

# Table of Contents

<b>YAIM functions dictionary.....</b>	<b>1</b>
config_add_pool_env.....	1
config_add_to_shift.....	1
config_amga_client.....	1
config_amga_server.....	1
config_amga_server_oracle.....	1
config_amga_server_postgres.....	1
config_apel_lsf.....	1
config_apel_pbs.....	2
config_apel_rgma.....	2
config_bdii_only.....	2
config_certs_userland.....	2
config_check_oraclelibs.....	3
config_condor_wms.....	3
config_cream_apel.....	3
config_cream_blah.....	3
config_cream_blparser.....	3
config_cream_ce.....	3
config_cream_cemon.....	4
config_cream_clean.....	4
config_cream_clean_users.....	4
config_cream_db.....	4
config_cream_drop.....	4
config_cream_gip.....	4
config_cream_glexec.....	4
config_cream_tomcat_user.....	4
config_cream_trustmanager.....	5
config_crl.....	5
config_DPM_disk.....	5
config_DPM_group.....	5
config_DPM_gsiftp.....	5
config_DPM_https.....	5
config_DPM_info.....	5
config_DPM_mgr.....	5
config_DPM_mysql.....	5
config_DPM_oracle.....	6
config_DPM_rfio.....	6
config_DPM_se.....	6
config_DPM_upgrade.....	6
config_DPM_user.....	6
config_DPM_xrootd.....	6
config_edgusers.....	7
config_FTA2_agents.....	7
config_FTM2.....	7
config_fts_client.....	7
config_FTS2_ws.....	7
config_gip_bdii_site.....	7
config_gip_bdii_top.....	7
config_gip_ce.....	7
config_gip_dpm.....	8
config_gip_fts.....	8
config_gip_lb.....	8
config_gip_lfc.....	8
config_gip_mon.....	8

# Table of Contents

## YAIM functions dictionary

config_gip_only.....	8
config_gip_sched_plugin_lsf.....	9
config_gip_sched_plugin_pbs.....	9
config_gip_scheduler_plugin.....	9
config_gip_service_release.....	9
config_gip_site.....	9
config_gip_software_plugin.....	9
config_gip_vobox.....	9
config_gip_vo_tag.....	10
config_gip_wms.....	10
config_glexec_wn.....	10
config_glite_env.....	10
config_glite_initd.....	10
config_glite_lb.....	10
config_glite_lbproxy.....	11
config_glite_locallogger.....	11
config_glite_ui.....	11
config_glite_wms.....	11
config_globus_clients.....	12
config_globus_devel.....	12
config_globus_gatekeeper.....	12
config_globus_gridftp.....	13
config_gridview_se.....	13
config_gsissh.....	13
config_host_certs.....	13
config_info_service_bdii_site.....	13
config_info_service_bdii_top.....	13
config_info_service_lb.....	14
config_info_service_px.....	14
config_info_service_wms.....	14
config_jobmanager.....	14
config_jobmon.....	15
config_lcas_lcmaps_gt4.....	15
config_lcgen.....	15
config_lcgce_dgas.....	15
config_ldconfig.....	15
config_lfc_dli.....	16
config_lfc_mysql.....	16
config_lfc_oracle.....	16
config_lfc_oracle_upgrade.....	16
config_lfc_server.....	16
config_lfc_upgrade.....	16
config_lfc_user.....	16
config_marshall.....	16
config_maui_cfg.....	17
config_mkgridmap.....	17
config_proxy_server.....	18
config_rfio.....	18
config_rgma_client.....	18
config_rgma_gin.....	18
config_rgma_server.....	18
config_root_email.....	19
config_sd2cache.....	19

# Table of Contents

## YAIM functions dictionary

config_sw_dir.....	19
config_sysconfig_edg.....	19
config_sysconfig_globus.....	19
config_sysconfig_lcg.....	19
config_tomcat.....	19
config_torque_client.....	20
config_torque_server.....	20
config_torque_submitter_ssh.....	20
config_ui_tar.....	21
config_users.....	21
config_vobox.....	21
config_vomsdir.....	21
config_vomses.....	21
config_vomsmap.....	21
config_wn.....	22
config_wn_tar.....	22
config_workload_manager_client.....	22

# YAIM functions dictionary

Please, note that only the functions used in 3.1 services and clients are described.

