:

- PBS, Torque
- LSF
- Condor

14.3 WORKER NODE INSTALLATION

It is possible to install the Worker Node as follows:

4. Install APT, if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite Worker Node by executing

```
apt-get install glite-wn-config
```

or

```
yum install glite-wn-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

5. This will install the following deployment modules:

- Worker Node
- R-GMA client (see section 6.3 for details)
- File Transfer Service Client (see section **Error! Reference source not found.** for details)
- I/O Client (see section 17.4 for the details)
- DGAs Client (see section for the details)
- Service Discovery (see section 7 for details)
- Security utils (see section 5 for details)

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

<i>gLite I/O Client</i>	in /opt/glite
<i>gLite LB Client</i>	in /opt/glite
<i>glite R-GMA Client</i>	in /opt/glite
<i>gLite WMS Checkpointing</i>	in /opt/glite
<i>gLite FTS client</i>	in /opt/glite
<i>gLite Service Discovery</i>	in /opt/glite
<i>gLite DGAS CLient</i>	in /opt/glite
<i>Globus</i>	in /opt/globus

The gLite Worker Node configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/glite-wn-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.

Since the WN is a collection of clients, the individual configuration files are also installed and they must be customized. Please refer to the appropriate chapters in this guide to configure the clients. All clients are configured automatically as part of the WN configuration.

14.4 WORKER NODE CONFIGURATION

1. Change to the configuration directory:

```
cd /opt/glite/etc/config
```

2. Copy the configuration file templates from the templates directory

```
cp templates/* .
```

3. Customize the configuration files by replacing the 'changeme' value in all user-defined parameters with the proper value:

- The file *glite-global.cfg.xml* contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.

- The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Refer to **Error! Reference source not found.** for the values that can be set and section 6 for the description about the R-GMA common configuration.
- The file *glite-rgma-client.cfg.xml* contains the R-GMA client specific configuration values. Refer to **Error! Reference source not found.** for the the values that can be set and section 6.3 for the description about the R-GMA client configuration.
- The file *glite-file-transfer-service-client.cfg.xml* contains the File Transfer Service client specific configuration values. Refer to Table X for the the values that can be set and section X for the description about the File Transfer Service client configuration.
- The file *glite-service-discovery.cfg.xml* contains the Service Discovery specific configuration values. Refer to Table 7 for the the values that can be set and section 7 for the description about the Service Discovery configuration.
- The file *glite-security-utils.cfg.xml* contains the security utils specific configuration values. Refer to Table 2 for the list of parameters and section 5 for the description of the security utils.
- The file *glite-wn.cfg.xml* contains the Worker Node specific configuration values. Table 16 shows the configuration values that can be set.
- No configuration is required for the DGAS Client

You will also find one or more instances for the file based service discovery. Refer to chapter 7.4 for the details about the configuration and Table 8 for the list of parameters that can be set.

Parameter	Default value	Description
User-defined Parameters		
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
custom.environment	<empty>	The entries specified in this array parameter are added to the glite environment file as additional export or setenv statements. The format of each entry must be key[space]value Example: MY_EXTRA_OPTION newvalue
System Parameters		
wn.ServiceList	<ul style="list-style-type: none"> • glite-file-transfer-service-client • glite-io-client • glite-rgma-client • glite-dgas-client • glite-lfc-client 	The gLite services, clients or applications that compose this worker node. Example: glite-rgma-client

Table 16: WN Configuration Parameters

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. Change to the script directory:

```
cd /opt/glite/etc/config/scripts
```

5. Configure the Worker Node by executing the Worker Node configuration script:

```
./glite-wn-config.py --configure
```

Running the configuration script will automatically configure the security utils, the service discovery as well as the different clients, so there is no need to run these configuration scripts as well.

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

```
The gLite Worker Node was successfully configured.
```

[New in gLite 1.5, released as a QF in gLite 1.4.1] The `glite_setenv.sh` file generated by the WN configuration script contains a protection statement to prevent the file from being running more than once. The first time the `glite_setenv.sh` file is sourced it sets the environment variable `GLITE_ENV_SET`. If this variable is set all other statements in the file are skipped. To source the file again after making modifications, it is necessary to unset the `GLITE_ENV_SET` variable from the environment.

6. Start the Worker Node:

```
./glite-wn-config.py --start
```

Check if any error message is displayed and if necessary fix the parameters values and restart the script.

7. Verify that the installation is successful by either running

```
./glite-wn-config.py --status
```

The Worker Node is completely configured and running.

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:

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

In what follows the installation instructions are given for a generic single catalog version. Whenever the steps or requirements differ for MySQL and Oracle it will be noted. Replace `gilte-data-single-catalog` with `gilte-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/>).
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 InstantClient

The Oracle Instant Client is required to run the Data Catalog (Fireman) service on Oracle. Due to license reasons, we cannot redistribute it. Version 10.1.0.3-1 can be downloaded from the Oracle Web Site.

15.3 SINGLE CATALOG INSTALLATION

1. Install APT, if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite Single Catalog by executing

```
apt-get install glite-data-single-catalog[-oracle]-config
```

or

```
yum install glite-data-single-catalog[-oracle]-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

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

gLite	in /opt/glite
Tomcat 5	in /var/lib/tomcat5

3. 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.
4. The gLite SC configuration script is installed in `$(GLITE_LOCATION)/etc/config/scripts/glite-data-single-catalog-config.py`. A template for the configuration is installed in `$(GLITE_LOCATION)/etc/config/templates/glite-data-single-catalog.cfg.xml`
5. Note that depending on the catalog you are using (Oracle or MySQL) the templates may have the oracle prefix (glite-data-single-catalog-oracle.cfg.xml for the oracle template and glite-data-single-catalog-oracle-config.py for the oracle python script). Though in the the following instructions this prefix is omitted it should be taken into account.

