LHC COMPUTING GRID

LCG - COMPUTING ELEMENT MANUAL INSTALLATION AND CONFIGURATION

Document identifier:	**LCG-GIS-MI-CE**
EDMS id:	**none**
Version:	**v3.1.0**
Date:	**August 3, 2004**
Section:	**LCG Grid Infrastructure Support**
Document status:	**ACTIVE**
Author(s):	Antonio Retico, Alessandro Usai, Oliver Keeble
File:	**CE**

Abstract: *Manual installation and set-up of a LCG2 Computing Element*

CONTENTS

REFERENCES

[1] D. 1.2.8.1. edg-lcas reference manual, 2003. `http://www.dutchgrid.nl/DataGrid/wp4/lcas/edg-lcas-1.1/apidoc/latex/ref\%man.pdf`

[2] 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`

[3] A. Retico and A. Usai. Lcg wn manual software installation and configuration, 2004. `http://www.cern.ch/grid-deployment/gis/release-docs/WN-index.html`

ABSTRACT

Manual steps for the installation and configuration of a LCG CE 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/CE-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 CE software you need:

- a working Linux Redhat7.3 on the Computing Element (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 CE machine
- references to software packages to be installed (see below)

2. COMPUTING ELEMENT SOFTWARE INSTALLATION

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

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

Two different methods are supported for CE software installation

- Centrally managed CE installation using the Debian *apt-get* tool
- Manual CE 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. CE INSTALLATION USING *apt-get*

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

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

2.2. CE 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/CE-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:

CE.pdf: this guide

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

CE_manual_install_forced_rpm: plain full list of those rpms needed to install the CE 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

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

CE_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 *CE_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 **CE_tools.tar.gz**. So they can be easily downloaded and used on target machines.

The provided *CE_manual_install_rpm* includes:

- all the packages which are needed in order to have a production Computing Element.
- all the general purpose objects (i.e. shared libraries or common applications) which the CE 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 **CE_tools.tar.gz**.
- extract the archived files.

```
> gunzip CE_tools.tar.gz
> tar xvf CE_tools.tar
```

- give execution permissions and run the script *CE_manual_install_wget*, which will then fetch all the rpms required into the subdirectories *CE_INSTALL* and *CE_INSTALL_FORCED*

```
> cd CE_tools
> chmod u+x CE_manual_install_wget
> ./CE_manual_install_wget
```

Install the downloaded rpms

- In the directory **CE_INSTALL**, where the rpms listed in *CE_manual_install_rpm* have been downloaded, run the command

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

- In the directory **CE_INSTALL_FORCED** where the rpms listed in *CE_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 *CE_manual_install_rpm* list.

Should you prefer to do an upgrade of the CE 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 CE 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.

2.3. Installing the Certification Authority rpms

As the list and structure of 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 LCG release procedure. You should consult the page

http://markusw.home.cern.ch/markusw/lcg2CAlist.html

in order to ascertain what the version number of the latest set of CA rpms is. At time of writing it is 1.6, so the full set of rpms can be downloaded with the following command;

```
> wget -l1 -nd -r http://grid-deployment.web.cern.ch/grid-deployment/download/RpmDir/security/index_r1.6.html
```

These rpms should then be installed in the usual way;

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

Changes and updates to these settings will be announced on the LCG-Rollout mailing list.

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
  ```

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 if you do not have one already.
Instruction to obtain a CE list can be found in

http://markusw.home.cern.ch/markusw/lcg2CAlist.html

From the CA list so obtained you should choose a CA close to you.

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

hostcert.pem

containing the machine public key 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
  ```

  (this command produces a huge amount of output)

4.3. Create Top-level EDG configuration 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 LCG CONFIGURATON FILE

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

```
# Usual value: /opt/lcg
LCG_LOCATION=/opt/lcg

# Directory for machine-specific files.
# Usual value: $LCG_LOCATION/var
LCG_LOCATION_VAR=/opt/lcg/var

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

4.5. CREATE TOP-LEVEL GLOBUS CONFIGURATION 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"
```

4.6. 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.
By a cron job, the information in this file is periodically verified and, if needed, updated by the script

/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 [2].

5. EDG USER CREATION

The special user/group *edguser* has to be created.

```
> groupadd -g 995 edguser
> useradd -c "EDG User" -u 995 -g 995 -d /home/edguser edguser
```

(the proposed choice of group ID and user ID is not mandatory)

6. POOL ACCOUNTS CREATION FOR CE

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 CE.

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 CE

User and Group Creation

Create groups and users as they have been defined in 6.1..
The group IDs can be chosen without any particular constraint, the only one being, of course, that the same group IDs be used when creating the corresponding users.

A suitable algorithm for user creation on Computing Element follows:

```
for each group defined in 6.1.1.
   create the group on the CE
for each user defined in 6.1.2.
   create the user on the CE with the proper group ID
```

An implementation example follows, which uses two scripts provided in 6.4., in order to get the group and user lists from a second CE used for reference. Using the following method the user and group IDs on the SE to be installed will match with those in use on the CE node taken as reference.
As already said, this is not a requirement and, should this implementation conflict with local policies, any customized implementation of the above algorithm is admitted.

- Create a file *VO_names.cfg*

```
atlas
alice
cms
lhcb
dteam
EOF
```

- Launch on the reference CE node the script *CE_createUGlists.sh* (see 6.4.) feeding it with the file *VO_names.cfg*

  ```
  > ./CE_createUGlists.sh VO_names.cfg
  ```

  CE_LCG_groups.lst: lists the Unix groups to be created on WN, with corresponding group IDs.

  CE_LCG_users.lst: lists the Unix users to be created on WN with corresponding user and group IDs

- Launch on the CE you are installing the script *createPoolAccounts.sh* (see 6.4.)

```
> ./createPoolAccounts.sh  CE_LCG_groups.lst CE_LCG_users.lst NOT_SHARED
```

It will produce two shell script:

createGroups: command lines to create groups

createUsers: command lines to create users

- Check the produced shell script for possible errors

```
> less createGroups
> less createUsers
```

- Execute the two shell scripts on the CE

```
> chmod u+x createGroups createUsers
> ./createGroups
> ./createUsers
```

NFS and firewall (just for Method2)

If you are creating pool accounts according to Method2 (home directories to be shared with WN), and you are using firewalls at your site, you should be sure yourself that the firewall on your WN machines allow the NFS mounting of part of the CE filesystem (see the configuration of Pool Accounts in [3])