## **config\_add\_pool\_env**

This function configures the pool accounts environment. It creates a symlink of the `grid-env. (c) sh` file stored under `GRID_ENV_LOCATION` in `/etc/profile.d`.

## **config\_add\_to\_shift**

This is an auxiliary function used by DPM.

## **config\_amga\_client**

This functions configures the AMGA client by defining in the environment the `PYTHONPATH` to point to `${INSTALL_ROOT}/glite/lib/python2.3/site-packages/amga`.

## **config\_amga\_server**

This function configures the AMGA server.

## **config\_amga\_server\_oracle**

This function configures the amga server service on SLC4 with the oracle database server. It creates the following files:

- `/etc/sysconfig/amgad`
- `${INSTALL_ROOT}/glite/etc/mdclient.config`
- `${INSTALL_ROOT}/glite/etc/amgad.conf`

It starts the following daemons:

- `/etc/init.d/mdservice`

## **config\_amga\_server\_postgres**

This function configures the amga server service on SLC4 with the postgres database server. It creates the following files:

- `/var/lib/pgsql/data/postgresql.conf`
- `/etc/odbc.ini`
- `${INSTALL_ROOT}/glite/etc/mdclient.config`
- `${INSTALL_ROOT}/glite/etc/amgad.config`

It starts the following daemons:

- `/etc/init.d/postgresql`
- `/etc/init.d/mdservice`

## **config\_apel\_lsf**

This function configures the apel parser for the LSF batch system. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite-apel-lsf/parser-config-yaim.xml`

The cron jobs that are created are:

- `edg-apel-lsf-parser` : it runs the apel parser and reports any error under the `/var/log/apel.log` log file.

## config\_apel\_pbs

This function configures the apel parser for the torque/pbs batch system. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite-apel-pbs/parser-config-yaim.xml`: depending on whether we are configuring Apel on an lcg CE or on a cream CE, yaim configures the apel parser to parse the log information in different log files:
  - ◆ `/opt/edg/var/gatekeeper` with log files containing prefix `grid-jobmap_` for lcg CE
  - ◆ `/opt/glite/var/log/accounting` with log files containing prefix `blahp.log-` for cream CE

Moreover, apel is also configured to read torque/pbs server log files under `/var/spool/pbs/server_priv/accounting`.

The cron jobs that are created are:

- `edg-apel-pbs-parser` : it runs the Apel parser and reports any error under the `/var/log/apel.log` log file.

## config\_apel\_rgma

This function configures the apel publisher which nowadays runs in the MON host. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite-apel-publisher/publisher-config-yaim.xml`

It also creates the Mysql DB that will contain the apel publisher tables.

The cron jobs that are creates are:

- `edg-apel-publisher`

## config\_bdii\_only

This functions configures the BDII. It creates the modifies the following files:

- `${INSTALL_ROOT}/bdii/etc/bdii.conf` : it changes `BDII_IS_CACHE=yes`.
- `${INSTALL_ROOT}/bdii/etc/schemas` : it updates the openldap version according to what it is installed.

It starts the following daemons as root:

- `/etc/init.d/bdii`

## config\_certs\_userland

This function installs the CA files in `X509_CERT_DIR` for the TAR UI. If the certificates are already installed in `X509_CERT_DIR`, this function does nothing. Otherwise it installas the CA files from the APT repository

defined in CA\_REPOSITORY.

## config\_check\_oraclelibs

This function checks that a set of oracle libraries are present in ldconfig for LFC, DPM and FTS. It gives a warning if the library is missing.

