



LHC COMPUTING GRID

LCG - WORKER NODE MANUAL INSTALLATION AND CONFIGURATION

<i>Document identifier:</i>	LCG-GIS-MI-WN
<i>EDMS id:</i>	none
<i>Version:</i>	v3.0.0
<i>Date:</i>	July 9, 2004
<i>Section:</i>	LCG Grid Infrastructure Support
<i>Document status:</i>	ACTIVE
<i>Author(s):</i>	Antonio Retico, Alessandro Usai, Oliver Keeble
<i>File:</i>	WN

Abstract: Manual installation and set-up of a LCG2 Worker Node



CONTENTS

1. PREREQUISITES FOR INSTALLATION	6
2. WORKER NODE SOFTWARE INSTALLATION	7
2.1. WN INSTALLATION USING <i>apt-get</i>	7
2.2. WN INSTALLATION USING <i>rpm</i>	7
3. NTP CLIENT INSTALLATION & CONFIGURATION	10
3.1. NTP SOFTWARE INSTALLATION	10
3.2. NTP CONFIGURATION	10
3.3. EXAMPLE OF NTP CLIENT INSTALLATION AND CONFIGURATION	11
4. LCG COMMON CONFIGURATION	13
4.1. CONFIGURE DYNAMIC LINKER RUN-TIME BINDINGS	13
4.2. CREATE TOP-LEVEL EDG CONFIGURATION FILE	13
4.3. CREATE TOP-LEVEL LCG CONFIGURATON FILE	13
4.4. SET-UP LCG ENVIRONMENT VARIABLES	14
4.5. CREATE TOP-LEVEL GLOBUS CONFIGURATION FILE	15
4.6. CONFIGURE GLOBUS	15
5. REPLICA MANAGER CONFIGURATION	17
6. POOL ACCOUNTS CREATION FOR WN	20
6.1. LCG GROUPS AND USERS IDS	20
6.2. POOL ACCOUNTS CREATION ON WN	21
6.3. EXAMPLE SCRIPTS FOR USER AND GROUP HANDLING	24
7. WORKLOAD MANAGER CONFIGURATION	28
8. WN SPECIFIC CONFIGURATION	29
8.1. EDITING OF /ETC/SSH/SSH_CONFIG	29
8.2. SYSTEM PARAMETERS TUNING	29
9. LRMS CLIENT CONFIGURATION	30
9.1. PBS INSTALLATION AND CONFIGURATION ON WN	30
9.2. PBS SOFTWARE CONFIGURATION	31
9.3. LSF INSTALLATION AND CONFIGURATION ON WN	33
9.4. CONDOR INSTALLATION AND CONFIGURATION ON WN	33
10. EXPERIMENT SOFTWARE MANAGER CONFIGURATION	34
11. CRON TABLE CONFIGURATION	35
12. UPGRADE FROM THE PREVIOUS VERSION	36
12.1. SOFTWARE INSTALLATION	36



12.2. UPGRADE CONFIGURATION.....	37
----------------------------------	----



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] E. DataGrid. Grid-mapfile, 2004. authorization team <sec-grid@infn.it>.
- [3] F. Donno and H. Stockinger. Lcg-manual-installation, 2004. Document identifier: CERN-LCG-GDEIS-412774.
- [4] L. Poncet. Cvs user guide, 2004. <http://grid-deployment.web.cern.ch/grid-deployment/cgi-bin/index.cgi?va\%r=documentation>
- [5] A. Retico. Lcg manual installation guides, 2004. <http://www.cern.ch/grid-deployment/gis/release-docs/MIG-index.html>.
- [6] 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>
- [7] 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>
- [8] A. Retico and A. Usai. Lcg ui manual software installation and configuration, 2004. <http://www.cern.ch/grid-deployment/gis/release-docs/UI-index.html>
- [9] 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>
- [10] A. Retico, A. Usai, and O. Keeble. Lcg se manual software installation and configuration, 2004. <http://www.cern.ch/grid-deployment/gis/release-docs/SE-index.html>



ABSTRACT

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

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



2. WORKER NODE SOFTWARE INSTALLATION

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

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

Two different methods are supported for WN software installation

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

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

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

2.2. WN 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/WN-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:

WN.pdf: this guide

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



WN_manual_install_forced_rpm: plain full list of those rpms needed to install the WN 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

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

WN_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 *WN_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 **WN_tools.tar.gz**. So they can be easily downloaded and used on target machines.

