

Table of Contents

Generic Installation and Configuration Guide for gLite 3.2.....	1
Support.....	2
Introduction to Manual Installation and Configuration.....	3
Installing the Operating System.....	4
Scientific Linux 5.....	4
Node synchronization, NTP installation and configuration.....	4
Cron and logrotate.....	5
Host Certificates.....	6
Installing the Middleware.....	7
Repositories.....	7
The middleware repositories.....	7
The Certification Authority repository.....	8
jpackage and the JAVA repository.....	8
The DAG repository.....	8
Installations.....	9
Note on the installation of the glite-APEL node.....	9
Updates.....	10
Normal updates.....	10
Important note on automatic updates.....	10
Upgrading from gLite 3.1.....	10
Configuring the Middleware.....	11
Using the YAIM configuration tool.....	11
Configuring multiple node types on the same physical host.....	11
Installing and Configuring a batch system.....	12
The Torque/PBS batch system.....	12
The WN for Torque/PBS.....	12
The UI.....	12
The LSF batch system.....	12
The WN for LSF.....	12
The lcg-CE for LSF.....	13
Note on site-BDII for LSF.....	13
The SGE batch system.....	14
The CREAM-CE for SGE.....	14
Configure CREAM-CE and SGE Qmaster in the same physical machine.....	14
Configure CREAM-CE and SGE Qmaster in different machines.....	15
Link the CREAM-CE with a running SGE Qmaster server.....	17
The WN for SGE.....	18
Note on lcg-vomscerts package.....	20
Note on hostname syntax.....	21
Note on openldap.....	22
Firewalls.....	23

Table of Contents

Documentation.....	24
--------------------	----

Generic Installation and Configuration Guide for gLite 3.2

This document is addressed to Site Administrators responsible for middleware installation and configuration. It is a generic guide to manual installation and configuration for any supported node types. This guide is for gLite release 3.2, if you configure gLite 3.1 services please check the previous version of this guide.

Support

Please open a GGUS ticket if you experience any Installation or Configuration problem.

Your contact point for technical support is your ROC <http://egee-sa1.web.cern.ch/egee-sa1/roc.html> but if you need to contact the release team, please send a mail to gd-release-team@cernNOSPAM.ch.

Introduction to Manual Installation and Configuration

This document is addressed to Site Administrators responsible for middleware installation and configuration. It is a generic guide to manual installation and configuration for any supported node types. It provides a fast method to install and configure the gLite middleware version 3.2.

The list of supported node types can be found in the gLite 3.2 [web pages](#).

Note that glite-UI and glite-WN are installed in compatibility mode and they also include 32bit versions of the following packages:

- WMS clients
- LB clients
- DPM client libraries and python bindings
- LFC client libraries and python bindings
- GFAL and lcg_utils, including python bindings
- VOMS APIs
- dCache client libraries

When installing a particular node type please also have a look at the specific release page of that node type (links are available on the main gLite 3.2 release page [to get specific installation information](#)).

The supported installation method for SL5 is the `yum` tool. Please note that **YAIM IS NOT SUPPORTING INSTALLATION** you have to configure yum repositories yourself and install the metapackages using your preferred way.

The configuration is performed by the YAIM tool. For a description of YAIM check YAIM guide

YAIM can be used by Site Administrators without any knowledge of specific middleware configuration details. They must define a set of variables in some configuration files, according to their site needs.

Installing the Operating System

Scientific Linux 5

The OS version of gLite Middleware version 3.2 is Scientific Linux 5 (SL). For more information please check:

<http://www.scientificlinux.org>

All the information to install the operating system can be found : <https://www.scientificlinux.org/download> [↗](#)
</verbatim> Middleware testing has been also carried out on Scientific Linux SL5 installed on Virtual Machines.

Node synchronization, NTP installation and configuration

A general requirement for the gLite nodes is that they are synchronized. This requirement may be fulfilled in several ways. If your nodes run under AFS they are most likely already synchronized. Otherwise, you can use the NTP protocol with a time server.

Instructions and examples for a NTP client configuration are provided in this section. If you are not planning to use a time server on your machine you can just skip this section.

Use the latest ntp version available for your system. If you are using APT, an apt-get install ntp will do the work.

- Configure the file /etc/ntp.conf by adding the lines dealing with your time server configuration such as, for instance:

```
restrict <time_server_IP_address> mask 255.255.255.255 nomodify notrap noquery  
server <time_server_name>
```

Additional time servers can be added for better performance results. For each server, the hostname and IP address are required. Then, for each time-server you are using, add a couple of lines similar to the ones shown above into the file /etc/ntp.conf.