## config\_condor\_wms

This function configures the condor configuration files for WMS. It modifies the following file:

- \$INSTALL\_ROOT/condor-c/local.\$LOCAL\_HOSTNAME/condor\_config.local
- var/local/condor/condor\_config.local

## config\_cream\_apel

This function configures apel for the cream CE. It modifies the following files:

- \${GLITE\_LOCATION}/etc/rgma/rgma.conf

## config\_cream\_blah

This function configures blah in the cream CE. It modifies the following files:

- /opt/glite/etc/blah.config
- /opt/glite/etc/glite-ce-cream/blah.sh

## config\_cream\_blparser

This function configures blparser in a Cream CE. It creates the following files:

- /opt/glite/etc/blparser.conf
- /etc/logrotate.d/glite-parser
- \${GLITE\_LOCATION}/etc/gLiteservices

It starts the following daemons:

- /etc/init.d/glite-ce-blparser

## config\_cream\_ce

This function configures main services of a Cream CE. It creates/modifies the following files:

- /etc/tomcat5/tomcat5.conf
- /etc/tomcat5/server.xml
- /opt/glite/etc/glite-ce-cream/cream-config-yaim.xml
- /usr/share/tomcat5/glexec-wrapper.sh

It starts the following daemons:

- etc/init.d/tomcat5

## config\_cream\_cemon

This function configures CEMON service of a Cream CE.

## config\_cream\_clean

This function cleans files before start Cream CE configuration.

## config\_cream\_clean\_users

This function deletes 'cream' and 'deleg' users and adds \$CREAM\_DB\_USER in the Mysql DB.

## config\_cream\_db

This function configures cream CE databases: creamdb and delegationdb.

It starts the following daemons:

- /etc/init.d/mysqld

## config\_cream\_drop

This function drops files in the Mysql DB before start Cream CE configuration.

## config\_cream\_gip

This function configures the generic information provider (GIP) for the cream CE. It creates the following files:

- \${INSTALL\_ROOT}/glite/etc/gip/plugin/glite-info-dynamic-ce : dynamic plugin for the corresponding batch system
- \${INSTALL\_ROOT}/glite/etc/gip/ldif/static-file-Cluster.ldif : static information for *GlueCluster*
- \${INSTALL\_ROOT}/glite/etc/gip/ldif/static-file-CE.ldif : static information for *GlueCE*
- \${INSTALL\_ROOT}/glite/etc/gip/ldif/static-file-CESEBind.ldif : static information for *GlueCESEBind*

## config\_cream\_glexec

This function configures glexec on a cream CE. It creates the following functions:

- /etc/ld.so.conf.d/glite.conf
- /etc/ld.so.conf.d/globus.conf
- /opt/glite/etc/lcmaps/lcmaps-suexec.db
- /opt/glite/etc/lcas/lcas-suexec.db
- /opt/glite/etc/glexec.conf
- /etc/logrotate.d/glexec

## config\_cream\_tomcat\_user

This function configures tomcat user as VO user on a Cream CE.

## config\_cream\_trustmanager

This function configures trustmanager on a Cream CE.

## config\_crl

This function creates the cron job to regularly update the CA crl files. The function exists without doing anything if we are configuring a TAR UI or TAR WN and the CA files are managed centrally and not installed by YAIM.

The cron jobs that are created are:

- `fetch-crl` : it regularly executes the `$INSTALL_ROOT/glite/libexec/fetch-crl.sh =` and writes in the log file `=/var/log/fetch-crl-cron.log`, which is also log rotated by YAIM.

## config\_DPM\_disk

This function configures the DPM disk.

## config\_DPM\_group

This function configures the table `Cns_groupinfo` for the virtual mapping of users in a DPM. The 'gid' start from 101 then +1 per each VO then `Cns_unique_id` contains the max gid.

## config\_DPM\_gsiftp

This function configures the gsiftp server. It creates the following file:

- `/etc/sysconfig/dpm-gsiftp`

It starts the following daemons:

- `/etc/init.d/dpm-gsiftp`

## config\_DPM\_https

This function configures the DPM secure http server.

## config\_DPM\_info

This function configures the MySQL DB for DPM info user.

## config\_DPM\_mgr

This function copies the host certificate in the necessary locations and sets the right ownership and permissions.

## config\_DPM\_mysql

This function configures Mysql for DPM.