The provided *WN_manual_install_rpm* includes:

- all the packages which are needed in order to have a production Worker Node.
- all the general purpose objects (i.e. shared libraries or common applications) which the WN 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 described 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 **WN_tools.tar.gz**.
- extract the archived files.

```
> gunzip WN_tools.tar.gz
> tar xvf WN_tools.tar
```

- give execution permissions and run the script *WN_manual_install_wget*, which will then fetch all the rpms required into the subdirectories *WN_INSTALL* and *WN_INSTALL_FORCED*

```
> cd WN_tools
> chmod u+x WN_manual_install_wget
> ./WN_manual_install_wget
```




Install the downloaded rpms

- In the directory **WN_INSTALL**, where the rpms listed in *WN_manual_install_rpm* have been downloaded, run the command

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

- In the directory **WN_INSTALL_FORCED** where the rpms listed in *WN_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 *WN_manual_install_rpm* list.

Should you prefer to do an upgrade of the WN 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 WN installation guide.

This procedure, when available, refers to changes occurring between the current version and the previous one. This also means that the procedure is not applicable if the software at your site is more than two tags old.



3. NTP CLIENT INSTALLATION & CONFIGURATION

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

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

3.1. NTP SOFTWARE INSTALLATION

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

- ntp
- libcap
- libcap-devel

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

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

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

3.2. NTP CONFIGURATION

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

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

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

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



```
137.138.16.69
137.138.17.69
```

- If you are running a kernel firewall, you will have to allow inbound communication on the NTP port. If you are using ipchains, you can do this by adding the following to */etc/sysconfig/ipchains*

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

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

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

- Activate the ntpd service with the following commands:

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

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

3.3. EXAMPLE OF NTP CLIENT INSTALLATION AND CONFIGURATION

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

- install NTP

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

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

the lines

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

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



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

the lines

```
137.138.16.69
137.138.17.69
```

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

- Edit */etc/sysconfig/ipchains* adding

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

then reload the firewall

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

- Activate the ntpd server

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

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

```
> ntpq -p
```



4. LCG COMMON CONFIGURATION

4.1. 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
/opt/lcg/lib
```

- Run the command:

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

(this command produces a huge amount of output)

4.2. 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
```

4.3. 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
```

4.4. SET-UP LCG ENVIRONMENT VARIABLES

Create and edit the files `/etc/profile.d/lcgen.sh` and `/etc/profile.d/lcgen.csh` as follows:

- Layout of `lcgen.sh`

```
#!/bin/sh
export LCG_GFAL_INFOSYS=<info-provider hostname>:2170
export VO_ALICE_SW_DIR=<alice-software-directory>
export VO_ATLAS_SW_DIR=<atlas-software-directory>
export VO_CMS_SW_DIR=<cms-software-directory>
export VO_LHCB_SW_DIR=<lhcb-software-directory>
export VO_DTEAM_SW_DIR=<dteam-software-directory>
```

- Layout of `lcgen.csh`