- Edit the file /etc/ntp/step-tickers adding a list of your time server(s) hostname(s), as in the following example:

```
137.138.16.69  
137.138.17.69
```

- If you are running a kernel firewall, you will have to allow inbound communication on the NTP port. If you are using iptables, you can add the following to /etc/sysconfig/iptables

```
-A INPUT -s NTP-serverIP-1 -p udp --dport 123 -j ACCEPT  
-A INPUT -s NTP-serverIP-2 -p udp --dport 123 -j ACCEPT
```

Remember that, in the provided examples, rules are parsed in order, so ensure that there are no matching REJECT lines preceding those that you add. You can then reload the firewall

```
# /etc/init.d/iptables restart
```

- Activate the ntpd service with the following commands:

```
# ntpdate <your ntp server name>  
# service ntpd start  
# chkconfig ntpd on
```

- You can check ntpd's status by running the following command

```
# ntpq -p
```

Cron and logrotate

Many middleware components rely on the presence of `cron` (including support for `/etc/cron.*` directories) and `logrotate`. You should make sure these utils are available on your system.

Host Certificates

All nodes except UI, WN and BDII require the host certificate/key files to be installed. Contact your national Certification Authority (CA) to understand how to obtain a host certificate if you do not have one already. Instructions to obtain a CA list can be found here:

- <http://grid-deployment.web.cern.ch/grid-deployment/lcg2CAlist.html>

Once you have obtained a valid certificate:

- `hostcert.pem` - containing the machine public key
- `hostkey.pem` - containing the machine private key

make sure to place the two files in the target node into the `/etc/grid-security` directory and check the access right for `hostkey.pem` is only readable by root and that the public key, `hostcert.pem`, is readable by everybody.

Installing the Middleware

Please before you proceed further **make sure** that Java is installed in your system.

As of SL5 the `yum` package manager is considered the to be the default installation tool. As the installation is not supported by YAIM, you have to install the metapackages on your own.

Repositories

For a successful installation, you will need to configure your package manager to reference a number of repositories (in addition to your OS);

- the middleware repositories
- the CA repository
- DAG
- SL

and to **REMOVE (!!!)** or **DEACTIVATE (!!!)**

- EPEL repository

Why? Because it contains `gLite/Globus/...` packages that are **not** certified for the `gLite` distribution and that will cause problems when they replace packages provided by the `gLite` repositories.

The middleware repositories

`gLite` is distributed in multiple `yum` repositories. Each node type has its own independent repository. Inside each node type repository there are in fact three repositories called `RPMS.release`, `RPMS.updates` and `RPMS.externals`, each of them with its own `repopdata`. These repositories contain only the relevant `rpms` for each node type. To save space, all the `rpms` are stored in a directory called `gLite-GENERIC` (with no `repopdata`) and there are symbolic links to the packages in `gLite-GENERIC` from the different repositories.

Due to this repository structure we recommend to use

```
yum >= 3.2.19
```

since some installation problems have been reported with lower versions of `yum`.

In `gLite 3.2`, packages in `RPMS.release` and `RPMS.updates` are signed with our `gpg` key:

The public key can be downloaded from: http://glite.web.cern.ch/glite/glite_key_gd.asc 

The fingerprint for the **new** key is:

```
pub 1024D/836AAC2B 2010-03-26 [expires: 2011-03-26]
    Key fingerprint = 84B8 7FDA 0208 63A0 BA63 D596 F701 EE87 836A AC2B
uid                               Gd Integration <gd-release-team@cern.ch>
sub 2048g/5F397E3E 2010-03-26 [expires: 2011-03-26]
```

The fingerprint for the **old** key is:

```
pub 1024D/D2734F10 2009-03-10 [expires: 2010-03-10]
    Key fingerprint = AC27 E294 BF80 6470 7229 C039 456B 965C D273 4F10
uid                               Gd Integration <gd-release-team@cern.ch>
sub 2048g/701792A5 2009-03-10 [expires: 2010-03-10]
```

The 3.2 repository can be found under:

`http://linuxsoft.cern.ch/EGEE/gLite/R3.2/`

To use yum, `wget` the yum repository for the node type you want to install from the following web address and copy it in `/etc/yum.repos.d`:

- <http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/3.2/>

Note that installation of several node types in the same physical host is not recommended. The repositories of each node type may not be synchronised for the same package and this can cause problems.

The Certification Authority repository

The most up-to-date version of the list of trusted Certification Authorities (CA) is needed on your node. As the list and structure of the Certification Authorities (CA) accepted by the LCG project can change independently of the middleware releases, the rpm list related to the CAs certificates and URLs has been decoupled from the standard gLite/LCG release procedure.

