LHC COMPUTING GRID

LCG RESOURCE BROKER MANUAL INSTALLATION AND CONFIGURATION

Document identifier:	**LCG-GIS-MI-RB**
EDMS id:	**none**
Version:	**v2.8.0**
Date:	**XX-Jun-2004**
Section:	**LCG Grid Deployment**
Document status:	**Status: Fully Working**
Author(s):	Alessandro Usai, Antonio Retico Alessandro.Usai@cern.ch, Antonio.Retico@cern.ch
File:	**RB**

Abstract: *LCG Resource Broker Manual Installation Guide*

CONTENTS

REFERENCES

[1] A. Retico and A. Usai. Lcg grid-mapfile handling on lcg nodes, 2004. http://www.cern.ch/grid-deployment/gis/release-docs/GMF-index.html

ABSTRACT

Manual steps for the installation and configuration of a LCG RB Node based on a standard (non CERN-like) Red Hat 7.3 installation

Link to this document:
This document is available on the *Grid Deployment* web site

http://www.cern.ch/grid-deployment/gis/release-docs/RB-index.html

There you need to select the version tag before downloading the corresponding version of the guide.
Oter documents labeled with ”cvs:”, in the *References*, are maintained in txt version into the LCG cvs repository.
They can be downloaded via web by browsing the LCG Deployment home page

http://www.cern.ch/grid-deployment/

following then the **CVS Deployment** link on the left side of the page.

1. PREREQUISITES FOR INSTALLATION

In order to install the RB software you need:

- a working Linux Redhat7.3 on the Resource Broker (This procedure has been tested on a basic RH7.3 and should not differ for more or less equivalent versions of RH. Should it be so please contact us)
- root account on RB machine
- references to software packages to be installed (see below)

2. RESOURCE BROKER SOFTWARE INSTALLATION

In order to have a running RB the following software must be installed

- RB package
- Other software which a group may rely on

Two different methods are supported for RB software installation

- Centrally managed RB installation using the Debian *apt-get* tool
- Manual RB installation using the Red Hat *rpm* tool.

The first method is simple and straightforward. Node-specific meta-packages define and hide the list of rpms to be installed. Dependencies among rpms, in particular, are easily managed by the *apt-get* tool. This may help if on your site you are running a customized version of Red Hat and you prefer not to solve dependencies manually.
On the other side, the installation using *apt-get* gives to site administrators poor control on installation and on the way the dependencies are solved.

The second method gives full control to site administrators for the installation, because it is based on an explicitly declared set of rpms to be installed on top of Red Hat 7.3
On the other hand, the installation procedure is slower and needs each unmanaged dependency to be manually solved.

2.1. RB INSTALLATION USING *apt-get*

in order to install your Resource Broker using *apt-get* please refer to the following document:

http://www.cern.ch/grid-deployment/gis/aptDB/

2.2. RB INSTALLATION USING *rpm*

All the tools needed for LCG releases are available on the *Grid Deployment* web site

http://www.cern.ch/grid-deployment/gis/release-docs/RB-index.html

You can start browsing by selecting the relevant version tag.
Once selected your tag, you will be moved to a page where you can find:

RB.pdf: this guide

RB_manual_install_rpm: plain full list of those rpms needed to install the RB software on top of a plain RH7.3 installation, whose dependencies are already dealt with.

RB_manual_install_forced_rpm: plain full list of those rpms needed to install the RB software on top of a plain RH7.3 installation, and whose dependencies are NOT dealt with, therefore requiring a forced installation. If this list is not present or void it means that no rpm needs to be forced

RB_manual_install_wget: a shell script which downloads, from the LCG mirror site at CERN, the rpms included into the list *RB_manual_install_rpm*

RB_manual_install_extra_rpm: a short list of rpms which RH7.3 does not install by default, but which have been turned out to be required in order to have the SE software installed. This list is provided just for reference, since the rpms from this list are already included into the list *RB_manual_install_rpm*

If any file of the mentioned list is not in the repository it means that it is not applicable within the installation of the current tag. So each further reference to it in this guide can be simply ignored.

All the above mentioned files are grouped in the archive file **RB_tools.tar.gz**. So they can be easily downloaded and used on target machines.

The provided *RB_manual_install_rpm* includes:

- all the packages which are needed in order to have a production Resource Broker.
- all the general purpose objects (i.e. shared libraries or common applications) which the RB installation needs in order to run and which are not included in the standard operative system. The composition of this group of rpms may vary depending on what is considered ”basic OS” at each site, and so it cannot be easily defined in a static way. Given the previous considerations, this part of the provided list may be not exhaustive.

A possible way to use the provided tools is decribed in the next sections

Download rpms from the repository in a local directory

- log in as *root* into the node to be installed and copy there the file **RB_tools.tar.gz**.
- extract the archived files.

  > *gunzip RB_tools.tar.gz*
  > *tar xvf RB_tools.tar*

- give execution permissions and run the script *RB_manual_install_wget*, which will then fetch all the rpms required into the subdirectories *RB_INSTALL* and *RB_INSTALL_FORCED*

  > *cd RB_tools*
  > *chmod u+x RB_manual_install_wget*
  > *./RB_manual_install_wget*

Install the downloaded rpms

- In the directory **RB_INSTALL**, where the rpms listed in *RB_manual_install_rpm* have been downloaded, run the command

  ```
  > rpm -ivh "*.rpm"
  ```

- In the directory **RB_INSTALL_FORCED** where the rpms listed in *RB_manual_install_forced_rpm* have been downloaded, run the command

  ```
  > rpm -ivh --nodeps "*.rpm"
  ```

  Due to possible missing rpms this operation might yield some missing dependencies which have to be analyzed and solved by adding/removing rpms to/from the *RB_manual_install_rpm* list.

Should you prefer to do an upgrade of the RB software already installed at your site instead of a new installation, the necessary steps are listed in the upgrade procedure usually reported in the last section of the RB installation guide.
This procedure, when available, refers to changes occurring between the current version and the previous one. This also means that the procedure is not applicable if the software at your site is more than two tags old.

3. NTP CLIENT INSTALLATION & CONFIGURATION

A general requirement for the LCG nodes is that they should be synchronized. This requirement may be fulfilled in several ways. If your nodes run under AFS most likely they are 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 it.

3.1. NTP SOFTWARE INSTALLATION

In order to install the NTP client, you need the following rpms to be installed:

- ntp
- libcap
- libcap-devel

The following versions of the above said rpms have been proven to work on our OS configuration (the list includes the corresponding links to download sites):

- ntp-4.1.1-1

  `http://grid-deployment.web.cern.ch/grid-deployment/download/RpmDir/release/ntp-4.1.1-1.i386.rpm`

- libcap-devel-1.10-8

  `http://grid-deployment.web.cern.ch/grid-deployment/download/RpmDir/release/libcap-devel-1.10-8.i386.rpm`

- libcap-1.10-8

  `http://grid-deployment.web.cern.ch/grid-deployment/download/RpmDir/release/libcap-1.10-8.i386.rpm`

A complete example of NTP rpms download and installation has been provided in 3.3..