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 17 to 23)
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 Fireman Instance parameters		
catalog-service-fr-mysql.DBNAME		Database name used for a catalog service
catalog-service-fr-mysql.DBUSER		Database user name owning the catalog database
catalog-service-fr-mysql.DBPASSWORD		Password for accessing the catalog database
System Parameters		
catalog-service-fr-mysql.DBURL	jdbc:mysql://localhost:3306/\${catalog-service-fr-mysql.DBNAME}?zeroDateTimeBehavior=convertToNull	URL of the database

Table 17: Fireman instances configuration parameters (MySQL)

User-defined Parameters

mysql.root.password		The root password of this MySQL installation. Leave this parameter empty or remove it if no password is required. If you set this parameter, it is recommended to define it in the local service configuration file on the node, not on the public site configuration file. <i>Example: verySecret [Type: 'string']</i>
---------------------	--	--

Advanced Parameters

glite.installer.verbose	True	Enable verbose output
glite.installer.checkcerts	True	Enable check of host certificates
set.mysql.root.password	False	If this parameter is true, then the root password of the mysql database is set to the value specified in mysql.root.password if it not yet set. This parameter has no effect if the database root password is already set. It can be used to ease automated installation and configuration of the service, if mysql is not managed in some other way

allow.unsecure.port	False	<p>[Example: false][Type: boolean] Enable using the unsecure port 8080. It can be true or false. Example: false</p>
catalog-service-fr-mysql.ADMIN_VOMS_ATTRIBUTES	<empty string>	<p>ADMIN/superuser settings. Note that the extra privileges defined by the settings below are combined in an OR fashion - a VOMS match OR a mapfile match will result in the client being given the extra authorization. If both of these are empty there is no superuser defined for the service and certain functionality is unavailable (i.e. if permissions on global default permissions are not tweaked, nobody can change them or create directories in the ROOT path. If a user's certificate contains this VOMS attribute, they are additionally permitted to do any operation upon the service including metadata bits operations. If a user's certificate contains any of these VOMS attributes, they are additionally permitted to do any operation upon the service including creating channel and VO managers</p>
catalog-service-fr-mysql.ADMIN_MAPFILE	<empty string>	<p>[Example: /opt/glite/etc/config/ADMIN_VOMS_ATTRIBUTES][Type: string] 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. [Example: /opt/glite/etc/config/ADMIN_MAPFILE][Type: string]</p>
System Parameters		
Catalog-service-fr-mysql.DOCBASE	<p>/\${GLITE_LOCATION}/share/java/glite-data-catalog-service-fr-mysql.war</p>	<p>Location of the glite-data-catalog-service-fr-mysql.war file</p>

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.MESSAGINGJNDIH OST		The host of the JNDI server that contains the messaging system connection factories and topic/queue objects.
catalog-service-fr- mysql.MESSAGINGJNDIP ORT		The port of the JNDI server that contains the messaging system connection factories and topic/queue objects.
catalog-service-fr- mysql.MESSAGINGJMSNA ME		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.
rgma.servicetool.activate	True	Turn on/off servicetool for the node. [Example: true] [Type: 'boolean']"
catalog-service-fr- mysql.httpconnector_maxT hreads	150	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.
catalog-service-fr- mysql.httpconnector_minSp areThreads	25	The number of request processing threads that will be created when this Connector is first started. The connector will also make sure it has the specified number of idle processing threads available. This attribute should be set to a value smaller than that set for maxThreads
catalog-service-fr- mysql.httpconnector_maxS pareThreads	75	The maximum number of unused request processing threads that will be allowed to exist until the thread pool starts stopping the unnecessary threads
catalog-service-fr-	100	The maximum queue length for

mysql.httpconnector_accept Count		incoming connection requests when all possible request processing threads are in use. Any requests received when the queue is full will be refused
catalog-service-fr- mysql.httpconnector_conne ctionTimeout	600000	The number of milliseconds this Connector will wait, after accepting a connection, for the request URI line to be presented

Table 18: Fireman common configuration parameters (MySQL)

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.DBUSER		Database user name owning the catalog database
catalog-service-fr.DBPASSWORD		Password for accessing the catalog database
catalog-service-fr.DBHOST		Hostname of the Oracle server ex: lxfs5502.cern.ch
catalog-service-fr.DBSERVICENAME		The database service name to connect to.
Advanced Parameters		
catalog-service-fr.DBPORT	1521	TCP port of the Oracle database.
catalog-service-fr.DBURL	Jdbc:oracle:thin:@ \${catalog-service-fr.DBHOST};\${catal og-service-fr.DBPORT};\${catal og-service-fr.DBSERVICENA ME}	URL of the database. <i>Example:</i> <i>jdbc:oracle:thin:@lxfs5502.cern.ch:1521:devegee3</i>

Table 19: Fireman instances configuration parameters (Oracle)

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.MESSAGINGJNDIH OST		The host of the JNDI server that contains the messaging system connection factories and topic/queue objects.
catalog-service-fr.MESSAGINGJNDIPO RT		The port of the JNDI server that contains the messaging system connection factories and topic/queue objects.
catalog-service-fr.MESSAGINGJMSNA ME		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
catalog-service-fr.oracle-instantclient.location	/usr/lib/oracle/10.1.0.3/client/	Location of the Oracle Instantclient installation
rgma.servicetool.activate	True	Turn on/off servicetool for the node

Table 20: Fireman common configuration parameters (Oracle)

From the gLite release 1.5 the VO-specific parameters have been moved to the separate vo-list-cfg.xml file. The Fireman instances are created automatically by iterating on all defined VOs. For more information about using the new VO configuration model refer to the VO Configuration Guide and to Chapter 4 of this Installation Guide. Also all R-GMA Servicetool instances have been removed from the configuration file, since the instances are now created and configured automatically by the configuration script. The instances can still be configured manually as in previous versions if the automatic values have to be overridden.

4. As root run the Single Catalog configuration file to configure the services
`$GLITE_LOCATION/etc/config/scripts/glite-data-single-catalog-config.py --configure`
5. As root run the Single Catalog configuration file to run the services
`$GLITE_LOCATION/etc/config/scripts/glite-data-single-catalog-config.py --start`
6. 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 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 Fireman module. The instance are automatically created and configured by the Fireman configuration script, but the values can be overridden by defining the instance manually as in previous versions.

For more details about the R-GMA Service Tool service refer to section 6.4 in this guide.

16 HYDRA

16.1 SERVICE OVERVIEW

The gLite Hydra service is a special metadata catalog that services using encrypted data can use to store encryption keys.

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

16.2.2 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.