Please note that the lcg-CA metapackage and repository is no longer maintained. The lcg-CA repository should be now replaced by the EGI trustanchors repository. All the details on how to install the CAs can be found in EGI IGTF release pages.

jpackage and the JAVA repository

As of SL 5.3 java is included in the Scientific Linux 5 distribution so the jpackage repository is not needed anymore.

Some gLite 3.2 node types need java. They don't contain any dependencies to automatically install java, so they rely on java to be installed in the machine. The relevant node types are:

- glite-APEL
- glite-ARGUS
- glite-CONDOR_utils
- glite-LSF_utils
- glite-SGE_utils
- glite-TORQUE_utils
- glite-UI (due to dCache clients + SAGA adpaters)
- glite-VOBOX (due to dCache clients)
- glite-WN (due to dCache clients + SAGA adpaters)
- glite-FTS_oracle

If you are installing any of these node types, please make sure java (OpenJDK 1.6 or Sun JDK 1.6) is previously installed in the machine.

The DAG repository

DAG is a maintained repository which provides a number of packages not available through Scientific Linux. If you have installed the CERN version of Scientific Linux, you will find that the relevant file is already installed in `/etc/yum.repos.d`. Otherwise, please use the following

```
[main]
[dag]
name=DAG (http://dag.wieers.com) additional RPMS repository
baseurl=http://linuxsoft.cern.ch/dag/redhat/el5/en/$basearch/dag
```

```
gpgkey=http://linuxsoft.cern.ch/cern/slc5X/$basearch/RPM-GPG-KEYS/RPM-GPG-KEY-dag
gpgcheck=1
enabled=1
```

Installations

Here it is an example on how to install a node:

```
cd /etc/yum.repos.d/
wget http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/3.2/glite-TORQUE_client.repo
yum update
yum install glite-TORQUE_client
```

The table below lists the available meta-packages and the associated repo file name in <http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/3.2/>.

Node Type	meta-package name	repo file	Comments
APEL	glite-APEL	glite-APEL.repo	
ARGUS	glite-ARGUS	glite-ARGUS.repo	
BDII	glite-BDII	glite-BDII.repo	
CREAM	glite-CREAM	glite-CREAM.repo	
GLEEXEC_wn	glite-GLEEXEC_wn	glite-GLEEXEC_wn.repo	The GLEEXEC_wn should always be installed together with a WN.
LB	glite-LB	glite-LB.repo	
LFC mysql	glite-LFC_mysql	glite-LFC_mysql.repo	
LFC oracle	glite-LFC_oracle	glite-LFC_oracle.repo	
MPI_utils	glite-MPI_utils	glite-MPI_utils.repo	
LSF_utils	glite-LSF_utils	glite-LSF_utils.repo	
DPM mysql	glite-SE_dpm_mysql	glite-SE_dpm_mysql.repo	
DPM disk	glite-SE_dpm_disk	glite-SE_dpm_disk.repo	
SCAS	glite-SCAS	glite-SCAS.repo	
SGE_utils	glite-SGE_utils	glite-SGE_utils.repo	
TORQUE client	glite-TORQUE_client	glite-TORQUE_client.repo	
TORQUE_server	glite-TORQUE_server	glite-TORQUE_server.repo	
TORQUE_utils	glite-TORQUE_utils	glite-TORQUE_utils.repo	
User Interface	glite-UI	glite-UI.repo	Use "yum groupinstall glite-UI"
VOBOX	glite-VOBOX	glite-VOBOX.repo	
VOMS_mysql	glite-VOMS_mysql	glite-VOMS_mysql.repo	
VOMS_oracle	glite-VOMS_oracle	glite-VOMS_oracle.repo	
Worker Node	glite-WN	glite-WN.repo	Use "yum groupinstall glite-WN"

For the TAR WN and the TAR UI, please check the following wiki pages:

- TAR WN Installation and Configuration
- TAR UI Installation and Configuration

Note on the installation of the glite-APEL node

In order to install the glite-APEL node you need to install manually the MySQL server. Please, run the following command:

```
yum install mysql-server
```

Updates

Normal updates

Updates to gLite 3.2 will be released regularly.

If an update has been released, a `yum update` should be all that is required to update the rpms. If you want to update the UI or the WN, you need to run `yum groupupdate glite-UI` or `yum groupupdate glite-WN` in order to properly get new dependencies as well.

NOTE that even if the recommendation is to use `yum update`, some sys admins are used to run `yum update metapackage-name`. This doesn't work in the last production releases due to a change in the way the dependencies are specified in the metapackage.