## config\_DPM\_oracle

This function configures oracle for DPM.

## config\_DPM\_rfio

This function configures rfio in the DPM. It modifies the following files:

- /etc/services: Add the port and protocol for the rfio service.
- /etc/sysconfig/rfiiod

It starts the following daemons:

- /etc/init.d/rfiiod

## config\_DPM\_se

This function configures a set of data storage services. It creates the following files:

- /etc/sysconfig/dpns
- /etc/sysconfig/dpm
- /etc/sysconfig/srmv1
- /etc/sysconfig/srmv2
- /etc/sysconfig/srmv2.2
- /etc/cron.monthly/create-default-dirs-DPM.sh

It starts the following daemons:

- /etc/init.d/dpns
- /etc/init.d/dpm
- /etc/init.d/srmv1
- /etc/init.d/srmv2
- /etc/init.d/srmv2.2

## config\_DPM\_upgrade

This function configures the upgrade actions needed for a db upgrade in DPM.

## config\_DPM\_user

This function creates the dpmmgr user needed by DPM.

## config\_DPM\_xrootd

This function configures xroot in DPM. It creates the following files:

- /etc/sysconfig/dpm-xrd

It starts the following daemons:

- /etc/init.d/dpm-olb
- /etc/init.d/dpm-xrd

## config\_edgusers

This function configures a set of special users needed by most of the grid services if `CONFIG_USERS=yes`. It configures the users listed under `${INSTALL_ROOT}/glite/yaim/examples/edgusers.conf`. For more information on user configuration, please check the YAIM guide -> User configuration.

## config\_FTA2\_agents

This function configures the FTS agent nodes. It detects which node you are on and configures the appropriate agents as found in the site-def file.

## config\_FTM2

This function configures the file transfer monitor that queries the FTS database to summarise the transfers.

## config\_fts\_client

This function configures the FTS client. It creates the configuration file:

```
${INSTALL_ROOT}/glite/etc/services.xml.
```

## config\_FTS2\_ws

This function configures the FTS web service.

## config\_gip\_bdii\_site

This function configures the information provider (GIP) for the site BDII. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/gip/site-urls.conf` : List of hostnames where there are services running and the site BDII should publish.
- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-info-provider-site`: dynamic plugin for the site BDII.

## config\_gip\_bdii\_top

This function configures the information provider (GIP) for the site BDII. It creates/modifies the following files:

- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-info-provider-egge`: dynamic plugin for the top level BDII.
- `${INSTALL_ROOT}/bdii/etc/bdii.conf`: BDII configuration file.

## config\_gip\_ce

This function configures the generic information provider (GIP) for the lcg CE. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/gip/plugin/glite-info-dynamic-ce` : dynamic plugin for the corresponding batch system
- `${INSTALL_ROOT}/glite/etc/gip/ldif/static-file-Cluster.ldif` : static information for *GlueCluster*
- `${INSTALL_ROOT}/glite/etc/gip/ldif/static-file-CE.ldif` : static information for *GlueCE*

- `${INSTALL_ROOT}/glite/etc/gip/ldif/static-file-CESEBind.ldif` : static information for *GlueCESEBind*

Due to the new cluster yaim module, the cluster information will be decoupled from the lcg CE. For more details please check the configuration of the information system in YAIM.

## config\_gip\_dpm

This function configures the generic information provider (GIP) for the DPM. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/gip/ldif/static-file-dSE.ldif` : static information for the DPM service.
- `${INSTALL_ROOT}/glite/etc/gip/plugin/glite-info-dynamic-se` : dynamic plugin for the DPM.
- `${INSTALL_ROOT}/glite/etc/gip/ldif/static-file-SE.ldif` : static information for *GlueSE*.

## config\_gip\_fts

This function configures the generic information provider (GIP) for the FTS. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-fts-provider`
- `${INSTALL_ROOT}/glite/var/glite-data-transfer-fts-info-provider.dbparams`

## config\_gip\_lb

This is an obsolete function that performs cleaning tasks, deleting old ldif and configuration files. The new function is `config_info_service_lb`.