16.3 HYDRA INSTALLATION

1. Install Hydra metapackage:

```
apt-get install glite-hydra-config
```

or

```
yum install glite-hydra-config
```

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

<i>gLite</i>	in	<code>/opt/glite</code>	<code>(\$GLITE_LOCATION)</code>
<i>MySQL-server</i>	in	<code>/usr</code>	
<i>MySQL-client</i>	in	<code>/usr</code>	
<i>Tomcat</i>	in	<code>/var/lib/tomcat5</code>	

3. The gLite Hydra configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/glite-hydra-config.py.
```

A template configuration file is installed in

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

16.4 HYDRA 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 23)

- Copy the configuration file templates from

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

Table 21 shows a list of the global hydra configuration variables that can be set:

Parameter	Default value	Description
User-defined Parameters		
hydra.DBNAME		Name of Database used for the catalog service. [Example: hydra] [Type: 'string']
hydra.DBUSER		Database user name to access the catalog database. [Example: hydraUser] [Type: 'string']
hydra.DBPASSWORD		Password of database user specified in 'hydra.DBUSER'. [Example: 'verySecret'] [Type: 'string']
Advanced Parameters		
System Parameters		
hydra.DBURL	<code>jdbc:mysql://\${HOSTNAME}:3306/\${hydra.DBNAME}</code>	URL of the database. [Example: <code>jdbc:mysql://\${HOSTNAME}:3306/\${hydra.DBNAME}</code>] [Type: 'string']
hydra.PATH	<code>\${vo.name}/glite-data-hydra-service</code>	Path to the web application. [Example: <code>\${vo.name}/glite-data-hydra-service</code>] [Type: 'string']

Table 21: Hydra instances configuration parameters

Parameter	Default value	Description
User-defined Parameters		
hydra.mysql.admin.password		MySQL root password. [Example: verySecret][Type: string]
Advanced Parameters		
glite.installer.verbose	True	Enable verbose output [Example : true][Type : boolean]
glite.installer.checkcerts	True	Enable check of host certificates [Example : true][Type : boolean]
rgma.servicetool.activate	True	Turn on/off servicetool for the node. [Example: true] [Type: 'boolean']
set.mysql.root.password	False	If this parameter is true, then the root password of the mysql database is set to the value specified in mysql.root.password if it not yet set. This parameter has no effect if the database root password is already set. It can be used to ease automated installation and configuration of the service, if mysql is not managed in some other way [Example : true][Type : boolean]
allow.unsecure.port	True	Enable using the unsecure port 8080. It can be true or false [Example : true][Type : boolean]
System Parameters		
hydra.DBDRIVERCLASS	org.gjt.mm.mysql.Driver	JDBC driver classname. [Example: org.gjt.mm.mysql.Driver] [Type: 'string']
hydra.DBRESOURCE_NAME	Meta	Name of the JNDI objectcs that is holding the DB connection object. [Example: meta] [Type: 'string']
hydra.DOCBASE	\${GLITE_LOCATION}/share/java/glite-data-hydra-service.war	Location of the glite-data-catalog-service-fr-mysql.war file. [Example: \${GLITE_LOCATION}/share/java/glite-data-hydra-service.war][Type: 'string']

hydra.ATTRIBUTE_HELPER_CLASS	org.glite.data.hydra.helpers.attribute.MySQLAttributeHelper"	Name of the class (including the package name) implementing the logic for operations on attributes (getAttributes, setAttributes, etc.). [Example: org.glite.data.hydra.helpers.AttributeHelper][Type: 'string']
hydra.CATALOG_HELPER_CLASS	org.glite.data.hydra.helpers.catalog.MySQLCatalogHelper	name of the class (including the package name) implementing the logic for operations on entries (createEntry and removeEntry). [Example: org.glite.data.hydra.helpers.CatalogHelper][Type: 'string']
hydra.SCHEMA_HELPER_CLASS	org.glite.data.hydra.helpers.schema.MySQLSchemaHelper	name of the class (including the package name) implementing the logic for operations on schemas (createSchema, dropSchema, etc.). [Example: org.glite.data.hydra.helpers.SchemaHelper][Type: 'string']
hydra.AUTHORIZATION_HELPER_CLASS	org.glite.data.hydra.helpers.authorization.MySQLAuthorizationHelper	name of the class (including the package name) implementing the logic for authorization (access control) on entries in the catalog (FASBase - setPermission, getPermission, etc... plus the internal policy for creation of new entries and schemas). [Example: org.glite.data.hydra.helpers.AuthorizationHelper][Type: 'string']
hydra.schemaFile	`\${GLITE_LOCATION}/etc/glite-data-hydra-service/schema/mysql/mysql-schema.sql	Location of hydra schema file. [Example: `\${GLITE_LOCATION}/etc/glite-data-hydra-service/schema/mysql/mysql-schema.sql][Type: 'string']

Table 22: Global Hydra configuration parameters

1. Configure the R-GMA servicetool by configuring the servicetool configuration file

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

2. As root run the Hydra configuration file with the --configure option in order to configure the services

`$GLITE_LOCATION/etc/config/scripts/glite-hydra-config.py --configure`

3. As root run the Hydra configuration file with the `--start` option so that all the services are started

`$GLITE_LOCATION/etc/config/scripts/glite-hydra-config.py --start`

The Metadata Catalog is now ready.

16.5 HYDRA CONFIGURATION WALKTHROUGH

The Hydra configuration script performs the following steps:

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

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

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

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

3. Configures the gLite Security Utilities module
4. 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:
12. Creates the DB user in MySQL
13. Configures the `context.xml` in Tomcat
14. Installs the web service for the VO
15. Configures the R-GMA `servicetool` and `servicetool` instances
16. Stops MySQL server

16.6 STARTING THE HYDRA SERVICES AT BOOT

When the Hydra 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-hydra-config.py --start* command and makes sure that all necessary services are started in the correct order.

16.7 PUBLISHING HYDRA SERVICES TO R-GMA

The Hydra 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 Hydra module. The instance are automatically created and configured by the Hydra configuration script, but the values can be overridden by defining the instance manually as in previous versions.

For more details about the R-GMA Service Tool service refer to section 6.4 in this guide.

17 GLITE I/O

17.1 GLITE I/O SERVER

17.1.1 Service Overview

GLite I/O server consists basically of 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.

17.1.2 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.

17.1.2.1 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/>).
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*

17.1.2.2 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*.

17.1.3 gLite I/O Server installation

1. Install APT if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite I/O Server by executing

```
apt-get install glite-io-server-config
```

or

```
yum install glite-io-server-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

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

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

3. 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

4. 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 6.4.

17.1.4 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 4: R-GMA servicetool configuration parameters.

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 rfiid. Adapt also these configuration values accordingly. You can find more information on the values and the R-GMA servicetool in section 6.4.

- 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, it is possible to create a separate instance for each VO by copy/paste the <instance> section in this file. Alternatively it is possible to let the configuration script to create automatically on instance per VO by iterating over the VOs defined in the vo-list.cfg.xml file. For more information refer to Chapter 4 of this guide and the document VO Configuration Guide.