If reconfiguration of any kind is necessary, just run the following command (**don't forget to list *all* node types installed in your host**):

```
/opt/glite/yaim/bin/yaim -c -s site-info.def -n <node_type> [ -n <node_type> ... ]
```

Important note on automatic updates

Several site use auto update mechanism. Sometimes middleware updates require non-trivial configuration changes or a reconfiguration of the service. This could involve database schema changes, service restarts, new configuration files, etc, which makes it difficult to ensure that automatic updates will not break a service. Thus

WE STRONGLY RECOMMEND NOT TO USE AUTOMATIC UPDATE PROCEDURE OF ANY KIND

on the gLite middleware repositories (you can keep it turned on for the OS). You should read the update docs and do the upgrade manually when an update has been released!

Upgrading from gLite 3.1

As gLite 3.2 is the first release of the middleware for SL5, there is no supported upgrade path from gLite 3.1 on SL4.

Configuring the Middleware

Using the YAIM configuration tool

For a detailed description on how to configure the middleware with YAIM, please check the YAIM guide.

The necessary YAIM modules needed to configure a certain node type are automatically installed with the middleware. However, if you want to install YAIM rpms separately, you can install the repository of the node type you are interested in, as explained in the section [Middleware repositories](#) and then run `yum install glite-yaim-node-type`. This will automatically install the YAIM module you are interested in together with `yaim core`, which contains the core functions and utilities used by all the YAIM modules.

Configuring multiple node types on the same physical host

Note that installation and configuration of several node types in the same physical host is not recommended. The repositories of each node type are now independent and may not be synchronised for the same package, which can cause problems.

Installing and Configuring a batch system

The Torque/PBS batch system

```
cd /etc/yum.repos.d/  
wget http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/3.2/glite-TORQUE_client.repo  
yum update  
yum install the_necessary_metapackage_
```

The WN for Torque/PBS

After fetching the `glite-WN` repository (see above) use the following commands for the 64bit architecture:

```
wget http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/3.2/glite-WN.repo  
yum groupinstall glite-WN  
yum install glite-TORQUE_client
```

In order to configure a Torque WN, you have to specify all the configuration target in one line:

```
yaim -c -s site-info.def -n glite-WN -n TORQUE_client
```

The UI

After fetching the `glite-UI` repository (see above) use the following commands for the 64bit architecture:

```
wget http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/3.2/glite-UI.repo  
wget http://grid-deployment.web.cern.ch/grid-deployment/glite/repos/3.2/dag.repo  
yum groupinstall glite-UI
```

In order to configure a Torque UI, you have to specify all the configuration target in one line:

```
yaim -c -s site-info.def -n glite-UI
```

The LSF batch system

You have to make sure that the necessary packages for submitting jobs to your LSF batch system are installed on your CE. By default, the packages come as tar balls. At CERN they are converted into rpms so that they can be automatically rolled out and installed in a clean way (in this case using Quattor).

Since LSF is a commercial software it is not distributed together with the gLite middleware. Visit the Platform's LSF home page [for](#) further information. You'll also need to buy an appropriate number of license keys before you can use the product.

The documentation for LSF is available on Platform Manuals [web](#) page. You have to register in order to be able to access it.

For questions related to LSF and LCG/gLite interaction, you can use the project-eu-egee-batchsystem-lsf@cern.NOSPAMPLEASE.ch mailing list.

The WN for LSF

Apart from the LSF specific configurations settings there is nothing special to do on the worker nodes. Just use the plain WN configuration target.

```
./yaim -c -s site-info.def -n glite-WN
```

The lcg-CE for LSF

There is some special configuration settings you need to apply when configuring your LSF batch system. The most important variables to set in YAIM's `site-info.def` file are:

```
JOB_MANAGER="lcglsf"
TORQUE_SERVER="machine where the gLite LSF log file parser runs"
BATCH_LOG_DIR="/path/to/where/the/lsf/accounting/and/event/files/are"
BATCH_BIN_DIR="/path/to/where/the/lsf/executables/are"
BATCH_VERSION="LSF_6.1"
CE_BATCH_SYS="lsf"
```

For gLite installations you may use the gLite LSF log file parser daemon to access LSF accounting data over the network. The daemon needs to access the LSF event log files which you can find on the master (or some common file system which you may use for fail over). By default, yaim assumes that the daemon runs on the CE in which case you have to make sure that the event log files are readable from the CE. But note that it is not a good idea to run the LSF master service on the CE.

Make sure that you are using `lcg-info-dynamic-lsf-2.0.36` or newer.

To configure your lcg-CE use:

```
./yaim -c -s site-info.def -n lcg-CE -n LSF_utils
```

Note on site-BDII for LSF

When you configure your site-BDII you have to populate the `[vomap]` section of the `/opt/lcg/etc/lcg-info-dynamic-scheduler.conf` file yourself. This is because LSF's internal group mapping is hard to figure out from yaim, and to be on the safe side the site admin has to crosscheck. Yaim configures the `lcg-info-dynamic-scheduler` in order to use the LSF info provider plugin which comes with meaningful default values. If you would like to change it edit the `/opt/glite/etc/lcg-info-dynamic-lsf.conf` file. After YAIM configuration you have to list the LSF group - VOMS FQAN - mappings in the `[vomap]` section of the `/opt/lcg/etc/lcg-info-dynamic-scheduler.conf` file.