```
#!/bin/csh
setenv LCG_GFAL_INFOSYS <info-provider hostname>:2170
setenv VO_ALICE_SW_DIR <alice-software-directory>
setenv VO_ATLAS_SW_DIR <atlas-software-directory>
setenv VO_CMS_SW_DIR <cms-software-directory>
setenv VO_LHCB_SW_DIR <lhcb-software-directory>
setenv VO_DTEAM_SW_DIR <dteam-software-directory>
```

The meaning of each parameter is:

<info-provider hostname>: hostname (comprehensive of domain name) of the MDS information server. It should be a BDII node.
If your site owns a BDII use its hostname. Else, if you have a reference site, ask them for indications. Otherwise, send a message to the "LCG-ROLLOUT@cclreclsv.RL.AC.UK" mailing list.
The same considerations are valid also for **<proxy-server hostname>**, hostname of the authentication proxy.



<**xxx-software-directory**>: Area on the WN for the installation of the experiment software.

If on your WNs you have predefined shared areas where VO managers can pre-install software, then these variables should point to these areas. If you do not have shared areas and each job must install the software, then these variables should contain a dot (.)

Production Example:

- `lcgen.sh`

```
#!/bin/sh
export LCG_GFAL_INFOSYS=lxn1178.cern.ch:2170
export VO_ALICE_SW_DIR=.
export VO_ATLAS_SW_DIR=.
export VO_CMS_SW_DIR=.
export VO_LHCB_SW_DIR=.
export VO_DTEAM_SW_DIR=.
```

- `lcgen.csh`

```
#!/bin/csh
setenv LCG_GFAL_INFOSYS lxn1178.cern.ch:2170
setenv VO_ALICE_SW_DIR .
setenv VO_ATLAS_SW_DIR .
setenv VO_CMS_SW_DIR .
setenv VO_LHCB_SW_DIR .
setenv VO_DTEAM_SW_DIR .
```

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
```

4.6. CONFIGURE GLOBUS

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

```
[common]
GLOBUS_LOCATION=/opt/globus
globus_flavor_name=gcc32dbg
```



[mds]

- Launch Globus Inizialization Script

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

This Globus script is meant to be launched on the whole series of LCG2 nodes, as a general configuration step. So that it may perform some configuration actions which are not specifically related to a WN. Because of this it can yield warning messages as well as some error ones. An example of warning messages to be considered "normal" in a WN configuration 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
```




5. REPLICATOR MANAGER CONFIGURATION

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

Before running the script, a configuration file (the default one installed by the rpm is

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

should be manually edited and used to input the script

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

that will automatically write the local configuration parameters in the *edg-replica-manager.conf* file

So two steps have to be done:

- Edit the local configuration file

The file */opt/edg/etc/edg-replica-manager/edg-replica-manager.conf.values*, as above said, is the default one and it could be used to feed the configuration script, but, since it is automatically created by the *edg-replica-manager-config* rpm it would be preferable if it were given a different name (e.g. *edg-replica-manager.conf.values_local*) so that possible upgrades of the rpm do not overwrite the already existing file.

The above mentioned file *edg-replica-manager.conf.values_local* should look like

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

the meaning of each parameter is:

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

LOCALDOMAIN: your domain name (site-dependent)

DEFAULT.SE: (site-dependent)

DEFAULT.CE: (site-dependent)

RLS.MODE: Replica Catalog type: The implementation of the replica catalog interface. Available values are:

LrcOnly: a single LRC per VO

WithRli: a full RLS deployed

INFOSERVICE: The implementation of the Information service interface. Available are:

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



RGMA: RGMA is contacted for all information service requests

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

STUBFILE: used if INFOSERVICE is set to *Stub*

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

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

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

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

empty: for 2.95

_gcc3_2_2: for 3.2 (this is the correct choice for lcg2)

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

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

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

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

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

Example of layout for file

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