6.4. Example scripts for user and group handling

VO_config

```
atlas
alice
cms
lhcb
dteam
EOF
```

CE_createUGlists.sh

```
#!/bin/sh
########################
#
# File: CE_createUGlists.sh
#
# Author: LCG Deployment Team
#
# Version: 1.2
#
# Date: 2004/02/18
#
# Description:
#        The script queries the /etc/passwd and /etc/group files of a LCG CE
#       node and creates the two lists of groups and users to
#       be created in the WN (or any other LCG node wich you want to have its
#       users and groups identical to CE).
#       The list descriptions follow:
#        1) "CE_LCG_groups.lst" lists the Unix groups to be created.
#           Its records have the format
#          <group Id>:<group name>
#        2) "CE_LCG_users.lst" lists the Unix users to be created.
#           Its records have the format
#         <user Id>:<user Name>:<group Id>
#       This script queries the /etc/passwd and /etc/group files
#       and fetches the lines matching the VOs listed in the VO file taken
#       by the script as input
#
# Syntax::
#        ./CE_createUGLists.sh <VO file>
#
# Launch Example:
#        ./CE_createUGLists.sh VO_config
#
########################

# program information
PROGRAM="CE_createUGLists.sh"
VERSION="1.2"
SYNTAX="./${PROGRAM} <VO file>"
LAUNCH="./${PROGRAM} VO_config"


if [ $# == 0 ]; then

       echo "Sorry, missing parameters"
       echo "Please make sure to run the script feeding in the VO file"

       echo "Syntax: ${SYNTAX}"
       echo "Launch Example: ${LAUNCH}"
       echo

       exit
fi

PASSWD_FILE="/etc/passwd"
GROUP_FILE="/etc/group"
USER_LIST_FILE="CE_LCG_users.lst"
GROUP_LIST_FILE="CE_LCG_groups.lst"

cat /dev/null >  $USER_LIST_FILE

cat /dev/null >  $GROUP_LIST_FILE

count=0

while read line; do
   arrayVO[$count]=$line

   (( count++ ))

done < $1

for i in 0 1 2 3 4 ;do

   grep "${arrayVO[$i]}"  ${PASSWD_FILE} | awk -F":" ’
              {print $3 ":" $1 ":" $4}’ >> ${USER_LIST_FILE}

   grep "${arrayVO[$i]}"  ${GROUP_FILE}| awk -F":" ’
              {print $3 ":" $1 }’ >> ${GROUP_LIST_FILE}

done

cat <<EOF

DONE

Please find the ouput lists in:

./${GROUP_LIST_FILE} #: record of groups to be created
./${USER_LIST_FILE}  #: record of users to be created

EOF

#EOF
```

createPoolAccounts.sh

```
----------------------------------
#!/bin/sh
########################
#
# File: createPoolAccounts.sh
```

```
#
# Author: LCG Deployment Team
#
# Version: 1.2
#
# Date: 2004/03/02
#
# Description:
#        This script is feeded by two files generated by the script
#        CE_createUGlists
#       and output two shell scripts to be manually launched in order to
#       create the desired users and groups
#       the specification for input files follows:
#
#       1) <CE_GroupFile> lists the Unix groups to be created.
#           Its records must be in the format
#          <group Id>:<group name>
#       2) <CE_UserFile> lists the Unix users to be created.
#           Its records nust be in the format
#          <user Id>:<user Name>:<group Id>
#       3) <share_flag>:
#           SHARED: home directories have to be shared with CE
#           NOT_SHARED; home directorier are not shared with CE
#
#       The output files are:
#        createGroups: command lines to create groups
#        createUsers: command lines to create users
#
# Syntax:
#       ./createPoolAccounts.sh  <CE_GroupFile> <CE_UserFile> <SHARED|NOT_SHARED>
#
# Launch Command:
#        ./createPoolAccounts.sh  CE_LCG_groups.lst CE_LCG_users.lst NOT_SHARED
#
########################

# program information
PROGRAM="createPoolAccounts.sh"
VERSION="1.2"
SYNTAX="./${PROGRAM} <CE_GroupFile> <CE_UserFile> <SHARED|NOT_SHARED>"
LAUNCH="./${PROGRAM} CE_LCG_groups.lst CE_LCG_users.lst NOT_SHARED"

CREATE_GROUPS_FILE=createGroups
CREATE_USERS_FILE=createUsers

echo "${PROGRAM} v${VERSION}"

if test $# -eq 3; then
    if [ ! -f ${1} ]; then
        echo "ERROR: file ${1} does not exist"
        exit
    else
        RET_CODE=‘awk -F":" ’{ if (NF != 2) {print "-1"; exit} } ’ ${1}‘

        if  [ ! ${RET_CODE} == ’’ ]; then
            echo "ERROR: file ${1} is not correctly formatted"
            exit
              fi
    fi

    if [ ! -f ${2} ]; then
        echo "ERROR: file ${2} does not exist"
        exit

    else
        RET_CODE=‘awk -F":" ’{ if (NF != 3) {print "-1"; exit} } ’ ${2}‘

        if  [ ! ${RET_CODE} == ’’ ]; then
            echo "ERROR: file ${2} is not correctly formatted"
            exit
              fi
    fi

    if [ ${3} != "SHARED" ] && [ ${3} != "NOT_SHARED" ] ; then
        echo "ERROR: unknown value ${3} for share_flag <SHARED|NOT_SHARED>"
        exit

    else
        if [ ${3} = "SHARED" ]; then
            HOMEFLAG="-M"
        else
            HOMEFLAG=""
        fi
    fi
```

```
else
    echo "Sorry, Missing parameters"
    echo "Syntax: ${SYNTAX}"
    echo "Launch Example: ${LAUNCH}"
    exit
fi

awk -F":" ’{print "groupadd -g "$1" "$2}’ ${1} > ${CREATE_GROUPS_FILE}
awk -F":" -v home=${HOMEFLAG} ’{print "useradd -c \"mapped user for group ID "$3"\" -u "$1" "home" -g "$3" "$2}’ ${2} > ${CREATE_USERS_FILE}

chmod u+x  ${CREATE_GROUPS_FILE}
chmod u+x  ${CREATE_USERS_FILE}

cat <<EOF
DONE

Please complete the creation of groups and accounts
by running, in the proposed sequence, the following two scripts:

./${CREATE_GROUPS_FILE} #: command lines to create groups
./${CREATE_USERS_FILE}  #: command lines to create users

EOF

# Execute Creation
#./${CREATE_GROUPS_FILE}
#./${CREATE_USERS_FILE}

#EOF
```

7. LRMS Client Configuration

In order to configure a specific LRMS client on the Computing Element please refer to one of the following subsections:

- PBS –> 7.1.
- LSF –> 7.4.
- Condor –> 7.5.

7.1. PBS installation and configuration on CE

PBS Software Installation

All the tools needed for PBS installation are available on the *Grid Deployment* web site

http://www.cern.ch/grid-deployment/gis/release-docs/CE-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:

CE_pbs_install_rpm: a plain full list of the rpms needed to install the PBS client on a plain RH7.3 operative system.

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

The *CE_pbs_install_rpm* list includes all the rpms which are needed in order to have the PBS system installed.