As an example you see here an extract from CERN's config file:

```
vomap :
  grid_ATLAS:atlas
  grid_ATLASSGM:/atlas/Role=lcgadmin
  grid_ATLASPRD:/atlas/Role=production
  grid_ALICE:alice
  grid_ALICESGM:/alice/Role=lcgadmin
  grid_ALICEPRD:/alice/Role=production
  grid_CMS:cms
  grid_CMSSGM:/cms/Role=lcgadmin
  grid_CMSPRD:/cms/Role=production
  grid_LHCB:lhcb
  grid_LHCBSGM:/lhcb/Role=lcgadmin
  grid_LHCBPRD:/lhcb/Role=production
  grid_GEAR:gear
  grid_GEARSGM:/gear/Role=lcgadmin
  grid_GEANT4:geant4
  grid_GEANT4SGM:/geant4/Role=lcgadmin
  grid_UNOSAT:unosat
  grid_UNOSAT:/unosat/Role=lcgadmin
  grid_SIXT:sixt
```

```

grid_SIXTSGM:/sixt/Role=lcgadmin
grid_EELA:eela
grid_EELASGM:/eela/Role=lcgadmin
grid_DTEAM:dteam
grid_DTEAMSGM:/dteam/Role=lcgadmin
grid_DTEAMPRD:/dteam/Role=production
grid_OPS:ops
grid_OPSSGM:/ops/Role=lcgadmin
module_search_path : ../lrms:../ett

```

For further details see the `/opt/glite/share/doc/lcg-info-dynamic-lsf` file.

The SGE batch system

DISCLAIMER: The SGE/gLite integration software was the result of the collaboration between 3 institutions: LIP, CESGA and LeSC. You use this software at your own risk. It may be not fully optimized or correct and therefore, should be considered as experimental. There is no guarantee that it is compatible with the way in which your site is configured.

For questions related to SGE and gLite interaction, you can use the `project-eu-egEE-batchsystem-sge@cernNOSPAMPLEASE.ch` mailing list.

The CREAM-CE for SGE

Note: The CREAMCE must be installed in a separate node from the SGE QMASTER, and the same SGE software version should be used in both cases.

Configure CREAM-CE and SGE Qmaster in the same physical machine

Note 1: This option is not recommended, but in any case, it could be done if it is necessary.

Note 2: In this example we do assume that no SGE NFS instalation will be used.

- Install SGE rpms (require **openmotif** and **xorg-x11-xauth** packages available in CERN SLC repositories). These rpms will install SGE files under `/usr/local/sge/pro`:

```
# yum localinstall sge-parallel-V62u1-1.i386.rpm sge-utils-V62u1-1.i386.rpm sge-docs-V6
```

- Install the `glite-CREAM*` and **glite-SGE_utils** meta rpm packages.

Note 3: Due to a dependency problem within the Tomcat distribution in SL5, it should be installed first the **xml-commons-apis** package before installing the `glite-CREAM` meta package.

```
# yum install xml-commons-apis
# yum install glite-CREAM glite-SGE_utils
```

- Download and install the **SGE server yaim interface** from ETICS repository:

```
# wget http://eticssoft.web.cern.ch/eticssoft/repository/org.glite/org.glite.yaim.sge-s
# yum localinstall glite-yaim-sge-server-4.1.1-1.noarch.rpm
```

- Set the following variables relevant in the **site-info.def** file: 💡 (Be also sure that you include the VOs and QUEUES information which you want to setup)

```

BATCH_SERVER="SGE Qmaster FQN"
BATCH_VERSION="SGE version"
BATCH_BIN_DIR="Directory where the SGE binary client tools are installed in the CE" Ex
BATCH_LOG_DIR="Path for the SGE accounting file". Ex: /usr/local/sge/pro/default/common
SGE_ROOT="The SGE installation dir". Default: /usr/local/sge/pro
SGE_CELL="SGE cell definition". Default: default
SGE_QMASTER="SGE qmaster port". Default: 536
SGE_EXECD="SGE execd port". Default: 537
SGE_SPOOL_METH="SGE spooling method". (Only the classic method is supported in the dist.
BLPARSER_WITH_UPDATER_NOTIFIER="true"
JOB_MANAGER=sge
CE_BATCH_SYS=sge

```

- Configure the CREAMCE, SGE_server and SGE_utils services: (in siteinfo/site-info.def the BATCH_SERVER variable should point to the CREAMCE machine)

```
# /opt/glite/yaim/bin/yaim -c -s siteinfo/site-info.def -n creamCE -n SGE_server -n SGE_utils
```

Note 4: If you run the SGE_server and SGE_utils YAIM configuration more than once, only the first action is valid. This is done to prevent overwriting the local site administrator local configuration tuning. In such cases, a warning is sent during the YAIM configuration procedure and a standard configuration template is stored in /tmp (which can be uploaded manually by the site administrator).