```
@EDG.LOCATION@|/opt/edg|location of edg the directory
@LOCALDOMAIN@|cern.ch|the local domain
@DEFAULT.SE@|adc0033.cern.ch|the host of the close SE
@DEFAULT.CE@|adc0029.cern.ch|the host of the close CE
@INFOSERVICE@|MDS|The info provider to use. It can be Stub, MDS or RGMA
@RLS.MODE@|LrcOnly|The mode the RLS should be run in. LrcOnly or WithRli
@STUBFILE@||The properties file for the static file 'info service'
@MDS.HOST@|adc0031.cern.ch|The host of the MDS info provider
@MDS.PORT@|2170|The port of the MDS info provider
@ROS.FAILURE@|false|Fail if no ROS is available
@CONF.GCC@|_gcc3_2_2|The gcc suffix as used on the build box (empty for 2.95, _gcc3_2_2 for 3.2.)
@IGNORE.PREFIX@|true|If true, the lfn and guid prefix will not be passed to the catalogs
@GRIDFTP.DCAU@|false|Does GridFTP use Data Channel Authentication (DCAU)
@GRIDFTP.STREAMS.SMALL@|1|The default number of stream to use for a small file
@GRIDFTP.STREAMS.BIG@|3|The default number of stream to use for a big file
@GRIDFTP.FILESIZE.THRESHOLD@|100|The Threshold (in MB) above which a file to transfer is considered "big"
```

- Run the configuration script (the `--verbose` flag is optional)

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



You can, eventually, check the file `/opt/edg/var/etc/edg-replica-manager/edg-replica-manager.conf` to have been modified accordingly by the script.



6. POOL ACCOUNTS CREATION FOR WN

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 WN.

Pool group names

The following groups have to be created:

```
alice
atlas
cms
lhcb
dteam
```

Please refer to 6.2. for details concerning to group IDs.

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

Please refer to 6.2. for details concerning to user IDs.

6.2. POOL ACCOUNTS CREATION ON WN

Creation of Groups and Users

Create groups and users as they have been defined in 6.1..

Notice that, for a correct implementation of both of the methods 1 and 2 defined in 6., LCG groups and users must be created with the same GroupID and UserID of the corresponding ones in the CE.

In order to do that a complete list of LCG users and groups has to be produced running a query on the CE (e.g. on the */etc/passwd* file).

A suitable algorithm for user creation on Worker Node follows:



- decide, according to the sharing Method (1 or 2) you are using, whether users have to be created with or without their "home" directory.
- implement the following algorithm (pseudo-code)

```
for each group defined in 6.1.1.  
    pick from the CE the corresponding group ID  
    create the group on the WN with the same group ID  
for each user defined in 6.1.2.  
    pick from the CE the corresponding group and user ID  
    create the user on the WN with the same group and user ID
```

An implementation example follows, which uses two scripts provided in 6.3..
should this implementation conflict with local policies, any customized implementation of the above algorithm is admitted.

- Create a file *VO_names.cfg* as follows:

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

- Launch on the CE node the script *CE_createUGlists.sh* (see 6.3.) feeding it with the file *VO_names.cfg*

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

It will produce two files:

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 WN the script *createPoolAccounts.sh* (see 6.3.)

```
> ./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 WN

```
> ./createGroups
> ./createUsers
```

NFS and firewall (just for Method2)

If you are creating pool accounts according to Method2 (home directories to be shared from CE), the home directory exported by the CE needs to be mounted on the Worker Node.

In order to do so operate the following steps:

- Most likely a local */home* directory on the Worker Node is already mounted. In this case the directory has to be unmounted

```
> umount /home
```

- Create a new mounting point (e.g. */scratch*) for the existing local */home* partition,

```
> mkdir /scratch
```

- Edit the file */etc/fstab* accordingly. For example:
replace line

```
LABEL=/home /home ext3 defaults 1 2
```

with line

```
LABEL=/home /scratch ext3 defaults 1 2
```

append line

```
<CE-hostname>:/home /home nfs rw 0 0
```

- if you are using firewalls at your site, you should be sure that the firewall on your WN machine allow the NFS mounting of part of the CE filesystem.
If you are using "ipchains" for firewall configuration, this can be obtained by editing the file */etc/sysconfig/ipchains* as shown below (create it if it does not exist).

