Installation and Configuration Guide

v. 1.5 (rev.4)

07 February 2006

TABLE OF CONTENT

1 INTRODUCTION

1.1 PURPOSE

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

1.2 TERMINOLOGY

Glossary

CE	Computing Element
FTA	File Transfer Agents
FTS	File Transfer Service
LB	Logging and Bookkeping
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
- DPM
- File & Replica Catalog (called Single Catalog in this release – SC)
- File Transfer Agents (FTA)
- File Transfer Service (FTS)
- gLite I/O Server and Client
- GPBox
- Hydra
- LFC
- 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).

- o 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
  - o Data is stored in an SRM based storage system.
  - o The gLite I/O Server allows posix-like access to files stored in the SRM
  - o The Single Catalog (SC) keeps track of the LFN:GUID:SURL (Logical File Names:Grid User Id:Storage URL) mapping of local files
  - o The Local Transfer Service and the File Transfer Agents provides file transfer/file placement service (FTS/FPS) and it is used for moving files.
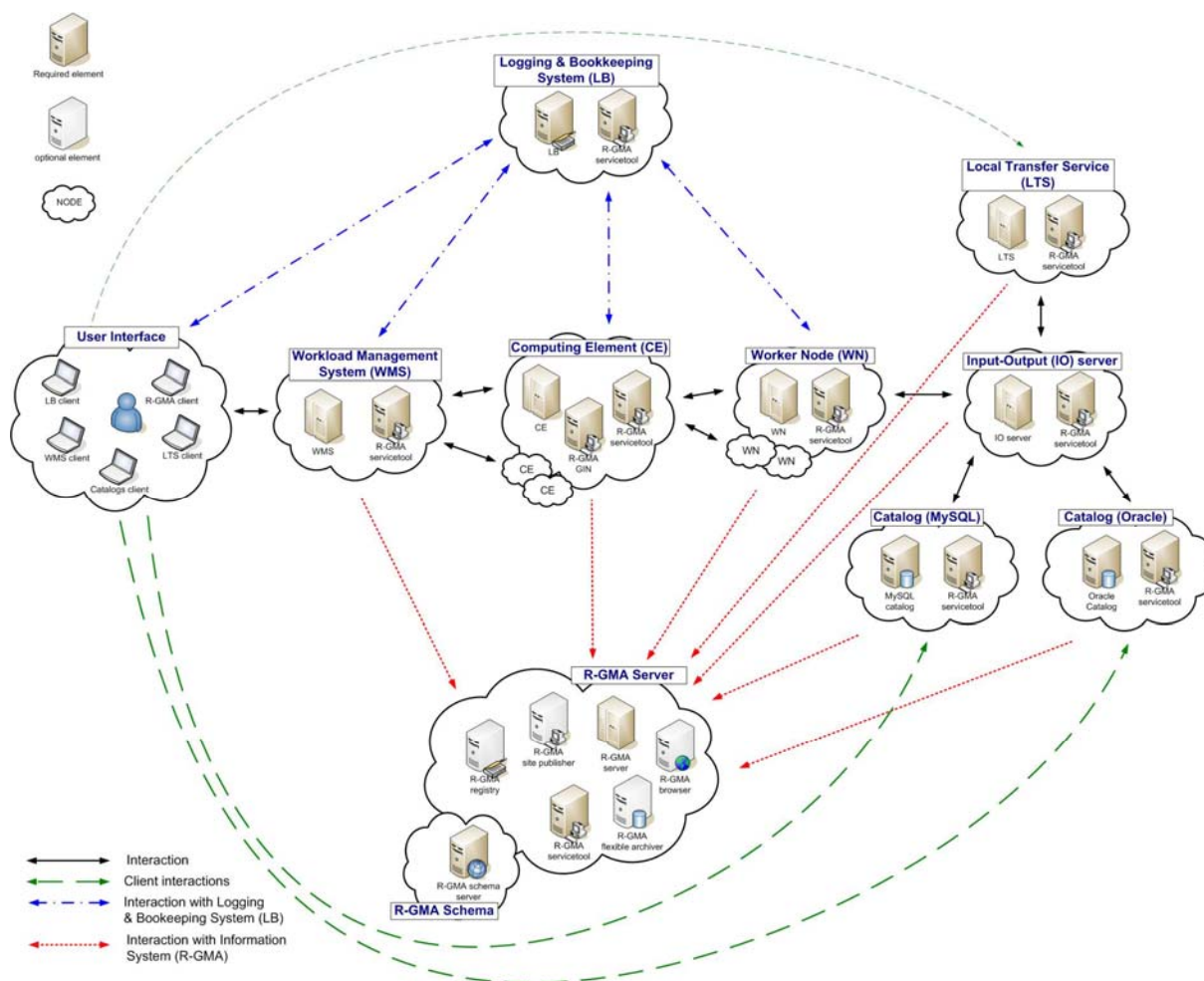
Figure 1: gLite Service Deployment Scenario

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

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

Instead of the distributed deployment of the catalogs (a local catalog and a global catalog) a centralized deployment of just a global catalog can be considered as well. This is actually the configuration supported in the gLite 1.2.

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

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

> jra1mw.cvs.cern.ch:/cvs/jra1mw

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

4   THE GLITE CONFIGURATION MODEL

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

Each module contains at least the following configuration RPMS:

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

In addition, a mechanism to load remote configuration files from URLs is provided. Refer to the Site Configuration section later in this chapter (4.3.4).

4.1   THE GLITE CONFIGURATION SCRIPTS

All configuration scripts are installed in:

$GLITE_LOCATION/etc/config/scripts

where $GLITE_LOCATION is the root of the gLite packages installation. 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 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

4.3   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

4.3.1   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)*

4.3.2 The Global Configuration File

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

Currently the global configuration file defines the following parameters:

Parameter	Default value	Description
User-defined Parameters		
site.config.url		The URL of the Site Configuration file for this node. The values defined in the Site Configuration file are applied first and are be overridden by values specified in the local configuration files. Leave this parameter empty or remove it to use local configuration only.
Advanced Parameters		
GLITE_LOCATION	/opt/glite	
GLITE_LOCATION_VAR	/var/glite	
GLITE_LOCATION_LOG	/var/log/glite	
GLITE_LOCATION_TMP	/tmp/glite	
GLOBUS_LOCATION	/opt/globus	Environment variable pointing to the Globus package.
GPT_LOCATION	/opt/gpt	Environment variable pointing to the GPT package.
JAVA_HOME	/usr/java/j2sdk1.4.2_08	Environment variable pointing to the SUN Java JRE or J2SE package.
CATALINA_HOME	/var/lib/tomcat5	Environment variable pointing to the Jakarta Tomcat package
host.certificate.file	/etc/grid-security/hostcert.pem	The host certificate (public key) file location
host.key.file	/etc/grid-security/hostkey.pem	The host certificate (private key) file location

ca.certificates.dir	/etc/grid-security/certificates	The location where CA certificates are stored
user.certificate.path	.certs	The location of the user certificates relative to the user home directory
host.gridmapfile	/etc/grid-security/grid-mapfile	Location of the grid mapfile
host.gridmap.dir	/etc/grid-security/gridmapdir	The location of the account lease information for dynamic allocation
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

4.3.3 The VO List File

gLite 1.5 introduces 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

4.3.4 The Site Configuration File

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

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

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

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

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

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.

4.3.5   Internal Configuration

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

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

4.3.6 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 logins in various ways (interactive, non-interactive or script).

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

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

4.3.7 Default Environment Variables

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

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

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

4.3.8 Configuration Overrides

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

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

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

5   GLITE SECURITY UTILITIES

5.1   OVERVIEW

The gLite Security Utilities module contains the CA Certificates distributed by the EU Grid PMA. In addition, it contains a number of utilities scripts needed to create or update the local grid mapfile from a VOMS server and periodically update the CA Certificate Revocation Lists. This module is presented first, since it is used by almost all other modules. However, it is not normally installed manually by itself, but automatically as part of the other modules.

5.1.1   CA Certificates

The CA Certificate are installed in the default directory

/etc/grid-security/certificates

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

5.1.2   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 an 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 by executing

    apt-get install glite-security-utils-config

2. *Installation via gLite installer scripts*
    a. Download from the gLite web site the latest version of the the gLite Security Utilities installation script *glite-security-utils_installer.sh*. Make the file executable (*chmod u+x glite-security-utils_installer.sh*) and execute it.
    b. Run the installation script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-security-utils next to the installation script and the installation procedure is started. If some RPMS are already installed, they upgraded if necessary. Check the screen output for errors or warnings.

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

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

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 4.3.2 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		
cron.mailto		E-mail address to which the stderr of the installed cron jobs is sent
Advanced Parameters		
glite.installer.verbose	true	Produce verbose output when running the script
glite.installer.checkcerts	true	Activate a check for host certificates and stop the script if not available. The certificates are looked for in the location specified by the global parameters *host.certificate.file* and *host.key.file*
fetch-crl.cron.tab	00 */6 * * *	The cron tab to use for the fetch-crl cron job.
install.fetch-crl.cron	true	Install the glite-fetch-crl cron job. Possible values are 'true' (install the cron job) or 'false'

		(do not install the cron job)
fetch-crl.script **[New in gLite 1.5]**	${EDG_LOCATION}/sbin/edg-fetch-crl	The full path of the fetch crl script script.
fetch-crl.cron.random.delay	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.
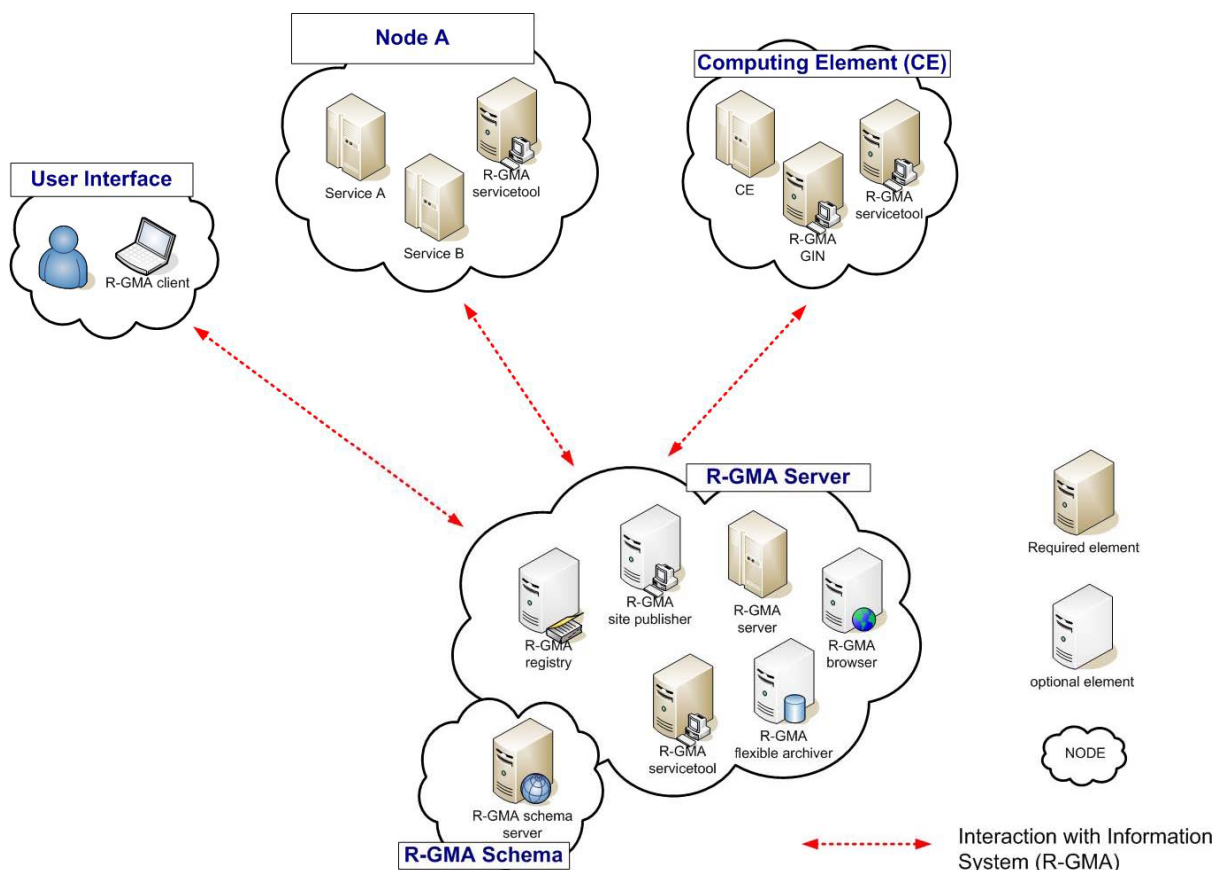
Figure 2 R-GMA components

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.

Deployment module	Contains	Used / included by
R-GMA server	R-GMA server	
	R-GMA registry server	
	R-GMA schema server	
	R-GMA browser	
	R-GMA site publisher	
	R-GMA archiver	
	R-GMA servicetool	
R-GMA client	RGMA client APIs	Service Discovery (SD) (Chapter 7)
		Worker Node (WN) (Chapter 13)
		User Interface (UI) (Chapter 20)
R-GMA servicetool	R-GMA servicetool	R-GMA server
		VOMS Server *(Chapter 8)*
		Logging & Bookkeeping *(Chapter 9)*
		Workload Management System *(Chapter 10)*
		Torque Server *(Chapter 11)*
		Computing Element *(Chapter 12)*
		Data Catalog *(Chapter 15)*
		File Transfer Service *(Chapter 16)*
		File Transfer Agents *(Chapter 17)*
		Hydra *(Chapter 18)*
		I/O-Server *(Chapter 19)*
R-GMA GIN	R-GMA GadgetIN	Computing Element *(Chapter 12)*

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

6.2.1   R-GMA Server deployment module overview

The R-GMA server is the central server of the R-GMA service infrastructure. It contains the four R-GMA server parts – server, schema, registry and browser (see section 6.1.1) as well as the R-GMA clients – R-GMA servicetool, site publisher and archiver (see section 6.1.2):

- The *R-GMA server* is always turned on to enable the server capabilities.

- For the *R-GMA schema,* the user can decide to turn it on or off. Remember there must be exactly one central schema server for the grid.

- For the *R-GMA registry,* the user can decide to turn it on or off – there can be one or more registries per grid.

- For the *R-GMA browser*, the user can decide to turn it on or off – it is recommended to turn it on to give the user the possibility to browse the information of the information system via a web browser.

- The *R-GMA servicetool* is always turned on to publish the server services.

- For the *R-GMA site publisher,* the user can decide to turn it on or off. There must be one site publisher enabled per site. If you run multiple R-GMA server on one site, turn on the site publisher for one server and off for the others.

- For the *R-GMA data archiver (flexible archiver)*, the user can decide to turn it on or off*.*

6.2.2   Installation Pre-requisites

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

6.2.2.1   Security Settings

The R-GMA server needs the list of Certificate Authorities as well as a host certificate:

Install one or more Certificate Authorities certificates in /etc/grid-security/certificates. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (http://www.gridpma.org/).

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

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

6.2.2.2 Java JRE/JDK

The Java JRE or JDK are required to run the R-GMA Server. This release requires v. 1.4.2 revision 08. The JDK/JRE version to be used is a parameter in the gLite global configuration file. Please change it according to your version and location.

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

6.2.3 R-GMA Server Installation

It is possible to install the R-GMA server 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 R-GMA server by executing

    > *apt-get install glite-rgma-server-config*

2. *Installation via gLite installer scripts*
    1. Download the latest version of the R-GMA server installation script

        > *glite-rgma-server_installer.sh*

        from the gLite web site. It is recommended to download the script in a clean directory.
    2. Make the script executable

        > *chmod u+x glite-rgma-server_installer.sh*

        and execute it or execute it with

        > *sh glite-rgma-server_installer.sh*

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

This will install the following deployment modules:

- R-GMA server
- R-GMA servicetool (see section 6.4 for details)
- 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
MySQL-server	in /usr
MySQL-client	in /usr
Tomcat	in /var/lib/tomcat5

The gLite R-GMA server configuration script is installed in

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

All the necessary template configuration files are installed into

$GLITE_LOCATION/etc/config/templates/

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

6.2.4 R-GMA 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/* .

For the configuration of the R-GMA server you don't need the configuration file *glite-rgma-servicetool-externalServices.cfg.xml* that is installed as part of the R-GMA Servicetool. This file is used to publish non-gLite services in R-GMA (for example myproxy or SE nodes) and must be therefore be used on the nodes to be published and not on the R-GMA Server itself. You can either delete it from the present directory or ignore it in the following instructions as it will not be taken into account.

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 4.3.2 for the description about the general configuration.

• The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values.Table 4 shows the configuration values that can be set.

• The file *glite-rgma-server.cfg.xml* contains the R-GMA server specific configuration values.

rgma.server.httpconnector_ maxPostSize	0	The maximum size in bytes of the POST which will be handled by the container FORM URL parameter parsing. The feature can be disbled by setting this attribute to a value inferior or equal to 0. If not specified, this attribute is set to 2097152 (2 megabytes). [Type: 'integer'] *Example: 0*

• Table 5 shows the configuration values that can be set.

- The file *glite-rgma-servicetool.cfg.xml* contains the R-GMA servicetool specific configuration values. Refer to Table 7 for the list of parameters that can be set and section 6.4 for the description of the R-GMA servicetool.
- 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.