3.2. NTP CONFIGURATION

- 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 ipchains, you can do this by adding the following to */etc/sysconfig/ipchains*

```
-A input -s <NTP-serverIP-1> -d 0/0 123 -p udp -j ACCEPT
-A input -s <NTP-serverIP-2> -d 0/0 123 -p udp -j ACCEPT
```

Remember that ipchains 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/ipchains restart
```

- Activate the ntpd service with the following commands:

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

A complete example of NTP configuration has been provided in 3.3.

3.3. EXAMPLE OF NTP CLIENT INSTALLATION AND CONFIGURATION

This section reports a complete example of NTP server configuration done in the CERN context.

- install NTP

```
> wget http://grid-deployment.web.cern.ch/grid-deployment/download/RpmDir/release/ntp-4.1.1-1.i386.rpm
> wget http://grid-deployment.web.cern.ch/grid-deployment/download/RpmDir/release/libcap-devel-1.10-8.i386.rpm
> wget  http://grid-deployment.web.cern.ch/grid-deployment/download/RpmDir/release/libcap-1.10-8.i386.rpm
> rpm -ivh libcap-1.10-8.i386.rpm libcap-devel-1.10-8.i386.rpm ntp-4.1.1-1.i386.rpm
```

- Configure the file */etc/ntp.conf*:

the lines

```
restrict 137.138.16.69 mask 255.255.255.255 nomodify notrap noquery
server 137.138.16.69
restrict 137.138.17.69 mask 255.255.255.255 nomodify notrap noquery
server 137.138.17.69
```

have been added to the file */etc/ntp.conf*

- Edit the file */etc/ntp/step-tickers*:

  the lines

  ```
  137.138.16.69
  137.138.17.69
  ```

  have been added to the file */etc/ntp/step-tickers*

- Edit */etc/sysconfig/ipchains* adding

  ```
  -A input -s 137.138.16.69 -d 0/0 123 -p udp -j ACCEPT
  -A input -s 137.138.17.69 -d 0/0 123 -p udp -j ACCEPT
  ```

  then reload the firewall

  ```
  > /etc/init.d/ipchains restart
  ```

- Activate the ntpd server

  ```
  > ntpdate ip-time-1.cern.ch
  > service ntpd start
  > chkconfig ntpd on
  ```

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

  ```
  > ntpq -p
  ```

As far as the disk partitioning of the machine at the moment of the RH 7.3 installation is concerned, it is recommended that the /tmp partition be given as much space as possible!

4. LCG COMMON CONFIGURATION

4.1. SET-UP HOST CERTIFICATES

CE, SE, PROXY, RB nodes require the host certificate/key files before you start their installation.
Contact your national certification authority (CA) to understand how to obtain a host certificate and a private key if you do not have them already.
Relevant information on all the trusted CAs can be found in

http://marianne.in2p3.fr/datagrid/ca/ca-table-ca.html

which lists the CAs used by edg, from which you should choose a CA close to you.

Once you have obtained a valid certificate, i.e. a file

hostcert.pem

containing the machine host certificate and a file

hostkey.pem

containing the machine private key, make sure to place the two files into the directory

/etc/grid-security

with the following permissions

```
> chmod 400 /etc/grid-security/hostkey.pem

> chmod 644 /etc/grid-security/hostcert.pem
```

It is IMPORTANT that permissions be set as shown, as otherwise certification errors will occur!!!

4.2. CONFIGURE DYNAMIC LINKER RUN-TIME BINDINGS

If not already present, append the following lines to the file */etc/ld.so.conf*

```
/opt/gcc-3.2.2/lib
/opt/globus/lib
/opt/edg/lib
/usr/local/lib
```

- Run the command:

```
> /sbin/ldconfig -v
```

(It produces a huge amount of output)

4.3. CREATE TOP-LEVEL EDG CONFIGURATON FILE

This file is parsed by EDG daemons to locate the EDG root directory and various other global properties.

Create and edit the file */etc/sysconfig/edg* as follows:

```
# Root directory for EDG software. (mandatory)
# Usual value: /opt/edg
EDG_LOCATION=/opt/edg

# Directory for machine-specific files.
# Usual value: $EDG_LOCATION/var

EDG_LOCATION_VAR=/opt/edg/var

# World writable directory for temporary files.  (mandatory)
# Usual value: /tmp
EDG_TMP=/tmp

# The directory containing trusted certificates and CRLs (CERTDIR).
# Usual value: /etc/grid-security/certificates


# Host certificate (X509_USER_CERT) for services which don't have their own.

# Usual value: /etc/grid-security/hostcert.pem

X509_USER_CERT=/etc/grid-security/hostcert.pem


# Host key (X509_USER_KEY) for services which don't have their own.

# Usual value: /etc/grid-security/hostkey.pem

X509_USER_KEY=/etc/grid-security/hostkey.pem


# Location of the grid mapfile (GRIDMAP).
# Usual value: /etc/grid-security/grid-mapfile
```

```
GRIDMAP=/etc/grid-security/grid-mapfile


# Location of the grid map directory for pooled accounts (GRIDMAPDIR).

# Usual value: /etc/grid-security/gridmapdir

GRIDMAPDIR=/etc/grid-security/gridmapdir/
```

4.4. CREATE TOP-LEVEL GLOBUS CONFIGURATON FILE

Create and edit the file */etc/sysconfig/globus* as follows:

```
GLOBUS_LOCATION=/opt/globus
GLOBUS_CONFIG=/etc/globus.conf
GLOBUS_TCP_PORT_RANGE="20000 25000"
```

4.5. CREATE GLOBUS CONFIGURATON FILE

Create and edit the file */etc/globus.conf* as follows below, entering the GIIS node at your site (this is for example a Computing Element):

```
[common]
GLOBUS_LOCATION=/opt/globus
globus_flavor_name=gcc32dbg
x509_user_cert=/etc/grid-security/hostcert.pem
x509_user_key=/etc/grid-security/hostkey.pem
gridmap=/etc/grid-security/grid-mapfile
gridmapdir=/etc/grid-security/gridmapdir/
[mds]
globus_flavor_name=gcc32dbgpthr
user=edginfo

[mds/gris/provider/edg]



[mds/gris/registration/site]
regname=cerntestlcg2
reghn=<GIIS_NODE>
```

```
[gridftp]
log=/var/log/globus-gridftp.log


[gatekeeper]
default_jobmanager=fork
job_manager_path=$GLOBUS_LOCATION/libexec

jobmanagers="fork "

[gatekeeper/fork]
type=fork
job_manager=globus-job-manager
```

4.6. LAUNCH GLOBUS INIZIALIZATION SCRIPT

- Define the *GLOBUS_LOCATION* environment variable (if needed)

```
> export GLOBUS_LOCATION=/opt/globus
```

- Launch setup script

```
> /opt/globus/sbin/globus-initialization.sh
```

This Globus script is meant to be launched on all the LCG nodes, as a general configuration step. Because of this it can yield warning messages as well as some error ones. An example of output to be considered normal in a RB configuration has been produced in Appendix 1.
Since the script produces a huge output it is recommended to re-direct the standard output into a log file:

```
> /opt/globus/sbin/globus-initialization.sh > globus-init.log 2>&1
```

4.7. GSI ACCESS CONTROL LIST (GRID-MAPFILE)

The plain text file

/etc/grid-security/grid-mapfile

maps a GSI (Grid Security Infrastructure) Credential to a local user's login name.

The information in this file is verified and if needed periodically updated by the cron job

```
> /opt/edg/sbin/edg-mkgridmap
```

whose configuration file is

/opt/edg/etc/edg-mkgridmap.conf

If this in turn is not properly configured, the relevant information of the authorised users will not be fetched or updated, resulting in the node not working properly!!!

Instructions on how to edit edg-mkgridmap.conf can be found in [1].

5. REPLICA MANAGER CONFIGURATION

The edg-replica-manager-config RPM installs the *edg-replica-manager-configure* script into the */opt/edg/sbin* directory. This script can be used to automatically generate the default configuration file.
Before running the script, the file

/opt/edg/etc/edg-replica-manager/edg-replica-manager.conf.values

should be manually edited as input file to the script

/opt/edg/sbin/edg-replica-manager-configure

which automatically writes the local configuration parameters into the *edg-replica-manager.conf* file.

So the two step to be done are outline below.

5.1. EDIT THE LOCAL CONFIGURATION FILE

The file

/opt/edg/etc/edg-replica-manager/edg-replica-manager.conf.values

as above said, should be manually edited. Since it is also created by the edg-replica-manager-config rpm it is advisable to rename it (e.g. edg-replica-manager.conf.values_local) so that possible upgrades of the rpm do not overwrite the already existing file.

The file *edg-replica-manager.conf.values* looks like