Common parameters

All parameters defined in this table are common to all instances.

Parameter	Default value	Description
User-defined Parameters		
<i>I/O Daemon initialization parameters</i>		
init.username		The username of the user running the I/O Daemon. If using a astor with a castor SRM, in some configurations this user must be a valid user on the Castor server. If the user doesn't exist on this I/O Server, it will be created. The uid specified in the 'init.uid' parameters may be used.
init.groupname		The groupname of the user running the I/O Daemon. If using a Castor SRM, in some configurations this group must be a valid user on the Castor server. If the group doesn't exist I/O Server, it will be created. The gid specified in the 'init.gid' parameters may be used.
init.uid		The userid of the user running the I/O Daemon. If using a Castor SRM, in some configurations the same uid of the Castor user specified in the 'init.username' parameter must be set. Leave this parameter empty or comment it out to use a system assigned uid.
init.gid		The gid of the user running the I/O Daemon. If using a Castor SRM, in some configurations the same gid of the Castor group specified in the

'init.groupname' parameter must be set. Leave this parameter empty or comment it out to use a system assigned gid.

Advanced Parameters

General gLite initialization parameters

glite.installer.verbose True
glite.installer.checkcerts True
rgma.servicetool.activate True

Enable verbose output
Enable check for host certificate
Turn on/off servicetool for the node.
[Example: true] [Type: 'boolean']

Security Utilities parameters

install.mkgridmap.cron True

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)

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.

user.certificate.path

The location of the user certificates relative to the user home directory. This parameter overrides the global one set in the glite-global.cfg.xml file

I/O Daemon parameters

io-daemon.MaxTransfers 20
io-resolve-common.SePort 8443

The maximum number of concurrent transfers
The port of the remote file operation server

io-resolve-common.RootPathRule	abs_dir	<p>The rule to be applied to define the path for creating new files. Allowed values are:</p> <ul style="list-style-type: none"> * 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>	<p>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</p>
io-authz-fas.FileGroup	<empty>	<p>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</p>
io-resolve-fireman.OverwriteOwnership	False	<p>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.</p>
io-resolve-fireman.FileOwner	<empty>	<p>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.</p>
io-resolve-fireman.FileGroup	<empty>	<p>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</p>

io-resolve-fr.OverwriteOwnership	False	string. 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		
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 the value starts with httpg://, GSI authentication will be used (using the CGI SOAP plugin), if it starts with https://, pure SSL authentication is performed, otherwise no authentication is requested. Please note that in case of a CASTOR SRM, you've always to use httpg, while in case of dCache https is required.

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. At the moment this must be set to the hostname of the SRM defined in the io-resolve-common.SrmEndPoint parameter.

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/

io-resolve-
common.SeProtocol

The protocol to be used to contact the remote file operation server. Currently the supported values are:

- * rfiio: use the remote file io (rfiio) protocol to access remotely the file
- * gsidcap: for secure access to a dCache SE
- * dcap: for unsecure access to a dCache SE
- * file: use normal posix operations to access a local file (useful only for testing purposes)

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-

The endpoint of the RMC catalog.

catalogs.RmcEndPoint

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.RIsEndPoint

The endpoint of the RIs 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-fireman.FiremanEndPoint

The endpoint of the FiReMan catalog. If that value starts with `httpg://` the GSI authentication will be used (using the CGSI GSOAP

FR parameters

io-resolve-
fr.ReplicaEndPoint

plugin); if it starts with https:// the SSL authentication will be used, using the CGI SOAP plugin in SSL compatible mode), otherwise no authentication is requested.

Example:

http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FiremanCatalog

The endpoint of the Replica catalog. If that value starts with http:// the GSI authentication will be used (using the CGI SOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGI SOAP plugin in SSL compatible mode), otherwise no authentication is requested.

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 http:// the GSI authentication will be used (using the CGI SOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGI SOAP plugin in SSL compatible mode), otherwise no authentication is requested. If that value is not set, the File Catalogs will not be contacted and the io-resolve-fr plug-in will managed only GUIDs.

Example:

http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FileCatalog

Advanced Parameters

instanceDescription \${vo.name}-
 \${init.CatalogType}

A short description of the instance used to create the different instance files

[Example: \${vo.name}-
 \${init.CatalogType}] [Type: string]

This parameter is a more general way of naming the I/O Server

		<p><i>instances. In previous releases the name was forced to be <code>#{vo.name}-#{init.CatalogType}</code>. Now this is the default value, but it can be replaced with any user string</i></p>
autocalculate.port	True	<p>If this value is true, the I/O Server port for each instance is calculated automatically starting from the value of the parameter <code>io-daemon.Port</code>. If the value is false, the <code>io-daemon.Port</code> value is taken without modifications. In this case, users must defined instance to have a different port configured in this file</p>
io-daemon.Port		<p>The port to be used to contact the server. This port is only used for authentication and session establishment messages. When the real data transfer will be performed using a QUANTA paralld TCP stream a pool of sockets are opened on the server side binding a tuple of available ports from 50000 to 51000.</p> <p>This port should not be higher than 9999 and different I/O Server instances should not run on contiguous ports (for example set one to 9999 and another one to 9989). If the parameter <code>autocalculate.port</code> is true or this parameter is absent or empty, the ports are automatically set by the configuration script following this rule and starting from 9999. If a value is given and the <code>autocalculate.port</code> parameter is true, the ports are set using the given value as port for the first instance and the other are calculated according to the rule. In all other case the value of this parameter is used without modifications</p>
log.Priority	DEBUG	<p>The <code>log4cpp</code> log level. Possible values are: <code>DEBUG</code>, <code>INFO</code>, <code>WARNING</code>, <code>ERROR</code>, <code>CRITICAL</code>, <code>ALERT</code>, <code>FATAL</code></p>

log.FileName	<pre> \${GLITE_LOCATION}/glite-io- server- \${instanceDescription}.log </pre>	The location of the log file for this instance
--------------	---	--

Table 23: gLite I/O Server Configuration Parameters

From the gLite release 1.5 the VO-specific parameters have been moved to the separate vo-list-cfg.xml file. The I/O Server instances are created automatically by iterating on all defined VOs. For more information about using the new VO configuration model refer to the VO Configuration Guide and to Chapter 4 of this Installation Guide. Also all R-GMA Servicetool instances have been removed from the configuration file, since the instances are now created and configured automatically by the configuration script. The instances can still be configured manually as in previous versions if the automatic values have to be overridden.