Parameter	Default value	Description
User-defined parameters		
rgma.server.hostname		Hostname of the R-GMA server. [Type: 'string'] *Example: lxb1420.cern.ch*
rgma.schema.hostname		Host name of the R-GMA schema service. [Type: 'string'] *Example: lxb1420.cern.ch* (See also configuration parameter 'rgma.server.run_schema_service' in the R-GMA server configuration file in case you install a server)
rgma.registry.hostname		Host name(s) of the R-GMA registry service. You must specify at least one. You can also specify a different port for a registry (e.g. lxb1420.cern.ch:8383) ***[Change in gLite 1.5: For the moment only one registry is allowed to be set]*** [Type: 'string'] *Example: lxb1420.cern.ch* (See also configuration parameter 'rgma.server.run_registry_service' in the R-GMA server configuration file in case you install a server).
Advanced Parameters		
rgma.secure.mode	true	Run R-GMA clients in secure mode (true|false). If you want to run the R-GMA clients in unsecure mode, make sure the R-GMA server is able to accept requests on the unsecure port by setting the corresponding 'allow.unsecure.port' to 'true' in the R-GMA server configuration. [Type: 'boolean'] *Example: true*
System Parameters		
rgma.user.name	rgma	Name of the user account used to run the R-GMA gLite services. [Type: 'string'] *Example: rgma*
rgma.user.group	rgma	Group of the user specified in the parameter 'rgma.user.name'. [Type: 'string']

Example: rgma

Table 4: R-GMA common configuration parameters

Parameter	Default value	Description
User-defined Parameters		
rgma.server. run_schema_service **[Modified in gLite 1.5, the possible values are now true|false instead of yes|no. However, the old format is still supported for compatibility]**		Run a schema server by yourself (true|false). If you want to run it on your machine set 'true' and set the parameter 'rgma.schema.hostname' to the hostname of your machine otherwise set it to 'false' and set the 'rgma.schema.hostname' to the host name of the schema server you want to use. [Type: 'boolean'] *Example: true*
rgma.server. run_registry_service **[Modified in gLite 1.5, the possible values are now true|false instead of yes|no. However, the old format is still supported for compatibility]**		Run a registry server by yourself (yes|no). If you want to run it on your machine set 'yes' and add your hostname to the parameter list 'rgma.registry.hostname' otherwise set it to 'no'. [Type: 'boolean'] *Example: true*
rgma.server. run_browser **[Modified in gLite 1.5, the possible values are now true|false instead of yes|no. However, the old format is still supported for compatibility]**		Run an R-GMA browser (yes|no). Running a browser is optional but useful. [Type: 'boolean'] *Example: true*
rgma.server. run_archiver **[Modified in gLite 1.5, the possible values are now true|false instead of yes|no. However, the old format is still supported for compatibility]**		Run the R-GMA data archiver (yes|no). Running an archiver makes the data from the site-publisher, servicetool and GadgetIN constantly available. If you turn on this option, by default the glue and service tables are archived. To change the archiving behaviour, you have to create/change the archiver configuration file and point the parameter 'rgma.server. archiver_configuration_file' to this location (see below). [Type: 'string'] *Example: yes*
rgma.server.		Run the R-GMA site-publisher (yes|no). Running the site-publisher

run_site_publisher **[Modified in gLite 1.5, the possible values are now true|false instead of yes|no. However, the old format is still supported for compatibility]**	publishes your site to R-GMA. [Type: 'boolean'] *Example: true*
rgma.server. mysql_root_password	MySQL root password. [Type: 'string'] *Example: verySecret*

site-publisher specific configuration values

rgma.site-publisher.description **[New in gLite 1.5]**	Human readable description of the site. [Type: 'string'] *Example: CERN testbed for prototype tests.*
rgma.site-publisher.siteName **[Changed in gLite 1.5]**	Human readable name of the site. [Example: CERN testbed] [Type: 'string'] *[This parameter was previously used as unique site id. Now it contains a human readable name, while the unique site id is represented by the new advanced parameter rgma.site-published.siteId]*
rgma.site-publisher.webLocation **[New in gLite 1.5]**	Web location of the site (e.g. the homepage). [Type: 'string'] *Example: www.mysite.com*
rgma.site-publisher. contact. system_administrator	Contact email address of the site system administrator. [Type: 'string'] *Example: systemAdministrator@mysite.com*
rgma.site-publisher. contact.user_support	Contact email address of the user support. [Type: 'string'] *Example: userSupport@mysite.com*
rgma.site-publisher. contact.site_security	Contact email address of the site security responsible. [Type: 'string'] *Example: security@mysite.com*
rgma.site-publisher.location **[New in gLite 1.5]**	Human readable location of the site. [Type: 'string'] *Example: CERN, Geneva*
rgma.site-publisher.	Latitude of your site. Please go to

location.latitude		'http://www.multimap.com/' to find the correct value for your site. [Type: 'Float'] *Example: 46.2341*
rgma.site-publisher. location.longitude		Longitude of your site. Please go to 'http://www.multimap.com/' to find the correct value for your site. [Type: 'Float'] *Example: 6.0447*

Advanced Parameters

glite.installer.verbose	true	Enable verbose output. [Type: 'boolean'] *Example : true*
set.mysql.root.password **[New in gLite 1.5]**	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. [Type: Boolean] *Example: false*
rgma.server. httpconnector. maxthread	1000	Maximum number of threads that are created for the tomcat http connector to process requests. This, in turn specifies the maximum number of concurrent requests that the connector can handle. [Type: 'integer'] *Example: 1000*
rgma.server. httpconnector_enableLooku ps	true	Set to true if you want calls to request.getRemoteHost() to perform DNS lookups in order to return the actual host name of the remote client. Set to false to skip the DNS lookup and return the IP address in String form instead (thereby improving performance). [Type: 'boolean'] *Example: true*
rgma.server.LD_ASSUME_ KERNEL	2.4.19	Version of linux threading libraries to be used for tomcat configuration [Type: 'string'] *Example: 2.4.19*
allow.unsecure.port	false	Enable using the unsecure port 8080. It can be true or false. [Type: 'boolean']

		Example: false
rgma.site-publisher.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.site-publisher.sitename
site-publisher specific configuration values		
rgma.site-publisher. site-name	${HOSTNAME}	Hostname of the site. It has to be a DNS entry owned by the site and does not have to be shared with another site (i.e it uniquely identifies the site). It normally defaults to the DNS name of the R-GMA Server running the Site Publisher service.
		[Type: 'string']
		Example: lxb1420.cern.ch
archiver specific configuration values		
rgma.archiver.db.name	arch0	Database name for flexible archiver. [Type: 'string']
		Example: arch0
rgma.archiver.db.user	test	User name for flexible archiver db access. [Type: 'string']
		Example: info
rgma.archiver.db. password	info	User password for flexible archiver db access.
		[Type: 'string']
		Example: info
rgma.archiver.db.history RetentionPeriod **[New in gLite 1.5]**	90	History retention period for flexible archiver db.
		[Type: 'integer'] [Unit: 'minutes']
		Example: 90
System Parameters		
rgma.server. webapp.path	R-GMA	Path under which R-GMA server should be deployed.
		[Type: 'string']
		Example: R-GMA
rgma.server. war.name	R-GMA.war	Name of war file for R-GMA server.
		[Type: 'string']
		Example: R-GMA.war
rgma.server.security. configurationFile	${GLITE_LOCATION} /etc/rgma-server/ServletAuthentication.props	Configuration file for R-GMA server security settings.
		[Type: 'string']
		Example: ${GLITE_LOCATION}/etc/rgma-server/ServletAuthentication.props

site-publisher specific configuration values		
rgma.site-publisher. configurationFile	${GLITE_LOCATION}/ etc/rgma-publish-site/site.props	Configuration file for R-GMA site-publisher settings. [Type: 'string'] *Example: ${GLITE_LOCATION}/etc/rgma-server/ServletAuthentication.props*
archiver specific configuration values		
rgma.archiver. configurationFile	${GLITE_LOCATION}/ etc/rgma-glue-archiver/glue.config	Configuration file for R-GMA flexible archiver archiving settings. [Type: 'string'] *Example: ${GLITE_LOCATION}/etc/rgma-glue-archiver/glue.config*
rgma.server.httpconnector_ maxPostSize	0	The maximum size in bytes of the POST which will be handled by the container FORM URL parameter parsing. The feature can be disbled by setting this attribute to a value inferior or equal to 0. If not specified, this attribute is set to 2097152 (2 megabytes). [Type: 'integer'] *Example: 0*

Table 5: R-GMA 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. Configure MySQL
   Make sure that the MySQL administrator password that you have specified in the configuration file matches the password that is set in the MySQL database. The configuration script does not set it for you. If you want to set a MySQL administrator password, you have to issue the following commands as root:

   /usr/bin/mysqladmin –u root password '*yourPassword*'

   /usr/bin/mysqladmin –u root –h *yourHostname* password '*yourPassword*'

   where *yourHostname* is the name of your host and *yourPassword* is the password that you want to set.
5. Change to the script directory:

   cd /opt/glite/etc/config/scripts
6. Configure the R-GMA server by executing the R-GMA Server configuration script:

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

   The configuration script will stop the services if they are running and configure the R-GMA server. Running the configuration script will also automatically configure the security utils as well as the R-GMA servicetool so there is no need to run these configuration scripts separately.

   By default the databases that have already been created are not replaced if you re-run the configuration script with the option *--configure* in order to protect the databa-

ses (e.g. the already stored information in your archiver). In case you want to force the recreation of the databases, you can run the configuration script with

> *./glite-rgma-server-config.py --configure --recreate_db*

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 Server was successfully configured.*

7. Start the R-GMA server by running:

> *./glite-rgma-server-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 either running

> *./glite-rgma-server-config.py --status*

or by connecting to the R-GMA Browser with your Internet browser at the address

> https://your.host.name:8443/R-GMA/

In the browser you should see the different R-GMA services and one site (if you enabled the site publisher) registered.

The R-GMA Server is completely configured and running.

If you want to stop the R-GMA server at one point, you can run

> *./glite-rgma-server-config.py --stop*

6.2.5 Configuration Walk-Through

After installing the gLite R-GMA 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: Set the password of the MySQL database

> /usr/bin/mysqladmin –u root password '*yourPassword*'

> /usr/bin/mysqladmin –u root –h *yourHostname* password '*yourPassword*'

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

Step 3: Change to the configuration directory:

> cd /opt/glite/etc/config

Step 4: Copy the configuration templates from the templates directory:

> cp templates/* .

Step 5: Customize the configuration files by replacing the changeme values with appropriate parameters according to the following table.

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

rgma.server. run_schema_service	<set it to true if you want to run a schema server, set it to false if you already have a schema server that you want to use>

rgma.server. run_registry_service	<set it to true if you want to run a registry server on this machine, otherwise set it to false>
rgma.server. run_browser	<set it to true if you want to be able to access the server via a web browser, otherwise set it to false>
rgma.server. run_archiver	<set it to true if you want to run an archiver on this server, otherwise set it to true>
rgma.server. run_site_publisher	<set it to true if you want to use this server to publish your site to R-GMA, otherwise set it to false>
rgma.server. mysql_root_password	<The root password of your MySQL database>
rgma.site-publisher.description	<Description of your site>
rgma.site-publisher.siteName	<Short name of your site>
rgma.site-publisher.webLocation	<The URL of your site web page>
rgma.site-publisher.location	<Short name of the location of your site>
rgma.site-publisher. location.longitude	<The physical location of your server (see 'http://www.multimap.com/ to find the coordinates)>

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

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

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

rgma.servicetool.siteId	<your site id as registered in R-GMA, must be the same as the R_GMA Server parameter rgma.site-publisher.siteId>

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

cron.mailto	<your own address>

Step 6: Change to the scripts directory:

cd /opt/glite/etc/config/scripts

Step 7: Execute the glite-rgma-server-config.py script:

./glite-rgma-server-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 server configuration was successfully completed

Step 8: Start the R-GMA server:

./glite-rgma-server-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 R-GMA server was successfully started

Step 9: Verify that the R-GMA server is working and that the R-GMA 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 a list of R-GMA services registered in the Glue Service table.

6.2.6 Querying the configuration

To check if you rgma-server is configured in the way you want, you can run the configuration script with the option –c

./glite-rgma-server-config.py -c

This will give you a list of the most important settings like the repository and server location.

6.3 R-GMA CLIENT DEPLOYMENT MODULE

6.3.1 Service Overview

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

6.3.2 Installation Pre-requisites

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

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

6.3.2.2 Java JRE/JDK

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

6.3.3 R-GMA Client Installation

If you install the client as part of another deployment module (e.g. the UI or WN), the R-GMA client is installed automatically and you can continue with the configuration description in the next section. In case you use the R-GMA client for the service discovery deployment module (see chapter 7) you have to install the R-GMA client by yourself. Otherwise, the R-GMA client can be installed via the following methods:

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 R-GMA client by executing

    > *apt-get install glite-rgma-client-config*

2. *Installation via gLite installer scripts*
    1. Download the latest version of the R-GMA client installation script

        > *glite-rgma-client_installer.sh*

        from the gLite web site. It is recommended to download the script in a clean directory.
    2. Make the script executable

        > *chmod u+x glite-rgma-client_installer.sh*

        and execute it or execute it with

        > *sh glite-rgma-client_installer.sh*

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

This will install the following deployment modules:
- R-GMA client
- 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
gLite-essentials- cpp	in $GLITE_LOCATION/externals/
swig-runtime	in $GLITE_LOCATION/externals/

The gLite R-GMA configuration script is installed in

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

All the necessary template configuration files are installed into

> *$GLITE_LOCATION/etc/config/templates/*

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

6.3.4 R-GMA Client Configuration

If you install the client as part of another deployment module (e.g. the UI or WN), the R-GMA client 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 4.3.2 for the description about the general configuration.
   - The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Table 4 shows the configuration values that can be set.
   - The file *glite-rgma-client.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.

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

Table 6: R-GMA Client 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 R-GMA client by executing the R-GMA client configuration script:

   > *./glite-rgma-client-config.py --configure*

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

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

   > *The gLite R-GMA client was successfully configured.*

6. To verify that the R-GMA client is running correctly, you can run

   > */opt/glite/bin/rgma-client-check*

   In order to have the correct environment set up to run this command, you can either source

   > */etc/glite/profile.d/glite_setenv.sh*

   or logout and login to your shell for the automatic update to take place.

The R-GMA Client is completely configured.

6.3.5 Configuration Walk-Through

After installing the gLite R-GMA client 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-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 client 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 client.

Step 5: Change to the scripts directory:

cd /opt/glite/etc/config/scripts

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

./glite-rgma-client-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 client service configuration was successfully completed.

6.3.6 Querying the configuration

To check if you rgma-client is configured in the way you want, you can run the configuration script with the option –c

./glite-rgma-client-config.py -c

This will give you a list of the most important settings like the repository and server location.

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.

Each 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*

b) *Installation via gLite installer scripts*
1. Download the latest version of the R-GMA servicetool installation script

> *glite-rgma-servicetool_installer.sh*

from the gLite web site. It is recommended to download the script in a clean directory.
2. Make the script executable

> *chmod u+x glite-rgma-servicetool_installer.sh*

and execute it or execute it with

> *sh glite-rgma-servicetool_installer.sh*

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

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 4.3.2 for the description about the general configuration.

    - The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Table 4 shows the configuration values that can be set.

    - The file *glite-rgma-servicetool.cfg.xml* contains the R-GMA client specific configuration values. Table 7 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 [**New in gLite 1.5** replaces rgma.servicetool.sitename]	${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'] *Example : true*
rgma.servicetool.activate	True	Turn on/off servicetool for the node. [Type: 'boolean'] *Example : true*
rgma.servicetool.enable **[New in gLite 1.5]**	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 **[New in gLite 1.5]**	<empty string>	Human-readable name for the service. Need not be globally unique. If value is empty/not specified, the serviceId 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. *Example: http://example.rl.uk/service?WSDL* [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 value if no value is defined in the individual servicetool instance. *Example: http://example.rl.ac.uk/service/semantics.html* [Type: 'string']
rgma.servicetool.vo	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. *Example: EGEE* [Type: 'string']
rgma.servicetool. associatedService	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. *Example: YOURhostname_YOURvoname_YOURservicetype* [Type: 'string']">
rgma.servicetool.param	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. *Example: yourkey=yourvalue* [Type: 'string']

System Parameters

Table 7: 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 [Changed in gLite 1.5]

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 4.3.2 for the description about the general configuration.

    - The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Table 4 shows the configuration values that can be set.

    - The file *glite-rgma-servicetool.cfg.xml* contains the R-GMA client specific configuration values. Table 7 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 8 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		
rgma.servicetool.enable	true	Publish this service via the R-GMA servicetool. If this varaiable set to false the other values below are not taken into account. *Example: true*
rgma.servicetool. service_type		The type of the service: • Unique string in reversed domain name structure. • For all gLite software the structure is

	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*
rgma.servicetool.name	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:* lxn5463.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		URI to contact the service at. This is a service specific string. If no URL is available a string 'not available' should be set. *Example: http://myService/homepage*
rgma.servicetool. publish_interval	3600	How often to publish the service details (like endpoint, version etc). in seconds. *Example: 3600*
rgma.servicetool. status_interval	30	How often check and publish service status (running/not running) in seconds. *Example: 30*
rgma.servicetool.url_wsdl		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. *Example: https://{$HOSTNAME}:8443/EGEE/gl ite-data-catalog-service-meta/services/MetadataCatalog?wsdl*
rgma.servicetool. url_semantics		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. *Example: http://egee-jra1-dm.web.cern.ch/egee-jra1-dm/*
rgma.servicetool.vo		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. *Example: EGEE* [Type: 'string']
rgma.servicetool. associatedService		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. *Example: YOURhostname_YOURvoname_YO URservicetype* [Type: 'string']">
rgma.servicetool.param		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.

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

**Table 8: 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 service-tool. 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 [Modified in gLite 1.5]

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 [New in gLite 1.5]

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 6.2.4 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

b) *Installation via gLite installer scripts*
   1. Download the latest version of the R-GMA GadgetIN installation script

      glite-rgma-gin_installer.sh

   from the gLite web site. It is recommended to download the script in a clean directory.
   2. Make the script executable

      chmod u+x glite-rgma-gin_installer.sh

and execute it or execute it with

sh glite-rgma-gin_installer.sh

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

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 4.3.2 for the description about the general configuration.
   - The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Table 4 shows the configuration values that can be set.
   - The file *glite-rgma-gin.cfg.xml* contains the R-GMA client specific configuration values. Table 9 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.

9. 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.

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

Table 9: 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 of 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),
- Berkely 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 no need to install it separately.

The following modules presently use Service Discovery:

- User Interface (UI) (Chapter 20)
- 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

b) Installation via gLite installer scripts

1. Download the latest version of the Service discovery installation script

    glite-service-discovery_installer.sh

    from the gLite web site. It is recommended to download the script in a clean directory.

2. Make the script executable

    chmod u+x glite-service-discovery_installer.sh

    and execute it or execute it with

    sh glite-service-discovery_installer.sh

    Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-service-discovery 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.

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 4.3.2 for the description about the general configuration.

- The file *glite-service-discovery.cfg.xml* contains the Service Discovery specific configuration values. Table 10 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 correspon-

ding configuration files. Table 11 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 10: 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'] *Example:* *my.hostname.com_myVO_org.glite.FiremanCatalog*
service-discovery.file. url_endpoint		URL endpoint of the service. [Type: 'string'] *Example:* *http://my.hostname.com:8443/myVO/glite-data-catalog-service-fr/services/FiremanCatalog*
service-discovery.file. service_version		Service version in the form 'major.minor.patch' of the used service. [Type: 'string'] *Example: 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. *Example:* *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. *Example:* *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. *Example: host.site.org* [Type: 'string'] List of supported vo for this service. You can specify zero, 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']"

Advanced Parameters

service-discovery.file. service_type	The service type of the used service. This must match the type used to publish the corresponding service. (see 'rgma.servicetool.service_typ e' for the corresponding service for R-GMA as information source) [Type: 'string'] *Example:* *org.glite.FiremanCatalog*
service-discovery.file. param	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 11: 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 11 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 6.3.3. 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 6.3.4 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 6.3.3)

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 PRE-REQUISITES

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

8.2.1 Security Settings

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

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

8.2.2 Java JRE/JDK

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

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

8.2.3 Oracle

If you want to use Oracle as a backend for the VOMS server you need:

1. Oracle Database backend

    If you want to use Oracle as a backend database, you will need to have the Oracle database already installed on the same or on a remote host.