```
@EDG.LOCATION@|<path>|location of edg middleware
@LOCALDOMAIN@|<domainname>|the local domain
@DEFAULT.SE@|< SE hostname>|the host of the close SE
@DEFAULT.CE@|< CE hostname>|the host of the close CE
@RLS.MODE@|<LrcOnly OR WithRli>|The mode the RLS should be run in. LrcOnly or WithRli
@INFOSERVICE@|<info-provider hostname>|The info provider to use. It can be Stub, MDS or RGMA
@STUBFILE@|<filepath>|The properties file for the static file 'info service'
@MDS.HOST@|<hostname-top-giis>|The host of the MDS info provide
@MDS.PORT@|<portnumber>|The port of the MDS info provider
@ROS.FAILURE@|false|Fail if no ROS is available
@CONF.GCC@|<_gcc3_2_2 OR empty>|The gcc suffix as used on the build box (empty for 2.95, _gcc3_2_2 for 3.2.)
@IGNORE.PREFIX@|true|If true, the lfn and guid prefix will not be passed to the
catalogs
@GRIDFTP.DCAU@|false|Does GridFTP use Data Channel Authentication (DCAU)
@GRIDFTP.STREAMS.SMALL@|1|The default number of stream to use for a small file
@GRIDFTP.STREAMS.BIG@|3|The default number of stream to use for a big file
@GRIDFTP.FILESIZE.THRESHOLD@|100|The Threshold (in MB) above which a file to transfer is considered "big"
```

The parameters have the following meaning:

<**EDG.LOCATION**>**:** location of edg middleware, normally set to /opt/edg, it must match with the previous definition in 4.2

<**LOCALDOMAIN**>**:** your domain name (site-dependent)

<**DEFAULT.SE**>**:** (site-dependent)

<**DEFAULT.CE**>**:** (site-dependent)

<**RLS.MODE**>**:** Replica Catalog type: the implementation of the replica catalog interface. Available choices are: - LrcOnly – a single LRC per VO - WithRli – a full RLS deployed

<**INFOSERVICE**>**:** The implementation of the Information Service Interface. The following choices are available:

<**Stub**>**:** a local configuration file specified by STUBFILE is used instead of an information service

<**RGMA**>**:** R-GMA is contacted for all information service requests

<**MDS**>**:** MDS.HOST is contacted on MDS.PORT for all information service requests (this is the correct choice for lcg2)

<**STUBFILE**>**:** used if INFOSERVICE is set to ”Stub”

<**MDS.HOST**>**:** (site-dependent) value used if INFOSERVICE is set to ”MDS”. It should be a BDII node. If your site owns a BDII use its hostname.
If you have a reference site, ask them for indications or send a message to the ”LCG-ROLLOUT@cclrclsv.RL.AC.U mailing list

<**MDS.PORT**>**:** used if INFOSERVICE is set to ”MDS” (see above)

<**ROS.FAILURE**>**:** Failure if ROS is NOT available. This may be set to ’false’ if you want the replica manager not to complain if ROS is unavailable (this is the correct choice for lcg2)

<**CONF.GCC**>**:** It is the gcc suffix to be concatenated to the name of the JNI library interfacing to gridFTP in the native implementation of the latter one. This dynamic library will be loaded from the classpath. Possible values are:

- empty – for 2.95

- _gcc3_2_2 – for 3.2 (this is the correct choice for lcg2)

IGNORE.PREFIX: This flag, if set to ’true’, allows the use of the new RLS, which ignores the prefixes. It should always be set to ’true’

GRIDFTP.DCAU: flag to indicate whether GridFTP uses Data Channel Authentication (DCAU) or not. In current implementation should be set to ’false’

GRIDFTP.STREAMS.SMALL: Default number of streams to use for the transfer of a ’small’ file

GRIDFTP.STREAMS.BIG: Default number of streams to use for the transfer of a ’big’ file

GRIDFTP.FILESIZE.THRESHOLD: The Threshold (given in MB) above which a file to transfer is considered to be ”big”

Example of layout for file
/opt/edg/etc/edg-replica-manager/edg-replica-manager.conf.values_local

```
@EDG.LOCATION@|/opt/edg|location of edg the directory
@LOCALDOMAIN@|cern.ch|the local domain
@DEFAULT.SE@|adc0033.cern.ch|the host of the close SE
@DEFAULT.CE@|adc0029.cern.ch|the host of the close CE
@INFOSERVICE@|MDS|The info provider to use. It can be Stub, MDS or RGMA
@RLS.MODE@|LrcOnly|The mode the RLS should be run in. LrcOnly or WithRli
@STUBFILE@||The properties file for the static file 'info service'
@MDS.HOST@|adc0031.cern.ch|The host of the MDS info provider
@MDS.PORT@|2170|The port of the MDS info provider
@ROS.FAILURE@|false|Fail if no ROS is available
@CONF.GCC@|_gcc3_2_2|The gcc suffix as used on the build box (empty for 2.95, _gcc3_2_2 for 3.2.)
```

5.2. RUN THE CONFIGURATION SCRIPT

Run (the –verbose flag is optional)

```
> /opt/edg/sbin/edg-replica-manager-configure /opt/edg/etc/edg-replica-manager/edg-replica-manager.conf.values_local [--verbose]
```

The file

/opt/edg/var/etc/edg-replica-manager/edg-replica-manager.conf

is modified accordingly as a result and should be checked to make sure it was edited properly!.

6. POOL ACCOUNTS CREATION FOR RB

Pool accounts are used to map Virtual Organizations to physical user on LCG nodes. Namely they have to be created on WN, CE, SE and RB node.

Configuration steps listed in chapter 6.1. are valid on all the nodes above said.
Specific configuration steps and/or constraints to be applied to a specific node type will be described in a following node-specific section.

User accounts for LCG may be handled in two ways within the PBS system; however in general (e.g. with LSF and Condor) just the first method can be used.

The two methods are outlined below:

1. Method1: creating users (and groups) with their own home directory
2. Method2: creating users (and groups) which share (by mounting) the /home directory of the CE machine

The first method is to be preferred for performance and scalability reasons.

6.1. LCG GROUPS AND USERS IDS

Pool accounts with the following characteristics have to be created on the RB.

Pool group names

The following groups have to be created:

```
alice
atlas
cms
lhcb
dteam
```

The following is a production example

```
> groupadd -g 2688 dteam
> groupadd -g 1307 atlas
> groupadd -g 1470 lhcb
> groupadd -g 1399 cms
> groupadd -g 1395 alice
```

Pool user names

The following users should be created (with also a corresponding home directory in ”/home”):

- Users of group alice:

```
alice001
...
...
alice050
```

- Users of group atlas:

```
atlas001
...
...
atlas050
```

- Users of group cms:

```
cms001
...
...
cms050
```

- Users of group lhcb:

```
lhcb001
...
...
lhcb050
```

- Users of group dteam:

```
dteam001
...
...
dteam050
```

The following is an example for creation of users with their own home directory

```
> useradd -c "mapped user for group ID 2688" -u 18118 -g 2688 dteam001
> ...
> ...
> useradd -c "mapped user for group ID 2688" -u 18774 -g 2688 dteam050

> useradd -c "mapped user for group ID 1307" -u 10761 -g 1307 atlas001
> ...
> ...
> useradd -c "mapped user for group ID 1307" -u 18543 -g 1307 atlas050

> useradd -c "mapped user for group ID 1470" -u 12238 -g 1470 lhcb001
> ...
> ...
> useradd -c "mapped user for group ID 1470" -u 18417 -g 1470 lhcb050

> useradd -c "mapped user for group ID 1399" -u 11410 -g 1399 cms001
> ...
> ...
> useradd -c "mapped user for group ID 1399" -u 18672 -g 1399 cms050

> useradd -c "mapped user for group ID 1395" -u 10417 -g 1395 alice001
> ...
> ...
> useradd -c "mapped user for group ID 1395" -u 18055 -g 1395 alice050
```

6.2. GLOBUS "GRIDMAPDIR" PATCH

- Create the directory */etc/grid-security/gridmapdir*

  ```
  > mkdir /etc/grid-security/gridmapdir
  ```

- For every Pool user name defined in 6.1., create a corresponding empty file with the same name, e.g.