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 4 shows a list of the parameters that can be set. More details can be found in section 6.4.

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 with the `--configure` option in order to configure the services

```
$GLITE_LOCATION/etc/config/scripts/glite-io-server-config.py --configure
```
5. As run the gLite I/O server configuration file with the `--start` option so that all the services are started

```
$GLITE_LOCATION/etc/config/scripts/glite-io-server-config.py --start
```
6. The gLite I/O server is now ready.

17.1.5 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 IO-SERVER 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

17.2 STARTING THE I/O SERVER AT BOOT

When the I/O Server 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-io-server-config.py -start` command and makes sure that all necessary services are started in the correct order.

17.3 PUBLISHING I/O SERVER SERVICES TO R-GMA

The I/O Server 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 I/O Server module. The instance are automatically created and configured by the I/O Server configuration script, but the values can be overridden by defining the instance manually as in previous versions.

For more details about the R-GMA Service Tool service refer to section 6.4 in this guide.

17.4 CLIENT

17.4.1 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.

17.4.2 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.

17.4.2.1 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/>).

17.4.3 gLite I/O Client installation

1. Install APT if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite I/O Client by executing

```
apt-get install glite-io-client-config
```

or

```
yum install glite-io-client-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

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

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

3. 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`

17.4.4 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 24)
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	`\${HOME}/.glite-io-client-\${vo.name}.log	The location of the log file. (Note that the double \$\$ means that the \${HOME} variable is not expanded to its real value, but it's left as it is)

Parameter	Default value	Description
User-defined Parameters		
Advanced Parameters		
glite.installer.verbose	True	Enable configuration script verbose output

System Parameters

Table 24: gLite I/O Client configuration parameters

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

18 AMGA SERVER

18.1 SERVICE OVERVIEW

AMGA is a metadata service for the Grid. In a more general way this is a database access service for Grid applications which allows user jobs running on the Grid to access databases by providing a Grid style authentication as well as an opaque layer which hides the differences of the different underlying database systems from the user. To achieve this, AMGA is a service sitting between the RDBMS and the user's client application.

In addition to this database translation layer, AMGA intends to solve another problem database services face on the Grid which is latencies. AMGA intends to provide a replication layer which makes databases locally available to user jobs and replicate the changes between the different participating databases. A simple implementation based on PostgreSQL asynchronous replication is already working.

18.2 INSTALLATION PRE-REQUISITES

18.2.1 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/>).

18.2.2 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.

18.2.3 Database backend ODBC drivers

AMGA server can support 4 different database plugins (mysql, Oracle, Postgress, SQLite). As a installation prerequisite are installed unixODBC package (part of the OS distribution) and the corresponding database ODBC driver.

18.2.4 Database backend configuration

AMGA server to its operation needs a database backend. It can be based on one of following database services: MySQL, Oracle, Postgress and SQLite. Since this database is an external dependency for the AMGA server it needs to be manually configured. This consists of:

1. database creation
 2. database user creation
 3. setting access rules
1. In case of MySQL these steps are
mysql> create database <DBName>;
mysql> grant all for <DBName>.* to <DBUser>@<AMGAServerNode> identified by <DBPass>;
 - 2) In case of Oracle these steps are:
Will be added as soon as amga deployment 1.1.0 will be released
 - 3) For Postgress and SQLite databases please refer to the corresponding administrator's guide

Note: As of version 1.0.X of the gLite AMGA server deployment module, only “mysql” database backend is supported.

18.3 AMGA SERVER INSTALLATION

It is possible to install the AMGA server as follows:

1. Install APT, if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite AMGA server Node by executing

```
apt-get install glite-amga-server-config
```

or

```
yum install glite-amga-server-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

2. This will install the following deployment modules:
 1. AMGA server
 2. AMGA client
 3. R-GMA servicetool
 4. Security utils (see section 5 for details)

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

<i>AMGA Server</i>	in /opt/glite
<i>AMGA Client</i>	in /opt/glite
<i>gLite R-GMA servicetool</i>	in /opt/glite

The gLite AMGA Server configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/glite-amga-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.

Since the AMGA server consist of set of modules, the individual configuration files are also installed and they must be customized. Please refer to the appropriate chapters in this guide to configure the additional modules. All additional modules are configured automatically as part of the AMGA server configuration.

18.4 AMGA SERVER CONFIGURATION

1. Change to the configuration directory:


```
cd /opt/glite/etc/config
```
2. Copy the configuration file templates from the templates directory


```
cp templates/* .
```
3. Customize the configuration files by replacing the ‘*changeme*’ value in all user-defined

parameters with the proper value:

1. The file *glite-global.cfg.xml* contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.
2. The file *glite-amga-client.cfg.xml* contains the definition of AMGA client specific values. Refer to .. for the description
- 1) The file *glite-security-utils.cfg.xml* contains the security utils specific configuration values. Refer to Table 2 for the list of parameters and section 5 for the description of the security utils.
- 2) The file *glite-amga-server.cfg.xml* contains the definition of AMGA server specific values. Table 16 shows the configuration values that can be set.

Parameter	Default value	Description
User-defined Parameters		
amga.server.DBUser		The user with which the server will contact the database backend.
amga.server.DBPass		The password the server will give when contacting to the database backend.
Amga.server.DBName		The database name created on the database server
Amga.server.DBHost		The host name on which the database server is running
Amga.server.DBSource	Mysql	Database backend type. Due to restrictions in the deployment script only mysql database backend is supported.
Advanced Parameters		
glite.installer.verbose	True	Enable verbose output
amga.server.Port	8822	The number of the port the server will listen on.
amga.server.MinProcesses	2	This is the minimum number of processes waiting for client connections the server must offer. When the server starts up or there are no client connections for some time, MinProcesses is the number of processes spawned waiting for connections.
amga.server.MaxProcesses	20	This is the maximum number of processes the server will spawn in total. The server always tries to have 1/3 of the processes in the awaiting connection state. To achieve this, the server will spawn new processes until the number of