2. Oracle client

    In order for the VOMS server to connect to the ORACLE database you will need to install the ORACLE instant client libraries for jdbc and sqlplus. This release requires v.10.1.0.3.

Due to license reasons, we cannot redistribute these libraries. Please download them from http://www.oracle.com and install them if you have not yet installed them yet

8.2.4 MySQL

If you want to use MySQL as a backend you don't need extra libraries. MySQL is downloaded and installed together with the MySQL version of the VOMS server.

8.3 VOMS SERVER INSTALLATION

Decide if you want to use the MySQL or the ORACLE version of VOMS. As the steps are identical for MySQL and for ORACLE, in the following only the steps for MySQL are described. If you want to use the ORACLE version, just replace 'mysql' with 'oracle' in the file and script names.

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 VOMS server by executing

   apt-get install glite-voms-server-mysql-config

2. *Installation via gLite installer scripts*
   3. Download the latest version of the VOMS server installation script

         glite-voms-server-mysql_installer.sh

      from the gLite web site. It is recommended to download the script in a clean directory.
   4. Make the script executable

      chmod u+x glite-voms-server-mysql_installer.sh

      and execute it or execute it with

      sh glite-voms-server-mysql_installer.sh

      Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-voms-server-mysql 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.

This will install the following deployment modules:
- VOMS server
- Security Utils (see chapter 5)
- R-GMA servicetool (see section 6.4)

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

gLite   in /opt/glite
Tomcat   in /var/lib/tomcat5
MySQL   in /usr/bin/mysql   (in case of the MySQL version)

The gLite VOMS Server and VOMS Admnistration configuration script is installed in

   $GLITE_LOCATION/etc/config/scripts/glite-voms-server-config.py.

A template configuration file is installed in

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

8.4 VOMS 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:
   - The file *glite-global.cfg.xml* contains global configuration values. Refer to 4.3.2 for the values that can be set and section Table 1 for the description about the general configuration.
   - The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Refer to chapter 6 for the description and Table 4 for the configuration values that can be set.
   - The file *glite-rgma-servicetool.cfg.xml* contains the R-GMA servicetool specific configuration values. Refer to Table 7 for the list of parameters that can be set and section 6.4 for the description of the R-GMA servicetool.
   - 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-voms-server.cfg.xml* contains the VOMS server specific configuration files. Since multiple instances of the VOMS Server can be installed on the same node (one per VO), some of the parameters refer to individual instances. Each instance is contained in a separate name
     tag. A default instance is already defined and can be directly configured. Additional instances can be added by simply copying and pasting the section, assigning a name and changing the parameters values as desired.

| voms.db.oracle.instantclient.location **[New in gLite 1.5]** | /usr/lib/oracle/10.1.0.3/client/ | Location of the Oracle Instantclient installation. [Example: /usr/lib/oracle/10.1.0.3/client/] [Type: 'string'] |

   Table 12 shows the list of parameters that can be set.
   The file *glite-voms-server.cfg.xml* also contains a set of
        <instance service="rgma-servicetool"/>
   parameter instances in order to publish the existence and status of the VOMS server to the information system R-GMA. A default instance is already defined and can be directly configured. Additional instances can be added by simply copying and pasting the <instance service="rgma-servicetool"/> for each VO that you want to configure and change all changeme values accordingly. Refer to Table 8 for the list of parameters for each instance that can/have to be set and section 6.4 for the description of the R-GMA servicetool.

Parameter	Default value	Description
User-defined Parameters		
voms.db.type		Database type to be used. Can be 'mysql|oracle'. This parameter

		cannot be specified separately per VO. *Example: mysql* [Type: 'string']
voms.db.host		Hostname of the database server. Put 'localhost' if you run the database on the same machine. This parameter can be specified also separately per VO. *Example: localhost* [Type: 'string']
voms.admin.smtp.host		Host to which voms-admin-service-generated emails should be submitted. Use 'localhost' if you have a fully configured SMTP server running on this host. Otherwise specify the hostname of a working SMTP submission service. This parameter can be specified also separately per VO. *Example: localhost* [Type: 'string']

MySQL configuration

If you use oracle as backend, please either change the value of the mysql parameters 'changeme' to an empty string or remove the parameters

voms.mysql.admin. password		Administrator login password for the MySQL database. This parameter can be specified also separately per VO. *Example: 'verySecret'* [Type: 'string']

Advanced Parameters

glite.installer.verbose	true	Enable verbose output. [Type: 'boolean']
glite.installer.checkcerts	true	Enable check of host certificates. [Type: 'boolean']
rgma.servicetool. activate	true	Turn on/off servicetool for the node. *Example: true* [Type: 'boolean']
set.mysql.root.passwor d **[New in gLite 1.5]**	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
voms.admin.install	true	Install and configure voms-admin. If value is set to false, only voms will be installed and configured. This parameter cannot be specified separately per VO. *Example: true* [Type: 'boolean']
voms.mysql.admin. name	root	Administrator login name for the MySQL database. This parameter can be specified also separately per VO. *Example: 'root'* [Type: 'string']
voms.db.mysql.port	3306	Port number of the database server for mysql. This parameter can be specified also separately per VO. Example: 3306 [Type: 'integer']
voms.db.oracle.port	1521	Port number of the database server for oracle. This parameter can be specified also separately per VO. *Example: 1521* [Type: 'integer']
voms.db.mysql.maxCo nnections **[New in gLite 1.5]**	500	The number of simultaneous client connections allowed. Increasing this value increases the number of file descriptors that mysqld requires. [Example: 500] [Type: integer]
System Parameters		
voms.db.mysql.library	${GLITE_LOCATIO N}/lib/libvomsmysql .so	Defines the MySQL VOMS library location Example: *${GLITE_LOCATION}/lib/libvomsm ysql.so* [Type: 'string']
voms.db.oracle.library	${GLITE_LOCATIO N}/lib/libvomsoracle .so	Location of the oracle voms libraries. *Example: ${GLITE_LOCATION}/lib/libvomsor acle.so* [Type: 'string']

VO Instances parameters

voms.vo.name	Name of the VO associated with the VOMS instance. *Example: EGEE* [Type: 'string']
voms.port.number	Port number listening for requests for this VO. *Example: 15001* [Type: 'string']
voms.db.name	Database name to be used to store VOMS information. *Example: VOMS_EGEE* [Type: 'string']
voms.db.user.name	Name of database user. This parameter can be specified also separately per VO. *Example: voUser* [Type: 'string']
voms.db.user.password	Password of database user defined in 'voms.db.user.name'. This parameter can be specified also separately per VO. *Example: verySecret* [Type: 'string']

VOMS admin specific parameters
If you have decided not to run the voms-admin by setting 'voms.admin.install' to false you can leave these parameters empty or remove them.

voms.admin.notification .e-mail		E-mail address that is used to send notification mails from the VOMS-admin. *Example: name.surname@domain.org* [Type: 'string']"
voms.admin.certificate		The certificate file (in pem format) of an initial VO administrator. The VO will be set up so that this user has full VO administration privileges. Remove parameter or leave parameter empty if you don't want to create an initial VO administrator. *Example: '/your/path/admincert.pem'* [Type: 'string']
voms.db.oracle.instantc lient.location	/usr/lib/oracle/10.1. 0.3/client/	Location of the Oracle Instantclient installation.

[New in gLite 1.5]

[Example: /usr/lib/oracle/10.1.0.3/client/]
[Type: 'string']

Table 12: VOMS 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 automatica values have to be overridden.

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

        /usr/bin/mysqladmin –u root password '*yourPassword*'

        /usr/bin/mysqladmin –u root –h *yourHostname* password '*yourPassword*'

        where *yourHostname* is the name of your host and *yourPassword* is the password that you want to set.
    b. Configure Oracle (if you use Oracle as backend)
        Create the necessary users and databases in ORACLE.
5. Change to the script directory:

    cd /opt/glite/etc/config/scripts

    Configure the VOMS server by executing the VOMS server configuration script:

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

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

    The gLite VOMS server was successfully configured.
6. Start the VOMS server:

    ./glite-voms-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 running

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

The VOMS server is completely configured and running.

8.4.1 Configuration Walk-Through

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

Step 1a - MySQL: If you want to use the MySQL version, set the password of the MySQL database

/usr/bin/mysqladmin –u root password '*yourPassword*'

/usr/bin/mysqladmin –u root –h *yourHostname* password '*yourPassword*'

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

Step 1b - ORACLE: If you want to use the ORACLE version, make sure you have the necessary users and databases created.

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.

For each VO that your VOMS server is supposed to support, create a separate <instance> by copy/pasting the instance in the glite-voms-server.cfg.xml.

If you want to publish the VOMS server for the different VOs, create a separate <instance service="rgma-servicetool"/> by copy/pasting the instance in the glite-voms-server-cfg.xml for each VO that your VOMS server is supposed to support.

File name: glite-voms-server.cfg.xml

voms.db.type	<Put *mysql* if you want to use MySQL as the database backend, put *oracle* if you want to use oracle as the backend>
voms.db.host	<Put *localhost* if your database is installed on the same machine, otherwise put the hostname of the remote database server. This parameter can be specified also separately per VO.>
voms.admin.smtp.host	<Put *localhost* if you have a fully configured SMTP server running on this host. Otherwise specify the hostname of a working SMTP submission service. This parameter can be specified also separately per VO.>
voms.mysql.admin. password	<put the root password for MySQL if you are using MySQL as the database backend, otherwise leave the parameter empty or remove it. This parameter can be specified also separately per VO.>

Create one instance out of the following set of parameters per VO by copy/paste

voms.vo.name	<the name of the vo>

voms.port.number	<the port number for the vo. Must be unique>
voms.db.name	<Database name to be used to store VOMS information.>
voms.db.user.name	<Name of database user for VOMS>
voms.db.user.password	<Password of database user defined in 'voms.db.user.name'.>
voms.admin.notification. e-mail	<E-mail address that should be used to send notification mails from the VOMS-admin>
voms.admin.certificate	<The place of the certificate file (in pem format) of an initial VO administrator. The VO will be set up so that this user has full VO administration privileges. Remove parameter or leave parameter empty if you don't want to create an initial VO administrator.>

Create one instance of the service type rgma-servicetool and change the parameters for each of the instances

vo.name	<name of vo to be published>

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:

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

9   LOGGING AND BOOKKEEPING SERVER

9.1   SERVICE OVERVIEW

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

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

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

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

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

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

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

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

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

9.2   INSTALLATION PRE-REQUISITES

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

9.2.1   Security Settings

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

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

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

9.3 JAVA JRE/JDK

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

Due to license reasons, we cannot redistribute Java. Please download it from the Sun Java web site and install it if you have not yet installed it.

9.4 LOGGING AND BOOKKEEPING SERVER INSTALLATION

1. Method 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-config

2. Method 2: Download from the gLite web site the latest version of the the gLite WMS installation script *glite-lb_installer.sh*. Make the file executable (*chmod u+x glite-lb_installer.sh*) and execute it

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

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

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

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

6. The gLite LB installs the R-GMA servicetool to publish its information to the information system R-GMA. The details of the installation of the R-GMA servicetool are described in section 6.4.

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 4.3.2 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 13). 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 13.

Parameter	Default value	Description
User-defined Parameters		
glite.user.name		The account used to run the LB daemons
glite.user.group		Group of the user specified in the 'glite.user.name' parameter. Leave it empty of comment it out to use the same as 'glite.user.name'
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 **[New in gLite 1.5]**	true	Turn on/off servicetool for the node. [Example: true] [Type: 'boolean']
set.mysql.root.password **[New in gLite 1.5]**	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: false][Type: boolean]
mysql.max_allowed_packet **[New in gLite 1.5]**	17	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 13: 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 automatica 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 7 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 8 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	
glite.user.name	<define your own, must be the same as in the WMS module if istalled on the same host>
glite.user.group	<define your own, must be the same as in the WMS module if istalled on the same host>
mysql.root.password	<define your own, must be the same as in the WMS module if istalled 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:

glite-lb-config.py --configure	Configures all LB services
glite-lb-config.py --start	Starts all LB services (or restart them if they are already running)
glite-lb-config.py –stop	Stops all LB services
glite-lb-config.py --status	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

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

2. 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 no 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 or installer script and they must be taken from the operating system distribution.

10.3 WORKLOAD MANAGER SYSTEM INSTALLATION

1. Method 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-config

2. Method 2: Download from the gLite web site the latest version of the the gLite WMS installation script *glite-wms_installer.sh*. Make the file executable (*chmod u+x glite-wms_installer.sh*) and execute it

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

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

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

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

6. The gLite WMS installs the R-GMA servicetool to publish its information to the information system R-GMA. The details of the installation of the R-GMA servicetool are described in section 6.4.3.

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 14.

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 **[New in gLite 1.5]**		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 **[New in gLite 1.5]**	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
GSIWUFTPPORT	2811	Port where the globus ftp server is listening
GSIWUFTPDLOG	${GLITE_LOCATION_LOG}/gsiwuftpd.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. Example: 120
wmproxy.MinPerusalTimeInterval **[New in gLite 1.5]**	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 **[New in gLite 1.5]**	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_U PDATE_INTERVAL	20	condor_negotiator update interval
condor.MASTER_UPDA TE_INTERVAL	20	condor_master update interval
condor.UPDATE_INTER VAL	20	Default update interval
condor.NEGOTIATOR_I NTERVAL	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 fil
condor.ENABLE_GRID_ MONITOR **[New in gLite 1.5]**	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 blahp poll for new jobs?
information.index.port	2170	Port number of the Information Index
information,index.base_d n	mds-vo-name=local, o=gris	Base DN of the information index LDAP server
disable.gris.purchasing	true	If this parameter is set to 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 **[New in gLite 1.5, but released as a QF in 1.4.1]**	org.glite.SEIndex	Service type of the gLite SEIndex service. Used in locating replicas with Fireman catalogs
wms.dli.service.type **[New in gLite 1.5, but released as a QF in 1.4.1]**	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 14: WMS Configuration Parameters

[New in gLite 1.5] In this release 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.

[New in gLite 1.5] 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 amanually 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 7 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

glite.user.name	\<specify an account name\>
glite.user.group	\<specify a group name or leave empty to use the same as account name\>
voms.voname	\<your VOMS server hostname\>
voms.vomsport	15000
voms.vomscertsubj	\<your VOMS server certificate subject\>
pool.account.basename	\<define your own\>
pool.account.group	\<define your own\>
pool.account.number	\<as many as you like\>
wms.cemon.port	5120
information.index.host	\<your BD-II server host name or empty if it is not used\>
wms.cemon.endpoint	\<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\>
lb.server	\<enter the hostname of your LB server or localhost if LB is running on the same host\>
mysql.root.password	\<define your own, must be the same as in the LB module if

	istalled 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:

glite-wms-config.py –configure	Configures all WMS services
glite-wms-config.py --start	Starts all WMS services (or restart them if they are already running)
glite-wms-config.py --stop	Stops all WMS services
glite-wms-config.py --status	Verifies the status of all services. The exit code is 0 if all services are running, 1 in all other cases
glite-wms-config.py --startservice=xxx	Starts the WMS xxx subservice. xxx can be one of the following:
	condor = the Condor master and daemons
	ftpd = the Grid FTP daemon
	lm = the gLite WMS Logger Monitor daemon
	wm = the gLite WMS Workload Manager daemon
	ns = the gLite WMS Network Server daemon
	jc = the gLite WMS Job Controller daemon
	pr = the gLite WMS Proxy Renewal daemon
	lb = the gLite WMS Logging & Bookkeeping client
	wmp = the WMProxy Server service
glite-wms-config.py --stopservice=xxx	Stops the WMS xxx subservice. xxx can be one of the following:
	condor = the Condor master and daemons
	ftpd = the Grid FTP daemon
	lm = the gLite WMS Logger Monitor daemon
	wm = the gLite WMS Workload Manager daemon
	ns = the gLite WMS Network Server daemon
	jc = the gLite WMS Job Controller daemon
	pr = the gLite WMS Proxy Renewal daemon
	lb = the gLite WMS Logging & Bookkeeping client
	wmp = the WMProxy 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

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. Method 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

2. Method 2: Download from the gLite web site the latest version of the the gLite Torque Server installation script glite-torque-server_installer.sh. Make the file executable (chmod u+x glite-torque-server_installer.sh) and execute it

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

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

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

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

6. The gLite torque-server installs the R-GMA servicetool to publish its information to the information system R-GMA. The details of the installation of the R-GMA servicetool are described in section 6.4.

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 15. The R-GMA servicetool related parameters can be found in **Table 7**.

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

a. *Common parameters* (first part of Table 15)

   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 15)

   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 15)

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

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

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

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

Common parameters

Parameter	Default value	Description
User-defined Parameters		
torque-server.force		This parameter specifies the behaviour of the pbs_server setting parameters and queue creation.In case it is True it will take the whole control of the queue creation/deletion. That means that if it's specified a queue in the config file and latter removed from the configuration file it will also be removed in the pbs_server configuration, on the contrary, no queue removal will be performed.
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output.
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 suers, other than job owner, are allowd 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 between jobs.
maui.server.port	40559	Port on which the Maui server will listen for client connections, by default

		40559.
maui.server.mode	NORMAL	Secifies how Maui interacts with the outside world. Possible values NORMAL, TEST AND SIMULATION.
maui.defer.time	00:01:00	Specifies amount of time a job will be held in the deferred state before being released back to the Idle job queue. Format [[[DD:]HH:]MM:]SS
maui.rm.poll.interval	00:00:10	Maui will refresh its resource manager information every 10 seconds. Ths parameter specifies the global poll interval for all resource managers.
maui.log.filename	${GLITE_LOCATION_LOG}/maui.log	Name of the maui log file
maui.log.max.size	10000000	Maximum allowed size (in bytes) the log file before it will be rolled.
maui.log.level	1	Specifies the verbosity of Maui logging where 9 is the most verbose (NOTE: each logging level is approximately an order of magnitude more verbose than the previous level. Values [0..9]"

System Parameters

Worker node instances

Torque-wn.name		Worker Node name to be used by the torque server. It can also be the CE itself. Example: lxb1426.cern.ch. [Type: string].
torque-wn.number.processors		Number of 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 enqueuing of jobs owned by members of the listed groups. The groups in the list are groups on the server host, not submitting hosts. Syntax: '[+|-group_name[,...]' Example: +iteam,+egee,-test authorizes the test group users to submit jobs to this queue..

Table 15: 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 4.3.2.

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:

glite-torque-server-config.py –configure	Configures all TORQUE SERVER services (pbs_server, maui, BLAHP log server and servicetool)
glite-torque-server-config.py –start	Starts all TORQUE CLIENT services (or restart them if they are already running, pbs_mom)
glite-torque-server-config.py –stop	Stops all TORQUE SERVER services (pbs_server, maui and servicetool)
glite-torque-server-config.py –status	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. Method 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
2. Method 2: Download from the gLite web site the latest version of the the gLite Torque
   Client installation script glite-torque-client_installer.sh. Make the file executable
   (chmod u+x glite-torque-client_installer.sh) and execute it

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

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

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

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

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 16)
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: ${HOSTNAME}.
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 16: TORQUE Client Configuration Parameters

4. 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.