## config\_gip\_lfc

This function configures the generic information provider (GIP) for the LFC. It creates the following files:

- `$INSTALL_ROOT/glite/var/tmp/gip/glite-info-static-lfc.conf`
- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-lfc-provider`

## config\_gip\_mon

This function configures the generic information provider (GIP) for the LFC. It creates the following files:

- `$INSTALL_ROOT/glite/etc/gip/ldif/glite-info-service-secondary-producer.ldif`
- `$INSTALL_ROOT/glite/etc/gip/ldif/glite-info-service-primary-producer.ldif`
- `$INSTALL_ROOT/glite/etc/gip/ldif/glite-info-service-consumer.ldif`
- `$INSTALL_ROOT/glite/etc/gip/ldif/glite-info-service-on-demand-producer.ldif`
- `$INSTALL_ROOT/glite/etc/gip/ldif/glite-info-service-browser.ldif`
- `${INSTALL_ROOT}/glite/libexec/glite-gip-dynamic-rgma`

## config\_gip\_only

This function configures general features of the generic information provider (GIP). It configures proper GIP related parameters, file ownership and permissions and creates/modifies the following files:

- `$INSTALL_ROOT/glite/etc/gip/glite-info-generic.conf`
- `${INSTALL_ROOT}/glite/etc/gip/ldif/stub-resource.ldif`
- `$INSTALL_ROOT/glite/libexec/glite-info-wrapper`

## config\_gip\_sched\_plugin\_lsf

This function configures the part of the dynamic scheduler plugin that has to do with LSF. It modifies the following files:

- `$(INSTALL_ROOT)/glite/etc/lcg-info-dynamic-scheduler.conf` : it adds the information related to LSF (calculates the Estimated Response Time or ERT).

## config\_gip\_sched\_plugin\_pbs

This function configures the part of the dynamic scheduler plugin that has to do with pbs/torque. It modifies the following files:

- `$(INSTALL_ROOT)/glite/etc/lcg-info-dynamic-scheduler.conf` : it adds the information related to torque/pbs (calculates the Estimated Response Time or ERT).

## config\_gip\_scheduler\_plugin

This function configures the dynamic scheduler plugin. It creates the following files:

- `$(INSTALL_ROOT)/glite/etc/gip/plugin/glite-info-dynamic-scheduler-wrapper` : dynamic plugin to store information about the running jobs.
- `$(INSTALL_ROOT)/glite/etc/lcg-info-dynamic-scheduler.conf` : configuration file used by the plugin containing information about the VOMS FQANs.

## config\_gip\_service\_release

This function configures the service provider that will publish information about the gLite release version for the service. It creates a symbolic link to `/opt/glite/libexec/glite-info-provider-release`.

## config\_gip\_site

This function configures the generic information provider (GIP) for a site. It creates the following files:

- `$(INSTALL_ROOT)/glite/etc/gip/ldif/glite-info-site.ldif`: static information for the *GlueSite*.

## config\_gip\_software\_plugin

This function configures the dynamic software information provider. It creates the following files:

- `$(INSTALL_ROOT)/glite/etc/gip/provider/glite-info-provider-software-wrapper` : information provider to store additional tags about supported VO applications.
- `$(INSTALL_ROOT)/glite/etc/gip/plugin/glite-info-dynamic-software-wrapper` : dynamic plugin to store information about the site runtime environment.

## config\_gip\_vobox

This function configures the generic information provider (GIP) for the VOBOX. It creates the following files:

- `$(INSTALL_ROOT)/glite/etc/gip/ldif/static-file-VOBOX.ldif`: static information for the VOBOX service.

## config\_gip\_vo\_tag

This function configures the VO tag directory per VO. It creates the following files:

- `${INSTALL_ROOT}/edg/var/info/<vo-name>/<vo-name>.list` : YAIM gives the right ownership and permissions to these files.

## config\_gip\_wms

This is an obsolete function that performs cleaning tasks, deleting old ldif and configuration files. The new function is `config_info_service_wms`.