  ```
  > touch /etc/grid-security/gridmapdir/atlas001
  ....
  > touch /etc/grid-security/gridmapdir/dteam050
  ```

  WARNING: Be careful not to create any other files than the pool user ones.
  Namely, the "sgm" users used by the experiment software management system DO NOT need to be associated to files in the */etc/grid-security/gridmapdir* directory.

- Edit the crontab

  ```
  > crontab -e
  ```

adding the line

```
0 5 * * * /opt/edg/sbin/lcg-expiregridmapdir.pl -v 1>>/var/log/lcg-expiregridmapdir.log 2>&1
```

6.3. POOL ACCOUNTS CREATION ON RB

Create groups and users as defined in 6.1..
Notice that the group Ids can be chosen differently, the only constraint being that the same group IDs be used when creating the corresponding users.

Follow what outlined in 6.2., CHANGING however the ownership of the *gridmapdir* directory to the user *edguser*, i.e.

```
> chown root:edguser gridmapdir
```

NOTICE however that the user and group ownerships of the files contained in the *gridmapdir* directory, e.g. *atlas001*, MUST be root, despite the fact that the *gridmapdir*'s group ID is *edguser*!

7. GRIS CONFIGURATION

This section deals with the configuration of the Gris which publishes information about the Resource Broker, rendering it visible within the Grid.

7.1. INFO-PROVIDER.CONF

Create the directory */opt/edg/var/info/*

```
> mkdir /opt/edg/var/info
```

Create and edit the file */opt/edg/var/info/info-provider.conf* as follows

```
EDG_LOCATION=/opt/edg
GRID_INFO_USER=
REP_MET_PRESENT=no
REP_LOC_PRESENT=no
REP_OPT_PRESENT=no
SE_PRESENT=yes
```

7.2. LCG-INFO-WRAPPER

Create and edit the file */opt/lcg/libexec/lcg-info-wrapper* as follows

```
#!/bin/sh
/opt/lcg/libexec/lcg-info-generic /opt/lcg/var/lcg-info-generic.conf
```

after which the command

```
> chmod a+wxr /opt/lcg/libexec/lcg-info-wrapper
```

should be launched.

7.3. INFORMATION PROVIDER

The information provider needs to be configured. In order to do so create and edit

/opt/lcg/var/lcg-info-generic.conf as follows:

```
ldif_file=/opt/lcg/var/lcg-info-static.ldif
generic_script=/opt/lcg/libexec/lcg-info-generic
wrapper_script=/opt/lcg/libexec/lcg-info-wrapper
template=/opt/lcg/etc/GlueService.template

dn: GlueServiceURI=gram://<NODE_NAME>:7772,Mds-vo-name=local,o=grid
GlueServiceAccessPointURL: gram://<NODE_NAME>:7772
GlueServiceType: ResourceBroker
GlueServicePrimaryOwnerName: LCG
GlueServicePrimaryOwnerContact: grid-testbed-managers@host.invalid
```

```
GlueServiceHostingOrganization: <HOSTING_ORG>
GlueServiceMajorVersion: 1
GlueServicePatchVersion: 1
GlueCEAccessControlBaseRule: <VIRTUAL_ORGANIZATION_1>
GlueCEAccessControlBaseRule: <VIRTUAL_ORGANIZATION_N>
GlueServiceInformationServiceURL: ldap://<NODE_NAME>:2135/mds-vo-name=local,o=grid
GlueServiceStatus: running
```

where <NODE_NAME> is the Resource Broker Node full Name; <HOSTING_ORG> is the Hosting Organization Name; <VIRTUAL_ORGANIZATION_1> to <VIRTUAL_ORGANIZATION_N> is the set of Virtual Organizations supported (so for every VO a corresponding line should be entered).
WARNING: PLEASE MAKE SURE THAT THE END OF LINE CHARACTER IS PRESENT AT THE END OF THE LAST LINE. IF NOT THE INFORMATION WILL BE BADLY PROCESSED AND IN THE END DISREGARDED!!!

A production example follows:

```
ldif_file=/opt/lcg/var/lcg-info-static.ldif
generic_script=/opt/lcg/libexec/lcg-info-generic
wrapper_script=/opt/lcg/libexec/lcg-info-wrapper
template=/opt/lcg/etc/GlueService.template

dn: GlueServiceURI=gram://lxshare0410.cern.ch:7772,Mds-vo-name=local,o=grid
GlueServiceAccessPointURL: gram://lxshare0410.cern.ch:7772
GlueServiceType: ResourceBroker
GlueServicePrimaryOwnerName: LCG
GlueServicePrimaryOwnerContact: mailto:hep-project-grid-cern-testbed-managers@cern.ch
GlueServiceHostingOrganization: CERN-TEST-LCG2
GlueServiceMajorVersion: 1
GlueServicePatchVersion: 1
GlueServiceAccessControlRule: alice
GlueServiceAccessControlRule: atlas
GlueServiceAccessControlRule: cms
GlueServiceAccessControlRule: lhcb
GlueServiceAccessControlRule: dteam
GlueServiceAccessControlRule: sixt
GlueServiceInformationServiceURL: ldap://lxshare0410.cern.ch:2135/mds-vo-name=local,o=grid
GlueServiceStatus: running
```

The script *lcg-info-generic-config* should then be launched in order to create the right configuration:

```
> /opt/lcg/sbin/lcg-info-generic-config  lcg-info-generic.conf
```

Notice that the file */opt/lcg/var/lcg-info-static.ldif* will be created as a result. This file SHOULD NOT BE EDITED MANUALLY.

7.4. INFO PROVIDER INITIALIZATION SCRIPT

This step is required so as to define which Information Provider should be used by the GRIS.
In order to do so launch the setup script

```
> /opt/edg/sbin/edg-info-provider-setup
```

and after that the globus initialization script

```
> /opt/globus/sbin/globus-initialization.sh > globus-init.log 2>&1
```

7.5. START THE INFORMATION SERVICE

```
> /sbin/chkconfig globus-mds on
> service globus-mds start
```

7.6. GRIS AND GIIS TEST

Test the GRIS and GIIS as follows:

```
> ldapsearch -h <RB-Hostname> -p 2135 -b "mds-vo-name=local,o=grid" -x
> ldapsearch -h <GIIS Node> -p 2135 -b "mds-vo-name=<your-registration-name>,o=grid" -x
```

NOTE CAREFULLY:
In order for the Resource Broker to be *visible* the BDII has to contain the information relative to the GIIS node (e.g. a computing element) which also requires that the RB node be entered in the file */etc/globus* of the GIIS node itself.

Also notice that should there be a sizeable time difference between the RB and BDII, the latter one will not update the RB information, which is therefore lost.

8. RESOURCE BROKER SPECIFIC CONFIGURATION

8.1.

Procede through the following steps as shown, to insure the correct ownership/permissions are set.

WARNING: Should any of the following steps be overlooked, the corresponding servers might not run properly and make any job submission fail as consequence!!!

Work Load Manager

In /var

```
> mkdir /var/edgwl
```

In /var/edgwl

```
> mkdir SandboxDir
> mkdir -p networkserver/log
> chmod g+w SandboxDir
> mkdir -p workload_manager/log
> mkdir -p jobcontrol/cond
```

In /var

```
> chown -R edguser:edguser edgwl
```

This last instruction is fundamental in setting the correct ownership!!!

/opt/edg/

Create the directory

/opt/edg/var/etc/profile.d

with

```
> mkdir -p /opt/edg/var/etc/profile.d
```

Copy from the directory

/opt/edg/etc/profile.d

the following files:

- *edg-wl.csh*

- *edg-wl.sh*

- *edg-wl-config.sh*

into the directory

/opt/edg/var/etc/profile.d

```
> cd /opt/edg/etc/profile.d
> cp  edg-wl.csh edg-wl.sh edg-wl-config.sh /opt/edg/var/etc/profile.d/
```

In /opt/edg/var

```
> mkdir run
> mkdir log
```

Condor

```
> mkdir -p /opt/condor/var/condor/log/GridLogs
```

In /opt/condor/var/condor/

```
> mkdir spool
```

Now it is IMPORTANT to do:

In /opt/condor/var/

```
> chown -R edguser:edguser condor
```

8.2. WORK LOAD CONFIG FILE

Create and edit the file */opt/edg/etc/edg_wl.conf* accordingly. The template of the file follows below. The file is standard and typically only your BDII node needs to be entered, i.e. <YOUR_BDII_NODE> in the *II_Contact* variable.