The following configuration of file */etc/sysconfig/ipchains* allows the mounting of CE exported */home* directory and it is required uniquely when the sharing method is chosen.



```
:input ACCEPT
:forward ACCEPT
:output ACCEPT
-A input -s 0/0 -d 0/0 22 -p tcp -y -j ACCEPT
-A input -s 0/0 -d 0/0 -i lo -j ACCEPT
-A input -s <CE-hostname> -d 0/0 -p tcp -y -j ACCEPT
-A input -s <CE-hostname> -d 0/0 -p udp -j ACCEPT
-A input -p tcp -s 0/0 -d 0/0 0:1023 -y -j REJECT
-A input -p tcp -s 0/0 -d 0/0 2049 -y -j REJECT
-A input -p udp -s 0/0 -d 0/0 0:1023 -j REJECT
-A input -p udp -s 0/0 -d 0/0 2049 -j REJECT
-A input -p tcp -s 0/0 -d 0/0 6000:6009 -y -j REJECT
-A input -p tcp -s 0/0 -d 0/0 7100 -y -j REJECT
```

where the <CE-hostname> is the Computing Element host name. Notice also the two lines needs to be placed before the first line ended with "REJECT".

After the configuration you should restart the ipchains daemon

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

- Re-mount all partitions

```
> mount -a
> chkconfig netfs on
> service netfs start
```

6.3. 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 must 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
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. 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/
```



8. WN SPECIFIC CONFIGURATION

8.1. EDITING OF /ETC/SSH/SSH_CONFIG

Once a job is completed on the WN, the files containing the output result of the job (the sand box) need to be copied to the RB node to be then retrieved by the UI. This requires the line

```
HostbasedAuthentication yes
```

to be inserted at the end of the file `/etc/ssh/ssh_config`, just after the line

```
Host *
```

Then the SSH daemon has to be re-started

```
> service sshd restart
```

8.2. SYSTEM PARAMETERS TUNING

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

- Increase maximum number of file handles.

The default allowed maximum number of open file on a RedHat node is only 26213. This number might be too small if users submit file-hungry jobs. So you may want to increase it on your WNs. The value currently in use at CERN is 256000. To set this parameter you can use this command:

```
> echo 256000 > /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 256000 > /proc/sys/fs/file-max
fi
```



9. LRMS CLIENT CONFIGURATION

In order to configure a specific LRMS client on a Worker Node please refer to one of the following sub-sections:

- PBS → 9.1.
- LSF → 9.3.
- Condor → 9.4.

9.1. PBS INSTALLATION AND CONFIGURATION ON WN

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/WN-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:

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

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

The *WN_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 *WN_pbs_install_wget* into an empty working directory.
- run the script *WN_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 *WN_pbs_install_rpm* list.



9.2. PBS SOFTWARE CONFIGURATION

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

Set Server Name

Create the file

```
/var/spool/pbs/server_name
```

containing the default CE hostname

```
> echo <CE hostname> > /var/spool/pbs/server_name
```

Configure ports for PBS services

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

```
pbs_mom          15002/tcp
pbs_remom        15003/tcp
pbs_remom        15003/udp
```

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.



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

- Configure and start the `pbs_mom` daemon

```
> 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
56 3,9,15,21 * * * /opt/edg/etc/cron/edg-fetch-crl-cron >> /var/log/edg-fetch-crl-cron.log 2>&1
56 1,7,13,19 * * * /opt/edg/sbin/edg-pbs-knownhosts
```

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  
}
```

9.3. LSF INSTALLATION AND CONFIGURATION ON WN

TBD

9.4. CONDOR INSTALLATION AND CONFIGURATION ON WN

TBD



10. 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 6.3. in order to create Pool Accounts using a CE as reference machine, the “sgm” users might have already been created on your node.