## config\_glexec\_wn

This function configures glexec in the WN for the pilot job framework. glexec is responsible for changing the identity of the pilot job user to execute the payload jobs owned by other users. This function configures the following files:

- `/opt/glite/etc/glexec.conf`: glexec configuration file where the paths for LCAS and LCMAPS configuration files are specified, as well as the allowed list of users to execute glexec.
- `/opt/glite/etc/lcas/lcas-glexec.db`: lcas configuration file.
- `/opt/glite/etc/lcmaps-glexec.db`: lcmaps configuration file.
- `/etc/ld.so.conf`: When configuring a TAR WN, it adds the glite and globus libraries to be able to run glexec. `LD_LIBRARY_PATH` is ignored by glexec. This is not needed in glite-WN since it already configures `ldconfig`.

glexec can be operated in setuid mode or logging only mode. The `glexec` user and group need to be created in the system.

## config\_glite\_env

This is an obsolete function that just performs cleaning tasks, deleting old environment files:

```
${GLITE_EXTERNAL_ROOT:-}/etc/profile.d/gliteenv.(c)sh.
```

## config\_glite\_initd

This function manages the configuration of service gLite on the node type that required it. The following files are created:

- `${GLITE_LOCATION}/etc/gLiteservices` : List of services.
- `/etc/rc.d/init.d/gLite` : gLite service startup function.

## config\_glite\_lb

This function configures the LB. It creates/modifies the following files:

- `/etc/my.cnf` : MySQL configuration file.
- `${GLITE_LOCATION}/etc/gLiteservices` : To add the LB service.

It starts the following daemons:

- `/etc/init.d/mysqld`
- `${GLITE_LOCATION}/etc/init.d/glite-lb-bkserverd`

It creates the following cron jobs:

- `/etc/cron.d/glite-lb-purge.cron`

## config\_glite\_lbproxy

This function configures the LB proxy server. It creates/modifies the following files:

- `/etc/my.cnf` : MySQL configuration file.
- `${GLITE_LOCATION}/etc/gLiteservices` : To add the LB proxy service.

It starts the following daemons:

- `/etc/init.d/mysqld`
- `${GLITE_LOCATION}/etc/init.d/glite-lb-proxy`

## config\_glite\_locallogger

This function configures the local logger that is responsible for reliable delivery of LB events from their source to the server. The function performs the necessary tasks to copy the host certificates into the right location, permissions and ownership. It creates/modifies the following file:

- `${GLITE_LOCATION}/etc/gLiteservices` : To add the local logger service.

The daemons that are started are:

- `${GLITE_LOCATION}/etc/init.d/glite-lb-locallogger`

The cron jobs that are created are:

- `/etc/cron.d/locallogger.cron`

## config\_glite\_ui

This function configures the UI. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite_wmsui_cmd_var.conf` : Python WMS UI client
- `$(GLITE_LOCATION)/etc/$vodir/glite_wmsui.conf` : WMS UI client
- `$(GLITE_LOCATION)/etc/$vodir/glite_wms.conf` : WMS proxy configuration

The WMS and LB are configured to be found by using service discovery.

## config\_glite\_wms

This function configures the WMS. It creates/modifies the following files:

- `${INSTALL_ROOT}/glite/etc/glite_wms.conf`
- `$(GLITE_LOCATION)/sbin/glite_wms_wmproxy_load_monitor`
- `$(INSTALL_ROOT)/glite/etc/glite_wms_wmproxy_httpd.conf`
- `$(INSTALL_ROOT)/glite/etc/glite_wms_wmproxy.gacl`
- `/opt/glite/etc/lcmaps/lcmaps.db` : it configures the lcmaps that will write in `/var/log/glite/lcmaps.log` log file.
- `/etc/sysconfig/globus`
- `${GLITE_LOCATION}/etc/gLiteservices`

The following cron jobs are created:

- /etc/cron.d/glite-wms-create-host-proxy.cron
- /etc/cron.d/glite-wms-purger.cron
- /etc/cron.d/glite-wms-wmproxy-purge-proxycache.cron
- /etc/cron.d/glite-wms-check-daemons.cron
- /etc/cron.d/wmproxy\_logrotate