```
[
   Common = [
      DGUser = "${EDG_WL_USER}";
      HostProxyFile = "${EDG_WL_TMP}/networkserver/ns.proxy";
      UseCacheInsteadOfGris = true;
   ];
   JobController = [
      CondorSubmit = "${CONDORG_INSTALL_PATH}/bin/condor_submit";
```

```
    CondorRemove = "${CONDORG_INSTALL_PATH}/bin/condor_rm";
    CondorQuery  = "${CONDORG_INSTALL_PATH}/bin/condor_q";
    CondorSubmitDag = "${CONDORG_INSTALL_PATH}/bin/condor_submit_dag";
    CondorRelease = "${CONDORG_INSTALL_PATH}/bin/condor_release";


    SubmitFileDir = "${EDG_WL_TMP}/jobcontrol/submit";
    OutputFileDir = "${EDG_WL_TMP}/jobcontrol/condorio";
    Input = "${EDG_WL_TMP}/jobcontrol/queue.fl";
    LockFile = "${EDG_WL_TMP}/jobcontrol/lock";
    LogFile = "${EDG_WL_TMP}/jobcontrol/log/events.log";
    ExternalLogFile = "${EDG_WL_TMP}/jobcontrol/log/external.log";

    LogLevel = 5;

    ContainerRefreshThreshold = 1000;
  ];
  LogMonitor = [
    JobsPerCondorLog = 1000;
    LockFile = "${EDG_WL_TMP}/logmonitor/lock";
    LogFile = "${EDG_WL_TMP}/logmonitor/log/events.log";
    LogLevel = 5;
    ExternalLogFile = "${EDG_WL_TMP}/logmonitor/log/external.log";
    MainLoopDuration = 5;

    CondorLogDir = "${EDG_WL_TMP}/logmonitor/CondorG.log";
    CondorLogRecycleDir = "${EDG_WL_TMP}/logmonitor/CondorG.log/recycle";

    MonitorInternalDir = "${EDG_WL_TMP}/logmonitor/internal";
    IdRepositoryName = "irepository.dat";

    AbortedJobsTimeout = 600;
  ];
  NetworkServer = [
    II_Port  = 2135;
    Gris_Port = 2135;
    II_Timeout = 30;
    Gris_Timeout = 20;
    II_DN    = "mds-vo-name=local, o=grid";
    Gris_DN   = "mds-vo-name=local, o=grid";
    II_Contact = "<YOUR_BDII_NODE>";

    ListeningPort = 7772;
```

```
    MasterThreads = 8;
    DispatcherThreads = 10;
    SandboxStagingPath = "${EDG_WL_TMP}/SandboxDir";

    LogFile = "${EDG_WL_TMP}/networkserver/log/events.log";
    LogLevel = 5;
    BacklogSize = 16;
    EnableQuotaManagement = false;
    MaxInputSandboxSize = 10000000;
    EnableDynamicQuotaAdjustment = false;
    QuotaAdjustmentAmount = 10000;
    QuotaInsensibleDiskPortion = 2.0;
  ];
  WorkloadManager = [
    PipeDepth = 1;
    NumberOfWorkerThreads = 1;
    DispatcherType = "filelist";
    Input = "${EDG_WL_TMP}/workload_manager/input.fl";
    LogLevel = 6;
    LogFile  = "${EDG_WL_TMP}/workload_manager/log/events.log";

    MaxRetryCount = 10;
  ];
]
```

8.3. CONDOR CONFIGURATION

The file

/opt/condor/etc/condor.conf

should be created and edited accordingly, as shown below:

```
#
# PART 1
#

CONDOR_HOST = $(FULL_HOSTNAME)

RELEASE_DIR = /opt/condor
LOCAL_DIR = /opt/condor/var/condor

CONDOR_ADMIN = \"hep-project-grid-cern-testbed-managers@cern.ch\"
```

```
MAIL = /bin/mail

UID_DOMAIN = $(FULL_HOSTNAME)
FILESYSTEM_DOMAIN = $(FULL_HOSTNAME)

#
# PART 2
#

DAEMON_LIST = MASTER, SCHEDD

FLOCK_NEGOTIATOR_HOSTS = $(FLOCK_TO)
FLOCK_COLLECTOR_HOSTS = $(FLOCK_TO)

HOSTALLOW_ADMINISTRATOR = $(CONDOR_HOST)
HOSTALLOW_OWNER = $(FULL_HOSTNAME), $(HOSTALLOW_ADMINISTRATOR)
HOSTALLOW_READ = *
HOSTALLOW_WRITE = $(FULL_HOSTNAME), $(GLIDEIN_SITES)
HOSTALLOW_NEGOTIATOR = $(NEGOTIATOR_HOST)
HOSTALLOW_NEGOTIATOR_SCHEDD = $(NEGOTIATOR_HOST), $(FLOCK_NEGOTIATOR_HOSTS)
HOSTALLOW_WRITE_COLLECTOR = $(HOSTALLOW_WRITE), $(FLOCK_FROM)
HOSTALLOW_WRITE_STARTD = $(HOSTALLOW_WRITE), $(FLOCK_FROM)
HOSTALLOW_READ_COLLECTOR = $(HOSTALLOW_READ), $(FLOCK_FROM)
HOSTALLOW_READ_STARTD = $(HOSTALLOW_READ), $(FLOCK_FROM)

LOCK = $(LOG)

MAX_SCHEDD_LOG = 64000000
SCHEDD_DEBUG = D_COMMAND
MAX_GRIDMANAGER_LOG = 64000000
GRIDMANAGER_DEBUG = D_COMMAND
MAX_COLLECTOR_LOG = 64000000
COLLECTOR_DEBUG = D_COMMAND
MAX_NEGOTIATOR_LOG = 64000000
NEGOTIATOR_DEBUG = D_MATCH
MAX_NEGOTIATOR_MATCH_LOG = 64000000
MAX_SHADOW_LOG = 64000000

#
# PART 3
#

MINUTE = 60
HOUR = (60 * $(MINUTE))
```

```
StateTimer = (CurrentTime - EnteredCurrentState)
ActivityTimer = (CurrentTime - EnteredCurrentActivity)
ActivationTimer = (CurrentTime - JobStart)
LastCkpt = (CurrentTime - LastPeriodicCheckpoint)

STANDARD = 1
PVM = 4
VANILLA = 5
IsPVM = (JobUniverse == $(PVM))
IsVANILLA = (JobUniverse == $(VANILLA))
IsSTANDARD = (JobUniverse == $(STANDARD))
NonCondorLoadAvg = (LoadAvg - CondorLoadAvg)
BackgroundLoad = 0.3
HighLoad = 0.5
StartIdleTime = 15 * $(MINUTE)
ContinueIdleTime = 5 * $(MINUTE)
MaxSuspendTime = 10 * $(MINUTE)
MaxVacateTime = 10 * $(MINUTE)
KeyboardBusy = (KeyboardIdle < $(MINUTE))
ConsoleBusy = (ConsoleIdle < $(MINUTE))
CPU_Idle = ($(NonCondorLoadAvg) <= $(BackgroundLoad))
CPU_Busy = ($(NonCondorLoadAvg) >= $(HighLoad))
BigJob = (ImageSize >= (50 * 1024))
MediumJob = (ImageSize >= (15 * 1024) && ImageSize < (50 * 1024))

SmallJob = (ImageSize < (15 * 1024))
JustCPU = ($(CPU_Busy) && ($(KeyboardBusy) == False))
MachineBusy = ($(CPU_Busy) || $(KeyboardBusy))