- The transferring of files between WN and CE is handled by a script, called sge_filestaging, which must be available in all WNs under /opt/glite/bin, and which you may find in your CreamCE installation under /opt/glite/bin/sge_filestaging. By default, this copy mechanism works with passwordless scp WN<->CreamCE but it is up to the site admin to set it up (YAIM will not take care of that task on your behalf). This script must be executed as prolog and epilog of your jobs. Therefore you should define /opt/glite/bin/sge_filestaging --stagein and /opt/glite/bin/sge_filestaging --stageout as prolog and epilog scripts, either in SGE global configuration "qconf -mconf" or in "each queue configuration "qconf -mq ". If you already have some prolog and epilog scripts defined, just add those definitions to your scripts. If your prolog and epilog scripts run as root, you will have to use su (for example, su -m -c "/opt/glite/bin/sge_filestaging --stageout" \$USER).

Configure CREAM-CE and SGE Qmaster in different machines

Note 1: In this example we do assume that no SGE NFS installation will be used.

- Install SGE rpms (require **openmotif** and **xorg-x11-xauth** packages available in CERN SLC repositories). These rpms will install SGE files under /usr/local/sge/pro:

```
# yum localinstall sge-parallel-V62u1-1.i386.rpm sge-utils-V62u1-1.i386.rpm sge-docs-V62u1-1.i386.rpm
```

- Install the glite-CREAM* and **glite-SGE_utils** meta rpm packages.

Note 2: Due to a dependency problem within the Tomcat distribution in SL5, it should be installed first the **xml-commons-apis** package before installing the glite-CREAM meta package.

```
# yum install xml-commons-apis
# yum install glite-CREAM glite-SGE_utils
```

- Set the following variables in the **site-info.def** file: 💡 (Be also sure that you include the VOs and QUEUES information which you want to setup)

```
BATCH_SERVER="SGE Qmaster FQN"
BATCH_VERSION="SGE version"
BATCH_BIN_DIR="Directory where the SGE binary client tools are installed in the CE" Ex
BATCH_LOG_DIR="Path for the SGE accounting file". Ex: /usr/local/sge/pro/default/common
SGE_ROOT="The SGE installation dir". Default: /usr/local/sge/pro
SGE_CELL="SGE cell definition". Default: default
SGE_QMASTER="SGE qmaster port". Default: 536
SGE_EXECD="SGE execd port". Default: 537
SGE_SPOOL_METH="SGE spooling method". (Only the classic method is supported in the dist.
BLPARSER_WITH_UPDATER_NOTIFIER="true"
JOB_MANAGER=sge
CE_BATCH_SYS=sge
```

- Configure the CREAMCE service (in siteinfo/site-info.def the BATCH_SERVER variable should point to the machine where your SGE Qmaster will run)

```
# /opt/glite/yaim/bin/yaim -c -s siteinfo/site-info.def -n creamCE -n SGE_utils
```

- Install all SGE rpms in the machine where the SGE Qmaster is supposed to run (require **openmotif** and **xorg-x11-xauth** packages available in CERN SLC repositories). These rpms will install SGE files under /usr/local/sge/pro:

```
# yum localinstall sge-parallel-V62u1-1.i386.rpm sge-utils-V62u1-1.i386.rpm sge-docs-V62u1-1.i
```

- Configure the SGE QMASTER service

```
# /opt/glite/yaim/bin/yaim -c -s siteinfo/site-info.def -n creamCE -n SGE_utils
```

- In the SGE Qmaster, declare the CE as an allowed submission machine:

```
# qconf -as <CE.MY.DOMAIN>
```

- If you have control of the SGE Qmaster, make sure that in the Qmaster configuration you have the following setting: **execd_params INHERIT_ENV=false**. This setting allows to propagate the environment of the submission machine (CE) into the execution machine (WN). It can be implemented in SGE QMASTER using:

```
# qconf -mconf
```

Note 3: If you run the SGE_server and SGE_utils YAIM configuration more than once, only the first action is valid. This is done to prevent overwriting the local site administrator local configuration tuning. In such cases, a warning is sent during the YAIM configuration procedure and a standard configuration template is stored in /tmp (which can be uploaded manually by the site administrator).