The simplest way to use the provided tools is:

- log in as *root* into the node to be installed and copy the file *CE_pbs_install_wget* into an empty working directory.
- run the script *CE_manual_install_wget*, which will fetch all the rpms required into the local directory
- Install the downloaded rpms, running the following command

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

  in the directory where the rpms have been downloaded
  Due to the possible misalignments between local installation of RH7.3 at different sites, this operation might yield some missing dependencies which have to be analyzed and solved by adding/removing rpms to/from the *CE_pbs_install_rpm* list.

7.2. PBS SOFTWARE CONFIGURATION

Several configuration steps are required in order to configure the PBS (Portable Batch System).

In case the PBS daemons are running, stop them

```
> /etc/rc.d/init.d/pbs_server stop
> /etc/rc.d/init.d/pbs_sched stop
> /etc/rc.d/init.d/pbs_mom stop
```

Set Server Name

Create the file

/var/spool/pbs/server_name

which should contain one line with the full name of the local host.

```
 > /bin/hostname > /var/spool/pbs/server_name
```

Configure ports for PBS services

Append to the file */etc/services* the following lines

```
pbs                  15001/tcp
pbs_sched             15004/tcp
pbs_mom              15002/tcp
pbs_remom             15003/tcp
pbs_remom             15003/udp
globus-gatekeeper      2119/tcp
```

Configure SSH deamon

If they are not yet, set up into the file */etc/ssh/sshd_config* the following lines, in order to enable host based authentication:

```
HostbasedAuthentication yes
IgnoreUserKnownHosts yes
IgnoreRhosts yes
```

Then re-start the SSH deamon

```
> service sshd restart
```

Configure Firewall

If you are using a firewall consider that the Pbs serve needs the following udp ports to open.
udp 1021 udp 1022 udp 1023 For example, if you are using ipchains for firewall configuration, you can,

- allow restrictively udp traffic incoming from your worker nodes. To do so, you should add **before the first ”REJECT”** line into the file */etc/sysconfig/ipchains*, a line

  ```
  -A input -p udp -s <WN-hostname> -d 0/0 1021:1023 -j ACCEPT
  ```

  For each worker node to be handled

- allow udp traffic incoming from your subnet. To do so, add **before the first ”REJECT”** line into the file */etc/sysconfig/ipchains*, a line

  ```
  -A input -p udp -s <your-subnet>/<type> -d 0/0 1021:1023 -j ACCEPT
  ```

  where <your-subnet> is your subnet address, as configured in the parameter *NETWORK* in the script

  /etc/sysconfig/network-scripts/ifcfg-eth0

  and<type> is a value which depends on the class of your network, with the following correspondence table.

netmask	type	Subnet
255.0.0.0	8	Class A
255.255.0.0	16	Class B
255.255.255.0	24	Class C
255.255.255.255	32	Point-to-point

  After this, the ipchains service has to be restarted:

  ```
  > service ipchains restart
  ```

Create the first PBS server configuration

- Edit the file var/spool/pbs/pbs_environment in the following way

  ```
  > cat <<-EOF > /var/spool/pbs/pbs_environment
  PATH=/bin:/usr/bin
  LANG=en_US
  LC_ALL=en_US
  EOF
  ```

This file should be automatically created by the running the *etc/rc.d/init.d/pbs_server start* command. The manual creation is a workaround due to a bug in the *pbs_server* initialization script.

- Create the first pbs configuration

  ```
  > /usr/sbin/pbs_server -t create
  ```

Configure the queues

PBS queues have to be configured by the PBS management tool *qmgr*. the In order to do so the *qmgr* application has to be given in input the following queue configuration commands:

```
set server scheduling = True
set server acl_host_enable = False
set server managers = root@<CE hostname>
set server operators = root@<CE hostname>
set server default_queue = short
set server log_events = 511
set server mail_from = adm
set server query_other_jobs = True
set server scheduler_iteration = 600
set server default_node = lcgpro
set server node_pack = False

create queue short
set queue short queue_type = Execution
set queue short resources_max.cput = 00:15:00
set queue short resources_max.walltime = 02:00:00
set queue short enabled = True
set queue short started = True

create queue long
set queue long queue_type = Execution
set queue long resources_max.cput = 12:00:00
set queue long resources_max.walltime = 24:00:00
set queue long enabled = True
set queue long started = True

create queue infinite
set queue infinite queue_type = Execution
set queue infinite resources_max.cput = 48:00:00
set queue infinite resources_max.walltime = 72:00:00
set queue infinite enabled = True
set queue infinite started = True
```

in order to do so quickly, you can write the lines above in a file (E.g. *temporary_file*) and then run the command

```
> /usr/bin/qmgr <CE hostname> < temporary_file
```

where <CE hostname> is the full hostname of your machine.

Configure worker nodes

Create the file */var/spool/pbs/server_priv/nodes* For each batch nodes (WNs) to be handled by the CE append a line to the file

/var/spool/pbs/server_priv/nodes

with the following syntax:

<WN1_hostname> np=2 lcgpro

<WN2_hostname> np=2 lcgpro

....

in example:

```
lxn1183.cern.ch np=2 lcgpro
lxn1184.cern.ch np=2 lcgpro
```

Then re-start PBS server

```
> service pbs_server restart
```

Configure the script enabling WNs to copy output back to the CE

- Copy the file

  /opt/edg/etc/edg-pbs-shostsequiv.conf.template

  into

  /opt/edg/etc/edg-pbs-shostsequiv.conf

  then edit this file and change parameters to your needs. Most sites will only have to set NODES to an empty string.

- Create the first version of the */etc/ssh/shosts.equiv* file by running

  ```
  > /opt/edg/sbin/edg-pbs-shostsequiv
  ```

  A cron job will update this file every 6 hours (see 7.2.13.).

Add nodes to ssh *"known-host"* list

- Create the file */opt/edg/etc/edg-pbs-knownhosts.conf* and insert the following lines

  ```
  NODES      = <CE-hostname> <SE-hostname>

  PBSBIN     = /usr/bin

  KEYTYPES   = rsa1,rsa,dsa

  KNOWNHOSTS = /etc/ssh/ssh_known_hosts
  ```

Customize the values of <CE-hostname> (default Computing Element) and <SE-hostname> (default Storage Element)

- Run the script:

  ```
  > /opt/edg/sbin/edg-pbs-knownhosts
  ```

  to update the relevant information about the hosts known by the Computing Element.

Note: every time you will add or remove WNs, do not forget to run

```
> /opt/edg/sbin/edg-pbs-shostsequiv
> /opt/edg/sbin/edg-pbs-knownhosts
```

on the CE or the new WNs will not work correctly till the next time cron runs them for you.

Configure PBS MOM (Machine Oriented Mini-server)

The execution of this configuration step is slightly different on whether the pool accounts on WN are sharing their home directory with CE or they are not (see 6.).
The difference in configuration is explained in the warning.

