LHC COMPUTING GRID

LCG - BDII - GENERIC CONFIGURATION REFERENCE

Document identifier:	**LCG-GIS-CR-BDII**
EDMS id:	**none**
Version:	
Date:	**January 16, 2006**
Section:	**LCG Grid Infrastructure Support**
Document status:	**ACTIVE**
Author(s):	LCG Deployment - GIS team;Retico,Antonio;Vidic,Valentin;
File:	**BDII**

Abstract: *Configuration steps done by the YAIM script 'configure_BDII'*

CONTENTS

1. INTRODUCTION

This document lists the manual steps for the installation and configuration of a LCG BDII Node. Furthermore it provides a specification of the YAIM functions used to configure the node with the script-based configuration.

The configuration has been tested on a standard Scientific Linux 3.0 Installation.

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

```
http://www.cern.ch/grid-deployment/gis/lcg-GCR/index.html
```

2. VARIABLES

In order to set-up a BDII node, you need at least the following variables to be correctly configured in the site configuration file (site-info.def):

BATCH_LOG_DIR : Your batch system log directory.

BDII_FCR : Set the URL of the Freedom of Choice for Rescources URL.

BDII_HOST : BDII Hostname.

BDII_HTTP_URL : URL pointing to the BDII configuration file.

BDII_REGIONS : List of node types publishing information on the bdii. For each item listed in the BDII_REGIONS variable you need to create a set of new variables as follows:

> **BDII_<REGION>_URL :** URL of the information producer (e.g.: BDII_CE_URL="URL of the CE information producer", BDII_SE_URL="URL of the SE information producer".

CE_BATCH_SYS : Implementation of site batch system. Available values are “torque”, “lsf”, “pbs”, “condor” etc.

CE_HOST : Computing Element Hostname.

CRON_DIR : Yaim writes all cron jobs to this directory. Change it if you want to turn off Yaim’s management of cron.

GRIDICE_SERVER_HOST : GridIce server host name (usually run on the MON node).

INSTALL_ROOT : Installation root - change if using the re-locatable distribution.

MON_HOST : MON Box Hostname.

MY_DOMAIN : site’s domain name.

SITE_NAME : Your GIIS.

USERS_CONF : Path to the file containing a list of Linux users (pool accounts) to be created. This file should be created by the Site Administrator, which contains a plain list of the users and IDs. An example of this configuration file is given in /opt/lcg/yaim/examples/users.conf.

VOS : List of supported VOs.

3. Create EDG users

Author(s): Retico,Antonio
Email : support-lcg-manual-install@cern.ch

This chapter describes the configuration steps done by the *yaim* function '*config_edgusers*'.

Many of the services running on LCG service nodes are owned by the user *edguser*. The user edguser belongs to the group *edguser* and it has got a home directory in */home*.

The user *edginfo* is required on all the nodes publishing information on the Information System. The user belongs to the group *edginfo* and it has got a home directory in */home*.

No special requirements exists for the ID of the above mentioned users and groups.

The function creates both*edguser* and *edginfo* groups and users.

- group *edguser*: the group is created with group ID 995.
- user *edguser*: the user is created with group ID 995 and its home is */home/edguser*.
- group *edginfo*: the group is created with group ID 999.
- user *edginfo*: the user is created with group ID 999 and its home is */home/edguser*.

3.1. Specification of function: config_edgusers

The function *'config_edgusers'* needs the following variables to be set in the configuration file:

INSTALL_ROOT : Installation root - change if using the re-locatable distribution.

USERS_CONF : Path to the file containing a list of Linux users (pool accounts) to be created. This file should be created by the Site Administrator, which contains a plain list of the users and IDs. An example of this configuration file is given in /opt/lcg/yaim/examples/users.conf.

VOS : List of supported VOs.

The original code of the function can be found in:

```
/opt/lcg/yaim/functions/config_edgusers
```

The code is reproduced also in 6.1..

4. SET-UP BDII

Author(s): Vidic,Valentin
Email : support-lcg-manual-install@cern.ch

This chapter describes the configuration steps done by the *yaim* function ’*config_bdii*’.

This functions is used to configure two types of BDIIs: site BDII and top BDDI.

At each site, a site BDII collects information about all resources present at a site (i.e. data from all GRISes of the site).

A top BDII collects all information coming from site BDIIs and stores them in a permanent database.

BDII is configured in *<INSTALL_ROOT>/bdii/etc/bdii.conf*. For top BDII the configuration is:

```
BDII_PORT_READ=2170
BDII_PORTS_WRITE="2171 2172 2173"
BDII_USER=edguser
BDII_BIND=mds-vo-name=local,o=grid
BDII_PASSWD=<random_password>
BDII_SEARCH_FILTER='*'
BDII_SEARCH_TIMEOUT=30
BDII_BREATHE_TIME=60
BDII_AUTO_UPDATE=yes
BDII_AUTO_MODIFY=no
BDII_DIR=<INSTALL_ROOT>/bdii/
BDII_UPDATE_URL=<BDII_HTTP_URL>
BDII_UPDATE_LDIF=http://
SLAPD=/usr/sbin/slapd
SLAPADD=/usr/sbin/slapadd
```

BDII is setup with the base DN of *mds-vo-name=local,o=grid* and automatic generation of *<INSTALL_ROOT>/bdii/e* *update.conf* by downloading from <BDII_HTTP_URL>. BDII password is generated using *mkpasswd* or from *RANDOM* shell variable.

For local BDII the configuration is:

```
BDII_PORT_READ=2170
BDII_PORTS_WRITE="2171 2172 2173"
BDII_USER=edguser
BDII_BIND=mds-vo-name=<SITE_NAME>,o=grid
BDII_PASSWD=<random_password>
BDII_SEARCH_FILTER='*'
BDII_SEARCH_TIMEOUT=30
BDII_BREATHE_TIME=60
BDII_AUTO_UPDATE=no
BDII_AUTO_MODIFY=no
BDII_DIR=<INSTALL_ROOT>/bdii/
BDII_UPDATE_URL=<BDII_HTTP_URL>
BDII_UPDATE_LDIF=http://
```

```
SLAPD=/usr/sbin/slapd
SLAPADD=/usr/sbin/slapadd
```

Base DN is set to *mds-vo-name=<SITE_NAME>,o=grid*. *<INSTALL_ROOT>/bdii/etc/bdii-update.conf* is created from the values of variables <BDII_REGIONS> and <BDII_<region>_URL>.

LDAP schemas are configured in *<INSTALL_ROOT>/bdii/etc/schemas*:

```
/etc/openldap/schema/core.schema
/opt/glue/schema/ldap/Glue-CORE.schema
/opt/glue/schema/ldap/Glue-CE.schema
/opt/glue/schema/ldap/Glue-CESEBind.schema
/opt/glue/schema/ldap/Glue-SE.schema
/opt/lcg/schema/ldap/SiteInfo.schema
```

BDII service is restarted and configured to start on boot.

4.1. Specification of function: config_bdii

The function *'config_bdii'* needs the following variables to be set in the configuration file:

BDII_FCR : Set the URL of the Freedom of Choice for Rescources URL.

BDII_HOST : BDII Hostname.

BDII_HTTP_URL : URL pointing to the BDII configuration file.

BDII_REGIONS : List of node types publishing information on the bdii. For each item listed in the BDII_REGIONS variable you need to create a set of new variables as follows:

> **BDII_<REGION>_URL :** URL of the information producer (e.g.: BDII_CE_URL="URL of the CE information producer", BDII_SE_URL="URL of the SE information producer".

INSTALL_ROOT : Installation root - change if using the re-locatable distribution.

SITE_NAME : Your GIIS.

The original code of the function can be found in:

```
/opt/lcg/yaim/functions/config_bdii
```

The code is also reproduced in 6.2..

5. Set-up GridICE agent

Author(s): Retico,Antonio
Email : support-lcg-manual-install@cern.ch

This chapter describes the configuration steps done by the *yaim* function ’*config_fmon_client*’.

The LCG nodes can 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 on the nodes (data producers), you should also run a GridICE collector server to collect information from your agents.

In the default LCG-2 configuration the MON node runs the GridICE collector node.

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

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_SERVER_HOST>
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_SERVER_HOST>** 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).

- start the GridICE agent

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

5.1. SPECIFICATION OF FUNCTION: CONFIG_FMON_CLIENT

The function *'config_fmon_client'* needs the following variables to be set in the configuration file:

BATCH_LOG_DIR : Your batch system log directory.

CE_BATCH_SYS : Implementation of site batch system. Available values are "torque", "lsf", "pbs", "condor" etc.

CE_HOST : Computing Element Hostname.

CRON_DIR : Yaim writes all cron jobs to this directory. Change it if you want to turn off Yaim's management of cron.

GRIDICE_SERVER_HOST : GridIce server host name (usually run on the MON node).

INSTALL_ROOT : Installation root - change if using the re-locatable distribution.

MON_HOST : MON Box Hostname.

MY_DOMAIN : site's domain name.

The original code of the function can be found in:

```
/opt/lcg/yaim/functions/config_fmon_client
```

The code is also reproduced in 6.3..

6. SOURCE CODE

6.1. CONFIG_EDGUSERS

```
config_edgusers(){

INSTALL_ROOT=${INSTALL_ROOT:-/opt}

check_users_conf_format

if ( ! id edguser > /dev/null 2>&1 ); then
    useradd -r -c "EDG User" edguser
    mkdir -p /home/edguser
    chown edguser:edguser /home/edguser
fi

if ( ! id edginfo > /dev/null 2>&1 ); then
    useradd -r -c "EDG Info user" edginfo
    mkdir -p /home/edginfo
    chown edginfo:edginfo /home/edginfo
fi

if ( ! id rgma > /dev/null 2>&1 ); then
    useradd -r -c "RGMA user" -m -d ${INSTALL_ROOT}/glite/etc/rgma rgma
fi

# Make sure edguser is a member of each group

awk -F: '{print $3, $4, $5}' ${USERS_CONF} | sort -u | while read gid groupname virtorg; do
    if ( [ "$virtorg" ] && echo $VOS | grep -w "$virtorg" > /dev/null ); then
# On some nodes the users are not created, so the group will not exist
# Isn't there a better way to check for group existance??
if ( grep "^${groupname}:" /etc/group > /dev/null ); then
    gpasswd -a edguser $groupname  > /dev/null
fi
    fi
done

return 0
}
```

6.2. CONFIG_BDII

```
config_bdii(){
#
# Configures the BDII.
#
# If SITE_BDII=yes configures as a site BDII otherwise top level
#
# Uses CE_HOST SE HOST RB_HOST and PX_HOST.
#
# These values should be changed the common ones.
```

```
#

requires BDII_HOST

INSTALL_ROOT=${INSTALL_ROOT:-/opt}

mkdir -p $INSTALL_ROOT/bdii/etc
chown edguser $INSTALL_ROOT/bdii/etc
mkdir -p $INSTALL_ROOT/bdii/var
chown edguser $INSTALL_ROOT/bdii/var

BDII_AUTO_MODIFY=no

if ( ! echo "${NODE_TYPE_LIST}" | grep BDII > /dev/null ); then

# We're a site BDII

requires BDII_REGIONS BDII__URL

BDII_BIND=mds-vo-name=$SITE_NAME,o=grid
BDII_AUTO_UPDATE=no

rm -f $INSTALL_ROOT/bdii/etc/bdii-update.conf
for REGION in $BDII_REGIONS; do
            echo "$REGION  `eval echo '$'BDII_${REGION}_URL`" >> $INSTALL_ROOT/bdii/etc/bdii-update.conf
done

else

# We're a top level BDII

requires BDII_HTTP_URL

if [ "$BDII_FCR" ]; then
    BDII_AUTO_MODIFY=yes
fi

BDII_BIND=mds-vo-name=local,o=grid
BDII_AUTO_UPDATE=yes

fi

pass=`mkpasswd -s 0 2> /dev/null` || pass=$RANDOM

cat << EOF > $INSTALL_ROOT/bdii/etc/bdii.conf
BDII_PORT_READ=2170
BDII_PORTS_WRITE="2171 2172 2173"
BDII_USER=edguser
BDII_BIND=$BDII_BIND
BDII_PASSWD=$pass
BDII_SEARCH_FILTER='*'
BDII_SEARCH_TIMEOUT=30
BDII_BREATHE_TIME=60
BDII_AUTO_UPDATE=$BDII_AUTO_UPDATE
```

```
BDII_AUTO_MODIFY=$BDII_AUTO_MODIFY
BDII_DIR=$INSTALL_ROOT/bdii/
BDII_UPDATE_URL=$BDII_HTTP_URL
BDII_UPDATE_LDIF=${BDII_FCR:-http://}
SLAPD=/usr/sbin/slapd
SLAPADD=/usr/sbin/slapadd

EOF

cat << EOF > $INSTALL_ROOT/bdii/etc/schemas
/etc/openldap/schema/core.schema
/opt/glue/schema/ldap/Glue-CORE.schema
/opt/glue/schema/ldap/Glue-CE.schema
/opt/glue/schema/ldap/Glue-CESEBind.schema
/opt/glue/schema/ldap/Glue-SE.schema
/opt/lcg/schema/ldap/SiteInfo.schema

EOF

/sbin/chkconfig --add bdii
/sbin/service bdii stop
/sbin/service bdii start

return 0
}
```

6.3. CONFIG_FMON_CLIENT

```
config_fmon_client(){

# Modified by Cristina Aiftimiei (aiftim <at> pd.infn.it):
# Modified by Enrico Ferro (enrico.ferro <at> pd.infn.it)
#  host kernel version no more published
#  user DN hidden by default
#  job monitoring resource refresh for jobs in on Q/R status disabled by default
#  support new job monitoring probe
#  support new LRMSInfo probe

INSTALL_ROOT=${INSTALL_ROOT:-/opt}

requires GRIDICE_SERVER_HOST

mkdir -p ${INSTALL_ROOT}/edg/var/etc
> ${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg

# Job-Monitoring parameters
JM_TMP_DIR=/var/spool/gridice/JM
LAST_HOURS_EXEC_JOBS=2
mkdir -p ${JM_TMP_DIR}/new
mkdir -p ${JM_TMP_DIR}/ended
mkdir -p ${JM_TMP_DIR}/subject
mkdir -p ${JM_TMP_DIR}/processed
```

```
#  Monitoring of processes/daemon with gridice
if ( echo "${NODE_TYPE_LIST}" | grep CE > /dev/null ); then
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[ce-access-node]
gsiftp ^[\s\w\/\.-]*ftpd
edg-gatekeeper ^[\s\w\/\.-]*edg-gatekeeper
globus-mds ^[\s\w\/\.-]*${INSTALL_ROOT}/globus/libexec/slapd
fmon-agent ^[\s\w\/\.-]*fmon-agent
lcg-bdii-fwd    ^[\s\w\/\.-]*bdii-fwd
lcg-bdii-update ^[\w\/\.-]*perl\s[\s\w\/\.-]*bdii-update
lcg-bdii-slapd  ^[\w\/\.-]*slapd\s[\s\w\/\.\-]*bdii
EOF

if [ "$CE_BATCH_SYS" = "torque" ] || [ "$CE_BATCH_SYS" = "pbs" ] || [ "$CE_BATCH_SYS" = "lcgpbs" ]; then
   cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
pbs-server ^[\s\w\/\.-]*pbs_server
maui ^[\s\w\/\.-]*maui
EOF
fi
if [ "$CE_BATCH_SYS" = "lsf" ]; then
   cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
lsf-lim ^[\s\w\/\.-]*lim
lsf-pim ^[\s\w\/\.-]*pim
lsf-res ^[\s\w\/\.-]*res
lsf-sbatchd ^[\s\w\/\.-]*sbatchd
EOF
MASTER="‘lsclusters |grep -v MASTER |awk ’{print \$3}’‘"
if [ "$CE_HOST" = "$MASTER" -o "$CE_HOST" = "$MASTER.$MY_DOMAIN" ]; then
       cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
lsf-mbatchd ^[\s\w\/\.-]*mbatchd
EOF
fi

fi
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[ce-access-node end]
EOF
fi

if ( echo "${NODE_TYPE_LIST}" | grep SE > /dev/null ); then
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[se-access-node]
gsiftp ^[\s\w\/\.-]*ftpd
globus-mds ^[\s\w\/\.-]*${INSTALL_ROOT}/globus/libexec/slapd.*2135.*
fmon-agent ^[\s\w\/\.-]*fmon-agent
[se-access-node end]
EOF
fi

if ( echo "${NODE_TYPE_LIST}" | grep SE_dpm_mysql > /dev/null ); then
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[dpm-master-node]
globus-mds ^[\s\w\/\.-]*/opt/globus/libexec/slapd.*2135.*
```

```
fmon-agent ^[\s\w\/\.-]*fmon-agent
dpm-master ^[\s\w\/\.-]*dpm
dpm-names ^[\s\w\/\.-]*dpnsdaemon
MySQL ^[\s\w\/\.-]*mysqld
srm-v1-interface ^[\s\w\/\.-]*srmv1
srm-v2-interface ^[\s\w\/\.-]*srmv2
gsiftp ^[\w,\/,-]*ftpd
rfio ^[\w,\/,-]*rfiod
[dpm-master-node end]
EOF
fi

if ( echo "${NODE_TYPE_LIST}" | grep SE_dpm_disk > /dev/null ); then
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[dpm-pool-node]
gsiftp ^[\w,\/,-]*ftpd
rfio ^[\w,\/,-]*rfiod
[dpm-pool-node end]
EOF
fi

if [ "X$GRIDICE_SERVER_HOST" = "X`hostname -f`" ]; then
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[gridice-collector]
gridice-mds ^[\s\w\/\.-]*${INSTALL_ROOT}/globus/libexec/slapd.*2136.*
fmon-server ^[\s\w\/\.-]*fmon-server
[gridice-collector end]
EOF
fi

if [ "X$MON_HOST" = "X`hostname -f`" ]; then
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[rgma-monbox]
ntpd ^[\s\w\/\.-]*ntpd
tomcat [\s\w\/\.-]tomcat
fmon-agent ^[\s\w\/\.-]*fmon-agent
[rgma-monbox end]
EOF
fi

if ( echo "${NODE_TYPE_LIST}" | grep RB > /dev/null ); then
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[broker]
ftp-server              ^[\s\w\/\.-]*ftpd
job-controller          ^[\s\w\/\.-]*edg-wl-job_controller
condor-master           ^[\s\w\/\.-]*condor_master
logging-and-bookkeeping ^[\s\w\/\.-]*edg-wl-bkserverd
condorg-scheduler       ^[\s\w\/\.-]*condor_schedd
log-monitor             ^[\s\w\/\.-]*edg-wl-log_monitor
local-logger            ^[\s\w\/\.-]*edg-wl-logd
local-logger-interlog   ^[\s\w\/\.-]*edg-wl-interlogd
network-server          ^[\s\w\/\.-]*edg-wl-ns_daemon
proxy-renewal           ^[\s\w\/\.-]*edg-wl-renewd
workload-manager        ^[\s\w\/\.-]*edg-wl-workload_manager
```

```
fmon-agent              ^[\s\w\/\.-]*fmon-agent
[broker end]
EOF
fi

if ( echo "${NODE_TYPE_LIST}" | grep BDII > /dev/null ); then
cat <<EOF >>${INSTALL_ROOT}/gridice/monitoring/etc/gridice-role.cfg
[bdii]
lcg-bdii-fwd    ^[\s\w\/\.-]*bdii-fwd
lcg-bdii-update ^[\w\/\.-]*perl\s[\s\w\/\.-]*bdii-update
lcg-bdii-slapd  ^[\w\/\.-]*slapd\s[\s\w\/\.\-]*bdii
fmon-agent      ^[\s\w\/\.-]*fmon-agent
[bdii end]
EOF
fi


# Configuration File for JobMonitoring
# If not definied before, use these defaults
GRIDICE_HIDE_USER_DN=${GRIDICE_HIDE_USER_DN:-yes}
GRIDICE_REFRESH_INFO_JOBS=${GRIDICE_REFRESH_INFO_JOBS:-no}

cat <<EOF >${INSTALL_ROOT}/gridice/monitoring/etc/JM.conf
##
## /opt/gridice/monitoring/etc/JM.conf
##

LRMS_TYPE=${CE_BATCH_SYS}

# --jm-dir=<$JM_TMP_PATH> (default /var/spool/gridice/JM) -- inside this directory
#                  will be created "new/" "ended/" "subject/" "processed/";
#                  when messlog_mon.pl is restarted it has to delete all
#                  "processed/.jmgridice*" files
JM_TMP_DIR=${JM_TMP_DIR}

# "--lrms-path=<LRMS_SPOOL_DIR>" (path for logs of batch-system)
LRMS_SPOOL_DIR=${BATCH_LOG_DIR}

# "--hide-subject=<yes|no>" (default: yes)
HIDE_USER_DN=${GRIDICE_HIDE_USER_DN}

# "--interval=<interval for ended jobs>", in hours (default: 2)
LAST_HOURS_EXEC_JOBS=${LAST_HOURS_EXEC_JOBS}

# <yes|no> (set the parameter "--no-update" if "no", otherwise no parameter is passed)
REFRESH_INFO_FOR_RUNNING_JOBS=${GRIDICE_REFRESH_INFO_JOBS}
EOF

# End configuration File for JobMonitoring

cat <<EOF >${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf
# template Sensor file for edg-fmonagent
# ** DO NOT EDIT **
# Generated from template: /usr/lib/lcfg/conf/fmonagent/sensors.cfg
```

```
MSA



Transport



UDP
Server ${GRIDICE_SERVER_HOST}
Port 12409
FilterMetrics KeepOnly
11001
11011
11021
11101
11202
11022
11031
11201
10100
10102
10103
10104
EOF
if ( echo "${NODE_TYPE_LIST}" | grep CE > /dev/null ); then
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf
TCP
Server ${GRIDICE_SERVER_HOST}
Port 12409
FilterMetrics KeepOnly
10106
10107
EOF
fi
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf

Sensors

edtproc
CommandLine ${INSTALL_ROOT}/gridice/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 ${INSTALL_ROOT}/edg/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
10102
MetricClass edt.alive
10103
MetricClass edt.regfiles
10104
MetricClass executeScript
Parameters
command ${INSTALL_ROOT}/gridice/monitoring/bin/CheckDaemon.pl --cfg ${INSTALL_ROOT}/gridice/monitoring/etc/gridice-
EOF
if ( echo "${NODE_TYPE_LIST}" | grep CE > /dev/null ); then
if [ "X$GRIDICE_REFRESH_INFO_JOBS" = "Xno" ]; then
  OPT_REFRESH=" --no-update"
fi
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf
10106
MetricClass executeScript
Parameters
command  ${INSTALL_ROOT}/gridice/monitoring/bin/CheckJobs.pl --lrms=${CE_BATCH_SYS} --lrms-path=${BATCH_LOG_DIR} --
EOF
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf
10107
```

```
MetricClass executeScript
Parameters
command ${INSTALL_ROOT}/gridice/monitoring/bin/LRMSinfo.pl --lrms=${CE_BATCH_SYS}
EOF
fi
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf

Samples
verylowfreq
Timing 3600 0
Metrics
10100
lowfreq
Timing 1800 0
Metrics
11001
EOF
if ( echo "${NODE_TYPE_LIST}" | grep CE > /dev/null ) && [ "X$GRIDICE_JM" = "Xyes" ]; then
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf
midfreq
Timing 1200 0
Metrics
10106
EOF
fi
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf
proc0
Timing 30 0
Metrics
10102
proc1
Timing 60 0
Metrics
11011
11021
11101
11202
11022
11031
11201
proc2
Timing 300 0
Metrics
10103
EOF
if ( echo "${NODE_TYPE_LIST}" | grep CE > /dev/null ); then
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf
10107
EOF
fi
cat <<EOF >>${INSTALL_ROOT}/edg/var/etc/edg-fmon-agent.conf
proc3
Timing 120 0
Metrics
```

```
10104
EOF

# Configure the job monitoring daemon only on CE
if ( echo "${NODE_TYPE_LIST}" | grep CE > /dev/null ); then
  /sbin/chkconfig gridice_daemons on
  /sbin/service gridice_daemons stop
  /sbin/service gridice_daemons start
fi

/sbin/chkconfig edg-fmon-agent on
/sbin/service edg-fmon-agent stop
/sbin/service edg-fmon-agent start

# The cron job required was originally installed under
# the spurious name edg-fmon-knownhosts
if [ -f ${CRON_DIR}/edg-fmon-knownhosts ]; then
    rm -f ${CRON_DIR}/edg-fmon-knownhosts
fi

if [ "X$GRIDICE_SERVER_HOST" = "X‘hostname -f‘" ]; then
/sbin/chkconfig edg-fmon-server on
/sbin/chkconfig gridice-mds on
/sbin/service edg-fmon-server stop
/sbin/service edg-fmon-server start
/sbin/service gridice-mds stop
/sbin/service gridice-mds start

cron_job edg-fmon-cleanspool root "41 1 * * * ${INSTALL_ROOT}/edg/sbin/edg-fmon-cleanspool &> /dev/null"

#Clean up any remaining sensitive information
find /var/fmonServer/ -name ’last.00010101’ -exec rm -f ’{}’ \;

fi


return 0
}
```