- The transferring of files between WN and CE is handled by a script, called sge_filestaging, which must be available in all WNs under /opt/glite/bin, and which you may find in your CreamCE installation under /opt/glite/bin/sge_filestaging. By default, this copy mechanism works with passwordless scp WN<->CreamCE but it is up to the site admin to set it up (YAIM will not take care of that task on your behalf). This script must be executed as prolog and epilog of your jobs. Therefore you should define **/opt/glite/bin/sge_filestaging --stagein** and **/opt/glite/bin/sge_filestaging --stageout** as prolog and epilog scripts, either in SGE global configuration "qconf -mconf" or in "each queue configuration "qconf -mq ". If you already have some prolog and epilog scripts defined, just add those definitions to your scripts. If your prolog and epilog scripts run as root, you will have to use

su (for example, `su -m -c "/opt/glite/bin/sge_filestaging --stageout" $USER`).

Link the CREAM-CE with a running SGE Qmaster server

- You should ensure that you are using the same SGE version for client and server tools, and that the SGE installation paths are the same in the CREAM-CE and in the SGE Qmaster server.
- If you are using a SGE installation shared via NFS or equivalent, and you do not want to change it with YAIM, you must set up the following variable in your `site-info.def` file. Its default value for this variable is "no", which means that SGE software WILL BE configured by YAIM.

```
# SGE_SHARED_INSTALL=yes
```

- If you are not using SGE installation shared via NFS or equivalent, install the SGE client tools in the CREAM-CE. For the SGE version described in this manual the following rpms should be deployed (require **openmotif** and **xorg-x11-xauth**, packages available in CERN SLC repositories). These rpms will install SGE files under `/usr/local/sge/pro`:

```
# yum localinstall sge-utils-V62u1-1.i386.rpm sge-V62u1-1.i386.rpm
```

- Install the `glite-CREAM*` and **glite-SGE_utils** meta rpm packages.

Note 1: Due to a dependency problem within the Tomcat distribution in SL5, it should be installed first the **xml-commons-apis** package before installing the `glite-CREAM` meta package.

```
# yum install xml-commons-apis
```

```
# yum install glite-CREAM glite-SGE_utils
```

- Change the following variables in the **site-info.def** file: 💡 (Be also sure that you include the VOs and QUEUES information which you want to setup)

```
BATCH_SERVER="SGE Qmaster FQN"
BATCH_VERSION="SGE version"
BATCH_BIN_DIR="Directory where the SGE binary client tools are installed in the CE" Ex
BATCH_LOG_DIR="Path for the SGE accounting file". Ex: /usr/local/sge/pro/default/common
SGE_ROOT="The SGE instalation dir". Default: /usr/local/sge/pro
SGE_CELL="SGE cell definition". Default: default
SGE_QMASTER="SGE qmaster port". Default: 536
SGE_EXECD="SGE execd port". Default: 537
SGE_SPOOL_METH="SGE spooling method". (Only the classic method is supported in the dist
BLPARSER_WITH_UPDATER_NOTIFIER="true"
JOB_MANAGER=sge
CE_BATCH_SYS=sge
```

- Configure the CREAMCE service

```
# /opt/glite/yaim/bin/yaim -c -s siteinfo/site-info.def -n creamCE -n SGE_utils
```

Note 2: If you run the `SGE_server` and `SGE_utils` YAIM configuration more than once, only the first action is valid. This is done to prevent overwriting the local site administrator local configuration tuning. In such cases, a warning is sent during the YAIM configuration procedure and a standard configuration template is stored in `/tmp` (which can be uploaded manually by the site administrator).

- In the SGE Qmaster, declare the CE as an allowed submission machine:

```
qconf -as <CE.MY.DOMAIN>
```

- If you have control of the SGE Qmaster, be sure that in the Qmaster configuration you have the following setting: **execd_params INHERIT_ENV=false**. This setting allows to propagate the environment of the submission machine (CE) into the execution machine (WN). It can be implemented in SGE QMASTER using:

```
# qconf -mconf
```

- The transferring of files between WN and CE is handled by a script, called `sgc_filestaging`, which must be available in all WNs under `/opt/glite/bin`, and which you may find in your CreamCE installation under `/opt/glite/bin/sgc_filestaging`. By default, this copy mechanism works with passwordless scp WN<->CreamCE but it is up to the site admin to set it up (YAIM will not take care of that task on your behalf). This script must be executed as prolog and epilog of your jobs. Therefore you should define `/opt/glite/bin/sgc_filestaging --stagein` and `/opt/glite/bin/sgc_filestaging --stageout` as prolog and epilog scripts, either in SGE global configuration "qconf -mconf" or in "each queue configuration "qconf -mq ". If you already have some prolog and epilog scripts defined, just add those definitions to your scripts. If your prolog and epilog scripts run as root, you will have to use `su` (for example, `su -m -c "/opt/glite/bin/sgc_filestaging --stageout" $USER`).