- Create and edit the file */var/spool/pbs/mom_priv/config* with the following lines (beware of the warning)

  ```
  $clienthost localhost
  $clienthost <CE-hostname>
  $restricted <CE-hostname>
  $logevent 255
  $ideal_load 1.6
  $max_load 2.1
  $usecp *.cern.ch:/home /home
  ```

  WARNING: The line $*usecp *.cern.ch:/home /home* is to be inserted if and only if the Worker Node mounts the CE home directory (see 6.).
  The CE configuration as well is necessarily different in the two cases

(Re-)start PBS server

```
> chkconfig pbs_server on
> service pbs_server restart

> chkconfig pbs_sched on
> service pbs_sched start
```

```
> chkconfig pbs_mom on
> /etc/rc.d/init.d/pbs_mom start
```

Add PBS services to the crontab

- Edit the crontab to add scheduled services

  ```
  > crontab -e
  ```

  This will allow the editing of the crontable, after which the following lines should be added:

  ```
  PATH=/sbin:/bin:/usr/sbin:/usr/bin
  30 1,7,13,19 * * * /opt/edg/etc/cron/edg-fetch-crl-cron >> /var/log/edg-fetch-crl-cron.log 2>&1
  30 2,8,14,20 * * * /sbin/service edg-wl-locallogger proxy
  03 1,7,13,19 * * * /opt/edg/sbin/edg-pbs-knownhosts
  05 1,7,13,19 * * * /opt/edg/sbin/edg-pbs-shostsequiv
  ```

  WARNING: The line containing */opt/edg/etc/cron/edg-fetch-crl-cron* and the line *PATH=/sbin:/bin:/usr/sbin:/usr/bin* may already be present in the crontab. In this case they do not need to be added.

- In order to handle the output of the scheduled scripts and so prevent a lot of e-mails to be regularly sent to root, an automatic rotation of the log file can be obtained editing, for example, the file

  /etc/logrotate.d/edg-fetch (create it if it is needed)

  An example of configuration of the */etc/logrotate.d/edg-fetch* for a monthly log rotation follows:

  ```
  /var/log/edg-fetch-crl-cron.log  {
      compress
      monthly
      rotate 12
      missingok
      ifempty
      create
  }
  ```

7.3. Basic test for PBS

Communication test

To test whether the configured worker nodes can be reached

- configure the WNs to use your Computing Element (see [3])
- run the command

```
> pbsnodes -a
```

The expected output is a list of worker nodes in the form

```
     <WN1-hostname>
          state = free
          np = 2
          speed = 0
          properties = lcgpro
          ntype = cluster
     ....

     <WNn-hostname>
          state = free
          np = 2
          speed = 0
          properties = lcgpro
          ntype = cluster
```

Job Submission test

To test whether simple jobs can be executed

- switch to a Pool Account (see 6.) eg. *atlas 001*

  ```
  > su - atlas001
  ```

- create a test script

  ```
  > cat << EOF > testscript.sh
  #!/bin/sh
  date
  hostname
  EOF
  ```

- submit the job

  ```
  > qsub testscript.sh
  ```

  The expected output is a line in the shape:

  <JobID>.<CE-hostname> (e.g. 1.adc0025.cern.ch)

- wait a reasonable time for the job to be excuted

- check the output in the file *testscript.sh.o<JobId>*
- check possible errors in the file *testscript.sh.e<JobId>*

7.4. LSF INSTALLATION AND CONFIGURATION ON CE

TBD

7.5. CONDOR INSTALLATION AND CONFIGURATION ON CE

TBD

8. GRIS CONFIGURATION

8.1. CREATE GROUP AND USER *edginfo*

Create both the user and group *edginfo* on the CE.
The user home directory should be */opt/edg/var*

```
> groupadd -g 999 edginfo
> useradd -c "EDG Info user" -u 999 -g 999 -d /opt/edg/var edginfo
```

(the proposed choice of group ID and user ID is not mandatory)

8.2. CREATE GLOBUS CONFIGURATION FILE

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

```
[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=<your-registration-name>
reghn=<CE-hostname>



[mds/giis/site]
name=<your-registration-name>
allowreg="<CE-hostname>:2135"
allowreg="<SE1-hostname>:2135"
allowreg="<SE2-hostname>:2135"
[mds/giis/site2]
name=site2


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


[gatekeeper]
default_jobmanager=fork
job_manager_path=$GLOBUS_LOCATION/libexec
globus_gatekeeper=/opt/edg/sbin/edg-gatekeeper
extra_options=\"-lcas_db_file lcas.db -lcas_etc_dir /opt/edg/etc/lcas/ -lcasmod_dir  \
/opt/edg/lib/lcas/ -lcmaps_db_file lcmaps.db -lcmaps_etc_dir /opt/\"

logfile=/var/log/globus-gatekeeper.log

jobmanagers="fork lcgpbs"

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


[gatekeeper/<your-batch-system>]
type=<your-batch-system>
```

WARNING the line in the template with the definition of the variable *extra_options* has been split for graphical reasons: So the two lines

```
extra_options=\"-lcas_db_file lcas.db -lcas_etc_dir /opt/edg/etc/lcas/ -lcasmod_dir  \
/opt/edg/lib/lcas/ -lcmaps_db_file lcmaps.db -lcmaps_etc_dir /opt/\"
```

should be written in the real configuration file as

```
extra_options=\"-lcas_db_file lcas.db -lcas_etc_dir /opt/edg/etc/lcas/ -lcasmod_dir /opt/edg/lib/lcas/ -lcma...\"
```

The meaning of the configurable parameters, reported in the template among angle brakets “<...>”, is:

your-registration-name: it is the name which will identify your site in the information system.

allowreg: the listed hosts will be allowed to publish information on your GRIS

your-batch-system: The implementation of your batch system. Available are:

- **lcgpbs** – if you are using PBS
- **lcglsf** – if you are using lsf

If you have doubts on how to configure the files above listed and you have a reference site, please ask them for indications.
Otherwise, send a message to the ”LCG-ROLLOUT@cclrclsv.RL.AC.UK” mailing list.

A production example of configuration for */etc/globus.conf* file follows:

```
#########################################################################
#                                                            #
# Globus configuraton.                                          #
#                                                            #
#########################################################################

[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=adc0029.cern.ch

[mds/giis/site]
name=cerntestlcg2
allowreg="adc0029.cern.ch:2135"
allowreg="adc0033.cern.ch:2135"
allowreg="adc0037.cern.ch:2135"
allowreg="adc0028.cern.ch:2135"
allowreg="lxn1182.cern.ch:2135"
allowreg="adc0031.cern.ch:2135"

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


[gatekeeper]
default_jobmanager=fork
job_manager_path=$GLOBUS_LOCATION/libexec
globus_gatekeeper=/opt/edg/sbin/edg-gatekeeper
extra_options=\"-lcas_db_file lcas.db -lcas_etc_dir /opt/edg/etc/lcas/ -lcasmod_dir \
/opt/edg/lib/lcas/ -lcmaps_db_file lcmaps.db -lcmaps_etc_dir /opt/edg/etc/lcmaps -lcmapsmod_dir /opt/edg/lib/lcmaps\"
logfile=/var/log/globus-gatekeeper.log

jobmanagers="fork lcgpbs"
```