11. 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
```

```
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  
}
```



12. 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.

12.1. SOFTWARE INSTALLATION

The list "WN_manual_install_rpm", which has been described in 2., contains the latest version of the WN-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

WN_upgrade_rpm

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

WN_upgrade_wget

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

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

WN_remove_rpm

The rpm command to remove the listed rpms is

```
> rpm -e `cat WN_remove_rpm`
```

Next, within the directory *WN_UPGRADE*, the command

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



should be launched.

Finally a further list of rpms is provided, namely

WN_upgrade_forced_rpm

containing those new rpms whose dependencies are not dealt with, therefore requiring the installation to be "forced".

The same "wget" file

WN_upgrade_wget

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

WN_UPGRADE_FORCED

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

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

12.2. UPGRADE CONFIGURATION

- Do system configuration actions specified in section 8.2..
- Change the file */etc/globus.conf* as specified in 4.6. and run the script

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

WARNING: Due to a known bug the script might issue the following output:

```
ERROR: Setup package not found. Looking for:
  name=globus_common_setup,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
  name=globus_mds_common_setup,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
  name=globus_mds_gris_setup,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
  name=globus_gatekeeper_setup,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
  name=globus_gram_job_manager_setup,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
  name=globus_gram_reporter_setup_fork,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
  name=globus_gram_reporter_setup_pbs,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
  name=globus_gram_reporter_setup_condor,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
  name=globus_gram_reporter_setup_lsf,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
```



```
name=globus_gram_job_manager_setup_fork,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
name=globus_gram_job_manager_setup_pbs,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
name=globus_gram_job_manager_setup_condor,setup_name=, and
setup version=
ERROR: Setup package not found. Looking for:
name=globus_gram_job_manager_setup_lsf,setup_name=, and
setup version=
loading cache ./config.cache
creating ./config.status
creating grid-cert-request-config
creating grid-security-config
```

this behaviour is not known to be fatal for the application.

- Fix Replica Manager configuration adding the four new parameters as specified in 5.. Remember to launch the configuration script after changing the configuration file.

Please check the document change history in order to apply configuration fixes possibly occurring between the document version released with the latest tag and 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 Worker Node 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
In: `'/opt/globus/sbin/globus-mds': File exists
/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 state file directory.
Done.
Reading gatekeeper configuration file...
Warning: Host cert file: /etc/grid-security/hostcert.pem not found. Re-run
setup-globus-gram-job-manager after installing host cert file.
Determining system information...
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... (cached) /usr/bin/mpirun
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 ./config.cache
creating ./config.status
creating grid-cert-request-config
creating grid-security-config
```



CHANGE HISTORY

The details in the History concerning version numbers before 2.10 have been removed for readability reasons. They can be retrieved, if needed, from the version 2.9 of this document

Table 1: Change History

<i>version</i>	<i>date</i>	<i>description</i>
v1.0	4/Nov/03	First Release
v1.1	11/Nov/03	
v1.2	19/Nov/03	
v1.3	28/Nov/03	
v1.4	08/Jan/04	
v1.5	09/Jan/04	
v1.6	09/Jan/04	
v1.7	11/Jan/04	
v1.8	14/Jan/04	
v2.0	20/Jan/04	
v2.1	21/Jan/04	
v2.2	21/Jan/04	
v2.3	26/Jan/04	
v2.4	26/Jan/04	
v2.5	27/Jan/04	
v2.6	12/Feb/04	
v2.7	15/Feb/04	
v2.8	24/Feb/04	
v2.9	01/Apr/04	
v2.10	23/Apr/04	4.4.: the configuration of the environment variables now covers the possibility to have the VO software installed on shared areas. 8.2.: section added dealing with system parameters fine tuning. Section numbers shifted up consequently.
v2.11	27/Apr/04	10.: Section inserted dealing with Experiment Software Mangement tool (SGM) configuration. Upgrade section changed accordingly. Numbering shifted up consequently.
v3.0	XX/May/04	Document re-styling 8.: mounting of SE shared area removed. SE should be accessed only with GridFTP. 4.6.: superfluous entries removed in <code>/etc/globus.conf</code> configuration file 5.: GridFTP configuration parameter added 2.: “apt-get” installation method added. Download tools and rpm lists now available via web. Changes to rpm installation method