#
# PART 4
#

DISABLE_AUTH_NEGOTIATION = true

LOG = $(LOCAL_DIR)/log
SPOOL = $(LOCAL_DIR)/spool
EXECUTE = $(LOCAL_DIR)/execute
BIN = $(RELEASE_DIR)/bin
LIB = $(RELEASE_DIR)/lib
SBIN = $(RELEASE_DIR)/sbin
HISTORY = $(SPOOL)/history

MASTER_LOG = $(LOG)/MasterLog
```

```
SCHEDD_LOG = $(LOG)/SchedLog
GRIDMANAGER_LOG = $(LOG)/GridLogs/GridmanagerLog.$(USERNAME)
SHADOW_LOG = $(LOG)/ShadowLog
COLLECTOR_LOG = $(LOG)/CollectorLog
NEGOTIATOR_LOG = $(LOG)/NegotiatorLog
NEGOTIATOR_MATCH_LOG = $(LOG)/MatchLog

SHADOW_LOCK = $(LOCK)/ShadowLock

RESERVED_DISK = 5

MASTER = $(SBIN)/condor_master
SCHEDD = $(SBIN)/condor_schedd
NEGOTIATOR = $(SBIN)/condor_negotiator
COLLECTOR = $(SBIN)/condor_collector

MASTER_ADDRESS_FILE = $(LOG)/.master_address

PREEN = $(SBIN)/condor_preen

PREEN_ARGS = -m -r

SHADOW = $(SBIN)/condor_shadow
SHADOW_PVM = $(SBIN)/condor_shadow.pvm
GRIDMANAGER = $(SBIN)/condor_gridmanager
GAHP = $(SBIN)/gahp_server

SCHEDD_ADDRESS_FILE = $(LOG)/.schedd_address

SHADOW_SIZE_ESTIMATE = 1800

SHADOW_RENICE_INCREMENT = 10

QUEUE_SUPER_USERS = root, condor

PVMD = $(SBIN)/condor_pvmd

PVMGS = $(SBIN)/condor_pvmgs

DEFAULT_UNIVERSE = globus
CRED_MIN_TIME_LEFT = 120

VALID_SPOOL_FILES = job_queue.log, job_queue.log.tmp, history, Accountant.log, Ac
```

```
INVALID_LOG_FILES = core

GLIDEIN_SERVER_NAME = gridftp.cs.wisc.edu
GLIDEIN_SERVER_DIR = /p/condor/public/binaries/gli dein

AUTHENTICATION_METHODS = CLAIMTOBE
ENABLE_GRID_MONITOR = TRUE
GRID_MONITOR = $(SBIN)/grid_monitor.sh
GRIDMANAGER_MINIMUM_PROXY_TIME = 600
GRIDMANAGER_MAX_SUBMITTED_JOBS_PER_RESOURCE = 32000
GRIDMANAGER_MAX_PENDING_SUBMITS_PER_RESOURCE = 5
GRIDMANAGER_MAX_PENDING_REQUESTS = 1000
GRIDMANAGER_GAHP_CALL_TIMEOUT = 900
GRID_MONITOR_HEARTBEAT_TIMEOUT = 300
GRID_MONITOR_RETRY_DURATION = 31536000
```

8.4. CONFIGURE THE MYSQL DATABASE FOR LBSERVER

```
> /sbin/chkconfig mysql on
> /etc/rc.d/init.d/mysql start
> mysqladmin password <YOUR_PASSWORD>
> mysqladmin -p create lbserver20
```

This command will require the password to be entered, e.g. ”datagrid”, as defined above!

Enter password: *****

```
> mysql -p lbserver20 < /opt/edg/etc/server.sql
```

This command will require the password to be entered, e.g. ”datagrid” as defined above!

Enter password: *****

```
> mysql -p lbserver20
```

This command will require the password to be entered, e.g. ”datagrid” as defined above!

Enter password: *****

After which mysql session will be started! Within the mysql session do the following:

```
> mysql> grant all on lbserver20.* to lbserver@localhost \g
> mysql> \qj
```

9. SERVERS START UP

9.1. FTPD

```
> /sbin/chkconfig edg-wl-ftpd on
> /etc/rc.d/init.d/edg-wl-ftpd start
```

after which the following command should be always issued:

```
> /etc/rc.d/init.d/edg-wl-ftpd status
```

to check that the server is properly running!!!

9.2. L&B SERVER

```
> /sbin/chkconfig edg-wl-lbserver on
> /etc/rc.d/init.d/edg-wl-lbserver start
```

After which the following command should be always issued:

```
> /etc/rc.d/init.d/edg-wl-lbserver status
```

to check that the server is properly running!!!

9.3. NETWORK SERVER

```
> /sbin/chkconfig edg-wl-ns on
> /etc/rc.d/init.d/edg-wl-ns start
```

After which the following command should be always issued:

```
> /etc/rc.d/init.d/edg-wl-ns status
```

to check that the server is properly running!!!

9.4. WORKLOAD MANAGER

```
> /sbin/chkconfig edg-wl-wm on
> /etc/rc.d/init.d/edg-wl-wm start
```

After which the following command should be always issued:

```
> /etc/rc.d/init.d/edg-wl-wm status
```

to check that the server is properly running!!!

9.5. JOBCONTROLLER DAEMON

```
> /sbin/chkconfig edg-wl-jc on
> /etc/rc.d/init.d/edg-wl-jc start
```

After which the following command should be always issued:

```
> /etc/rc.d/init.d/edg-wl-jc status
```

to check that the server is properly running!!!

Notice that the Job controller relies on the file

/opt/condor/etc/condor.conf

being created and properly edited (see 7.2)!!!

9.6. LOGMONITOR DAEMON

```
> /sbin/chkconfig edg-wl-lm on
> /etc/rc.d/init.d/edg-wl-lm start
```

After which the following command should be always issued:

```
> /etc/rc.d/init.d/edg-wl-lm status
```

to check that the server is properly running!!!

9.7. LOCALLOGGER SERVER

```
> /sbin/chkconfig edg-wl-locallogger on
> /etc/rc.d/init.d/edg-wl-locallogger start
```

After which the following command should be always issued:

```
> /etc/rc.d/init.d/edg-wl-locallogger status
```

to check that the server is properly running!!!

10. FABRIC MONITORING BY GRIDICE

The CE, SE, and RB LCG nodes produce data for the GridICE monitoring system. The data are then sent to a collector server node which will then be queried by the LCG central GridICE monitoring service.
If you are running agents you should also run a GridICE collector server to collect information from your agents.
In the default LCG-2 configuration the SE node runs the GridICE collector node. In Section 10.1. you can find the configuration of agents while section 10.2. deals with the configuration details for the GridICE collector server.
If you are doing a defaul LCG-2 configuration you should apply section 10.2. just to the SE.

Before going forward with configuration, please assure the following RPMs to be installed (they should have been distributed with the node RPMs).

edg-fabricMonitoring edt_sensor

10.1. SET-UP OF GRIDICE AGENTS

In order to enable GridICE agent on a LCG node:

- Create and configure the file /opt/edg/var/etc/edg-fmon-agent.conf as follows:

```
# Sensor file for edg-fmonagent
MSA


      Transport


            UDP
                  Server  <GridICE-collector-hostname>
                  Port 12409
                  FilterMetrics KeepOnly
                        11001
                        11011
                        11021
                        11101
                        11202
                        11013
                        11022
                        11031
                        11201
                        10100
                        10101
                        10102
                        10103
                        10104
                        10105


      Sensors

            edtproc
                  CommandLine /opt/edt/monitoring/bin/GLUEsensorLinuxProc
                  MetricClasses
                        edt.uptime
                        edt.cpu
                        edt.memory
                        edt.disk
                        edt.network
```

```
                edt.ctxint
                edt.swap
                edt.processes
                edt.sockets
                edt.cpuinfo
                edt.os
                edt.alive
                edt.regfiles

        sensor1
                CommandLine $(EDG_LOCATION)/libexec/edg-fmon-sensor-systemCheck
                MetricClasses
                        executeScript