		MaxProcess is reached. Please make sure that your database backend can support as many client connections.
amga.server.MaxConnectsPerProcess	"	To prevent any very rare memory leaks or other resource leaks to reduce the stability of the service, server processes can be asked to terminate themselves after serving a certain number of connections.
amga.server.Sessions	Allow	This allows sessions. Sessions create an overhead on the protocol if they are enforced, so the performance of individual clients may reduce while you will be able to support more clients which share the available connections (there is a maximum of MaxProcesses connections, if they are all hogged by a client, then no new clients will be able to connect). Such a denial-of-service situation can be prevented by forcing sessions. Values are: no, allow, force.
amga.server.IdleTimeout	20 min	Timeout for an idle connection (that is a connection that waits for a client command) in seconds. There are no timeouts currently for database queries apart from how the database is configured.
amga.server.SessionTimeout	1 day	Timeouts for session. The lifetime of a session in seconds.
amga.server.UseSSL	1	Whether the server will offer SSL as a connection protocol. This is also required to allow certificate based authentication and if you want to use passwords this is recommended if you want to be sure no one listens in. Note that you cannot force the client to use an SSL connection.
amga.server.RequireAuthentication	No	Whether users need to be authenticated.
amga.server.AllowCertificateAuthentication	No	Whether you allow users to authenticate with their certificate. The CA are automatically installed on the gLite installation and corresponding parameters like (TrustedCertDir, etc) are set. See

<p>amga.server.AllowPassword Authentication</p>	<p>No</p>	<p>AMGA server users guide “User Management” (p. 15). Allow authentication with a password. You need a user manager module running for this to work. See AMGA server users guide “User Management” (p. 15).</p>
<p>amga.server.AllowGridProxy Login</p>	<p>No</p>	<p>Whether you allow users to authenticate with a proxy certificate.</p>

Table 25: AMGA server Configuration Parameters

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. Change to the script directory:

```
cd /opt/glite/etc/config/scripts
```

5. Configure the AMGA server by executing the AMGA server configuration script:

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

Running the configuration script will automatically configure the security utils and the AMGA client, so there is no need to run these configuration scripts as well.

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

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

6. Start the AMGA server:

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

Check if any error message is displayed and if necessary fix the parameters values and restart the script.

7. Verify that the installation is successful by either running

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

The AMGA server is completely configured and running.

19 AMGA CLIENT

19.1 SERVICE OVERVIEW

CLI and C++ client to the AMGA server

19.2 INSTALLATION PRE-REQUISITES

19.2.1 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/>).

19.3 AMGA CLIENT INSTALLATION

It is possible to install the AMGA client as follows:

1. Install APT, if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite AMGA server Node by executing

```
apt-get install glite-amga-client-config
```

or

```
yum install glite-amga-client-config
```

[New in gLite 3.0] Starting from gLite release 3.0 the installation via gLite installer scripts is not supported.

2. This will install the following deployment modules:
 1. AMGA client
 2. Security utils (see section 5 for details)

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

AMGA Client in `/opt/glite`

The gLite AMGA Client configuration script is installed in

```
$GLITE_LOCATION/etc/config/scripts/glite-amga-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.

Since the AMGA client consist of set of modules, the individual configuration files are also installed and they must be customized. Please refer to the appropriate chapters in this guide to configure the additional modules. All additional modules are configured automatically as part of the AMGA client configuration.

19.4 AMGA CLIENT CONFIGURATION

1. Change to the configuration directory:

```
cd /opt/glite/etc/config
```


2. Copy the configuration file templates from the templates directory `cp templates/*`.
3. Customize the configuration files by replacing the `'changeme'` value in all user-defined parameters with the proper value:
 - The file `glite-global.cfg.xml` contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.
 - The file `glite-security-utils.cfg.xml` contains the security utils specific configuration values. Refer to Table 2 for the list of parameters and section 5 for the description of the security utils.
 - The file `glite-amga-client.cfg.xml` contains the definition of AMGA client specific values. Table 16 shows the configuration values that can be set.

Parameter	Default value	Description
User-defined Parameters		
<code>amga.client.Host</code>		The name of the host to connect to. This option can be overridden on the command line of <code>mdclient</code> . (Default: localhost)
<code>amga.client.Login</code>		The login name of the user on the AMGA server. All entries created in the catalogue will have this owner. This is also the user which you need to authenticate to the AMGA server if authentication is enabled. (Default: NULL which gives the default role when authenticating with a VO certificate)
Advanced Parameters		
<code>glite.installer.verbose</code>	True	
<code>amga.client.Port</code>	8822	Port of the <code>mdserver</code> to connect to. Can be overridden on the command line of <code>mdclient</code> using the <code>-p</code> option.
<code>amga.client.PermissionMask</code>	<code>rw-</code>	A 3 character string giving the owner permissions of newly created entries in the metadata catalogue.
<code>amga.client.GroupMask</code>	<code>r--</code>	A 3 character string giving the group permissions of newly created entries in the metadata catalogue.
<code>amga.client.Home</code>	<code>/</code>	The home-directory.
<code>amga.client.UseSSL</code>	Require	Possible values are <code>no</code> , <code>try</code> , <code>require</code> (synonym is <code>yes</code>). Default is <code>no</code> . Needed for any authentication

amga.client.AuthenticateWith Certificate	No	<p>using certificates (also proxy certificate). You want this if you intend to use passwords which are not sent in plain text. If you use SSL the entire session will be encrypted. Some servers may require you to use SSL to connect. If you want to be sure that SSL is always used you need to set this to require or yes.</p> <p>Set this to 1 to enable certificate based authentication, also grid-proxy certificates. You will need to either enable normal certificates via a Cert-File, KeyFile option pair, or use a grid proxy certificate via the UseGridProxy option. If you specify both, then the grid proxy gets precedence.</p>
amga.client.UseGridProxy	No	<p>Tries to use the a grid proxy certificate in /tmp/x509up_u[user-id].</p>
amga.client.VerifyServerCert	No	<p>Verifies the server certificate against CA certificates.</p>
amga.client.RequireDataEncryption	No	

Table 26: AMGA server Configuration Parameters

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. Change to the script directory:
`cd /opt/gLite/etc/config/scripts`
5. Configure the AMGA client by executing the AMGA client configuration script:
`./glite-amga-client-config.py --configure`
 Running the configuration script will automatically configure the security utils and the AMGA client, so there is no need to run these configuration scripts as well.
 Check if any error message is displayed and if necessary fix the parameters values and restart the script. If the configuration is successful you should see at the end the message:
The gLite AMGA client was successfully configured.

The AMGA client is completely configured

6. Before the usage of the AMGA client, the following link must be created:

In -s \$GLITE_LOCATION/etc/mdclient.config \$HOME/.mdclient.config

20 USER INTERFACE

From glite release 3.0 the glite-UI metapackage contains a merge of the gLite UI and LCG UI. Since the `glite-ui-config.py` configures only the gLite part of the worker node it is recommended to use YAIM target 'UI' to configure the worker node. Any direct usage of the `glite-ui-config.py` script can cause unexpected misfunctionality of the service.

20.1 SERVICE OVERVIEW

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

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

20.2 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.

20.2.1 Security Settings

A security module called *glite-security-utils* is installed and configured automatically by <http://www.glite.org/> by the UI. The module contains a number of certificate and security utilities. The latest version of the CA certificates should be installed manually. 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.

20.2.2 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.

20.3 UI INSTALLATION

From the gLite release 3.0 the gLite User Interface can be installed only as root.

It is possible to install the glite user interface as follows:

1. Install APT, if not yet installed following the instructions at <http://glite.web.cern.ch/glite/packages/APT.asp> and install the gLite user interface by executing