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


[gatekeeper/lcgpbs]
type=lcgpbs
```

WARNING the line in the template with the definition of the variable *extra_options* has been splitted for graphical reasons: So the two lines

```
extra_options=\"-lcas_db_file lcas.db -lcas_etc_dir /opt/edg/etc/lcas/ -lcasmod_dir \
/opt/edg/lib/lcas/ -lcmaps_db_file lcmaps.db -lcmaps_etc_dir /opt/
```

should be written in the real configuration file as

```
extra_options=\"-lcas_db_file lcas.db -lcas_etc_dir /opt/edg/etc/lcas/ -lcasmod_dir /opt/edg/lib/lcas/ -lcma....
```

8.3. Create directories

create the directories:

/opt/edg/var/info/

/opt/edg/var/etc/

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

8.4. Create information provider configuration files

Create the following files:

- */opt/edg/libexec/ceinfo-wrapper.sh* This shell script launches the *edg-ce-all* command with the apprpriate parameters. It is the configuration file for the Computing Element Information Provider, which is claimed to gather pbs queues information from the CE.
  Remember to give the file the execution permissions.
  The layout is:

  ```
  #!/bin/sh
  /opt/edg/libexec/edg-ce-all                         \
    -lrms pbs                                  \
    -cluster <CE-hostname>                      \
    -queue short long infinite                   \
    -globus-config-file /opt/globus/etc/globus-gatekeeper.conf \
    -static /opt/edg/var/etc/ce-static.ldif         \
    -auth-users-from-grid-mapfile /dev/null        \
    -globus-gatekeeperport 2119      \
  ```

```
  -globus-gatekeeperhost <CE-hostname>    \
  -globus-jobmanager lcgpbs   \
  -grisport 2135   \

  -ttl 120   \
  -cesebind '.' <SE-hostname> /storage
```

```
   > chmod ugo+x /opt/edg/libexec/ceinfo-wrapper.sh
```

- */opt/edg/var/info/siteinfo.ldif* This file deals with site information and version of the software installed in the site.
  The layout is:

```
siteName:            <SITENAME>
sysAdminContact:      "hep-project-grid-cern-testbed-managers@cern.ch"
userSupportContact:   "hep-project-grid-cern-testbed-managers@cern.ch"
siteSecurityContact: "hep-project-grid-cern-testbed-managers@cern.ch"
dataGridVersion:      <version tag>
installationDate:     <YYYYMMDDhhmmss>Z
```

  The >version-tag> is the cvs tag you are installing.
  A production example of configuration for */opt/edg/var/info/siteinfo.ldif* file follows:

```
siteName:            CERN-LCG2
sysAdminContact:      "hep-project-grid-cern-testbed-managers@cern.ch"
userSupportContact:   "hep-project-grid-cern-testbed-managers@cern.ch"
siteSecurityContact: "hep-project-grid-cern-testbed-managers@cern.ch"
dataGridVersion:      LCG-2_0_0beta
installationDate:     20040106120000Z
```

- /opt/edg/var/etc/ce-static.ldif

  This file contains static information on the CE machine. When submitting a job, users of LCG are supposed to state in their jdl the minimal hardware resources (memory, scratch disk space, CPU time) required to run the job. These requirements are matched by the RB with the information on the BDII to select a set of available CEs where the job can run.

  For this schema to work, each CE must publish some information about the hardware configuration of the WNs connected to it. This means that site managers must collect information about WNs available at the site and insert it in the information published by the local CE.

In general, WNs are added to a batch system at different times and with heterogeneous hardware configurations.
All these WNs often end up being part of a single queue, so that when a LCG job is sent to the batch system, there is no way to ask for a specific hardware configuration (note: LSF and other batch systems offer ways to do this but the current version of the Globus gatekeeper is not able to take advantage of this possibility). This means that the site manager has to choose a single WN as ”representative” of the whole batch cluster. In general it is strongly recommended that this node is chosen among the ”least powerful” ones, to avoid sending jobs with heavy hardware requirements to under-spec nodes. The values in the configuration file below labeled with the “FABRIC” prefix refer to this “representative” Worker Node

The layout of file */opt/edg/var/etc/ce-static.ldif* is:

```
dn: GlueCEUniqueID=<CE-hostname>:2119/jobmanager-lcgpbs-short
GlueCEAccessControlBaseRule: VO:alice
GlueCEAccessControlBaseRule: VO:atlas
GlueCEAccessControlBaseRule: VO:cms
GlueCEAccessControlBaseRule: VO:lhcb
GlueCEAccessControlBaseRule: VO:dteam

dn: GlueCEUniqueID=<CE-hostname>:2119/jobmanager-lcgpbs-long
GlueCEAccessControlBaseRule: VO:alice
GlueCEAccessControlBaseRule: VO:atlas
GlueCEAccessControlBaseRule: VO:cms
GlueCEAccessControlBaseRule: VO:lhcb
GlueCEAccessControlBaseRule: VO:dteam

dn: GlueCEUniqueID=<CE-hostname>:2119/jobmanager-lcgpbs-infinite
GlueCEAccessControlBaseRule: VO:alice
GlueCEAccessControlBaseRule: VO:atlas
GlueCEAccessControlBaseRule: VO:cms
GlueCEAccessControlBaseRule: VO:lhcb
GlueCEAccessControlBaseRule: VO:dteam

dn: GlueSubClusterUniqueID=<CE-hostname>, GlueClusterUniqueID=<CE-hostname>
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2
GlueHostNetworkAdapterOutboundIP: <FABRIC-outbound-allowed>
GlueHostNetworkAdapterInboundIP: <FABRIC-inbound-allowed>
GlueHostArchitectureSMPSize: <FABRIC-CPU-number>
GlueHostOperatingSystemName: <FABRIC-OS-name>
GlueHostOperatingSystemRelease: <FABRIC-OS-release>
GlueHostOperatingSystemVersion: <FABRIC-OS-version>
GlueHostBenchmarkSF00: <FABRIC-SpecInt2000>
GlueHostBenchmarkSI00: <FABRIC-SpecFloat2000>
```