Metrics
        11001
                MetricClass edt.uptime
        11011
                MetricClass edt.cpu
        11021
                MetricClass edt.memory
        11101
                MetricClass edt.disk
        11202
                MetricClass edt.network
                Parameters
                        interface       eth0
        11013
                MetricClass edt.ctxint
        11022
                MetricClass edt.swap
        11031
                MetricClass edt.processes
        11201
                MetricClass edt.sockets
        10100
                MetricClass edt.cpuinfo
        10101
                MetricClass edt.os
        10102
                MetricClass edt.alive
        10103
                MetricClass edt.regfiles
        10104
                MetricClass executeScript
                Parameters
                        command       /opt/edt/monitoring/bin/CheckDaemon.pl --cfg /opt/edt/monitoring/etc/gridice-role.cfg
        10105
                MetricClass executeScript
                Parameters
                        command       /opt/edt/monitoring/bin/PoolDir.pl


Samples
        verylowfreq
                Timing 3600 0
                Metrics
                        10100
                        10101
        lowfreq
                Timing 1800 0
                Metrics
                        11001
        proc0
                Timing 30 0
                Metrics
                        10102
        proc1
                Timing 60 0
                Metrics
                        11011
                        11021
                        11101
                        11202
                        11013
                        11022
                        11031
                        11201
        proc2
                Timing 300 0
                Metrics
                        10103
```

```
                10105
proc3
        Timing 120 0
        Metrics
                10104
```

WARNING: be very careful not to use <SPACE> characters to indent lines in this configuration file. Use <TAB> (or nothing) instead. The edg-fmon-agent does not allow spaces at the beginning of a row in the configuration file.

The parameter <GridICE-collector-hostname> is the complete hostname of the node that runs the GridICE collector server and publishes the data on the information system. The collector node will have to run a plain GRIS for this.
The information is sent to the collector node via UDP (port 12409)

If you already have a GridICE collector node at your site use that, else set up one as described in section 10.1..

- start the GridICE agent

```
> chkconfig edg-fmon-agent on
> service edg-fmon-agent start
```

10.2. SET-UP OF A GRIDICE COLLECTOR SERVER

The Fabric Monitoring server can be installed on every node running a plain GRIS (e.g. Computing Element, Storage Element)

Fabric Monitoring server configuration

- Create (if needed) the directory

/opt/edg/var/etc

```
> mkdir /opt/edg/var/etc
```

- Create and edit as follows the file

/opt/edg/var/etc/edg-fmon-server.conf

```
SERVER_SPOOLDIR=/var/fmonServer
SERVER_PORT=12409
MR_SOAP_PORT=12411
```

- Start the Fabric Monitoring server

```
> chkconfig edg-fmon-server on
> service edg-fmon-server start
```

Start the GridICE Information service

```
> chkconfig gridice-mds on
> service gridice-mds start
```

Configure extra info in the standard GRIS

- Edit the file /etc/globus.conf and insert the line

```
[mds/gris/provider/gridice]
```

between the line

```
[mds/gris/provider/edg]
```

and the line

```
[mds/gris/registration/site]
```

- Re-start the standard MDS

```
> service globus-mds restart
```

Set up Cron Table

The repository is cleaned daily : data is kept for last 5 days only

- Add the service to the crontab

```
> crontab -e
```

This will allow the editing of the crontable with the default editor, after which the following line has to be added:

```
43 1 * * * /opt/edg/sbin/edg-fmon-cleanspool &> /dev/null
```

11. EXPERIMENT SOFTWARE MANAGER CONFIGURATION

For each VO group defined in [**?**] a special account to handle experiment software has to be created.

The list of SGM user to be created follows:

---

alicesgm
atlassgm
cmssgm
lhcbsgm
dteamsgm

The following lines are an example for user creation

```
useradd -c "mapped user for group ID 2688" -u 18946 -g 2688 dteamsgm

useradd -c "mapped user for group ID 1307" -u 18943 -g 1307 atlassgm

useradd -c "mapped user for group ID 1470" -u 18945 -g 1470 lhcbsgm
useradd -c "mapped user for group ID 1399" -u 18944 -g 1399 cmssgm
useradd -c "mapped user for group ID 1395" -u 18941 -g 1395 alicesgm
```

12. CRON TABLE CONFIGURATION

The following production example of crontab should be taken as reference:

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin

26 1,7,13,19 * * * /opt/edg/sbin/edg-mkgridmap --output=/etc/grid-security/grid

26 1,7,13,19 * * * /opt/edg/etc/cron/edg-fetch-crl-cron
26 2,8,14,20 * * * /sbin/service edg-wl-locallogger proxy
26 2,8,14,20 * * * /sbin/service edg-wl-lbserver proxy
26 2,8,14,20 * * * /sbin/service edg-wl-proxyrenewal proxy
26 2,8,14,20 * * * /sbin/service edg-wl-ns proxy
26 */1 * * 1-6 /opt/edg/libexec/edg-wl-purgestorage.sh hourly
26 */4 * * 0 /opt/edg/libexec/edg-wl-purgestorage.sh weekly
*/5 * * * * /opt/edg/libexec/edg-wl-check-daemons.sh
```

All those lines missing in the crontab of the machine under configuration should be added with the command

```
> crontab -e
```

via the crontab standard editor.

13. UPGRADE FROM THE PREVIOUS VERSION

This procedure is meant to help site administrators upgrade the node without starting an installation ”from scratch”, whenever possible.
The aim is to define those ”delta” steps dealing both with installation and configuration phases. It is worthy of reminding that here ”delta” refers to two contiguous software tags, the latter of which is the current one. Therefore, the procedure is not applicable to software versions more than two tags old.
WARNING: As a requirement for a successful upgrade, all the active servers which might be affected by the upgrade should be stopped.
The servers should be restarted once the installation and configuration described in the following sub paragraphs have been successfully done.

13.1. SOFTWARE INSTALLATION

The list ”RB_manual_install_rpm”, which has been described in 2., contains the latest version of the RB-related rpms to be installed. If the *rpm -i* command were launched as described in that section, a large number of conflicts would be issued because most of the packages would be reported to be already installed, some of which with the same version number, some others with an older one.
The easiest way to handle this situation is to download those rpms which should update the ones already installed on the node as well as any new ones.
The list of these rpms is contained in the file

RB_upgrade_rpm

whereas the list with the complete command lines for the download is contained in the file

RB_upgrade_wget

WARNING: This script will create a directory tree where the rpms are downloaded and needs to be launched once only.

After downloading the rpms, the first thing to do is uninstall any obsolete packages; this is particularly likely to be necessary with CA-related rpms. For coherence and security reasons, therefore, you should download from CVS the list of obsoleted rpms provided with the current release, in the file

RB_remove_rpm

The rpm command to remove the listed rpms is

> rpm -e ‘cat RB_remove_rpm‘

Next, within the directory *RB_UPGRADE*, the command

```
> rpm -U --oldpackage "*.rpm"
```

should be launched.