5. *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:

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

3. Add the torque and maui ports to /etc/services.
4. Create the required ssh configuration (modifying the /etc/ssh/ssh_config file) to allow the torque client (Worker Nodes) used HostbasedAuthentication in order to copy its output back to the Torque Server.
5. Look for duplicated key entries in /etc/ssh/ssh_known_hosts.
6. Create the configuration file /opt/edg/etc/edg-pbs-knownhosts.conf. This file contains the nodes which keys will be added to the /etc/ssh/ssh_known_hosts file apart from the torque client nodes (which are taken directly from the torque server via the pbsnodes –a command).
7. Create the edg-pbs-knownhosts script. This script contains a crontab entry to call periodically the /opt/edg/sbin/edg-pbs-knownhosts script. This file is then added to the /etc/cron.d/ directory.
8. Create the pbs_mom configuration file under /var/spool/pbs/mom_priv/config.
9. Start the pbs_mom service.

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:

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

12 COMPUTING ELEMENT

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 Utilites module). The module contains the latest version of the CA certificates plus a number of certificate utilities and security utilities. In particular this module installs the *fetch-crl*, *glite-mkgridmap* and *mkgridmap.py* scripts and sets up cron jobs that periodically check for updated revocation lists and grid-mapfile entries if required). Please note that the use of the glite-mkgridmap script is not normally required on the CE node, since VOMS entries are used instead of individual user DN mappings.

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

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

4. The CE Service may require modification to the server firewall settings. The following iptables instructions must be executed.

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

```
-I <Chain_Name> 2 -m state --state NEW -m tcp -p tcp --dport 3878 -j ACCEPT
-I <Chain_Name> 3 -m state --state NEW -m tcp -p tcp --dport 3879 -j ACCEPT
-I <Chain_Name> 4 -m state --state NEW -m udp -p udp --dport 3879 -j ACCEPT
-I <Chain_Name> 5 -m state --state NEW -m tcp -p tcp --dport 3882 -j ACCEPT
-I <Chain_Name> 6 -m state --state NEW -m udp -p udp --dport 1020 -j ACCEPT
-I <Chain_Name> 7 -m state --state NEW -m udp -p udp --dport 1021 -j ACCEPT
-I <Chain_Name> 8 -m state --state NEW -m udp -p udp --dport 1022 -j ACCEPT
-I <Chain_Name> 9 -m state --state NEW -m udp -p udp --dport 1023 -j ACCEPT
-I <Chain_Name> 10 -m state --state NEW -m tcp -p tcp --dport 32768:65535
-I <Chain_Name> 11 -m state --state NEW -m udp -p udp --dport 32768:65535
```

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

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. Method 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-config

2. Method 2: Download from the gLite web site the latest version of the the gLite Computing Element installation script glite-ce_installer.sh. Make the file executable (chmod u+x glite-ce_installer.sh) and execute it

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

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

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

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

6. The gLite CE installs the R-GMA servicetool to publish its information to the
   information system R-GMA. The details of the installation of the R-GMA servicetool
   are described in section 6.4.

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

		Management System
cemon.cetype **[Modified in gLite 1.5]**		he type of Computing Element. It can be blah, condor or gram. If this parameter is absent or empty, the default type is blah.
cemon.cluster		The cluster entry point host name. Normally this is the CE host itself
cemon.cluster-batch-system-bin-path		The path of the lrms commands. For example: '/usr/pbs/bin' or '/usr/local/lsf/bin'. This value is also used to set the PBS_BIN_PATH or LSF_BIN_PATH variables depending on the value of the '*cemon.lrms*' parameter
cemon.cesebinds		The CE-SE bindings for this CE node. The format is: 'queue[|queue]' se se_entry point A '.' character for the queue list means all queues. Example: '.' EGEE::SE::Castor /tmp
cemon.queues		A list of queues defined on this CE node. Examples are: long, short, infinite, etc.
use.log.parser		Set this option to true to use a separate log parser. Valid values for this parameter are true or false. [Example: false] [Type: boolean]
log.parser.address		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.
lb.user		The account name of the user that runs the local logger daemon. If the user doesn't exist it is created. In the current version, the host certificate and key are used as service certificate and key and are copied in this user's home in the directory specified by the global parameter 'user.certificate.path' in the glite-global.cfg.xml file
iptables.chain		The name of the chain to be used for configuring the local firewall. If the chain doesn't exist, it is created and the rules are assigned to this chain. If the chain exists, the rules are appended to the existing chain
Advanced Parameters		
glite.installer.verbose	True	Enable verbose output

glite.installer.checkcerts	True	Enable check of host certificates
rgma.servicetool.activate **[new in gLite 1.5]**	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 **[new in gLite 1.5]**	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 **[new in gLite 1.5]**	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 mormally

		automatically read from the server host certificate. However it can be set here. For example: 'C=ORG, O=DOMAIN, OU=GRID, CN=host/server.domain.org'
condor.version	6.7.10	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.
System Parameters		
ce-monitor.DOCBASE	${GLITE_LOCATION}/share/webapps/ce-monitor.war	Location of the ce-monitor.war file.

Table 17: CE Configuration Parameters

[New in gLite 1.5] In this release 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.

[New in gLite 1.5] 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 amanually 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 7 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 8 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:

glite-ce-config.py --configure   Configure the CE services

glite-ce-config.py --start   Starts all CE services (or restart them if they are already running)

glite-ce-config.py --stop   Stops all CE services

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

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 indipendent 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 aconomic 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 the 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 a 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. Method 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

2. Method 2: Download from the gLite web site the latest version of the the gLite DGAS Server installation script glite-dgas-server_installer.sh. Make the file executable (chmod u+x glite-dgas-server_installer.sh) and execute it

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

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

   gLite   in /opt/glite ($GLITE_LOCATION)

5.

6. 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

7.

8. The gLite dgas-server installs the security utils rpms as well as the security utils config RPM.

9.

10. 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 versión 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.

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

		'boolean']
pa-server.enabled **[Moved from the user section in 1.5]**	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 mimimun 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 **[Moved from the user section in 1.5]**		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_VA	default HLR lock file

	R}/hlr.lock	
hlr-server.dgas.var	${GLITE_LOCATION_VAR}/dgas	
hlr-server.had.lockfile	${GLITE_LOCATION_VAR}/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_LOG}/hlr_qmgrd.log	Default log file for the transaction manager daemon
hlr-transaction-manager.lockfile	${GLITE_LOCATION_VAR}/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_paPriceAlgMan.so	specifies which shared library contains the algorithm used for pricing

Table 18: DGAS Server configuration parameters

[New in gLite 1.5] In this release 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.

[New in gLite 1.5] 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 amanually 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..

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:

glite-dgas-server-config.py – configure	Configures the DGAS SERVER services (PA and/or HLR) and servicetool.
glite-dgas-server-config.py – start	Starts all DGAS services and servicetool if rgma.servicetool.activate is set to true.
glite-dgas-server-config.py – stop	Stops all DGAS services (PA and/or HLR) and servicetool
glite-dgas-server-config.py – status	Checks the status of the DGAS SERVER services

13.4 DGAS CLIENT

13.4.1 DGAS Client Installation

1. Method 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

2. Method 2: Download from the gLite web site the latest version of the the gLite DGAS Client installation script glite-dgas-Client_installer.sh. Make the file executable (chmod u+x glite-dgas-client_installer.sh) and execute it

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

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

   gLite   in /opt/glite ($GLITE_LOCATION)

5. 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

Parameter	Default Value	Description
		that is responsible for 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.lsfAcctLogDir		This specifies on LSF systems where to find the accountig 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 [{path_lsf}/mnt/work/{nome cluster}/logdir]. 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.economicAccounting	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.lockFileName **[New in 1.5]**	${GLITE_LOCATION_VAR}/dgas_ce_Serverd.lock	Lock File for the daemon
dgas-client.CeServerd.logFileN	${GLITE_LOCATION_VAR}/dgas_ce_Serverd.log	Log File for the daemon

Parameter	Default Value	Description
ame **[New in 1.5]**		
<dgas-client.CeServerd.hadlockFileName **[New in 1.5]**	${GLITE_LOCATION_VAR}/dgas_ce_ServerdHAD.lock	Lock File for the had lock daemon
dgas-client.ce_pushd.gridUser **[New in 1.5]**	dgas	User account used by the pushd for using the user proxy certificates to contact the User HLR server
dgas-client.ce_pushd.dgasURDir	${GLITE_LOCATION_VAR}/dgasURBox/	Specifies the spool directory where the daemon searches for the job usage records and user proxies
dgas-client.ce_pushd.dgasErrDir	${GLITE_LOCATION_VAR}/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.lockFileName	${GLITE_LOCATION_VAR}/dgas_ce_pushd.lock	Lock File for the daemon
dgas-client.ce_pushd.mainPollInterval	10	Time, in seconds, between two usage record processing cycles
dgas-client.ce_pushd.queuePollInterval	50	Time, in seconds, after which the system processes lower priority usage records in the queue
dgas-client.ce_pushd.forceLocalFirst	no	Specifies an alternate routing for the usage record forwarding process

Parameter	Default Value	Description
		between CE, User HLR and Resource HLR. If it is set to yes, usage records are signed with the CE's 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 executied 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 **[New in 1.5]**	${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)

Parameter	Default Value	Description
dgas-client.gianduia.queuePollInterval	600	Interval between two cleanups of the UR directory. During cleanup the daemon checks for garbage in the UR 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	GlueHostBenchmarkSF0 0,GlueHostBenchmarkSI0 0	Comma-seperated list of parameters that we want gianduia to retrieve from an ldif file specified in the 'ldifDefaultFiles' and 'glueLdifFile' files. The key/value pairs will be appended to the usage record. Example: keyList = GlueHostBenchmarkSF0 0,GlueHostBenchmarkSI0 0
dgas-client.gianduia.ldifDefaultFiles	/opt/glite/etc/glite-ce-ce-plugin/out.ldif	Comma-seperated list of files to be searched for the keys specified in the 'keyList' parameter. Example: ldifDefaultFiles = /opt/glite/etc/glite-ce-ce-plugin/out.ldif
dgas-client.gianduia.glueLdifFile		File to search for the parameters specified in 'keyList'. It overrides the contents of ldifDefaultFiles. Example: glueLdifFile = '/opt/glite/etc/glite-ce-ce-plugin/out.ldif'
dgas-client.getAcctLogd.aclFile **[New in 1.5]**	"${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.listeningPort **[New in 1.5]**	56565	The listening port used by the daemon.

Parameter	Default Value	Description
dgas-client.getAcctLogd.logFileName **[New in 1.5]**	${GLITE_LOCATION_VAR}/getAcctLogd.log	The log file where the listener logs its activities.
dgas-client.getAcctLogd.lockFileName **[New in 1.5]**	${GLITE_LOCATION_VAR}/getAcctLogd.lock	Lock file name
dgas-client.getAcctLogd.outputFile **[New in 1.5]**	${GLITE_LOCATION_VAR}	The file where the listening daemon writes the contents of the file sent by the client.
dgas-client.getAcctLogd.outputDir **[New in 1.5]**		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.gianduiottiBox	${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 19: DGAS Server configuration parameters

[New in gLite 1.5] In this release 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.

[New in gLite 1.5] 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 amanually as in previous versions if the automatic values have to be overridden.

8. 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.aclFile 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:

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

14 WORKER NODE

14.1 SERVICE OVERVIEW

The gLite Standard Worker Node is a set of clients required to run jobs sent by the gLite Computing Element via the Local Resource Management System. It currently includes the gLite I/O Client, the Logging and Bookeeping Client, the R-GMA Client and the WMS Checkpointing library. The gLite Torque Client module can be installed together with the WN module if Torque is used as a batch system.

14.2 INSTALLATION PRE-REQUISITES

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

14.2.1 Security Settings

Install one or more Certificate Authorities certificates in */etc/grid-security/certificates*. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (http://www.gridpma.org/). A special security module called *glite-security-utils* can be installed by downloading and running from the gLite web site (http://www.glite.org/) the script *glite-security-utils_installer.sh* (Chapter 13). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl* script and sets up a crontab that periodically check for updated revocation lists.

14.2.2 Java JDK/JRE

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

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:

6. Method 1: 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

7. Method 2: Download from the gLite web site the latest version of the the gLite Worker Node installation script glite-wn_installer.sh. Make the file executable (chmod u+x glite-wn_installer.sh) and execute it

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

9. This will install the following deployment modules:

    - Worker Node
    - R-GMA client (see section 6.3 for details)
    - File Transfer Service Client (see section 16 for details)
    - I/O Client (see section 19.4 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:

gLite I/O Client	in /opt/glite
gLite LB Client	in /opt/glite
glite R-GMA Client	in /opt/glite
gLite WMS Checkpointing	in /opt/glite
gLite FTS client	in /opt/glite
gLite Service Discovery	in /opt/glite
gLite DGAS CLient	in /opt/glite
Globus	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 4.3.2 for the description about the general configuration.

- The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Refer to Table 4 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 Table 6 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 10 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 20 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 11 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 **[Modified in gLite 1.5]**	- 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 20: 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:

gilte-data-single-catalog   ➔   MySQL version

gilte-data-single-catalog-oracle   ➔   Oracle version

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

15.2 INSTALLATION PRE-REQUISITES

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

15.2.1 Security Settings

1. Install one or more Certificate Authorities certificates in */etc/grid-security/certificates*. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (http://www.gridpma.org/). A special security module called *glite-security-utils* can be installed by downloading and running from the gLite web site (http://www.glite.org) the script *glite-security-utils_installer.sh* (Chapter 13). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl* script and sets up a crontab that periodically check for updated revocation lists

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

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

15.2.2 Java JDK

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

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

15.2.3 Oracle 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. Method 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-ctalog[-oracle]-config

2. Method 2: Download from the gLite web site the latest version of the the gLite Single Catalog installation script glite-data-single-catalog[-oracle]_installer.sh. Make the file executable (chmod u+x glite- data-single-ctalog[-oracle]_installer.sh) and execute it

3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-data-local-transfer-service next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.

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

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

5. Download and install the Oracle Instant Client Basic version 10.1.0.3-1 (oracle-instantclient-basic-10.1.0.3-1.i386.rpm) and Oracle Instant Client SQL-Plus (oracle-instantclient-sqlplus-10.1.0.3-1.i386.rpm). To do so, connect to the Oracle Web Site.

6. The gLite 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

7. 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 ommitted 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 21 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 acessing the catalog database
System Parameters		
catalog-service-fr-mysql.DBURL **[Modified in gLite 1.5]**	jdbc:mysql://localhost:3306/${catalog-service-fr-mysql.DBNAME}?zeroDateTimeBehavior=convertToNull	URL of the database

Table 21: 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. *Example: verySecret [Type: 'string']*
Advanced Parameters		
glite.installer.verbose	True	Enable verbose output
glite.installer.checkcerts	True	Enable check of host certificates
set.mysql.root.password **[New in gLite 1.5]**	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: false][Type: boolean]
allow.unsecure.port	False	Enable using the unsecure port 8080. It can be true or false. Example: false
catalog-service-fr-mysql.ADMIN_VOMS_ATT RIBUTES **[New in gLite 1.5]**	<empty string>	ADMIN/superuser settings. Note that the extra priviliges 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 iss 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 [Example: /opt/glite/etc/config/ ADMIN_VOMS_ATTRIBUTES][T ype: string]
catalog-service-fr-mysql.ADMIN_MAPFILE **[New in gLite 1.5]**	<empty 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_MAP FILE][Type: string]
System Parameters		

Catalog-service-fr-mysql.DOCBASE	${GLITE_LOCATION}/share/java/glite-data-catalog-service-fr-mysql.war	Location of the glite-data-catalog-service-fr-mysql.war file
Catalog-service-fr-mysql.DBDRIVERCLASS	org.gjt.mm.mysql.Driver	JDBC driver classname
Catalog-service-fr-mysql.MODULE.NAME	glite-data-catalog-service-fr-mysql	Catalog service module name
catalog-service-fr-mysql.MESSAGINGON	False	If 'true', then a connection to the specified messaging system is attempted and messages will be produced.
catalog-service-fr-mysql.MESSAGINGJNDIHOST		The host of the JNDI server that contains the messaging system connetion factories and topic/queue objects.
catalog-service-fr-mysql.MESSAGINGJNDIPORT		The port of the JNDI server that contains the messaging system connetion factories and topic/queue objects.
catalog-service-fr-mysql.MESSAGINGJMSNAME		The JNDI name of the 'local' messaging server to connect to.
catalog-service-fr-mysql.MESSAGINGTOPIC		The JNDI name of the topic that the messages should be produced on.
rgma.servicetool.activate	True	Turn on/off servicetool for the node. [Example: true] [Type: 'boolean']"
catalog-service-fr-mysql.httpconnector_maxThreads	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_minSpareThreads	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-	75	The maximum number of unused

mysql.httpconnector_maxSpareThreads		request processing threads that will be allowed to exist until the thread pool starts stopping the unnecessary threads
catalog-service-fr-mysql.httpconnector_acceptCount	100	The maximum queue length for 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_connectionTimeout	600000	The number of milliseconds this Connector will wait, after accepting a connection, for the request URI line to be presented

Table 22: 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 acessing 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:@${catalog-service-fr.DBHOST}:${catalog-service-fr.DBPORT}:${catalog-service-fr.DBSERVICENAME}	URL of the database. *Example: jdbc:oracle:thin:@lxfs5502.cern.ch:1521:devegee3*

Table 23: 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.MESSAGINGJNDIHOST		The host of the JNDI server that contains the messaging system connetion factories and topic/queue objects.
catalog-service-fr.MESSAGINGJNDIPORT		The port of the JNDI server that contains the messaging system connetion factories and topic/queue objects.
catalog-service-fr.MESSAGINGJMSNAME		The JNDI name of the 'local' messaging server to connect to.
catalog-service-fr.MESSAGINGTOPIC		The JNDI name of the topic that the messages should be produced on.
System Parameters		
catalog-service-fr.DOCBASE	${GLITE_LOCATION}/share/java/glite-data-catalog-service-fr.war	Location of the glite-data-catalog-service-fr-mysql.war file
catalog-service-fr.DBDRIVERCLASS	oracle.jdbc.driver.OracleDriver	JDBC driver classname
catalog-service-fr.MODULE.NAME	glite-data-catalog-service-fr	Catalog service module name
catalog-service-fr.oracle-jdbc.classpath	${CATALINA_HOME}/common/lib	Path to the Oracle JDBC drivers
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 24: Fireman common configuration parameters (Oracle)

[New in gLite 1.5] In this release 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.