```
apt-get install glite-UI
```

or

```
yum install glite-UI
```

2. If the installation is performed successfully, the following components are installed:
 - gLite* in /opt/glite (= GLITE_LOCATION)
 - Globus* in /opt/globus (= GLOBUS_LOCATION)
 - GPT* in /opt/gpt (= GPT_LOCATION)
3. The script will install the following deployment modules:
 - R-GMA client (see section 6.3 for details)
 - File Transfer Service Client (see section **Error! Reference source not found.** for details)
 - File Placement Service Client (see section **Error! Reference source not found.** for details)
 - Service Discovery (see section 7 for details)
 - Security utils (see section 5 for details)

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

<i>gLite I/O Client</i>	in /opt/glite
<i>gLite LB Client</i>	in /opt/glite
<i>glite R-GMA Client</i>	in /opt/glite
<i>glite DGAS Client</i>	in /opt/glite
<i>gLite WMS Checkpointing</i>	in /opt/glite
<i>gLite FTS client</i>	in /opt/glite
<i>gLite Service Discovery</i>	in /opt/glite
<i>Globus</i>	in /opt/globus

4. The gLite User Interface configuration script is installed in


```
$GLITE_LOCATION/etc/config/scripts/glite-ui-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.

20.4 UI CONFIGURATION

1. Change to the configuration directory:


```
cd /opt/glite/etc/config
```
2. Copy the configuration file templates from the templates directory


```
cp templates/* .
```
3. Customize the configuration files by replacing the '*changeme*' value in all user-defined parameters with the proper value:
 - The file *glite-global.cfg.xml* contains global configuration values. Refer to Table 1 for the values that can be set and section 4.2.3 for the description about the general configuration.
 - The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Refer to **Error! Reference source not found.** for the values that can be set and section 6 for the description about the R-GMA common configuration.
 - The file *glite-rgma-client.cfg.xml* contains the R-GMA client specific configuration values. Refer to **Error! Reference source not found.** for the the values that can be set and section 6.3 for the description about the R-GMA client configuration.

- The file *glite-file-transfer-service-client.cfg.xml* contains the File Transfer Service client specific configuration values. Refer to Table X for the the values that can be set and section X for the description about the File Transfer Service client configuration.
- The file *glite-service-discovery.cfg.xml* contains the Service Discovery specific configuration values. Refer to Table 7 for the the values that can be set and section 7 for the description about the Service Discovery configuration.
- The file *glite-security-utils.cfg.xml* contains the security utils specific configuration values. Refer to Table 2 for the list of parameters and section 5 for the description of the security utils.
- The DGAS Client doesn't require any configuration.
- The file *glite-ui.cfg.xml* contains the User Interface specific configuration values. The configuration file contains:
 - one common <parameters> section used for all Vos. Table 28 shows the common configuration values that can be set.
 - one or more <set> sections, one per each VO that the UI must be configured for. Table 27 shows the parameters per VO.
 - one or more instances for the file based service discovery. Refer to chapter 7.4 for the details about the configuration and Table 8 for the list of parameters that can be set.

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.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
ui.wms-proxy.endpoints		List of endpoints URL of WMPProxy to contact. [Example: https://ghemon.cnaf.infn.it:7443/glite_wms_wmproxy_server] [Type: 'string']
ui.MyProxyServer		MyProxy server to use

ui.HLRLocation	<p>Location of the HLR accounting server. Optional parameter. The syntax is hostname:port: and default port is 56568</p> <p>[Example: lxb0001.cern.ch:56568:]</p> <p>[Type: string]</p>
----------------	---

Table 27: UI VO specific configuration parameters – defined in one or several <set> sections

Parameter	Default value	Description
User-defined Parameters		
py-ui.DefaultVo		Default VO to connect
lfc.server		LFC server. [Example: lxb0755.cern.ch] [Type: string]
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
py-ui.requirements	other.GlueCEStateStatus == 'Production'	Requirements for job matchmaking for this VO
py-ui.rank	-	Matchmaking rank.
	other.GlueCEStateEstimatedResponseTime	
py-ui.RetryCount	3	Number of retries.
py-ui.ErrorStorage	\$\$GLITE_LOCATION_TMP}/glite-ui	Storage of the errors.
py-ui.OutputStorage	\$\$GLITE_LOCATION_TMP}/glite-ui	Storage of the output.
py-ui.ListenerStorage	\$\$GLITE_LOCATION_TMP}/glite-ui	Storage of the outputs.
py-ui.LoggingTimeout	10	Timeout for logging.
py-ui.LoggingSyncTimeout	10	Timeout for logging synchronization.
py-ui.NSLoggerLevel	1	Level of the NS Logger.
py-ui.DefaultStatusLevel	1	Default status level.
py-ui.DefaultLogInfoLevel	1	Default level of logging.

<p>wmproxy.ShallowRetry Count</p>	<p>10</p>	<p>Maximum number of shallow job re-submissions to be done in case of job failure. If this parameter is empty or missing a default value of 10 is used. [Example: 10][Type: integer]</p>
<p>wmproxy.AllowZippedI SB</p>	<p>true</p>	<p>When set to true makes the WMS client commands archive and compress all job input sandbox files into a single tar, gzipped file that is then transferred to the WMS. If this parameter is empty or missing a default value is true. [Example: true][Type: boolean]</p>
<p>wmproxy.PerusalFileEn able [Changed in gLite 3.0]</p>	<p>false</p>	<p>When set to true enables the job file perusal support in the WMS. If this parameter is empty or missing a default value is true. [Example: true][Type: boolean]</p>
<p>ui.ClientList</p>	<ul style="list-style-type: none"> • glite-file-transfer-service-client • glite-io-client • glite-rgma-client • glite-lfc-client 	<p>The gLite clients or applications that compose this user interface. [Type: 'string'] Example: glite-rgma-client</p>

System Parameters

Table 28: UI common configuration parameters

4. Run the UI configuration file

```
$GLITE_LOCATION/etc/config/scripts/glite-ui-config.py
```

The gLite User Interface is now ready.

20.5 CONFIGURATION FOR THE UI USERS

To get the environment configured correctly, each gLite UI user MUST run the

```
$GLITE_LOCATION/etc/config/scripts/glite-ui-config.py
```

configuration script before using the glite UI for the first time.

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.

20.6 NOTE

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

- 3) to source the *glite_setenv.[sh|csh]* file in */etc/glite/profile.d/* or *\$HOME/.glite* directory depending on the type of installation
- 4) log off and log in. The file with UI environment variables will be sourced automatically.

21 THE GLITE FUNCTIONAL TEST SUITES

Functional test suites in gLite release 3.0 are supported only partially.

21.1 OVERVIEW

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

21.2 I/O TEST SUITE

21.2.1 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.

21.2.2 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.

21.2.3 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.

21.2.4 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.

21.2.5 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)

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

21.3 CATALOG TEST SUITE

21.3.1 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.

21.3.2 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.

21.3.3 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.

21.3.4 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

21.3.5 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)

21.3.6 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.

21.4 WMS TEST SUITE

21.4.1 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.

21.4.2 Installation Pre-requisites

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

21.4.3 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.

21.4.4 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`.

21.4.5 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/glite-wms/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/glite-wms/opt/edg/tests/etc/config_tests_conf
/regressionTest.reg RTest | tee MyLogFile
```

21.4.6 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.

21.5 WMS VALIDATION TEST SUITE

21.5.1 Test suite description

The WMS validation test suite currently consists of a single regression test for bug number 8663.

21.5.2 Installation Pre-requisites

The WMS test suite depends on the VOMS and WMS client being there, and has been designed to be executed from a UI machine.

21.5.3 Installation

This test suite is installed using glite-testsuites-wms-validation 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 test under \$GLITE_LOCATION/test/bin directory.

21.5.4 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)
LD_LIBRARY_PATH (including: $GLITE_LOCATION/lib:$GLOBUS_LOCATION/lib)
PATH (including: $GLITE_LOCATION/bin:$GLOBUS_LOCATION/bin)
```

The user should be authorized to execute a job on the grid.

Also, the user must have a voms-proxy to run the tests in batch mode, typing: voms-proxy-init -voms your_vo_name. If a voms proxy cannot be found the test will try to create one, prompting for the certificate passphrase.

21.5.5 Execution

You can run the tests from the command line, executing the binary:

```
$GLITE_LOCATION/test/bin/job-list-match-bug-8663-test.sh [OPTIONS]
```

The test will perform a series of glite-job-list-match for a configurable amount of time, with a configurable time step.

The parameters that can be set from the command line are:

- the time one wants the test to last (with -t)

- the time one wants the test to sleep between successive matches (with -s)
- the VO name (with -v)
- the parent directory where one wants the directory containing the results (with -d, this parameter is optional, the default being the directory from which the test is executed)

The test tries to find a computing element for a very simple jdl, with no requirements, it is just the echo of “Hello World”, and so the match returns the list of all CEs available at that time.

21.5.6 Test results

The result of the test is a pdf file showing a plot of the available CEs during the time of the test. It also stores the file called “matched_sites.out” on which the plot is based showing the number of matching CEs as a function of time, and a file called “matched_sites.txt” giving the names of the CEs with attached queues as a function of time.

21.6 R-GMA TEST SUITE

21.6.1 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?).

21.6.2 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.

21.6.3 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.

21.6.4 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"; one of them, `TEST_API`, selects the R-GMA API source code to use. The valid values are `CPP`, `C` (default) and `JAVA`. There are additional parameters to allow timings to be adjusted if tests fail due to very slow systems causing timeouts. You should not normally need to change these.

21.6.5 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.

21.6.6 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.

22 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>

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

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

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

      <value>EGEE</value>

    </voms.voname>

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

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

    </voms.vomsnode>

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

      <value>17001</value>

    </voms.vomsport>

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

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

    </voms.voms-certsub>

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

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

      <value>egee</value>

    </pool.account.basename>

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

      <value>egee-gr</value>

    </pool.account.group>
```

```

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

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

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

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

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

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

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

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

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

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

    <cemon.queues
description="A space-separated list of the queues defined on this CE node
Example: blah-pbs-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,
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"/>
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
<!-- Advanced parameters - Change them if you know what you're doing -->
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
<!-- gLite configuration -->
<glite.installer.verbose
description="Enable verbose output"
value="true"/>
<glite.installer.checkcerts
description="Enable check of host certificates"
value="true"/>
<!-- PBS configuration
The following parameters are specific to PBS. They may have to be set
depending on your local PBS configuration.
If PBS is not used, remove this section -->
<PBS_SPOOL_DIR
description="The PBS spool directory"
value="/usr/spool/PBS"/>
<!-- LSF configuration
The following parameters are specific to LSF. They may have to be set
depending on your local LSF configuration.
If LSF is not used, remove this section -->
<LSF_CONF_PATH
description="The directory where the LSF configuration file is located"
value="/etc"/>
<!-- Globus configuration -->
<globus.osversion
description="The kernel id string identifying the system installed on this node.
For example: '2.4.21-20.ELsmp'. This parameter is normally automatically detected,
but it can be set here"
value=""/>
<globus.hostdn
description="The host distinguished name (DN) of this node. This is normally
automatically
read from the server host certificate. However it can be set here. For example:
'C=ORG, O=DOMAIN, OU=GRID, CN=host/server.domain.org'"
value=""/>

```

```

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

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

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

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

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

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

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

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

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

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

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

</parameters>
</config>

```


23 SITE CONFIGURATION FILE EXAMPLE

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

```
<config/>
```

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

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

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

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

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

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

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

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

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

  </parameters>

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

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

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

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

        <value>EGEE</value>

      </voms.voname>

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

```

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

</voms.vomsnode>

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

  <value>15001</value>

</voms.vomsport>

<voms.vomscertsbj
  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.vomscertsbj>

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

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

  <value>egee</value>

</pool.account.basename>

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

  <value>egee</value>

</pool.account.group>

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

  <value>40</value>

</pool.account.number>

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

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

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

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

```



```

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

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

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

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

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

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

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

</cemon.cesebinds>

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

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

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

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

  <value></value>

</pool.lsfgid>

-->

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

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

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

```

chain.

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

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

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

<condor.blahpollinterval

```

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

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

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

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