```
GlueHostMainMemoryRAMSize: <FABRIC-RAM-size>
GlueHostMainMemoryVirtualSize: <FABRIC-VM-size>
GlueHostProcessorClockSpeed: <FABRIC-clock-speed>
GlueHostProcessorModel: <FABRIC-CPU-model>
GlueHostProcessorVendor: <FABRIC-CPU-vendor>
```

the parameters among square brackets “<...>” will be directly published by the Information system according to the GLUE schema definitions. Their meaning is:

CPU-model: model of the CPU used by the CE. This parameter is a string whose domain is not defined yet in the GLUE Schema. The value used for Pentium III is “PIII”.

vendor: vendor of the CPU. used by the CE. This parameter is a string whose domain is not defined yet in the GLUE Schema. The value used for Intel is “intel”.

FABRIC-outbound-allowed: TRUE if outbound connectivity is enabled at your site, FALSE otherwise

FABRIC-inbound-allowed: TRUE if inbound connectivity is enabled at your site, FALSE otherwise

FABRIC-CPU-number: number of cpus in an SMP box

FABRIC-clock-speed: clock frequency in Mhz

FABRIC-RAM-size: RAM size in kblocks

FABRIC-VM-size: Virtual Memory size in kblocks

FABRIC-OS-name: Operating System name

FABRIC-OS-release: Operating System release

FABRIC-OS-version: Operating System version

FABRIC-SpecInt2000: Performance index of your fabric in SpecInt 2000. For some examples of Spec values see http://www.specbench.org/osg/cpu2000/results/cint2000.html

FABRIC-SpecFloat2000: Performance index of your fabric in SpecFloat 2000. For some examples of Spec values see http://www.specbench.org/osg/cpu2000/results/cint2000.html

A productione example for the file */opt/edg/var/etc/ce-static.ldif* follows:

```
dn: GlueCEUniqueID=adc0029.cern.ch:2119/jobmanager-lcgpbs-short
GlueCEAccessControlBaseRule: VO:alice
GlueCEAccessControlBaseRule: VO:atlas
GlueCEAccessControlBaseRule: VO:cms
GlueCEAccessControlBaseRule: VO:lhcb
GlueCEAccessControlBaseRule: VO:dteam

dn: GlueCEUniqueID=adc0029.cern.ch:2119/jobmanager-lcgpbs-long
GlueCEAccessControlBaseRule: VO:alice
GlueCEAccessControlBaseRule: VO:atlas
GlueCEAccessControlBaseRule: VO:cms
GlueCEAccessControlBaseRule: VO:lhcb
GlueCEAccessControlBaseRule: VO:dteam

dn: GlueCEUniqueID=adc0029.cern.ch:2119/jobmanager-lcgpbs-infinite
GlueCEAccessControlBaseRule: VO:alice
GlueCEAccessControlBaseRule: VO:atlas
GlueCEAccessControlBaseRule: VO:cms
```

```
GlueCEAccessControlBaseRule: VO:lhcb
GlueCEAccessControlBaseRule: VO:dteam

dn: GlueSubClusterUniqueID=adc0029.cern.ch, GlueClusterUniqueID=adc0029.cern.ch
GlueHostApplicationSoftwareRunTimeEnvironment: LCG-2-TEST
GlueHostNetworkAdapterOutboundIP: TRUE
GlueHostNetworkAdapterInboundIP: FALSE
GlueHostArchitectureSMPSize: 2
GlueHostOperatingSystemName: Redhat
GlueHostOperatingSystemRelease: 7.3
GlueHostOperatingSystemVersion: 3
GlueHostBenchmarkSF00: 400
GlueHostBenchmarkSI00: 380
GlueHostMainMemoryRAMSize: 512
GlueHostMainMemoryVirtualSize: 1024
GlueHostProcessorClockSpeed: 1000
GlueHostProcessorModel: PIII
GlueHostProcessorVendor: intel
```

- */opt/edg/var/info/info-provider.conf* The layout is:

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

If you have doubts about the values to be configured into the files above listed and you have a reference site, please ask them for indications. Otherwise, send a message to the ”LCG-ROLLOUT@cclrclsv.RL.AC.UK” mailing list.

8.5. Launch Info Provider Initialization Script

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

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

- Launch the setup script

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

8.6. Launch Globus Inizialization Script

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

This Globus script is meant to be launched on the whole series of LCG1 nodes, as a general configuration step. So that it may perform some configuration actions which are not specifically related to a CE. Because of this it can yield warning messages as well as some error ones. Other typical error messages you can get on a CE are due to the fact that the configuration script will try and perform configuration

steps for every kind of GRAM reporter, and of course it will cause errors since it may look up for a GRAM reporter that you are currently not using. An example of warning messages to be considered ”normal” in a CE configuration using PBS has been produced in Appendix 1.
Since the script produces a huge output, in order to check it, it is recommended to re-direct the standard output in a log file, using, for instance, the command

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

8.7. Start the Information service

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

8.8. Test the GRIS and GIIS

Run the following *ldap* queries:

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

If the output shows some entries like the characteristics of the PBS queues that you setup, then it is very possible that the GRIS just configured on the machine works.

9. GLOBUS GATEKEEPER AND GLOBUS GRIDFTP

Start the Globus Gatekeeper and the Globus GridFtp Server.
To do so, ron the following commands

```
> /sbin/chkconfig globus-gatekeeper on
> /sbin/chkconfig globus-gridftp on
> service globus-gatekeeper start
> service globus-gridftp start
```

10. REPLICA MANAGER CONFIGURATION

If the CE is going to be used as a Worker Node as well you will need to install the Replica Manager Client on the node. Please refer to [3]

11. LCAS (Local Centre Authorization Service) configuration

For further information on the LCAS system please refer to [1].

11.1. Configure users allowed to run in the fabric

Create and configure the file

/opt/edg/etc/lcas/allowed_users.db

This file is used to configure the globus user ids that are allowd to run on the fabric.
The layout of the file is

```
#
# This file contains the globus user ids that are ALLOWED on this fabric
#
# NOTE: The current version of LCAS does *NOT* use this file. It uses the
#       gridmap file instead. So, if this file is empty, that's ok.
#

"<GSI Credential>"
"<GSI Credential>"
"<GSI Credential>"
...
```

The **GSI Credential** is the subject of the user X509 certificate issued by the local GSI administrator. Usually the *usercert.pem* is stored into the user's $*HOME/.globus/* directory.
The piece of information in the subject can be retrieved from the certificate itself by the following command:

```
> grid-cert-info -subject -file ~/.globus/usercert.pem
```

Since LCAS does not currently use this file and so this file can be left empty, you can just copy the template *allowed_users.db.in* contained in the same path.

```
> cp /opt/edg/etc/lcas/allowed_users.db.in /opt/edg/etc/lcas/allowed_users.db
```

11.2. Configure users NOT allowed to run in the fabric

Create and configure the file */opt/edg/etc/lcas/ban_users.db* . This file is used to configure the globus user ids that are not allowed to run on the fabric.
It should contain the GSI credentials (subject of the GSI certificate) of those users which are specifically banned from this CE.
The layout of the file is

```
#
# This file contains the globus user ids that are BANNED from this fabric
#

"<GSI Credential>"
"<GSI Credential>"
"<GSI Credential>"
...
```

The meaning of *GSI Credential* has been given in 11.1. If you do not have any particular requirement for users to be banned, you can just copy the template *ban_users.db.in* contained in the same path.

```
> cp /opt/edg/etc/lcas/ban_users.db.in /opt/edg/etc/lcas/ban_users.db
```

11.3. SET-UP FABRIC WORKING TIMESLOTS

Create and configure the file */opt/edg/etc/lcas/timeslots.db* .
This file is used to configure the time slots within which the fabric is available for Grid jobs.

Configuration instruction and several examples are produced in the header of the template *timeslots.db.in* contained in the same path.
The layout of the template file is re-produced below:

```
#
# This file contains the time slots for which the fabric
# is available for Grid jobs
# Format:
#       minute1-minute2 hour1-hour2 mday1-mday2 month1-month2 year1-year2 wday1-wday2
# max range:  [0-59]       [0-23]      [1-31]      [1-12]    [1970-...]     [0-6]
#
# wday:
#  0-6 = Sunday-Saturday
#  5-3 = Friday-Wednesday
#
# '*' means the maximum range
# <val>- means from <val> to maximum value
#
# The wall clock time should match at least one time slot for authorization
# The wall clock time matches if:
#          (hour1:minute1)      <= (hour:minute)     <= (hour2:minute2)
#      AND (year1.month1.mday1) <= (year.month.mday) <= (year2.month2.mday2)
#      AND (wday1)             <= (wday)           <= (wday2)
#
# If the fabric is open on working days from 8:30-18:00 h, from 1 July 2002 till 15 January 2003
# the following line should be added:
#     30-0          8-18       1-15        7-1      2002-2003      1-5
# If the fabric is open from 18:00-7:00 h, two time slots should be used:
#     18:00-24:00 and 0:00-7:00
#
#      0-0         18-24        *           *           *          *
#      0-0          0-7        *           *           *          *
# If the fabric is always open the following line should be uncommented:
#       *            *         *           *           *          *
# minute1-minute2 hour1-hour2 mday1-mday2 month1-month2 year1-year2 wday1-wday2
#
#* * * * * *
```

Note that you will have to configure a policy in order for jobs to run. Uncommenting the final line will mean that your fabric is always open.

11.4. CONFIGURE LCAS POLICY

Create and configure the file */opt/edg/etc/lcmaps/lcmaps.db* as follows:

```
# LCMAPS policyfile
# @(#)/opt/edg/etc/lcmaps/lcmaps.db
#
# where to look for modules
path = /opt/edg/lib/lcmaps/modules

# module definitions
localaccount = "lcmaps_localaccount.mod  -gridmapfile /etc/grid-security/grid-mapfile"
poolaccount = "lcmaps_poolaccount.mod -override_inconsistency  -gridmapfile /etc/grid-security/grid-mapfile -gridmapdir /etc/grid-security/gridmapdi
posixenf = "lcmaps_posix_enf.mod -maxuid 1 -maxpgid 1 -maxsgid 32 "

# policies
standard:
localaccount -> posixenf | poolaccount
poolaccount -> posixenf
```

A template *lcmaps.db.in* is contained in the same path.

11.5. CONFIGURE LCAS DATABASE/PLUGIN LIST

Create and configure the file */opt/edg/etc/lcas/lcas.db* as follows:

```
# LCAS database/plugin list
#
# Format of each line:
# pluginname="<name/path of plugin>", pluginargs="<arguments>"
#
#
pluginname=lcas_userallow.mod,pluginargs=allowed_users.db
pluginname=lcas_userban.mod,pluginargs=ban_users.db
pluginname=lcas_timeslots.mod,pluginargs=timeslots.db
pluginname=lcas_plugin_example.mod,pluginargs=arguments
```

The template *lcas.db.in* is contained in the same path.

12. Workload Manager Configuration

- Create the directory */opt/edg/var/etc/profile.d*

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

- Copy from the directory

  /opt/edg/etc/profile.d

  into the directory

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

  the following files:

  – edg-wl.csh
  – edg-wl.sh

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

- Start the Local Logging Service:

  ```
  > chkconfig edg-wl-locallogger on
  > service edg-wl-locallogger start
  ```

  The following output can be considered ”normal”

```
Starting LocalLogger: edg-wl-interlogd and edg-wl-logd.
rm: cannot remove ‘/tmp/dglogd_sock_*’: No such file or directory
This is LocalLogger, part of Workload Management System in EU DataGrid.Copyright (c) 2002 CERN, INFN and CESNET on behalf of the EU DataGrid.
Initializing...
Parse messages for correctness...yes.
Send messages also to inter-logger...yes.
Store messages with the filename prefix "/var/tmp/dg20logd_"...yes.
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root     10219  3.0  0.5  2544 1308 pts/1    S    16:36   0:00 /bin/sh /etc/i
edguser  10239  0.0  0.7 15384 2028 ?        S    16:36   0:00 /opt/edg/sbin/
edguser  10243  0.0  0.7 15384 2028 ?        S    16:36   0:00 /opt/edg/sbin/
edguser  10244  0.0  0.7 15384 2028 ?        S    16:36   0:00 /opt/edg/sbin/
edguser  10245  0.0  0.7 15384 2028 ?        S    16:36   0:00 /opt/edg/sbin/
edguser  10246  0.0  0.7  4004 1916 ?        S    16:36   0:00 /opt/edg/sbin/
```

13. 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 13.1. you can find the configuration of agents while section 13.2. deals with the configuration details for the GridICE collector server.
If you are doing a defaul LCG-2 configuration you should apply section 13.2. 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

13.1. 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)

Should you need to set up a GridICE collector node at your site use refer to section 13.1..

- start the GridICE agent

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

13.2. 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
```

14. EXPERIMENT SOFTWARE MANAGER CONFIGURATION

For each Virtual Organization supported by the site, a special account to handle the VO software has to be created.

The mapping of SGM users to the LCG VOs follows:

VO name	SGM user
alice	alicesgm
atlas	atlassgm
cms	cmssgm
lhcb	lhcbsgm
dteam	dteamsgm

The following lines are an example for user creation

```
> useradd -c "SGM user for dteam" -u 18946 -g 2688 dteamsgm
> useradd -c "SGM user for atlas" -u 18943 -g 1307 atlassgm
> useradd -c "SGM user for lhcb" -u 18945 -g 1470 lhcbsgm
> useradd -c "SGM user for cms" -u 18944 -g 1399 cmssgm
> useradd -c "SGM user for alice" -u 18941 -g 1395 alicesgm
```

WARNING: If you have used the example scripts produced in 6.4. in order to create Pool Accounts using a CE as reference machine, the “sgm” users might have already been created on your node.

In addition the following sub-directories have to be created:

/opt/edg/var/info/alice

/opt/edg/var/info/atlas

/opt/edg/var/info/cms

/opt/edg/var/info/lhcb

/opt/edg/var/info/dteam

```
> mkdir /opt/edg/var/info/alice
> mkdir /opt/edg/var/info/atlas
> mkdir /opt/edg/var/info/cms
> mkdir /opt/edg/var/info/lhcb
> mkdir /opt/edg/var/info/dteam
```

15. System parameters tuning

In order to to improve the CE scalability, some system parameters have to be increased.