Finally a further list of rpms is provided, namely

RB_upgrade_forced_rpm

containing those new rpms whose dependecies are not dealt with, therefore requiring the installation to be ”forced”.
The same ”wget” file

RB_upgrade_wget

takes also care to download the rpms of this final list in the directory

RB_UPGRADE_FORCED

The rpms listed in this directory should be installed with the command

```
> rpm -Uvh --force --nodeps "*.rpm"
```

13.2. UPGRADE CONFIGURATION

- Referring to 4.2. the file */etc/ld.so.conf* has changed! - Referring to 4.5. the file */etc/globus.conf* has changed! - Referring to 5.1. the file *edg-replica-manager.conf.values* has changed! - Perform actions specified in 7. - Referring to 8.1.2. there are changes in the number of the copied scripts

14. APPENDIX 1: STANDARD OUTPUT OF GLOBUS-INITIALIZATION.SH

```
creating globus-sh-tools-vars.sh
creating globus-script-initializer
creating Globus::Core::Paths
checking globus-hostname
/opt/globus/libexec/globus-libc-hostname: error while loading shared libraries:
libgcc_s.so.1: cannot open shared object file: No such file or directory
Done

Creating...
        /opt/globus/etc/grid-info.conf
/opt/globus/libexec/globus-libc-hostname: error while loading shared libraries:
libgcc_s.so.1: cannot open shared object file: No such file or directory
Done
/opt/globus/libexec/globus-libc-hostname: error while loading shared libraries:
libgcc_s.so.1: cannot open shared object file: No such file or directory

Creating...
       /opt/globus/sbin/SXXgris
       /opt/globus/libexec/grid-info-script-initializer
       /opt/globus/libexec/grid-info-mds-core
       /opt/globus/libexec/grid-info-common
       /opt/globus/libexec/grid-info-cpu*
       /opt/globus/libexec/grid-info-fs*
       /opt/globus/libexec/grid-info-mem*
       /opt/globus/libexec/grid-info-net*
       /opt/globus/libexec/grid-info-platform*
       /opt/globus/libexec/grid-info-os*
       /opt/globus/etc/grid-info-resource-ldif.conf
       /opt/globus/etc/grid-info-resource-register.conf
       /opt/globus/etc/grid-info-resource.schema
       /opt/globus/etc/grid.gridftpperf.schema
       /opt/globus/etc/gridftp-resource.conf
       /opt/globus/etc/gridftp-perf-info
       /opt/globus/etc/grid-info-slapd.conf
/opt/globus/libexec/globus-libc-hostname: error while loading shared libraries:
libgcc_s.so.1: cannot open shared object file: No such file or directory
       /opt/globus/etc/grid-info-site-giis.conf
       /opt/globus/etc/grid-info-site-policy.conf
       /opt/globus/etc/grid-info-server-env.conf
       /opt/globus/etc/grid-info-deployment-comments.conf
Done
Creating gatekeeper configuration file...
Done
Creating grid services directory...
Done
Creating state file directory.
Done.
Reading gatekeeper configuration file...
Warning: Host cert file: /etc/grid-security/hostcert.pem not found.  Re-run
        setup-globus-gram-job-manager after installing host cert file.
Determining system information...
/opt/globus/libexec/globus-libc-hostname: error while loading shared libraries:
libgcc_s.so.1: cannot open shared object file: No such file or directory
Creating job manager configuration file...
Done
Setting up fork gram reporter in MDS
----------------------------------------
Done

Setting up pbs gram reporter in MDS
----------------------------------------
configure: error: Cannot locate qstat
loading cache /dev/null
checking for qstat... no
Error locating pbs commands, aborting!
Setting up condor gram reporter in MDS
----------------------------------------
configure: error: Cannot locate condor_q
loading cache /dev/null
checking for condor_q... no
Error locating condor commands, aborting!
Setting up lsf gram reporter in MDS
----------------------------------------
configure: error: Cannot locate lsload
loading cache /dev/null
checking for lsload... no
Error locating LSF commands, aborting!
configure: warning: Cannot locate mpirun
loading cache ./config.cache
checking for mpirun... no
updating cache ./config.cache
```

```
creating ./config.status
creating fork.pm
configure: warning: Cannot locate mpirun
configure: error: Cannot locate qdel
loading cache /dev/null
checking for mpirun... no
checking for qdel... no
Error locating PBS commands, aborting!
configure: error: Cannot locate condor_submit
loading cache /dev/null
checking for condor_submit... no
Error locating condor commands, aborting!
configure: warning: Cannot locate mpirun
configure: error: Cannot locate bsub
loading cache /dev/null
checking for mpirun... no
checking for bsub... no
Error locating LSF commands, aborting!
loading cache ./config.cache
creating ./config.status
creating grid-cert-request-config
creating grid-security-config
```

15. APPENDIX 2: NON GENERAL LCG USERS AND GROUPS IDS

–Groups:

edguser:x:995:
sshd:x:74:
edginfo:x:999:

Notice that reported above are the entries appearing in the file */etc/group*

–Users:

User *edguser* :

The user *edguser* belongs to the group *edguser* as well as *atlas*, *alice*, *cms*, *lhcb* and *dteam*

The user *edguser* has also a home directory in */home*.

Notice that many of the servers which run within the RB node are owned by the user *edguser* and so its presence is fundamental!!!

User *sshd*:

The user *sshd* belongs to the group *sshd*

The user *sshd* has NOT a home directory in */home*. Rather, its home directory is */var/empty/sshd* instead. Also notice that the default shell for this user should be */bin/false*, i.e. no login is allowed under the *sshd* user.

The user sshd is normally created by newer versions of opelssl and so is not created in a standard red hat 7.3.

User *edginfo*:

The user *edginfo* belongs to the group *edginfo*

The user *edginfo* has NOT a home directory in */home*.
Rather, its home directory is */opt/edg/var* instead.

Notice also that edginfo is not strictly required for the correct functioning of the node.

15.1. PRODUCTION EXAMPLE

Reported below are the standard steps required to create all the relevant non general groups and users.

Groups creation

```
>  groupadd -g 2688 dteam
>  groupadd -g 1307 atlas
>  groupadd -g 1470 lhcb
>  groupadd -g 1399 cms
>  groupadd -g 1395 alice

>  groupadd -g 995 edguser
>  groupadd -g 74 sshd
>  groupadd -g 999 edginfo
```

Users creation

```
>  useradd  -M -u 74 -g 74 -s /bin/false -d /var/empty/sshd sshd
>  useradd  -u 995 -g 995 -s /bin/bash edguser
>  useradd  -M -u 999 -g 999 -s /bin/bash -d /opt/edg/var edginfo
```

Group members extension

```
>  gpasswd -a edguser dteam
>  gpasswd -a edguser atlas
>  gpasswd -a edguser lhcb
>  gpasswd -a edguser cms
>  gpasswd -a edguser alice
```

CHANGE HISTORY

Table 1: Change History

version	*date*	*description*
v1.0	29/Jan/04	First Release
v1.1	29/Jan/04	Minor Corrections in 7) **??**
v1.2	30/Jan/04	Restructure of paragraph **??** and insertion in ”gridmapdir” Dir of LCG users files creation!
v1.3	30/Jan/04	Minor Corrections.
v1.4	30/Jan/04	Reshaping of the document and Minor Corrections.
v1.5	30/Jan/04	Minor Corrections.
v1.6	30/Jan/04	Minor Corrections in Appendix **??**.
v1.7	02/Feb/04	Ref. [1] specified. Minor corrections.
v1.8	02/Feb/04	Minor corrections.
v1.9	03/Feb/04	Appendix **??**: edguser and edginfo creation corrected.
v2.0	04/Feb/04	Paragraph **??**: Few corrections of the mkdir instructions.
v2.1	06/Feb/04	Minor corrections. Paragraph **??**: Few mistakes corrected! Section **??** added.
v2.2	09/Feb/04	Minor corrections.
v2.3	15/Feb/2004	Appendix **??** inserted. Reference to appendix **??** inserted
v2.4	23/Feb/2004	Appendix **??** changed: rpm lists have been changed.
v2.5	12/Mar/2004	Appendix **??** changed
v2.5.2	15/Mar/2004	**??** GLOBUS_LOCATION environment variable to be definied before launching the initialization script
v2.5.3	16/Mar/2004	**??** Installation procedure updated
v2.5.4	18/Mar/2004	Chapter **??** and **??** changed
v2.6.0	01/Apr/2004	**??**: cvs references changed
v2.6.1	20/Apr/2004	**??** and **??** paragraph names changed
v2.7.0	21/Apr/2004	Correction in **??** for the file */opt/edg/etc/edg_wl.conf*: II_Port = 2170;
v2.8.0	26/Apr/04	**??**: section inserted dealing with GridICE agent configuration. Upgrade section changed accordingly. Numbering changed consequently. - 9: section inserted dealing with Experiment Software Mangement tool (SGM) configuration. Upgrade section changed accordingly Numbering shifted up consequently.