[New in gLite 1.5] 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 amanually 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:$GLOBUS_LOCATION/bin:$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 FILE TRANSFER SERVICE

16.1 SERVICE OVERVIEW

The data movement services of gLite are responsible to securely transfer files between Grid sites. The transfer is performed always between two Storage Elements having the same transfer protocol available to them (usually gsiftp). The gLite File Transfer Service is composed of the File Transfer Service web service (responsible for managing data transfers and placements), and a number of file transfer agents (see Chapter 17). As of gLite 1.4 there are no separate File Transfer and Placement services/client

The File Transfer Service is responsible for the actual transfer of the file between the SEs. It takes the source and destination names as arguments and performs the transfer. The FTS is managed by the site administrator, i.e. there is usually only one such service serving all VOs.

FTS supports two underlying database backends: Oracle and MySQL.

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 the server host certificate *hostcert.pem* and key *hostkey.pem* in */etc/grid-security*
2. Install the VOMS Server(s) host certificate in the directory */etc/grid-security/vomsdir*. This is necessary to extract the VOMS information from the VOMS proxies.

16.2.2 Java JRE/JDK

The Java JRE or JDK are required to run the FTS. This release requires v. 1.4.2 (revision 08 or greater). The JDK/JRE version to be used is a parameter in the glite-global.cfg.xml configuration file. Please change it according to your version and location.

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

16.2.3 Database backend

FTS supports two database backend services: Oracle and MySQL. In either case the database must be installed and configured with appropriate databases and database user accounts. Choice of the database backend is done in the FTS configuration. The database process must be running prior to starting the configuration of FTS.

16.2.4 Oracle

16.2.4.1 Oracle InstantClient

The Oracle Instant Client is required to run the File Transfer Service. Due to license reasons, we cannot redistribute it. Version 10.1.0.3-1 (or newer) can be downloaded from the Oracle Web Site.

16.2.4.2 Oracle Database Configuration

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

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

You may otionally grant debugging privileges:

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

16.2.5 MySQL

16.2.5.1 MySQL Database Configuration

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

```
create database <DBNAME>;

grant  all  privilegies  on  <DBNAME>.*  to  <DBUSER>  identified  by
"<DBPASSWORD>";

grant all privilegies on <DBNAME>.* to <DBUSER>@localhost identified
by "<DBPASSWORD>";

grant all privilegies on <DBNAME>.* to <DBUSER>@<FTSNODE> identified
by "<DBPASSWORD>";

grant all privilegies on <DBNAME>.* to <DBUSER>@<FTANODE> identified
by "<DBPASSWORD>";
```

NOTE: If MySQL database is not set up, configuration script fails with the error message:
`[ERROR] Error during communication with the database`

16.2.6 R-GMA client (in case of the R-GMA based service discovery)

If R-GMA-based Service Discovery is used, the R-GMA client must be installed before the FTS service is configured (see Chapter 7 for more details).

16.3 FILE TRANSFER SERVICE

16.3.1 File Transfer Service Installation

1. Method 1: Install APT, if not yet installed following the instructions at http://glite.web.cern.ch/glite/packages/APT.asp and install the gLite File Transfer Service by executing

apt-get install glite-file-transfer-service-config

2. Method 2: Download from the gLite web site the latest version of the the gLite File Transfer Service installation script glite-file-transfer-service_installer.sh. Make the file executable (chmod u+x glite-file-transfer-service_installer.sh) and execute it

3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-file-transfer-service next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.

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

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

5. In case of Oracle database backend, download and install the Oracle Instant Client Basic version 10.1.0.3-1 (for example using the oracle-instantclient-basic-10.1.0.3-1.i386.rpm) and Oracle Instant Client SQL-Plus (for example using the oracle-instantclient-sqlplus-10.1.0.3-1.i386.rpm). To do so, connect to the Oracle Web Site.

6. The gLite FTS configuration script is installed in $GLITE_LOCATION/etc/config/scripts/glite-file-transfer-service-config.py. A template configuration file is installed in $GLITE_LOCATION/etc/config/templates/glite-file-transfer-service.cfg.xml

16.3.2 File Transfer Service ORACLE Configuration

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

2. Copy the configuration file template from *$GLITE_LOCATION/etc/config/templates/glite-file-transfer-service.cfg.xml* to *$GLITE_LOCATION/etc/config/glite-file-transfer-service.cfg.xml* and modify the parameters values as necessary (Table 21)

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:

Global Parameters

Parameter	Default value	Description
User-defined Parameters		
file-transfer.DBBackend		Type of the underlying database backend. Supported values: oracle and mysql
file-transfer.DBHOST		Hostname of the transfer database
file-transfer.DBUSER		Name of the database user owning the transfer database

file-transfer.DBPASSWORD		Password for accessing the transfer database
file-transfer.DBSERVICENAME		The database (MySQL) or database service (Oracle) name to connect to. It must be the same as the corresponding FTA DB service name

Advanced Parameters

file-transfer.DBPORT	''	TCP port of the database. Default value is chosen in function of the used database backend. Oracle: 1521 MySQL: 3306 To use this default value the value of this parameter must be empty.
file-transfer.DBURL	''	URL to connect to. Default value is chosen in function of the used database backend. To use this default value the value of this parameter must be empty.
file-transfer.SECURITY_ENABLED	true	If set to 'false', no authorization will be made at all, regardless of the attribute settings below and regardless of whether a secure connector is used or not. Setting to 'true' will requires the use of a secure connector and the use of an appropriately authorized certificate.
file-transfer.SUBMIT_VOMS_ATTRIBUTES		Any user with these voms attributes may submit to the service
file-transfer.SUBMIT_MAPFILE	${GLITE_LOCATION}/etc/glite-data-transfer-submit-mapfile	If a client's certificate subject name is listed in this file, a client may submit jobs to the service
file-transfer.ADMIN_VOMS_ATTRIBUTES		If a user's certificate contains this VOMS attribute, they are additionally permitted to do any operation upon the service including manage channels
file-transfer.ADMIN_MAPFILE	${GLITE_LOCATION}/etc/glite-data-transfer-manager-mapfile	If a client's certificate subject name is listed in this file, they are additionally permitted to do any operation upon the service including manage channels.
file-transfer.VETO_MAPFILE	${GLITE_LOCATION}/etc/glite-data-	Path to a file containing a list of client DNs to veto. The file should contain one DN per line.

		veto-mapfile
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
watchdog.enable	true	Flag to enable or disable the watchdog cron job
System Parameters		
file-transfer-fts.DOCBASE	${GLITE_LOCATION}/share/java/glite-data-transfer-fts.war	Location of the FTS WAR file
file-transfer-fts.DBDRIVERCLASS	<empty string>	Java class name of the JDBC driver. Default value is chosen automatically as a function of used database backend. To use default value, this parameter should be empty
watchdog.fts.check-command	${GLITE_LOCATION}/bin/glite-transfer-channel-list -s https://${HOSTNAME}:8443/glite-data-transfer-fts/services/ChannelManagement %%%%%%%%	The command to be executed by the watchdog daemon to check the component status
watchdog.fts.return-string	list: getChannel: Channel name '%%%%%%%%' does not exist	The expected return code from the watchdog command
file-transfer.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 25: File Transfer Service Configuration Parameters

[New in gLite 1.5] 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 amanually as in previous versions if the automatic values have to be overridden.

4. Put in the mapfiles (submit, cancel, manager and veto) the DNs of the users allowed to perform the specified operations with the foloowing syntax, e.g.:
   "/C=CH/O=CERN/OU=GRID/CN=Maite Barroso Lopez 5660" .egtest

5. As root run the FTS configuration file with the *--configure* option:
   $GLITE_LOCATION/etc/config/scripts/glite-file-transfer-service-config.py --configure
6. As root run the FTS configuration file with the *--start* option:
   $GLITE_LOCATION/etc/config/scripts/glite-file-transfer-service-config.py--start
7. The File Transfer Service is now ready.

16.3.3 File Transfer Service Configuration Walkthrough

The File Transfer Service configuration script performs the following steps:

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

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

   GLITE_LOCATION_VAR
   GLITE_LOCATION_LOG
   GLITE_LOCATION_TMP
   CATALINA_HOME

3. Load the global and the File Transfer Service configuration files
   $GLITE_LOCATION/etc/config/glite-file-transfer-service.cfg.xml

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

5. Copy the Oracle jar file in the Tomcat location[1]

6. Configure Tomcat.

7. Check the existence of the Oracle JDBC drivers[1].

8. Install the security utils.

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

10. Copy the host certificates to the user account.

11. Configure the File Transfer Service instances

    a. Checking environment variables (CATALINA_HOME)
    b. Creates the context.xml file

---

[1] If Oracle backend is used

    c. Checks if the VETO mapfile exists, if not it creates the VETO mapfile indicated in the variable file-transfer.VETO_MAPFILE

    d. Checks if the schemas already exist

    e. Uploads the Data Base Schema if it does not exist

16.3.4 Publishing FILE TRANSFER Services to R-GMA

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

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

16.4 FILE TRANSFER SERVICE CLIENT

16.4.1 Service Overview

16.4.2 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.

16.4.2.1 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* can be installed by downloading and running from the gLite web site (http://www.glite.org/) the script *glite-security-utils_installer.sh* (Chapter 5). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl* script and sets up a crontab that periodically check for updated revocation lists

16.4.3 File Transfer Client installation

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

    apt-get install glite-file-transfer-service-client-config

2. Method 2: Download from the gLite web site the latest version of the the gLite File Transfer Service Client installation script glite-file-transfer-service-client_installer.sh. Make the file executable (chmod u+x glite-file-transfer-service-client_installer.sh) and execute it

3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.

4. If the installation is performed successfully, the following components are installed:
gLite in /opt/glite
Globus in /opt/globus

5. The File Transfer client configuration script is installed in
$GLITE_LOCATION/etc/config/glite-file-transfer-service-client-config.py. A template
configuration file is installed in $GLITE_LOCATION/etc/config/templates/glite-file-
transfer-service-client.cfg.xml

16.4.4 File Transfer 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 *$GLITE_LOCATION/etc/config/templates/glite-
file-transfer-service-client.cfg.xml* to *$GLITE_LOCATION/etc/config/* and modify the
parameter values as necessary (Table 26)

3. Some parameters have default values; others must be changed by the user. All
parameters that must be changed have a token value of *changeme*.
The following parameters can be set:

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

Table 26: File Transfer Client configuration parameters

[New in gLite 1.5] The Service Discovery parameters for file-based discovery have
been removed from the configuration file. File-based discovery is only supported for
test or debugging. The normal mode of operation is based on R-GMA or BD-II
discovery.

4. Run the File Transfer client configuration file with the --configure option

*$GLITE_LOCATION/etc/config/
scripts/glite-file-transfer-service-client-config.py --configure*

5. The File Transfer client is now ready.

17 FILE TRANSFER AGENTS

17.1 SERVICE OVERVIEW

The File Transfer Agents perform Data Transfer and Placement actions. We distinguish two different kinds of agent: the Channel Agent and the VO Agent. The Channel Agent is responsible for managing all the file transfers through a channel, i.e. the entity that represent the phisical, monodirectional link between two sites: this agent will fetch the files transfer requests from a Queue and submit them to the configured File Transfer Service. The other type of agent, the VO Agent, is in charge of performing all the actions that are related to a specific Virtual Organization, which involves applying VO policies, managing catalog interactions and running VO custom actions. Moreover, we distinguish between two possible VO Agent deployment types:

- File Transfer Service (FTS) Agent: This agent manages jobs where the source and destination contains Physical File Names (SURLs or TURLs). No catalog interaction is required
- File Placement Service (FPS) Agent: Extend the previous model adding the interaction with a Catalog Service, in order to retrieve the source and destination physical file names from the Logical File Names (LFNs and GUIDs) and source and destination sites. Once a transfer is completed, the new replicas are registered to the appropriate catalog.

One Channel Agent is needed for each channel available on the site, and one VO Agent is needed for each VO that what to perfoms data transfer requests. All of these agents share the same Queue, but the FTA framework guarantees that they interact with each other in the proper way: a VO Agent is allowed to see just the jobs (and related information) that belongs to itself, in the same way a Channel Agent is not able to process requests belonging to a different channel. You can imagine that each agent act on a view of the entire Queue:

```
              /---------------\
              |     Queue     |
              \---------------/
              |--------       |
VO_1          || Vo_1 |       |
Agent =====> View |-------|
              |-------- Ch_1 ||    Channel_1
              |       | View <=====  Agent
              |--------       ||
VO_2          || Vo_2 |-------|
Agent =====> View |-------|
              |-------- Ch_2 ||    Channel_2
              |       | View <=====  Agent
              |--------       ||
VO_3          || Vo_2 |-------|
Agent =====> View |       |
              |--------       |
              \---------------/
```

The way of the Channel and the VO Agent work is the same: they periodically run some action in order to perform the step required to transfer data. The Channel Agent actions are:

- Fetch: Submit new File transfer request to the TransferService
- Check: Check the state of all the active File transfer requests and update the Queue with the retrieved information
- Cancel: Revoke active file transfers marked as canceling on the Queue

The VO Agent actions are:

- File Transfer Service:
  - o Allocate: Allocate a transfer job to a channel based on the source and destination of the related files
  - o Retry: Reschedule failed transfers that are in waiting state
  - o Cancel: Revoke pending (i.e. not yet processed by the Channel Agent) files transfers marked as canceling on the Queue.
- File Placement Service: adds the following actions to the FTS ones:
  - o Resolve: Resolve the source Logical File Names into an SURL and generate the destination SURL looking at the information provided by the Service Discovery
  - o Register: When a Transfer Job is completed, register the new replicas on the Catalog Service

In addition, the VO Agent allows the possibility to schedule additional custom actions that the VO may want to provide in order to perform VO-apecific tasks.

The GLite Data Transfer Agents module provides a default action for all of these types, but it would also allow extending the behavior of an agent by configuring different actions.

17.2 INSTALLATION PRE-REQUISITES

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

17.2.1 Security Settings

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

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

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

17.2.2 Database backend

File transfer agents support two database backend services: Oracle and MySQL. In either case the database must be installed, configured and running. Database is configured during the FTS configuration (see Chapter 16). To be able to start the channel agents it is necessary to have corresponding channels added into the database. The choice of the database backend is done in the FTA configuration.

17.3 DATA TRANSFER AGENTS INSTALLATION

1. Method 1: Install APT, if not yet installed following the instructions at http://glite.web.cern.ch/glite/packages/APT.asp and install the gLite File Transfer Agents by executing

   apt-get install glite-file-transfer-agents-config

2. Method 2: Download from the gLite web site the latest version of the the gLite File Transfer Agents installation script glite-file-transfer-agents_installer.sh. Make the file executable (chmod u+x glite-file-transfer-agents_installer.sh) and execute it

3. If the database backend is Oracle, download and install the Oracle Instant Client Basic version 10.1.0.3-1 (for example using the oracle-instantclient-basic-10.1.0.3-1.i386.rpm) and Oracle Instant Client SQL-Plus (for example using the oracle-instantclient-sqlplus-10.1.0.3-1.i386.rpm). The latest version of the Oracle InstantClient tools is available on the Oracle Web Site.

4. Make the script executable (chmod u+x glite-data-transfer-agents _installer.sh) and execute it or execute it with sh glite-data-transfer-agents _installer.sh

5. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-data-transfer-agents 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.

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

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

7. The gLite FTA configuration script is installed in $GLITE_LOCATION/etc/config/scripts/ glite-file-transfer-agents-config.py. A template configuration file is installed in $GLITE_LOCATION/etc/config/templates/ glite-file-transfer-agents.cfg.xml

17.4 DATA TRANSFER AGENTS CONFIGURATION

3. 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)

4. Copy the configuration file template from $GLITE_LOCATION/etc/config/templates/glite glite-file-transfer-agents.cfg.xml to $GLITE_LOCATION/etc/config/ glite-file-transfer-agents.cfg.xml and modify the parameters values as necessary (Table 27 and Table 28)

5. Some parameters have default values; others must be changed by the user. All parameters that must be changed have a token value of *changeme.* The following parameters can be set:

Instance configuration:

There are four types of agents:

> transfer-vo-agent-fps
> transfer-vo-agent-fts
> transfer-channel-agent-urlcopy
> transfer-channel-agent-srmcopy

The agents instances must be defined in the configuration file by defining one agent instance per each existing VO and channel (only one of transfer-vo-agent-fps or transfer-vo-agent-fts must be defined per each VO)

[New gLite 1.5] The agent VO instances are created automatically by iterating over the VO list defined in the vo-list.cfg.xml. However, the instances can still be configured manually using the same parameters as in previous versions if the values have to be overridden.