The WN for SGE

- If you are using a SGE installation shared via NFS or equivalent, and you do not want to change it with YAIM, you must set up the following variable in your `site-info.def` file. Its default value for this variable is "no", which means that SGE software WILL BE configured by YAIM.

```
# SGE_SHARED_INSTALL=yes
```

- If you are not using SGE installation shared via NFS or equivalent, install the following SGE rpms (require **openmotif** and **xorg-x11-xauth** packages available in CERN SLC repositories). These rpms will install SGE files under `/usr/local/sgc/pro`:

```
# yum localinstall sgc-parallel-V62u1-1.i386.rpm sgc-V62u1-1.i386.rpm sgc-utils-V62u1-1
```

- Install the `*glite-WN*`

```
# yum groupinstall glite-WN
```

- Download the **SGE client yaim interface** from ETICS repository and install it in the machine where the SGE client is supposed to run:

```
# yum http://eticssoft.web.cern.ch/eticssoft/repository/org.glite/org.glite.yaim.sgc-cl
```

- Configure the glite-WN and SGE client services:

```
# /opt/glite/yaim/bin/yaim -c -s siteinfo/site-info.def -n glite-WN -n SGE_client
```

- The transferring of files between WN and CE is handled by a script, called `sge_filestaging`. If you are using `glite-yaim-sge-client` to configure your WNs the `sge_filestaging` will be present by default in your WNs, and its execution is triggered by invoking `"/usr/local/sge/pro/default/queues_conf/prolog.sh"` and `"/opt/glite/bin/sge_filestaging --stageout"` as prolog and epilog. Both these files are constructed by YAIM and properly defined in the SGE QMASTER if you are also using the ETICS distributed solution for the SGE QMASTER installation.

Note 1: The transferring of files between WN and CE is handled by a script, called `sge_filestaging`, which must be available in all WNs under `/opt/glite/bin`, and which you may find in your CreamCE installation under `/opt/glite/bin/sge_filestaging`. By default, this copy mechanism works with passwordless scp WN<->CreamCE but it is up to the site admin to set it up (YAIM will not take care of that task on your behalf). This script must be executed as prolog and epilog of your jobs. If you are not using the **SGE client yaim interface** from ETICS repository, you should define `"/opt/glite/bin/sge_filestaging --stagein"` and `"/opt/glite/bin/sge_filestaging --stageout"` as prolog and epilog scripts, either in SGE global configuration `"qconf -mconf"` or in "each queue configuration `"qconf -mq "`. If you already have some prolog and epilog scripts defined, just add those definitions to your scripts. If your prolog and epilog scripts run as root, you will have to use **su** (for example, `su -m -c "/opt/glite/bin/sge_filestaging --stageout" $USER`).

Note on lcg-vomscerts package

lcg-vomscerts package is no longer installed in gLite 3.2. Instead, services rely on *.lsc files. These files are a description of the server certificate rather than the actual certificate. They contain DN/CA couples. The difference is that *.lsc files do not need to get updated if the server cert is renewed but keeps the same DN and is issued by the same CA. They are used to verify that the copy of the server certificate that all ACs contains respect that profile.

YAIM creates automatically the *.lsc files so sys admins do not need to worry about them.

Note on hostname syntax

The WLCG middleware assumes that hostnames are case sensitive. Site administrators **MUST** not choose mix case hostnames because of that. Actually all hostnames **MUST** be in lowercase since most of the WLCG middleware depends on Globus and in particular on the `globus_hostname` function that lowercases all hostnames. If hostnames are assigned using mix cases or uppercases, then any middleware that will compare hostnames as returned by the `globus_hostname` function and as provided by clients will fail.

Note on openldap

`openldap` is no longer installed as part of the middleware but it's taken from the operating system and it's installed with `openldap-clients`.

Firewalls

No automatic firewall configuration is provided by this version of the configuration scripts. If your nodes are behind a firewall, you will have to ask your network manager to open a few "holes" to allow external access to some service nodes. A complete map of which port has to be accessible for each service node is maintained in CVS;

http://jra1mw.cvs.cern.ch:8180/cgi-bin/jra1mw.cgi/org.glite.site-info.ports/doc/?only_with_tag=HEAD , or you can have a look to it's html version.

Documentation

For further documentation you can visit:

- the LCG Directory [↗](#)
- the LCG Troubleshooting Guide
- Sysadmin wiki [↗](#)
- or just find The Answer [↗](#) 😊

This topic: LCG > GenericInstallGuide320

Topic revision: r51 - 2011-04-08 - MariaALANDESPRADILLO



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback