LHC COMPUTING GRID

LCG - STORAGE ELEMENT MANUAL INSTALLATION AND CONFIGURATION

Document identifier:	**LCG-GIS-MI-SE**
EDMS id:	**none**
Version:	**v2.5.0**
Date:	**August 20, 2004**
Section:	**LCG Grid Infrastructure Support**
Document status:	**ACTIVE**
Author(s):	Antonio Retico, Alessandro Usai, Oliver Keeble, Guillermo Diez-Andino
File:	**SE**

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

CONTENTS

REFERENCES

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

[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 SE 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/SE-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 SE software you need:

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

Though it is not mandatory, it is strongly recommended to create, during Red Hat installation, a separate ext3 partition for the storage space resource. The configuration steps listed below, when needed, identify this storage space with the mount point:

```
/storage
```

2. STORAGE ELEMENT SOFTWARE INSTALLATION

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

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

Two different methods are supported for SE software installation

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

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

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

2.2. SE 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/SE-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:

SE.pdf: this guide

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

SE_manual_install_forced_rpm: plain full list of those rpms needed to install the SE 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

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

SE_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 *SE_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 **SE_tools.tar.gz**. So they can be easily downloaded and used on target machines.

The provided *SE_manual_install_rpm* includes:

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

  > *gunzip SE_tools.tar.gz*
  > *tar xvf SE_tools.tar*

- give execution permissions and run the script *SE_manual_install_wget*, which will then fetch all the rpms required into the subdirectories *SE_INSTALL* and *SE_INSTALL_FORCED*

  > *cd SE_tools*
  > *chmod u+x SE_manual_install_wget*
  > *./SE_manual_install_wget*

Install the downloaded rpms

- In the directory **SE_INSTALL**, where the rpms listed in *SE_manual_install_rpm* have been downloaded, run the command

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

- In the directory **SE_INSTALL_FORCED** where the rpms listed in *SE_manual_install_forced_rpm* have been downloaded, run the command

  ```
  > rpm -Uvh --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 *SE_manual_install_rpm* list.

Should you prefer to do an upgrade of the SE 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 SE 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 -Uvh *.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 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"
```

4.5. 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. POOL ACCOUNTS CREATION FOR SE

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 5.1. 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.

5.1. LCG GROUPS AND USERS IDS

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

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
```

5.2. GLOBUS "GRIDMAPDIR" PATCH

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

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

- For every Pool user name defined in 5.1., 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
```

5.3. POOL ACCOUNTS CREATION ON SE

User and Group Creation

Create groups and users as they have been defined in 5..
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 Storage Element follows:

```
for each group defined in 5.1.1.
   create the group on the SE
for each user defined in 5.1.2.
   create the user on the SE with the proper group ID
```

An implementation example follows, which uses two scripts provided in 5.4., in order to get the group and user lists from another node (e.g. another SE or a 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 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 node the script *CE_createUGlists.sh* (see 5.4.) 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 SE you are installing the script *createPoolAccounts.sh* (see 5.4.)

  ```
  > ./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 SE

  ```
  > 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])

5.4. 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
```

6. STORAGE DIRECTORIES CREATION

- Create the storage resource directory

  The parent storage directory to which all the configuration steps refer is */storage*.
  If it has not been already created as a separate partition (see 1.), create it

  ```
  > mkdir /storage
  ```

- Create a storage sub-directory for each VO

```
> mkdir /storage/alice
> chown :alice /storage/alice
> chmod g+rwx /storage/alice

> mkdir /storage/atlas
> chown :atlas /storage/atlas
> chmod g+rwx /storage/atlas

> mkdir /storage/cms
> chown :cms /storage/cms
> chmod g+rwx /storage/cms

> mkdir /storage/lhcb
> chown :lhcb /storage/lhcb
> chmod g+rwx /storage/lhcb

> mkdir /storage/dteam
> chown :dteam /storage/dteam
> chmod g+rwx /storage/dteam
```

7. GRIS CONFIGURATION

7.1. 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)

7.2. 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-site-name>
reghn=<CE-hostname>

[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
```

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

your-site-name: Name of your site within the Information System (as it is defined in */etc.globus.conf* on the CE Installation Guide [1])

If you have doubts on how to configure the file above 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


[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
```

7.3. CREATE DIRECTORIES

create the directories:

/opt/edg/var/info/

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

7.4. LCG INFO PROVIDER STATIC CONFIGURATION

- Create the file

  /opt/lcg/var/lcg-info-generic.conf

  WARNING: be careful not to use blank spaces at the end of each line.
  The layout is:

```
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/GlueSE.template
template=/opt/lcg/etc/GlueService.template
```

```
dynamic_script=/opt/lcg/libexec/lcg-info-dynamic-classic <SE-hostname> <VO1-name>:<VO1-storage-area> <VO2-name>:<VO2-storage-area> ...

GlueSEName: <your-site-name>:disk
GlueSEPort: 2811
GlueInformationServiceURL: ldap://<SE-hostname>:2135/Mds-Vo-name=local,o=grid
GlueSLName: <your-site-name>
GlueSLArchitectureType: disk
GlueSAPolicyFileLifeTime: permanent
GlueSAPolicyMaxFileSize: 100000
GlueSAPolicyMinFileSize: 00
GlueSAPolicyMaxData: 00
GlueSAPolicyMaxNumFiles: 00
GlueSAPolicyMaxPinDuration: 10

dn: GlueSEUniqueID=<SE-hostname>,Mds-vo-name=local,o=grid
GlueForeignKey: GlueSLUniqueID=<SE-hostname>

dn: GlueSLUniqueID=<SE-hostname>,Mds-vo-name=local,o=grid
GlueForeignKey: GlueSEUniqueID=<SE-hostname>

dn: GlueSEAccessProtocolType=rfio,GlueSEUniqueID=<SE-hostname>,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=<SE-hostname>
GlueSEAccessProtocolPort: 5001

dn: GlueSEAccessProtocolType=gsiftp,GlueSEUniqueID=<SE-hostname>,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=<SE-hostname>
GlueSEAccessProtocolPort: 2811

dn: GlueSARoot=<VO1-name>:<VO1-group-name>,GlueSEUniqueID=<SE-hostname>,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=<SE-hostname>
GlueSAAccessControlBaseRule: <VO1-group-name>

dn: GlueSARoot=<VO2-name>:<VO2-group-name>,GlueSEUniqueID=<SE-hostname>,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=<SE-hostname>
GlueSAAccessControlBaseRule: <VO2-groupname>

...
```

- Configure appropriately the variables:

  <**SE-hostname**> : hostname of the Storage Element

  <**your-site-name**> : Name of your site within the Information System (as it is defined in */etc.globus.conf* on the CE Installation Guide [1])

  <**VOn-name**> : Name of the n-th supported VOs

  <**VOn-group-name**> : Group name Name of the n-th supported VOs as it is defined in 5.1.1.

  <**VOn-storage-area**> : VO Private storage area as defined in 6.

  A production example of the file */opt/lcg/var/lcg-info-generic.conf* 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/GlueSE.template
template=/opt/lcg/etc/GlueService.template

dynamic_script=/opt/lcg/libexec/lcg-info-dynamic-classic adc0033.cern.ch alice:/storage/alice atlas:/storage/atlas dteam:/storage/dteam

GlueSEUniqueID: adc0033.cern.ch
GlueSEName: CERN-TEST-LCG2:disk
```

```
GlueSEPort: 2811
GlueInformationServiceURL: ldap://adc0033.cern.ch:2135/Mds-Vo-name=local,o=grid
GlueSLUniqueID: adc0033.cern.ch
GlueSLName: CERN-TEST-LCG2
GlueSLArchitectureType: disk
GlueSAPolicyFileLifeTime: permanent
GlueSAPolicyMaxFileSize: 2000000
GlueSAPolicyMinFileSize: 00
GlueSAPolicyMaxData: 00
GlueSAPolicyMaxNumFiles: 00
GlueSAPolicyMaxPinDuration: 10

dn: GlueSEUniqueID=adc0033.cern.ch,Mds-vo-name=local,o=grid
GlueForeignKey: GlueSLUniqueID=adc0033.cern.ch

dn: GlueSLUniqueID=adc0033.cern.ch,Mds-vo-name=local,o=grid
GlueForeignKey: GlueSEUniqueID=adc0033.cern.ch

dn: GlueSEAccessProtocolType=rfio,GlueSEUniqueID=adc0033.cern.ch,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=adc0033.cern.ch
GlueSEAccessProtocolPort: 5001

dn: GlueSEAccessProtocolType=gsiftp,GlueSEUniqueID=adc0033.cern.ch,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=adc0033.cern.ch
GlueSEAccessProtocolPort: 2811

dn: GlueSARoot=alice:alice,GlueSEUniqueID=adc0033.cern.ch,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=adc0033.cern.ch
GlueSAAccessControlBaseRule: alice

dn: GlueSARoot=atlas:atlas,GlueSEUniqueID=adc0033.cern.ch,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=adc0033.cern.ch
GlueSAAccessControlBaseRule: atlas

dn: GlueSARoot=cms:cms,GlueSEUniqueID=adc0033.cern.ch,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=adc0033.cern.ch
GlueSAAccessControlBaseRule: cms

dn: GlueSARoot=dteam:dteam,GlueSEUniqueID=adc0033.cern.ch,Mds-Vo-name=local,o=grid
GlueChunkKey: GlueSEUniqueID=adc0033.cern.ch
GlueSAAccessControlBaseRule: dteam
```

- Run the script

```
> /opt/lcg/sbin/lcg-info-generic-config /opt/lcg/var/lcg-info-generic.conf /opt/lcg/etc/GlueSE.template
```

7.5. EDG INFO PROVIDER CONFIGURATION

Create the file

/opt/edg/var/info/info-provider.conf

with the following content:

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

7.6. 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
  ```

7.7. 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
```

7.8. START THE INFORMATION SERVICE

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

7.9. TEST THE GRIS AND GIIS

Run the following *ldap* queries:

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

The output should show both the static values you configured in 7.4. (e.g *GlueSAPolicyMaxFileSize*) and dynamic values (e.g. *GlueSAStateUsedSpace*)

8. "RFIO" AND "GLOBUS GRIDFTP" SERVICES

8.1. CONFIGURE PORTS FOR RFIO DAEMON

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

```
rfio  5001/tcp
rfio  5001/udp
```

8.2. START THE GLOBUS GRIDFTP SERVER

Run the following commands

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

8.3. FIREWALL CONFIGURATION

If you are using a firewall consider that the following tcp/udp ports need to be opened:

- rfio

  ```
  udp 5001
  tcp 5001
  ```

- globus-gridftp

  ```
  tcp 2811
  ```

  For example, if you are using ipchains for firewall configuration, you could open the listed ports to your subnet.
  To do so, add, **before the first REJECT line**, the lines

  ```
  -A input -p udp -s <your-subnet>/<type> -d 0/0 5001 -j ACCEPT
  -A input -p tcp -s <your-subnet>/<type> -d 0/0 5001 -j ACCEPT
  -A input -p tcp -s <your-subnet>/<type> -d 0/0 2811 -j ACCEPT
  ```

into the file

/etc/sysconfig/ipchains

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
```

9. 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 9.1. you can find the configuration of agents while section 9.2. deals with the configuration details for the GridICE collector server.
If you are doing a defaul LCG-2 configuration you should apply section 9.2. 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

9.1. 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 9.1..

- start the GridICE agent

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

9.2. 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
```

10. RGMA CLIENT INSTALLATION

In what follows the installation and configuration of a standard RGMA producer is described.
The producer component takes care of publishing information following the general RGMA architecture and as such should be installed in all those machines which need to be monitored (RGMA can in principle be used as an information system as well, however for the purpose of this document this will not be considered).

10.1. RGMA CONFIGURATION

ld.so.conf

Add to the file */etc/ld.so.conf* the following lines

```
/opt/edg/lib
/opt/gcc-3.2.2/lib
```

Environmental Variables

Define the following environmental variables

```
> export RGMA_PROPS=/opt/edg/var/edg-rgma
> export EDG_LOCATION_VAR=/opt/edg/var
> export EDG_TMP=/tmp
> export EDG_LOCATION=/opt/edg
```

Also MAKE SURE that */opt/edg/bin is in the PATH*

File /etc/sysconfig/edg

Create and edit the file

/etc/sysconfig/edg

as

```
# 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


# Host key (X509_USER_KEY) for services which don't have their own.
# Usual value: /etc/grid-security/hostkey.pem


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


# Location of the grid map directory for pooled accounts (GRIDMAPDIR).
# Usual value: /etc/grid-security/gridmapdir
```

Directory edg-rgma

Create the directory (if it does not exist already)

/opt/edg/var/edg-rgma

```
> mkdir -p /opt/edg/var/edg-rgma
```

Notice that this step is also performed by *edg-rgma-config* (See 10.2.)

RGMA Default Values

Create and edit the file

/opt/edg/var/edg-rgma/rgma-defaults

```
RGMA_PROPS=/opt/edg/var/edg-rgma
CATALINA_HOME=/var/tomcat4
JUNIT_JAR=_undefined_
```

```
LOG4J_JAR=/usr/share/java/log4j.jar
MYSQL_JAR=/usr/share/java/mm.mysql-2.0.14.jar
NETLOGGER_JAR=/usr/share/java/netlogger.jar
PREVAYLER_JAR=/usr/share/java/prevayler.jar
EDG_TRUST_SERVER_JAR=/opt/edg/share/java/edg-java-security-trustmanager.jar
EDG_AUTHORIZ_SERVER_JAR=/opt/edg/share/java/edg-java-security-authorization.jar
EDG_TRUST_CLIENT_JAR=/opt/edg/share/java/edg-java-security-trustmanager-client.jar
BOUNCYCASTLE_JAR=/usr/share/java/bcprov-jdk14.jar
JAVACC_HOME=_undefined_
TEXDOCLET_JAR=_undefined_
INFO_JAR=/opt/edg/share/java/info.jar
SQLUTIL_JAR=/opt/edg/share/java/sqlutil.jar
rgma_c_API=_undefined_
RGMA_RPM_VERSION=3.4.31
GCC_VERSION=
SERLOC=<MON_NODE_NAME>
REGLOC=<REGISTRY_NODE_NAME>
SECURE=n
RegistryHosts="<REGISTRY_NODE_NAME>"
RGMA_ROOT=/opt
```

where <MON_NODE_NAME> is the MON BOX machine name and <REGISTRY_NODE_NAME> is the REGISTRY machine name, whose default value is *lcgic01.gridpp.rl.ac.uk* in the current lcg version.

10.2. RGMA CONFIGURATION CREATION

Launch the following script to configure rgma

```
> /opt/edg/sbin/edg-rgma-config -q
```

11. EXPERIMENT SOFTWARE MANAGER CONFIGURATION

There is no real need to have Experiment Software Manager accounts on the SE.
The only problem is the way the grid-mapfile is automatically generated.
The ESM could be mapped to any account belonging to the VO group.
However, if such account does not exist, than the ESM cannot perform any operation on the SE.

So, eventhough the only real requirement is to have the ESM certificate subject mapped to any account belonging to that VO, because of the way the *grid-mapfile* is generated, 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 5.4. 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
```

12. 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:

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
32 1,7,13,19 * * * /opt/edg/sbin/edg-mkgridmap --output=/etc/grid-security/grid-mapfile --safe
56 3,9,15,21 * * *  /opt/edg/etc/cron/edg-fetch-crl-cron >> /var/log/edg-fetch-crl-cron.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
}
```

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 ”SE_manual_install_rpm”, which has been described in 2., contains the latest version of the SE-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

SE_upgrade_rpm

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

SE_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 *SE_UPGRADE*, by issuing the command

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

A further list of rpms is provided, namely

SE_upgrade_forced_rpm

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

SE_upgrade_wget

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

SE_UPGRADE_FORCED

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

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

Finally the list of rpms *SE_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 SE_remove_rpm'

13.2. 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.

13.3. UPGRADE CONFIGURATION

- Check section 10. for the RGMA client installation.
- Edit the file */opt/edg/var/etc/edg-fmon-agent.conf*
  and replace the line

  ```
  CommandLine $(EDG_LOCATION)/libexec/edg-fmon-sensor-systemCheck
  ```

  with the line

```
CommandLine $(EDG_LOCATION)/libexec/gridice-edg-fmon-sensor-systemCheck
```

Beware not to use spaces at the beginning of the line but to use three <TAB> characters.

Then re-start the GridICE agent

```
> service edg-fmon-agent restart
```

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
----------------------------------------
configure: error: Cannot locate qstat
Error locating pbs commands, aborting!
loading cache /dev/null
checking for qstat... no
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
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
Error locating PBS commands, aborting!
loading cache /dev/null
checking for mpirun... no
```

```
checking for qdel... no
configure: error: Cannot locate condor_submit
Error locating condor commands, aborting!
loading cache /dev/null
checking for condor_submit... no
configure: warning: Cannot locate mpirun
configure: error: Cannot locate bsub
Error locating LSF commands, aborting!
loading cache /dev/null
checking for mpirun... no
checking for bsub... 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*
v1.0.0	06/Apr/04	First Release
v1.1.0	27/Apr/04	9: section inserted dealing with GridICE agent configuration.Upgrade section changed accordingly. Numbering changed consequently. 10: Section inserted dealing with Experiment Software Mangement tool (SGM) configuration. Upgrade section changed accordingly. Numbering shifted up consequently.
v2.0.0	24/Jun/04	Document re-styling Section “Edg User creation” removed (not needed on SE) Section “Experiment Software Manager Configuration” removed (not needed on SE)
v2.1.0	12/Jul/04	11.: Section “Experiment Software Manager Configuration” re-inserted. Numbering shifted up consequently.
v2.2.0	12/Jul/04	13.3.: changes to be done in file */opt/lcg/var/lcg-info-generic.conf* added to the list of instructions for node upgrade .
v2.3.0	3/Aug/04	2.,13.: rpms dealing with CAs no more included in rpm lists.
v2.4.0	5/Aug/04	10.: rgma client installation included 9.1.: file /opt/edg/var/etc/edg-fmon-agent.conf changed (see 13.3. for details).
v2.4.1	10/Aug/04	10.: RGMA client configuration changed
v2.4.2	12/Aug/04	10.: configuration of file /opt/edg/var/edg-rgma/rgma-defaults changed
v2.5.0	20/Aug/04	7.4.: configuration of file */opt/lcg/var/lcg-info-generic.conf* changed