- Increase maximum number of filehandles.

  ```
  > echo 120000 > /proc/sys/fs/file-max
  ```

- Configure changes to be re-done at boot

  - Create a back-up copy of the file */etc/rc.d/rc.local*

    ```
    > cp -f /etc/rc.d/rc.local /etc/rc.d/rc.local.orig
    ```

  - Add the following lines to the file */etc/rc.d/rc.local* in order to let the configuration survive reboots:

    ```
    # Increase some system parameters to improve LCG WN scalability
    if [ -f /proc/sys/fs/file-max ]; then
       echo 120000 > /proc/sys/fs/file-max
    fi
    ```

16. Cron Table Configuration

Edit the crontab to add scheduled services

```
> crontab -e
```

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

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

If needed, an automatic rotation of the log file can be obtained editing the file */etc/logrotate.d/edg-fetch* (create it if it is needed)
An example of configuration of the file */etc/logrotate.d/edg-fetch* for a monthly log rotation follows:

```
/var/log/edg-fetch-crl-cron.log  {
    compress
    monthly
    rotate 12
    missingok
    ifempty
    create
}
```

17. 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.

17.1. SOFTWARE INSTALLATION

The list ”CE_manual_install_rpm”, which has been described in 2., contains the latest version of the CE-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

CE_upgrade_rpm

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

CE_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 to update the rpms within the directory *CE_UPGRADE*, by issuing the command

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

A further list of rpms is provided, namely

CE_upgrade_forced_rpm

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

CE_upgrade_wget

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

CE_UPGRADE_FORCED

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

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

Finally the list of rpms *CE_remove_rpm*

is provided which contains those packages which need to be removed; this is particularly likely to be necessary with CA-related rpms.
The rpm command to remove the listed rpms is

> *rpm -e ‘cat CE_remove_rpm‘*

17.2. INSTALLING THE UP-TO-DATE CERTIFICATION AUTHORITY RPMS

As the list and structure of 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 LCG release procedure. You should consult the page

http://markusw.home.cern.ch/markusw/lcg2CAlist.html

in order to ascertain what the version number of the latest set of CA rpms is. At time of writing it is 1.6, so the full set of rpms can be downloaded with the following command;

```
> wget -l1 -nd -r http://grid-deployment.web.cern.ch/grid-deployment/download/RpmDir/security/index_r1.6.html
```

These rpms should then be installed in the usual way;

```
> rpm -Uvh *.rpm
```

Changes and updates to these settings will be announced on the LCG-Rollout mailing list.

17.3. UPGRADE CONFIGURATION

No configuration change is needed to upgrade your Computing Element from the previous tag to the current one.

APPENDIX 1: EXAMPLE OF GLOBUS INITIALIZATION OUTPUT

The following output produced by the script *globus-initialization.sh* can be considered *normal* on a Computing Element and it does not indicate any harmful issue:

```
creating globus-sh-tools-vars.sh
creating globus-script-initializer
creating Globus::Core::Paths
checking globus-hostname
Done

Creating...
        /opt/globus/etc/grid-info.conf
Done

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/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...
Determining system information...
Creating job manager configuration file...
Done
Setting up fork gram reporter in MDS
-----------------------------------------
Done

Setting up pbs gram reporter in MDS
----------------------------------------
loading cache /dev/null
checking for qstat... /usr/bin/qstat
updating cache /dev/null
creating ./config.status
creating /opt/globus/libexec/globus-script-pbs-queue
Done

Setting up condor gram reporter in MDS
----------------------------------------
configure: error: Cannot locate condor_q
Error locating condor commands, aborting!
loading cache /dev/null
checking for condor_q... no
Setting up lsf gram reporter in MDS
----------------------------------------
configure: error: Cannot locate lsload
Error locating LSF commands, aborting!
loading cache /dev/null
checking for lsload... no
loading cache ./config.cache
checking for mpirun... /usr/bin/mpirun
updating cache ./config.cache
creating ./config.status
creating fork.pm
loading cache /dev/null
checking for mpirun... /usr/bin/mpirun
checking for qdel... /usr/bin/qdel
checking for qstat... /usr/bin/qstat
checking for qsub... /usr/bin/qsub
checking for ssh... /usr/bin/ssh
updating cache /dev/null
creating ./config.status
creating /opt/globus/lib/perl/Globus/GRAM/JobManager/pbs.pm
configure: error: Cannot locate condor_submit
Error locating condor commands, aborting!
loading cache /dev/null
checking for condor_submit... no
configure: error: Cannot locate bsub
Error locating LSF commands, aborting!
loading cache /dev/null
checking for mpirun... /usr/bin/mpirun
checking for bsub... no
loading cache /dev/null
checking for mpirun... /usr/bin/mpirun
checking for qdel... /usr/bin/qdel
checking for qmsg... /usr/bin/qmsg
checking for qstat... /usr/bin/qstat
checking for qsub... /usr/bin/qsub
checking for ssh... /usr/bin/ssh
updating cache /dev/null
creating ./config.status
creating /opt/globus/lib/perl/Globus/GRAM/JobManager/lcgpbs.pm
configure: error: Cannot locate bsub
Error locating LSF commands, aborting!
loading cache /dev/null
checking for mpirun... /usr/bin/mpirun
checking for bsub... no
configure: error: Cannot locate condor_submit
Error locating condor commands, aborting!
loading cache /dev/null
checking for condor_submit... no
Setting up lcgpbs gram reporter in MDS
----------------------------------------
loading cache /dev/null
checking for qstat... /usr/bin/qstat
updating cache /dev/null
creating ./config.status
creating /opt/globus/libexec/globus-script-lcgpbs-queue
Done

Setting up lcglsf gram reporter in MDS
----------------------------------------
configure: error: Cannot locate lsload
Error locating LSF commands, aborting!
loading cache /dev/null
checking for lsload... no
Setting up lcgcondor gram reporter in MDS
----------------------------------------
configure: error: Cannot locate condor_q
Error locating lcgcondor commands, aborting!
loading cache /dev/null
checking for condor_q... no
loading cache ./config.cache
creating ./config.status
creating grid-cert-request-config
creating grid-security-config
```

CHANGE HISTORY

Table 1: Change History

version	*date*	*description*
v1.0	06/Apr/04	First Release
v1.2.0	26/Apr/04	12: section inserted dealing with GridICE agent configuration.Upgrade section changed accordingly. Numbering changed consequently. 13: Section inserted dealing with Experiment Software Mangement tool (SGM) configuration. Upgrade section changed accordingly. Numbering shifted up consequently.
v1.3.0	28/Apr/04	12: reference to external document inserted.
v3.0.0	24/May/04	Document re-styling 4.2.: lib /usr/lib/qt-3.0.5/lib configured. 8.2.: reference to regional GIIS removed 8.4.: paragraph name changed from ”Create information provider configuration files” to ”Create static information provider configuration” 8.4.:file /opt/edg/var/etc/ce-static.ldif changed 8.4.:file /opt/edg/libexec/ceinfo-wrapper.sh changed 10.:Replica Manager configuration removed. Refer to [3]
v3.1.0	3/Aug/04	2.,17.: rpms dealing with CAs no more included in rpm lists.