Service defaults:
Each of four agent types has its corresponding defaults:

transfer-vo-agent-fts

Parameter	Default value	Description
transfer-vo-agent.Contact	""	The contact information of the Administrator responsible for that agent

System Parameters

transfer-agent.name.prefix	transfer-vo-agent	Internal parameter, cannot be changed

transfer-vo-agent-fps

Parameter	Default value	Description
transfer-vo-agent.Contact	""	The contact information of the Administrator responsible for that agent

System Parameters

transfer-agent.name.prefix	transfer-vo-agent	`Internal parameter, cannot be changed`

transfer-channel-agent-urlcopy

Parameter	Default value	Description
transfer-channel-agent.Contact	""	The contact information of the Administrator responsible for that agent
transfer-agent-ts-urlcopy.MaxTransfers	50	The maximum number of transfer request that the Transfer Service can process simultaneously
transfer-agent-ts-urlcopy.Streams	10	The maximum number of parallel streams that could be used during the transfer. A transfer is performed with the number of streams specified in the channel table, if lesser than 'Streams', otherwise this threshold would be used. In any case the value can be overwritten by a job using the job's parameter '-p'
transfer-agent-ts-urlcopy.TcpBlockSize	0	The TCP Block Size that should be used during the transfer.If set to 0, the default is used. This parameter can be overwritten by a job using the job's parameter '-tcpbs'
transfer-agent-ts-urlcopy.TransferTimeout	600	The timeout value (in seconds) that should be used to detect that a transfer is hanging. 0 means no timeout. In case the specified value is -1, the glite-url-copy default will apply. Please note that in case of SrmCopy bulk requests the global transfer timeout is computed by multiplying this value for the size of the request.
transfer-agent-ts-urlcopy.SrmPutTimeout	60	The timeout value (in seconds) that should be used for an SRM Put request. 0 means no

Parameter	Default value	Description
		timeout. In case the specified value is -1, the glite-url-copy default will apply
transfer-agent-ts-urlcopy.SrmGetTimeout	60	The timeout value (in seconds) that should be used for an SRM Get request. 0 means no timeout. In case the specified value is -1, the glite-url-copy default will apply
transfer-agent-ts-urlcopy.SrmPutDoneTimeout	60	The timeout value (in seconds) that should be used for trying to set the SRM status of the Destination File to Done. 0 means no timeout. In case the specified value is -1, the glite-url-copy default will apply
transfer-agent-ts-urlcopy.SrmGetDoneTimeout	60	The timeout value (in seconds) that should be used for trying to set the SRM status of the Source File to Done. 0 means no timeout. In case the specified value is -1, the glite-url-copy default will apply
transfer-agent-ts-urlcopy.TransferMarkersTimeout	120	The timeout value (in seconds) between two Transfer Markers that should be used to detect that a transfer is hanging. 0 means no timeout. In case the specified value is -1, the glite-url-copy default will apply
System Parameters		
transfer-agent.name.prefix	transfer-channel-agent	Internal parameter, cannot be changed

transfer-channel-agent-srmcopy

Parameter	Default value	Description
transfer-channel-agent.Contact	""	The contact information of the Administrator responsible for that agent
transfer-agent-ts-urlcopy.MaxTransfers	50	The maximum number of transfer request that the

		Transfer Service can process simultaneously
transfer-agent-ts-urlcopy.TransferTimeout	0	The timeout value (in seconds) that should be used to detect that a transfer is hanging. 0 means no timeout. In case the specified value is -1, the glite-url-copy default will apply. Please note that in case of SrmCopy bulk requests the global transfer timeout is computed by multiplying this value for the size of the request.
transfer-agent-ts-urlcopy.SrmCopyTimeout	120	The timeout value (in seconds) that should be used for an SRM Copy request. 0 means no timeout.In case the specified value is -1, the glite-url-copy default will apply.
transfer-agent-ts-urlcopy.SrmStatusTimeout	600	The time interval (in seconds) that should be used in order to abort an srm copy call: when the SRM can't be contacted for the time specified by that parameter, the transfer will be aborted. 0 means no timeout.
transfer-agent-ts-urlcopy.MaxBulkSize	100	The maximum size for a bulk SrmCopy request. This value is ignored is the 'TransferType' parameter is different from 'srmcopy'
System Parameters		
transfer-agent.name.prefix	transfer-channel-agent	Internal parameter, cannot be changed

Table 27: File Transfer Agents Configuration Parameters (agent-specific parameters)

Parameter	Default value	Description
file-transfer.DBBackend		Backend database type: 'oracle' or 'mysql'
transfer-agent.username		The username of the user running the agents daemons. Example: myuser
transfer-agent.groupname		The groupname of the user

		running the agents daemons.Example: mygroup
transfer.agent.user.uid		The userid of the user running the agents daemons. This may be needed by some SRM implmentations. Leave it empty if not used
transfer-agent.group.gid		The gid of the user running the agent daemons. This may be needed by some SRM implmentations. Leave it empty if not used
transfer-agent.DBUser		The name of the user that should be used to connect to the DB
transfer-agent.DBPassword		The password of the user that should be used to connect to the DB
transfer-agent.DBHost		Database host name/address
transfer-agent.DBService		Database/database service name
Advanced parameters		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
transfer-agent.log.Priority	WARN	WARN, DEBUG, INFO
watchdog.enable	true	Flag to enable or disable the watchdog cron job. – true: enable the watchdog – false: disable the watchdog
transfer-agent.DBPort	""	TCP port of the database. It must be the same as the corresponding FTS DB. Leave empty to use a default value for a chosen DB backend Oracle: 1521 MySQL: 3306 In the current implementation of the FTA this parameter is ignored for the MySQL backend and allways the default port is used
transfer-agent-dao-oracle.ConnectString	""	The Oracle ConnectString identifying the DB default : ${transfer-agent.DBHost}:${transfer-agent.DBPort}/${transfer-

		agent.DBService} To use default value, leave the parameter value empty
transfer-agent-myproxy.Server	""	The host name of the MyProxy Server. If that parameter is not set or is empty, the Myproxy Server is looked up using the Service Discovery and then, if not found, the myproxy default will apply (MYPROXY_SERVER environment variable)
transfer-agent-myproxy.Port	0	The port of the MyProxy Server. If that parameter is not set or is 0, the myproxy default will applies
transfer-agent-myproxy.ProxyLifetime	86400	The lifetime in seconds of the proxy certificates that will be created
transfer-agent-myproxy.Repository	/tmp	The location where the certificates retrieved from the MyProxy Service will be stored. That location must already exist
transfer-agent-myproxy.MinValidityTime	3600	The minimum validity time (in seconds) an already existent certificate should have before submitting a new job. In case the certificate couldn't satisfy that requirement, a new certificate will be retrieved from the MyProxy Service
transfer-agent.logdir	${GLITE_LOCATION_LOG}	The location of the log files
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

fta.certificate.path	/etc/grid-security	The location of the user certificates relativeto the user home directory or absolute path to the location of the user certificates. This parameter overrides the global one set in the glite-global.cfg.xml file
file-transfer -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.
transfer-agent-dao-mysql.SocketName	""	Socket name for MySQL

Table 28: File Transfer Agents Configuration Parameters (global parameters)

[New in gLite 1.5] In this release the VO-specific parameters have been moved to the separate vo-list-cfg.xml file. The FTA 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.

[New in gLite 1.5] 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 amanually as in previous versions if the automatic values have to be overridden.

[New in gLite 1.5] The Service Discovery parameters for file-based discovery have been removed from the configuration file. File-based discovery is only supported for test or debugging. The normal mode of operation is based on R-GMA or BD-II discovery.

4. As root run the FTA configuration file with the –configure option:
   $GLITE_LOCATION/etc/config/scripts/ glite-data-transfer-agents-config.py –configure
5. For each VO agent run the command:
   `glite-transfer-channel-setvoshare <channel> <vo> <share size>`
6. REMINDER: To start channel agents, the corresponding channels must be added to the FTS database
7. As root run the FTA configuration file with the –start option:
   $GLITE_LOCATION/etc/config/scripts/ glite-data-transfer-agents-config.py -–start
8. The File Transfer Agents Oracle is now ready.

17.5 PER-INSTANCE CONFIGURATION OF DATA TRANSFER AGENTS

Starting from gLite Release 1.4 the FTA module provides the functionality of adding, configuring, starting and stopping individual FTA instances.
The general syntax is:

```
$GLITE_LOCATION/etc/config/scripts/glite-data-transfer-agents-
config.py --instance <command> <instance_name> [<instance_type>]
```

where:

<command> is one of:

- add
- configure
- start
- stop
- status

<instance_name> is a unique name of the instance

<instance_type> is a parameter required only for the "add" command and specifies the type of the agent.

The action of adding an agent automatically modifies the configuration file with the necessary entries.

17.5.1 Adding FTA instance

Adding of the FTA instance corresponds to the modification of the local configuration file by adding of new data transfer agent instance. By default this instance will be configured by using the default values. To customise the instance the manual configuration file modification is necessary. Note that this mechanism is not compatible with the automatic creation of agent vo instances by iteration over the defined VOs.

```
$GLITE_LOCATION/etc/config/scripts/glite-data-transfer-agents-
config.py --instance add <instance_name> <instance_type>
```

<instance_name> is the unique instance name
<instance_type> is one of following transfer aget types:

```
transfer-channel-agent-urlcopy
transfer-channel-agent-srmcopy
transfer-vo-agent-fps
transfer-vo-agent-fts
```

WARNING: Since this command modifies (writes into) the configuration file, it is necessary that this file is saved locally on the node. Therefore this command will not work with the siteconfig file. Nevertheless the combination of site configuration and local file configuration is possible. In addition, the generated xml text can be copied and pasted in a site configuration file.

17.5.2 Configuring FTA instance

Simple FTA instance can be configured by following commands:

```
$GLITE_LOCATION/etc/config/scripts/glite-data-transfer-agents-
config.py --instance configure <instance_name>
```

<instance_name> is the name of given instance

17.5.3 Starting/stopping FTA instance

Simple FTA instance can be started/stopped by following command:

```
$GLITE_LOCATION/etc/config/scripts/glite-data-transfer-agents-
config.py --instance start <instance_name>
```

or

```
$GLITE_LOCATION/etc/config/scripts/glite-data-transfer-agents-
config.py --instance stop <instance_name>
```
<instance_name> is the name of given instance

17.6 DATA TRANSFER AGENTS CONFIGURATION WALKTHROUGH

The Data Transfer Agents configuration script performs the following steps:

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

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

    GLITE_LOCATION_VAR
    GLITE_LOCATION_LOG
    GLITE_LOCATION_TMP
    CATALINA_HOME

14. Load the global and the File Transfer Agents configuration files
    $GLITE_LOCATION/etc/config/ glite-file-transfer-agents.cfg.xml

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

16. Copy the Oracle jar file in the Tomcat location

17. Configure Tomcat.

18. Check the existence of the Oracle JDBC drivers.

19. Install the security utils.

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

21. Copy the host certificates to the user account.

22. Configure the Data Transfer Agents

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

18 HYDRA

18.1 SERVICE OVERVIEW

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

18.2 INSTALLATION PRE-REQUISITES

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

18.2.1 Security Settings

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

18.2.2 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.

18.3 HYDRA INSTALLATION

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

gLite	in	/opt/glite	($GLITE_LOCATION)
MySQL-server	in	/usr	
MySQL-client	in	/usr	
Tomcat	in	/var/lib/tomcat5	

5. The gLite 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

18.4 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)

2. 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 29 shows a list of the global hydra configuration variables that can be set: **Error! Reference source not found.** 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	jdbc:mysql://${HOSTNAME}:3306/${hydra.DBNAME}	URL of the database. [Example: jdbc:mysql://${HOSTNAME}:3306/${hydra.DBNAME}] [Type: 'string']
hydra.PATH	${vo.name}/glite-data-hydra-service	Path to the web application. [Example: ${vo.name}/glite-data-hydra-service] [Type: 'string']

Table 29: 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.DBRESOURCEN AME	meta	Name of the JNDI objetcs that is holding the DB connection object. [Example: meta] [Type: 'string']
hydra.DOCBASE	${GLITE_LOCATI ON}/share/java/glit e-data-hydra-service.war	Location of the glite-data-catalog-service-fr-mysql.war file. [Example: ${GLITE_LOCATION}/share/java/gl ite-data-hydra-service.war][Type: 'string']
hydra.ATTRIBUTE_HE LPER_CLASS	org.glite.data.hydra .helpers.attribute.M ySQLAttributeHelp er"	Name of the class (including the package name) implementing the logic for operations on attributes (getAttributes, setAttributes, etc.). [Example: org.glite.data.hydra.helpers.Attribut eHelper][Type: 'string']
hydra.CATALOG_HEL PER_CLASS	org.glite.data.hydra .helpers.catalog.My SQLCatalogHelper	name of the class (including the package name) implementing the logic for operations on entries (createEntry and removeEntry). [Example: org.glite.data.hydra.helpers.Catalo gHelper][Type: 'string']
hydra.SCHEMA_HELP ER_CLASS	org.glite.data.hydra .helpers.schema.M ySQLSchemaHelp er	name of the class (including the package name) implementing the logic for operations on schemas (createSchema, dropSchema, etc.). [Example: org.glite.data.hydra.helpers.Schem aHelper][Type: 'string']
hydra.AUTHORIZATIO N_HELPER_CLASS	org.glite.data.hydra .helpers.authorizati on.MySQLAuthoriz ationHelper	name of the class (including the package name) implementing the logic for authorization (acess control) on entries in the catalog (FASBase - setPermission, getPermission, etc... plus the internal policy for creation of new entries and schemas). [Example: org.glite.data.hydra.helpers.Authori zationHelper][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 30: 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.

18.5 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*:

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

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

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

3. Configures the gLite Security Utilities module
4. Verifies the JAVA installation
5. Checks the configuration values

6. Stops MySQL server if it is running
7. Starts mySQL server
8. Sets the MySQL root password
9. Stops Tomcat
10. Configures Tomcat
11. Configures the different VO instances inside Tomcat:
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

18.6 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.

18.7 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.

19 GLITE I/O

19.1 GLITE I/O SERVER

19.1.1 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.

19.1.2 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.

19.1.2.1 Security Settings

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

2. Customize the *mkgridmap* configuration file *$GLITE_LOCATION/etc/glite-mkgridmap.conf* by adding the required VOMS server groups. The information in this file is used to run the *glite-mkgridmap* script during the Security Utilities configuration to produce the */etc/grid-security/grid-mapfile*

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

19.1.2.2 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.

19.1.3 gLite I/O Server installation

1. Method 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

2. Method 2: Download from the gLite web site the latest version of the the gLite I/O Server installation script glite-io-server_installer.sh. Make the file executable (chmod u+x glite-io-server_installer.sh) and execute it

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

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

5. gLite I/O Server in /opt/glite

6. Globus   in /opt/globus

7. 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

8. 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.

19.1.4 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 31 gives an overview of the parameters that can be set. The R-GMA servicetool related parameters can be found in Table 7: R-GMA servicetool configuration parameters.

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

   a. *Common parameters (first part of Table 31)*

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

Also you will find a section for the R-GMA servicetool to publish information about the rfiod. Adapt also these configuration values accordingly. You can find more information on the values and the R-GMA servicetool in section 6.4.

b. *VO dependant gLite I/O Server parameters (second part of Table 31)*

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/O Daemon initialization parameters		
init.username		The username of the user running the I/O Daemon. If using a astor with a castor SRM, in some configurations this user must be a valid user on the Castor server. If the user doesn't exist on this I/O Server, it will be created. The uid specified in the 'init.uid' parameters may be used.
init.groupname		The groupname of the user running the I/O Daemon. If using a Castor SRM, in some configurations this group must be a valid user on the Castor server. If the group doesn't exist I/O Server, it will be created. The gid specified in the 'init.gid' parameters may be used.
init.uid		The userid of the user running the I/O Daemon. If using a Castor SRM, in some configurations the

		same uid of the Castor user specified in the 'init.username' parameter must be set. Leave this parameter empy or comment it out to use a system assigned uid.
init.gid		The gid of the user running the I/O Daemon. If using a Castor SRM, in some configurations the same gid of the Castor group specified in the 'init.groupname' parameter must be set. Leave this parameter empy or comment it out to use a system assigned gid.

Advanced Parameters

General gLite initialization parameters

glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check for host certificate
rgma.servicetool.activate **[New in gLite 1.5]**	true	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	The maximum number of concurrent transfers
io-resolve-common.SePort	8443	The port of the remote file operation server
io-resolve-common.RootPathRule	abs_dir	The rule to be applied to define the path for creating new files. Allowed values are: * abs_dir: The file name will be created by appending the file name to the path specified by RootPath configuration parameter * user_home_dir: the file name will be created by appending the file name to a path specified by the RootPath configuration parameter, a directory with the user name first letter and then the complete user name. [Note: Since at the moment the user name that is retrieved is the distinguished name, using that option is not suggested]
io-authz-fas.FileOwner	<empty>	When checking the credentials, perform an additional check on that name to verify it was the user's name. Default value is an empty string, that means that this additional test is not performed
io-authz-fas.FileGroup	<empty>	When checking the credentials, perform an additional check on that name to verify it was one of the user's groups. Default value is an empty string, that means that this additional test is not performed
io-resolve-fireman.OverwriteOwnership	false	Overwrite the ownership of the file when creating it. If set to true, the newly created file will have as owner the values set by the FileOwner and FileGroup configuration parameters.
io-resolve-fireman.FileOwner	<empty>	The name of the group that will own any newly created file. This parameter is meaningful only if OverwriteOwnership is set to true. In case this parameter is not set, the Replica Catalog default will apply. Default value is an empty

		string.
io-resolve-fireman.FileGroup	<empty>	The name of the group of any newly created file. This 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.OverwriteOwnership	false	Overwrite the ownership of the file when creating it. If set to true, the newly created file will have as owner the values set by the FileOwner and FileGroup configuration parameters. Default value is false.
io-resolve-fr.FileOwner		The name of the user that will own any newly created file. This parameter is meaningful only if OverwriteOwnership is set to true. In case this parameter is not set, the Replica Catalog default will apply. Default value is an empty string.
io-resolve-fr.FileGroup		The name of the group of any newly created file. This parameter is meaningful only if OverwriteOwnership is set to true. In case this parameter is not set, the Replica Catalog default will apply. Default value is an empty string

System Parameters

I/O Daemon parameters

io-daemon.EnablePerfMonitor	false	Enable the Performace Monitor. If set to true, a process will be spawned to monitor the performance of the server and create some of the statistics.
io-daemon.PerfMonitorPort	9998	The Performace Monitor port
io-daemon.CacheDir	<empty>	The directory where cached files should be stored
io-daemon.CacheDirSize	0	The maximum size of the directory where cached files should be stored
io-	5000000	The size of the preloaded cache

Parameter	Default value	Description
daemon.PreloadCache Size		
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.CacheCleanup Threshold	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.CacheCleanup Limit	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 CGSI GSOAP 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: * rfio: use the remote file io (rfio) 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-catalogs.RmcEndPoint	The endpoint of the RMC catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. This is also the value of the 'io-resolve-catalogs.RmcEndpoint' parameter. *Example: https://lxb2028:8443/VO-NAME/edg-replica-metadata-catalog/services/edg-replica-metadata-catalog*
io-resolve-catalogs.RlsEndpoint	The endpoint of the Rls catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. *Example: https://lxb2028:8443/VO-NAME/edg-local-replica-catalog/services/edg-local-replica-catalog*

Parameters required by the Fireman and FR catalogs.

io-authz-fas.FasEndpoint	The endpoint of the Fas catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. *Examples:* *http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FAS (for FR)* *http://lxb2024.cern.ch:8080/glite-*

	data-catalog-service-fr/services/FiremanCatalog (for Fireman)
Fireman parameters	
io-resolve-fireman.FiremanEndpoint	The endpoint of the FiReMan catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. *Example: http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FiremanCatalog*
FR parameters	
io-resolve-fr.ReplicaEndPoint	The endpoint of the Replica catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. *Example: http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/ReplicaCatalog*
io-resolve-fr.FileEndPoint	The endpoint of the File catalog. If that value starts with httpg:// the GSI authentication will be used (using the CGSI GSOAP plugin); if it starts with https:// the SSL authentication will be used, using the CGSI GSOAP plugin in SSL compatible mode), otherwise no authentication is requested. If that value is not set, the File Catalogs will not be contacted and the io-resolve-fr plug-in will managed only GUIDs. *Example: http://lxb2024.cern.ch:8080/glite-data-catalog-service-fr/services/FileCatalog*
Advanced Parameters	

instanceDescription **[New in gLite 1.5]**	${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 instances. In previous releases the name was forced to be ${vo.name}-${init.CatalogType}.* *Now this is the default value, but it can be replaced with any user string*
autocalculate.port **[New in gLite 1.5]**	true	If this value is true, the I/O Server port for each instance is calculated automatically starting from the value of the parameter io-daemon.Port. If the value is false, the io-daemon.Port value is taken without modifications. In this case, users must defined instance to have a different port configured in this file
io-daemon.Port **[Modified in gLite 1.5]**		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 perfomed using a QUANTA paralled TCP stream a pool of sockets are opened on the server side binding a tuple of available ports from 50000 to 51000. This port should not be higher than 9999 and different I/O Server instances should not run on contiguos ports (for example set one to 9999 and another one to 9989). If the parameter autocalculate.port 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 autocalculate.port 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
log.Priority	DEBUG	The log4cpp log level. Possible values are: DEBUG, INFO, WARNING, ERROR, CRITICAL, ALERT, FATAL
log.FileName **[Modified in gLite 1.5]**	${GLITE_LOCATION_LOG}/glite-io-server-${instanceDescription}.log	The location of the log file for this instance

Table 31: gLite I/O Server Configuration Parameters

[New in gLite 1.5] In this release 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.

[New in gLite 1.5] 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 amanually 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 7 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.

19.1.5 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

19.2 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.

19.3 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.

19.4 CLIENT

19.4.1 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.

19.4.2 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.

19.4.2.1 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* can be installed by downloading and running from the gLite web site (http://www.glite.org/) the script *glite-security-utils_installer.sh* (Chapter 13). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl* script and sets up a crontab that periodically check for updated revocation lists

19.4.3 gLite I/O Client installation

1. Method 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

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

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

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

19.4.4 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 32)

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 32: gLite I/O Client configuration parameters

[New in gLite 1.5] The Service Discovery parameters for file-based discovery have
been removed from the configuration file. File-based discovery is only supported for

test or debugging. The normal mode of operation is based on R-GMA or BD-II discovery.

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

20 THE DPM SERVER

20.1 SERVICE OVERVIEW

The Disk Pool Manager (DPM) has been developed as a lightweight solution for disk storage management. A priori, there is no limitation on the amount of disk space that the DPM can handle. The DPM offers an implementation of the Storage Resource Manager (SRM) specifications, for version 1.1 and version 2. For details about the SRM specifications, see http://sdm.lbl.gov/srm-wg.

The DPM handles the storage on disk servers. In fact, it handles **pools**: a pool is a group of file systems, located on one or more disk servers. The way file systems are grouped to form a pool is up to the DPM administrator.

The can handle two different kinds of file systems:

- **volatile** : the files contained in a volatile file system can be removed by the system at any time, unless they are pinned by a user.
- **permanent** : the files contained in a permanent file system cannot be removed by the system.

DPM can work with two different backends, MySQL and Oracle. Additionally, the DPM is security enabled : the basic GSI security stack (Globus RPMs, pool accounts, etc.) has to be installed on the DPM server machines, as well as on the disk servers.

The DPM consists of:

- DPM socket client,
- SRM server (srmv1 and/or srmv2) : receives the SRM requests and pass it on to the DPM server
- DPM server : keeps track of all the requests
- DPM name server (DPNS) : handles the namespace for all the files under the DPM control
- DPM RFIO server : handles the transfers for the RFIO protocol
- DPM Gridftp server : handles the transfer for the Gridftp protocol.

20.1.1 DPM Server Overview

The DPM has been split into two different deployment modules. The dpm-server one and dpm-disk-server.

This dpm-server deployment module contains and configures the following services: dpns, dpm, srmv1 and srmv2. It is also responsible for registering these services into RGMA via the servicetool deployment module. Additionally, the dpm-server deployment module configures rgma-gin in order to publish the DPM information into RGMA.

The dpm-server can work with two different backends, MySQL (dpm-server-mysql) and Oracle (dpm-server-oracle). There are two different deployment rpms in order to install DPM with an Oracle or MySQL backend.

The second DPM module is the dpm-disk-server deployment module. It needs to be installed and configured in all the disk servers that are added to the DPM server.

20.1.2 DPM Disk Server Overview

This deployment module configures the rfiod and gsi-ftpd daemons. These two services are required in any host (disk-server) that contains a file system used by the DPM server.

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 Oracle

If you want to use Oracle as a backend for the DPM server you need:

1. Oracle Database backend

    If you want to use Oracle as a backend database, you will need to have the Oracle database already installed on the same or on a remote host.

2. Oracle client

    In order for the DPM server to connect to the ORACLE database you will need to install the ORACLE instant client libraries for jdbc and sqlplus. This release requires v.10.1.0.3.

    Due to license reasons, we cannot redistribute these libraries. Please download them from http://www.oracle.com and install them if you have not yet installed them yet

20.2.2 MySQL

If you want to use MySQL as a backend you don't need extra libraries. MySQL is downloaded and installed together with the MySQL version of the DPM server.

20.3 DPM SERVER

20.3.1 DPM Server Installation

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

    a. If the backend is **MySQL**

        apt-get install glite-dpm-server-mysql-config

    b. If the backend is **Oracle**

apt-get install glite-dpm-server-oracle-config

2. Method 2: Download from the gLite web site the latest version of the the gLite DPM Server installation script.

    c. In order to configure DPM with a MySQL backend download:

        glite-dpm-server-mysql_installer.sh

    d. If the backend is Oracle, the installer script is:

        glite-dpm-server-oracle_installer.sh

Make the file executable (chmod u+x glite-dpm-server-mysql_installer.sh or glite-dpm-server-oracle_installer.sh) and execute it

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

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

gLite   in /opt/glite ($GLITE_LOCATION)
dpm   in /opt/lcg/
MySQL   in /usr/bin/mysql   (in case of the MySQL version)

5. The gLite dpm-server configuration script (the same configuration script is used to configure dpm with a MySQL and Oracle backend) is installed in *$GLITE_LOCATION/etc/config/scripts/glite-dpm-server-config.py*. A template configuration file is installed in *$GLITE_LOCATION/etc/config/templates/glite-dpm-server.cfg.xml*

6. The gLite dpm-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.

7. The gLite dpm-server installs the RGMA-GIN module to publish the DPM information in the R-GMA.

20.3.2 DPM 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-dpm-server.cfg.xml

    to

    $GLITE_LOCATION/etc/config/glite-dpm-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 15. The R-GMA servicetool related parameters can be found in **Table 7**

    The parameters in the file are the following ones:

Name	Default Value	Description
User-defined Parameters		
db.type		Database backend to be used. It can be mysql or oracle
db.user		DPM user to access the database. Example dpm
db.password		DPM user password to access the database. Example dpm_password
db.host		Name of the machine running the database. Example: localhost, oradev10.cern.ch
mysql.root.password		Password (clear) of the root user of the MySQL server used for the

Name	Default Value	Description
		database creation. A password has to be provided.
db.name		Database name. Example: devdb10. If the database type is mysql this parameter can be left empty
db.sid		If the database type is mysql this parameter can be left empty
db.port		Por where the database server will be waiting for connections. If the database type is mysql this parameter can be left empty
dpm.uid		User ID of the dpm user. Example: 21088
dpm.gid		Group ID of the dpm group. Example: 21088
disk-server.list		List of disk-servers which file systems will be added to the dpm pools.
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output. [Example: 'true'] [Type: 'boolean']
rgma.servicetool.activate	true	Turn on/off servicetool for the node.[Example: true] [Type: 'boolean']
lcg.providers.location	/opt/lcg	The locations were the LCG Info Providers are installed
dpm.username	dpmmgr	DPM user name
dpm.groupname	dpmmgr	DPM group name
dpns-server.enabled	true	Select this option if you want to configure the pa server. Format: true, false
dpns-server.host	${HOSTNAME}	dpns server hostname

Name	Default Value	Description
dpns-server.logfile	/var/log/dpns/log	dpns server log file
dpns-server.configfile	/opt/lcg/etc/NSCONFIG	dpns server configuration file
dpm-server.enabled	true	Select this option if you want to configure the dpm server. Format: true, false
dpm-server.host	${HOSTNAME}	dpm server hostname
dpm-server.logfile	/var/log/dpm/log	dpm server log file
dpm-server.configfile	/opt/lcg/etc/DPMCONFIG	dpm server configuration file
srmv1-server.enabled	true	Select this option if you want to configure the srmv1 server. Format: true, false
srmv1-server.host	${HOSTNAME}	server where the srmv1 server is running
srmv1-server.logfile	/var/log/srmv1/log	srmv1 server daemon log file
srmv2-server.enabled	true	Select this option if you want to configure the srmv2 server. Format: true, false
srmv2-server.host	${HOSTNAME}	Host where the srmv2 server is running
srmv2-server.logfile	/var/log/srmv2/log	srmv2 server daemon log file
db.tnsadmin		Location of the tnsnames.ora file. If left empty the deployment script will create it automatically. Otherwise, a correct lcoation to the tnsnames file must be specified.
dpm.oracle-instantclient.location	/usr/lib/oracle/10.1.0.3/client	Location of the Oracle Instantclient installation
System parameters		
db.protocol	TCP	Database protocol.

In case you want to allow an IO-server submit requests to the DPM server you need to create an io-server instances.

IO Server instances		
io-server.voname		VO of the io-server using dpm.
io-server.hostname		IO Server hostname. Example: lxb1427.cern.ch. In the io-server configuration the io-resolve-common.RootPath parameter should point to the this location (without the domain). i.e /dpm/domain/home/vo/io-server /dpm/cern.ch/home/egee/lxb1427
io-server.certificate.subject		IO Server certificate subject. Example: /C=CH/O=CERN/OU=GRID/CN=host/lxb1427.cern.ch

Table 33: DPM Server Configuration Parameters

4. 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 1 shows a list of the parameters that can be set. More details can be found in section 4.3.2.

   For dpm-server-mysql or dpm-server-oracle the following sub-services are published via the R-GMA servicetool:
   - i. dpns
   - ii. dpm
   - iii. srmv1
   - iv. srmv2

5. Configure the RGMA-GIN:

Copy the RGMA-GIN configuration file template

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

to

$GLITE_LOCATION/etc/config

Modify the rgma-gin configuration file by specifying in the *rgma.gin.run_generic_info_provider* the value yes. The *rgma.gin.run_fmon_provider* should have the value no.

RGMA-gin is used to publish DPM information in

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

6. As root run the DPM Server Configuration script (with the –configure option in order to configure the service) */opt/glite/etc/config/scripts/glite-dpm-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.

7. *As* root start the DPM Server services by running the configuration script with the –start option.
> */opt/glite/etc/config/scripts/glite-dpm-server-config.py --start*

20.3.3 DPM Server Configuration Walkthrough

The DPM Server configuration script performs the following steps:

1. Load the DPM Server configuration file $GLITE_LOCATION/etc/config/glite-dpm-server.cfg.xml and the servicetool configuration file $GLITE_LOCATION/etc/config/glite-rgma-servicetool.cfg.xml
2. If the backend is Oracle:
   - Check that the Oracle client is installed.
   - Create the tnsnames Oracle file

3. Stop the services that are running
4. Configure the security utils
5. Configure the grid-mapfile
6. Create the group-mapfile
7. Check the host certificates
8. Create the dpm group
9. Create the dpm user

10. Create the dpm certificates directory, copy them and assign the right permissions
11. Check the gridmap-dir and gridmap-file
12. Add the io-server´s DNs to the gridmap-file
13. Create the pool accounts
14. If the backend is MySQL:
    - Start MySQL
    - Set the root password
15. For each service to be configured (dpns, dpm, srmv1, srmv2)
    - Create the configuration file
    - Assign the permissions
    - Create the dpns or dpm databases (if the service being configured is dpns or dpm)
16. Create the servicetool instances and configure the servicetool
17. Create the LCG info provider configuration file
18. configure RGMA-GIN
19. Start dpns
20. Create the dpns namespace and the io-server instances namespace
21. Create the /etc/shift.conf file
22. Stop MySQL

20.3.4 Managing the DPM Server Service

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

glite-dpm-server-config.py – configure	Configures all DPM SERVER services (dpns, dpm, srmv1, srmv2, servicetool and rgma-gin)
glite-dpm-server-config.py – ldif	Generates the DPM info provider configuration file
glite-dpm-server-config.py – start	Starts all DPM SERVER services (or restart them if they are already running)
glite-dpm-server-config.py – stop	Stops all DPM SERVER services (dpns, dpm, srmv1, srmv2, servicetool, rgma-gin)
glite-dpm-server-config.py – status	Checks the status of the DPM SERVER services

20.4 DPM DISK SERVER

20.4.1 DPM DISK SERVER Installation

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

   apt-get install glite-dpm-disk-server-config

2. Method 2: Download from the gLite web site the latest version of the the gLite DPM
   Disk Server installation script glite-dpm-disk-server_installer.sh. Make the file
   executable (chmod u+x glite-dpm-disk_server_installer.sh) and execute it

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

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

gLite	in /opt/glite ($GLITE_LOCATION)
dpm	in /opt/lcg

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

20.4.2 DPM DISK 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 (see
   Table 16)
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-dpm-disk-server.cfg.xml to
   $GLITE_LOCATION/etc/config/glite-dpm-disk-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 following parameters 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

The dpm-disk server configuration parameters values are the following:

Name	Default Value	Description
User-defined Parameters		
dpm.uid		User ID of the dpm user. Example: 21088
dpm.gid		Group ID of the dpm group. Example: 21088
dpm-server.host		DPM Server hostname
dpns-server.host		DPNS Server hostname
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output. [Example: 'true'] [Type: 'boolean']
rgma.servicetool.activate	false	Turn on/off servicetool for the node.[Example: true] [Type: 'boolean']
dpm.username	dpmmgr	DPM user name
dpm.groupname	dpmmgr	DPM group name
dpm-gsiftp-server.enabled	true	Select this option if you want to configure the srmv1 server. Format: true, false
rfiod-server.enabled	true	Select this option if you want to configure the srmv2 server. Format: true, false
rfiod-server.logfile	/var/log/rfiod/log	

Table: DPM Disk Server Configuration Parameters

4. As root run the DPM Disk Server Configuration file with the –configure option
   /opt/glite/etc/config/scripts/glite-dpm-disk-servert-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.

5. *As* root start the DPM Disk Server services (rfiod and dpm-gsiftp) by running the DPM Disk Server Configuration File:

    /opt/glite/etc/config/scripts/glite-dpm-disk-server-config.py --start

20.4.3 DPM DISK SERVER Configuration Walkthrough

The DPM Disk Server configuration script performs the following steps:

1. Load the DPM Disk Server configuration file *$GLITE_LOCATION/etc/config/glite-dpm-disk-server.cfg.xml* and the servicetool configuration file $GLITE_LOCATION/etc/config/glite-rgma-servicetool.cfg.xml
2. Stop all the services (rfiod and gsi-ftp)
3. Configure the security utils
4. Check the host certificates
5. Create the dpm group
6. Create the dpm user
7. Create the dpm certificates directory, copy them and assign the right permissions
8. Check the gridmap-dir
9. Check the gridmap-file
10. Create the pool accounts
11. Configure the gsiftp service
    - Create the dpm-gsiftp configuration file
12. Configure the rfiod service
    - Create the rfiod configuration file
    - Add /opt/globus/lib to ld.so.conf
    - Execute ldconfig
13. Create the servicetool instances and configure the servicetool
    23. Create the /etc/shift.conf file

20.4.4 Managing the DPM DISK SERVER

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

glite-dpm-disk-server-config.py --configure	Configures all DPM DISK SERVER services (dpm-gsiftp and rfiod)
glite-dpm-disk-server-config.py --start	Starts all DPM DISK SERVER services (or restart them if they are already running)
glite-dpm-disk-server-config.py --stop	Stops all DPM DISK SERVER services (dpm-gsiftp and rfiod
glite-dpm-disk-server-config.py --status	Checks the status of the DPM DISK SERVER services

21 THE LFC SERVER

21.1 SERVICE OVERVIEW

The LCG File Catalog is provided by the CERN IT Grid Deployment (IT-GD) group. It is a high performance file catalog. It fixes the performance and scalability problems seen with the EDG catalogs. For instance, it provides:
- Cursors for large queries
- Timeouts and retries from the client.

The LFC provides more features than the RLS:
- User exposed transaction API,
- Hierarchical namespace and namespace operations,
- Integrated GSI Authentication and Authorization,
- Access Control Lists (Unix Permissions and POSIX ACLs),
- Sessions
- Checksums

The LFC supports Oracle and Mysql as database backends, and the integration with GFAL and lcg util has been done by the Grid Deployment group.

The LFC has a completely different architecture from the RLS framework. Like the EDG catalog, it contains a GUID (Globally Unique Identifier) as an identifier for a logical file, but unlike the EDG catalog it stores both logical and physical mappings for the file in the same database. This speeds up operations which span both sets of mappings. It also treats all entities as files in a UNIX-like filesystem. The API is similar to that of the UNIX filesystem API, with calls such as creat, mkdir and chown.

There is a global hierarchical namespace of Logical File Names (LFNs) which are mapped to the GUIDs. GUIDs are mapped to the physical locations of file replicas in storage (Storage File Names or SFNs). System attributes of the files (such as creation time, last access time, file size and checksum) are stored as attributes on the LFN, but user-defined metadata is restricted to one field, as the authors believe that user metadata should be stored in a separate metadata catalog. Multiple LFNs per GUID are allowed as symbolic links to the primary LFN.

Bulk operations are supported, with transactions, and cursors for handling large query results. As there is only one catalog, transactions are possible across both LFN and SFN operations, which was impossible with the EDG RLS. In case of momentary loss of connection to the catalog, timeouts and retries are supported.

In the secure version of the LFC, authentication is by Kerberos 5 or Grid Security Infrastructure (GSI), which allows single sign-on to the catalog with users Grid certificates.
The client domain name is mapped internally to a uid/gid pair which is then used for authorization.

21.1.1 LFC Server Overview

This LFC-server deployment module contains and configures the following services: dli and lfcdaemon. It is also responsible for registering these services into RGMA via the servicetool deployment module. Additionally, the lfc deployment module configures rgma-gin in order to publish the LFC information into RGMA.

The lfc can work with two different backends, MySQL (lfc-mysql) and Oracle (lfc-oracle). There are two different deployment rpms in order to install LFC with an Oracle or MySQL backend.

21.1.2 LFC Client Overview

This deployment module configures the lfc client in order to contact a LFC server.

21.2 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.

21.2.1 Oracle

If you want to use Oracle as a backend for the LFC server you need:

1. Oracle Database backend

   If you want to use Oracle as a backend database, you will need to have the Oracle database already installed on the same or on a remote host.

2. Oracle client

   In order for the LFC server to connect to the ORACLE database you will need to install the ORACLE instant client libraries for jdbc and sqlplus. This release requires v.10.1.0.3.

   Due to license reasons, we cannot redistribute these libraries. Please download them from http://www.oracle.com and install them if you have not yet installed them yet

21.2.2 MySQL

If you want to use MySQL as a backend you don't need extra libraries. MySQL is downloaded and installed together with the MySQL version of the LFC server.

21.3 LFC SERVER

21.3.1 LFC Server Installation

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

   e. If the backend is **MySQL**

   apt-get install glite-lfc-mysql-config

     f.   If the backend is **Oracle**

       apt-get install glite-lfc-oracle-config

2. Method 2: Download from the gLite web site the latest version of the the gLite LFC installation script.

     g.   In order to configure LFC with a MySQL backend download:
         glite-lfc-mysql_installer.sh

     h.   If the backend is Oracle, the installer script is:
         glite-lfc-oracle_installer.sh

   Make the file executable (chmod u+x glite-lfc-mysql_installer.sh or glite-lfc-oracle_installer.sh) and execute it

3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-lfc-mysql or glite-lfc-oracle next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.

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

   gLite   in /opt/glite ($GLITE_LOCATION)
   lfc   in /opt/lcg/
   MySQL   in /usr/bin/mysql   (in case of the MySQL version)

5. The gLite lfc configuration script (the same configuration script is used to configure lfc with a MySQL and Oracle backend) is installed in *$GLITE_LOCATION/etc/config/scripts/glite-lfc-config.py*. A template configuration file is installed in *$GLITE_LOCATION/etc/config/templates/glite-lfc.cfg.xml*

6. The gLite lfc 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.

7. The gLite lfc installs the RGMA-GIN module to publish the LFC information in the R-GMA.

21.3.2 LFC 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-lfc.cfg.xml

   to

   $GLITE_LOCATION/etc/config/glite-lfc.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 15. The R-GMA servicetool related parameters can be found in **Table 7**

   The parameters in the file are the following ones:

Name	Default Value	Description
User-defined Parameters		
db.type		Database backend to be used. It can be mysql or oracle
db.lfc.user		LFC user to access the database. Example lfc
db.lfc.password		LFC user password to access the database. Example lfc_password
db.host		Name of the machine running the database. Example: localhost, oradev10.cern.ch
mysql.root.password		The mysql root password
db.name		Database name. Example: devdb10. If

Name	Default Value	Description
		the database type is mysql this parameter can be left empty
db.sid		If the database type is mysql this parameter can be left empty
db.port		Por where the database server will be waiting for connections. If the database type is mysql this parameter can be left empty
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
glite.installer.checkcerts	true	Enable check of host certificates
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)
lcg.providers.location	/opt/lcg	The locations were the LCG Info Providers are installed
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
set.mysql.root.passwor	true	If this parameter is true,

Name	Default Value	Description
d		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
db.tnsadmin		Location of the tnsnames.ora file. If left empty the deployment script will create it automatically. Otherwise, a correct lcoation to the tnsnames file must be specified.
oracle-instantclient.location	/usr/lib/oracle/10.1.0.3/ client	Location of the Oracle Instantclient installation
trusted.hosts		List of the trusted host of this LFC server
lfc.user	lfcmgr	The user name used to run the lfc services
lfc.group	lfcmgr	The group name of the user used to run the lfc services
System Parameters		
db.protocol	TCP	Database protocol.
rgma.servicetool.activate	true	Turn on/off servicetool for the node. [Example: true] [Type: 'boolean']

Table 34: LFC Configuration Parameters

4. 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 1 shows a list of the parameters that can be set. More details can be found in section 4.3.2.

The dli and lcfdaemon are published via the R-GMA servicetool:

    v.   dli

    vi. lcfdaemon

5. Configure the RGMA-GIN:

Copy the RGMA-GIN configuration file template

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

to

$GLITE_LOCATION/etc/config

Modify the rgma-gin configuration file by specifying in the *rgma.gin.run_generic_info_provider* the value yes. The *rgma.gin.run_fmon_provider* should have the value no.

RGMA-gin is used to publish LFC information in

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

6. As root run the LFC Configuration script (with the –configure option in order to configure the service) */opt/glite/etc/config/scripts/glite-lfc-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 LFC Server services by running the configuration script with the –start option.

    /opt/glite/etc/config/scripts/glite-lfc-config.py --start

21.3.3 LFC Server Configuration Walkthrough

The LFC Server configuration script performs the following steps:

1. Load the LFC configuration file $GLITE_LOCATION/etc/config/glite-lfc.cfg.xml and the servicetool configuration file $GLITE_LOCATION/etc/config/glite-rgma-servicetool.cfg.xml

2. Stop the services that are running
3. If the backend is Oracle:
    - Check that the Oracle client is installed.
    - Create the tnsnames Oracle file
4. Create the lfc user and group
5. Create the lfc certificates directory, copy them and assign the right permissions
6. Configure the grid-mapfile
7. Create the pool accounts
8. Configure the security utils
9. If the backend is MySQL:
    - Start MySQL
    - Check and set the root password
10. Create the /etc/shift.conf file
11. Check if the lfc database exists and create it in case it does not
12. Create the lfcdaemon configuration file
13. Start the lfc daemon
14. Create the initial file system taking into account the existing Vos
15. Create the dli configuration file
16. Create the servicetool instances and configure the servicetool
17. Create the LCG info provider configuration file
18. Configure RGMA-GIN

21.3.4 Managing the LFC Server Service

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

glite-lfc-config.py –configure	Configures all the LFC services (lfcdaemon, dli, servicetool and rgma-gin)
glite-lfc-config.py –ldif	Generates the LFC info provider configuration file
glite-lfc-config.py –start	Starts all LFC services (or restart them if they are already running)
glite-lfc-config.py –stop	Stops all LFC services (lfcdaemon, dli, servicetool and rgma-gin)
glite-lfc-config.py –status	Checks the status of the LFC services

21.4 LFC CLIENT

21.4.1 LFC CLIENT Installation

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

   apt-get install glite-lfc-client-config

2. Method 2: Download from the gLite web site the latest version of the the gLite LFC Client installation script glite-lfc-client_installer.sh. Make the file executable (chmod u+x glite-lfc-client_installer.sh) and execute it

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

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

   gLite   in /opt/glite ($GLITE_LOCATION)
   LFC client   in /opt/lfc

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

21.4.2 LFC 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

2. Copy the configuration file template from *$GLITE_LOCATION/etc/config/templates/glite-lfc-client.cfg.xml* to *$GLITE_LOCATION/etc/config/glite-lfc-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*

The lfc-client configuration parameters values are the following:

Name	Default Value	Description
User-defined Parameters		
lfc.server		LFC server. [Example: 'lxb0755.cern.ch'] [Type: 'string']
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output. [Example: 'true'] [Type: 'boolean']

Table: LFC Client Configuration Parameters

3. As root run the LFC Client Configuration file with the –configure option
   /opt/glite/etc/config/scripts/glite-lfc-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.

4. *As* root start the LFC Client by running the LFC Client Configuration File:
   /opt/glite/etc/config/scripts/glite-lfc-client-config.py --start

21.4.3 LFC Client Configuration Walkthrough

The LFC Client configuration script performs the following steps:

1. Load the LFC Client configuration file *$GLITE_LOCATION/etc/config/glite-lfc-client.cfg.xml*
2. Set up the environment and create the LFC_HOST environment variable
3. Configure the security utils

21.4.4 Managing the LFC CLIENT

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

glite-lfc-client-config.py configure	--	Configures the LFC Client
glite-lfc-client-config.py --subservice		This option is mainly used by services calling a sequence of clients to be configured. This option should be used with the –configure option. Example: glite-lfc-client -config.py –subservice --configure

22 AMGA SERVER

22.1 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.

22.2 INSTALLATION PRE-REQUISITES

22.2.1 Security Settings

Install one or more Certificate Authorities certificates in */etc/grid-security/certificates*. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (http://www.gridpma.org/). A special security module called *glite-security-utils* can be installed by downloading and running from the gLite web site (http://www.glite.org/) the script *glite-security-utils_installer.sh* (Chapter 13). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl* script and sets up a crontab that periodically check for updated revocation lists.

22.2.2 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.

22.2.3 Database backend ODBC drivers

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

22.2.4 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 SQLlite. 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 SQLlite 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.

22.3 AMGA SERVER INSTALLATION

It is possible to install the AMGA server as follows:

1. Method 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

2. Method 2: Download from the gLite web site the latest version of the the gLite AMGA server installation script glite-amga-server_installer.sh. Make the file executable (chmod u+x glite-amga-server_installer.sh) and execute it
3. Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-amga-server next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4. 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:

AMGA Server	in /opt/glite
AMGA Client	in /opt/glite
gLite R-GMA servicetool	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.

22.4 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 4.3.2 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 20 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.RequireAuthenti	no	Whether users need to be

cation		authenticated.
amga.server.AllowCertificate Authentication	no	Whether you allow users to authenticate with their certificate. The CA are automatically installed on the gLite installation and corresponding parameters loke (TrustedCertDir, etc) are set. See AMGA server users guide "User Management" (p. 15).
amga.server.AllowPassword Authentication	no	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).
amga.server.AllowGridProxy Login	no	Whether you allow users to authenticate with a proxy certificate.

Table 35: 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.

23   AMGA CLIENT

23.1   SERVICE OVERVIEW

CLI and C++ client to the AMGA server

23.2   INSTALLATION PRE-REQUISITES

23.2.1   Security Settings

Install one or more Certificate Authorities certificates in */etc/grid-security/certificates*. The complete list of CA certificates can be downloaded in RPMS format from the Grid Policy Management Authority web site (http://www.gridpma.org/). A special security module called *glite-security-utils* can be installed by downloading and running from the gLite web site (http://www.glite.org/) the script *glite-security-utils_installer.sh* (Chapter 13). The module contains the latest version of the CA certificates plus a number of certificate and security utilities. In particular this module installs the *glite-fetch-crl* script and sets up a crontab that periodically check for updated revocation lists.

23.3   AMGA CLIENT INSTALLATION

It is possible to install the AMGA client as follows:

1.   Method 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

2.   Method 2: Download from the gLite web site the latest version of the the gLite AMGA client installation script glite-amga-client_installer.sh. Make the file executable (chmod u+x glite-amga-client_installer.sh) and execute it
3.   Run the script as root. All the required RPMS are downloaded from the gLite software repository in the directory glite-amga-client next to the installation script and the installation procedure is started. If some RPM is already installed, it is upgraded if necessary. Check the screen output for errors or warnings.
4.   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.

23.4 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 4.3.2 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 20 shows the configuration values that can be set.

Parameter	Default value	Description
User-defined Parameters		
amga.client.Host		The name of the host to connect to. This option can be overridden on the command line of mdclient. (Default: localhost)
amga.client.Login		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		
glite.installer.verbose	true	
amga.client.Port	8822	Port of the mdserver to connect to. Can be overridden on the command line of mdclient using the -p option.
amga.client.PermissionMask	rw-	A 3 character string giving the owner permissions of newly created entries in the metadata catalogue.
amga.client.GroupMask	r--	A 3 character string giving the group permissions of newly created

		entries in the metadata catalogue.
amga.client.Home	/	The home-directory.
amga.client.UseSSL	require	Possible values are no, try, require (synonym is yes). Default is no. Needed for any authentication 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.
amga.client.AuthenticateWith Certificate	no	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.
amga.client.UseGridProxy	no	Tries to use the a grid proxy certificate in /tmp/x509up_u[user-id].
amga.client.VerifyServerCert	no	Verifies the server certificate against CA certificates.
amga.client.RequireDataEncr yption	no	

Table 36: 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:

ln -s $GLITE_LOCATION/etc/mdclient.config $HOME/.mdclient.config

24 USER INTERFACE

24.1 SERVICE OVERVIEW

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

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

24.2 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.

24.2.1 Security Settings

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

24.2.2 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.

24.3 UI INSTALLATION

The gLite User Interface can be installed as root or as non-privileged user. The installation procedure is virtually identical. The root installation installs by default the UI RPMS in the standard location /opt/glite.

The location of the gLite RPMS can be changed by means of the prefix command line switch.

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

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

The destination directory of both root and user installations can be modified by using of the basedir=<path> option of the ui installer script, where the <path> MUST be an absolute path. The installation steps are the same in both the root and no-root installation cases:

1. Download the latest version of the UI installation script

    glite-ui_installer.sh

    from the gLite web site. It is recommended to download the script in a clean directory.

2. Make the script executable

    chmod u+x glite-ui_installer.sh

    and execute it or execute it with

    sh glite-ui_installer.sh

    If needed, pass the basedir=<path> option to specify the target installation directory.

3. Run the script as root or as normal user. All the required RPMS are downloaded from the gLite software repository in the directory *glite-ui* next to the installation script and the installation procedure is started.
    If some RPM is already installed, they are upgraded if necessary. Check the screen output for errors or warnings. This step can fail in case if some of the OS RPMs are missing. These RPMs MUST be installed manually by the user from the OS distribution CD, or by apt/yum tools.

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

    a) *Root installation*
gLite	in /opt/glite	(= GLITE_LOCATION)
Globus	in /opt/globus	(= GLOBUS_LOCATION)
GPT	in /opt/gpt	(= GPT_LOCATION)

    b) *User installation*

    gLite, Globus and GPT (unless already installed) are installed in the tree from `pwd`/glite_ui by removing the /opt/[glite, globus, gpt] prefix.

    The GLITE_LOCATION, GLOBUS_LOCATION and GPT_LOCATION variables are set to the `pwd`/glite_ui value. If Globus and GPT are already installed before installing the gLite UI, they are not reinstalled and the existing GLOBUS_LOCATION and GPT_LOCATION can be used.

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

6. The script will install the following deployment modules:

    - Worker Node
    - R-GMA client (see section 6.3 for details)
    - File Transfer Service Client (see section 0 for details)
    - File Placement Service Client (see section 0 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:

gLite I/O Client	in /opt/glite
gLite LB Client	in /opt/glite
glite R-GMA Client	in /opt/glite
glite DGAS Client	in /opt/glite

gLite WMS Checkpointing in /opt/glite
gLite FTS client in /opt/glite
gLite Service Discovery in /opt/glite
Globus in /opt/globus

7. 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.

24.4 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 4.3.2 for the description about the general configuration.

- The file *glite-rgma-common.cfg.xml* contains the common R-GMA configuration values. Refer toTable 4 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 Table 6 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 10 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 38 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 37 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 11 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 WMProxy to contact. [Example: https://ghemon.cnaf.infn.it:7443/glite_wms_wmproxy_server] [Type: 'string']
ui.MyProxyServer		MyProxy server to use
ui.HLRLocation		Location of the HLR accounting server. Optional parameter. The syntax is hostname:port: and default port is 56568 [Example: lxb0001.cern.ch:56568:] [Type: string]

**Table 37: 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 **[New in gLite 1.5]**		LFC server. [Example: lxb0755.cern.ch] [Type: string]
Advanced Parameters		
glite.installer.verbose	true	Enable verbose output
py-ui.requirements **[Moved to Advanced parameters in gLite 1.5]**	other.GlueCEStateStatus == 'Production'	Requirements for job matchmaking for this VO

py-ui.rank	-other.GlueCEStateEstimatedResponseTime	Matchmaking rank.
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 Loggger.
py-ui. DefaultStatusLevel	1	Default status level.
py-ui. DefaultLogInfoLevel	1	Default level of logging.
wmproxy.ShallowRetryCount	10	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]
wmproxy.AllowZippedISB	true	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]
wmproxy.PerusalFileEnable	true	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]

ui.ClientList **[Modified in gLite 1.5]**	• glite-file-transfer-service-client • glite-io-client • glite-rgma-client • glite-lfc-client	The gLite clients or applications that compose this user interface. [Type: 'string'] Example: glite-rgma-client

System Parameters

Table 38: UI common configuration parameters

[New in gLite 1.5] The Service Discovery parameters for file-based discovery have been removed from the configuration file. File-based discovery is only supported for test or debugging. The normal mode of operation is based on R-GMA or BD-II discovery.

4. Run the UI configuration file

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

The gLite User Interface is now ready.

24.5 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.

24.6 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.

25 THE GLITE FUNCTIONAL TEST SUITES

25.1 OVERVIEW

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

25.2 I/O TEST SUITE

25.2.1 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.

25.2.2 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.

25.2.3 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.

25.2.4 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.

25.2.5 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)

25.2.6 Test results

a) From the command line:

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

b) Using TestManager:

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

25.3 CATALOG TEST SUITE

25.3.1 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.

25.3.2 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.

25.3.3 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.

25.3.4 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

25.3.5 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)

25.3.6 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.

25.4 WMS TEST SUITE

25.4.1 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.

25.4.2 Installation Pre-requisites

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

25.4.3 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.

25.4.4 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.

25.4.5 Execution

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

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

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

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

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

25.4.6 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.

25.5 WMS VALIDATION TEST SUITE

25.5.1 Test suite description

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

25.5.2 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.

25.5.3 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.

25.5.4 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.

25.5.5 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 executes)

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.

25.5.6 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.

25.6 R-GMA TEST SUITE

25.6.1 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?).

25.6.2 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.

25.6.3 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.

25.6.4 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.

25.6.5 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.

25.6.6 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.

26 SERVICE CONFIGURATION FILE EXAMPLE

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

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

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

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

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

                        <value>EGEE</value>

                </voms.voname>

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

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

                </voms.vomsnode>

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

                        <value>17001</value>

                </voms.vomsport>

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

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

                </voms.vomscertsubj>

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

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

                        <value>egee</value>

                </pool.account.basename>

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

                        <value>egeegr</value>

                </pool.account.group>
```

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

                        <value>40</value>

                </pool.account.number>

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

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

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

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

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

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

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

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

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

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

                </cemon.cesebinds>

                <cemon.queues
                        description="A space-separated list of the queues defined on this CE node
                        Example: blah-pbs-egee-high"
                        value=" blah-pbs-egee-high "/>
<!--            <!-- LSF configuration
                         The following parameters are specific to LSF. They may have to be set
                         depending on your local LSF configuration.
                         If LSF is not used, remove this section -->

                <pool.lsfgid
```

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

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

                        <value>changeme</value>

                </pool.lsfgid>

-->

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

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

                <!-- Firewall configuration -->
                <iptables.chain
                        description="The name of the chain to be used for configuring the local firewall.
                        If the chain doesn't exist, it is created and the rules are assigned to this
chain.
                        If the chain exists, the rules are appended to the existing chain"
                        value="EGEE-DEFAULT-INPUT"/>

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

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

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

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

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

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

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

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

                <globus.hostdn
                        description="The host distinguished name (DN) of this node. This is mormally
automatically
                        read from the server host certificate. However it can be set here. For example:
                        'C=ORG, O=DOMAIN, OU=GRID, CN=host/server.domain.org'"
                        value=""/>
```

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

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

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

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

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

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

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

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

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

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

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

    </parameters>
</config>
```

27 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>

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

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

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

              <!-- 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"/>

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

      </parameters>

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

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

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

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

                      <value>EGEE</value>

              </voms.voname>

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

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

            </voms.vomsnode>

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

                    <value>15001</value>

            </voms.vomsport>

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

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

            </voms.vomscertsubj>

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

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

                    <value>egee</value>

            </pool.account.basename>

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

                    <value>egeegr</value>

            </pool.account.group>

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

                    <value>40</value>

            </pool.account.number>

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

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

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

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

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

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

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

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

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

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

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

            </cemon.cesebinds>

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

            <!-- LSF configuration
                         The following parameters are specific to LSF. They may have to be set
                         depending on your local LSF configuration.
                         If LSF is not used, remove this section -->
<!--            <pool.lsfgid
                        description="The gid of the groups to be used for the pool accounts on some LSF
installations,
                        on per each pool account group. This parameter is an array of values containing
one value
                        for each VO served by this CE node. The list must match
                        the corresponding lists in the VOMS configuration section

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

                        <value></value>

            </pool.lsfgid>

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

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

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

```
                      If the chain doesn't exist, it is created and the rules are assigned to this
  chain.
                      If the chain exists, the rules are appended to the existing chain"
                      value="EGEE-DEFAULT-INPUT"/>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

           <condor.blahpollinterval
```

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

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

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

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