The following daemons are started:

- /etc/init.d/globus-gridftp
- /etc/init.d/glite-wms-wm
- /etc/init.d/glite-wms-jc
- /etc/init.d/glite-wms-ice
- /etc/init.d/glite-wms-lm
- /etc/init.d/glite-proxy-renewald
- /etc/init.d/glite-wms-wmproxy

It also calls the condor launcher by running `su - $GLITE_USER -c "condorc-initialize"`

## config\_globus\_clients

This functions configures the globus clients. It runs the following setup functions:

- `${GLOBUS_LOCATION}/setup/globus/setup-tmpdirs`
- `${GLOBUS_LOCATION}/setup/globus/setup-globus-common`

It also defines and exports the necessary environment variables to be able to run the globus clients.

## config\_globus\_devel

This function configures globus. It runs the following script `${INSTALL_ROOT}/gpt/sbin/gpt-build -force -nosrc gcc32 gcc32pthr gcc32dbg gcc32dbgpthr`

## config\_globus\_gatekeeper

This function configures the GT4 globus gatekeeper. The task of the gatekeeper is to authenticate the the job and its owner, map the owner to a local user, start a job manager and pass the request so that the job manager can manage it. The function creates the following files:

- `${GLOBUS_LOCATION}/etc/globus-gatekeeper.conf` : globus gatekeeper configuration file

It runs the following setup functions:

- `${GLOBUS_LOCATION}/setup/globus/setup-globus-gram-job-manager`
- `${GLOBUS_LOCATION}/setup/globus/setup-globus-job-manager-fork`

It starts the following daemons as root:

- `/opt/globus/sbin/globus-gatekeeper`

This function also implements log rotate for the globus gatekeeper log files in `/var/log/globus-gatekeeper.log`.

## config\_globus\_gridftp

This function starts the globus gridftp service:

- `/etc/init.d/globus-gridftp`

## config\_gridview\_se

This function configures the gridview service client for SEs. It creates the following files:

- `${INSTALL_ROOT}/gridview/etc/gridview-gridftp-conf.pl`

It starts the following daemons:

- `/etc/init.d/gridview-wsclient-gridftp`

## config\_gsissh

This function configures the gsissh client and server (only for the VOBOX). It creates or modifies the following files:

- `${INSTALL_ROOT}/globus/sbin/SXXsshd` : gsissh start up script. It should contain the VOBOX\_PORT as an argument.
- `${INSTALL_ROOT}/globus/etc/ssh/sshd_config` : gsissh server configuration
- `${INSTALL_ROOT}/globus/etc/ssh/ssh_config` : gsissh client configuration

In the case of the gsissh server it starts the following daemons:

- `/etc/init.d/gsisshd` : which is a symlink to `${INSTALL_ROOT}/globus/sbin/SXXsshd`

## config\_host\_certs

This function checks that the host certificate files exist and they have the right permissions.

## config\_info\_service\_bdii\_site

This function configures the service provider for the site BDII. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite-info-service-bdii-site.conf` : configuration file containing the service information that needs to be published.
- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-info-provider-service-bdii-site-wrapper` : wrapper that will contain the call to the `glite-info-service` function using the previous configuration file.

For more information on how to configure the service provider please check the YAIM Feature list -> `config_info_service_<node-type>` .

## config\_info\_service\_bdii\_top

This function configures the service provider for the top level BDII. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite-info-service-bdii-top.conf` : configuration file containing the service information that needs to be published.



- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-info-provider-service-bdii-top-wrapper` : wrapper that will contain the call to the `glite-info-service` function using the previous configuration file.

For more information on how to configure the service provider please check the YAIM Feature list -> `config_info_service_<node-type>` .

## **config\_info\_service\_lb**

This function configures the service provider for the LB. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite-info-service-lbserver.conf` : configuration file containing the service information that needs to be published.
- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-info-provider-service-lbserver-wrapper` : wrapper that will contain the call to the `glite-info-service` function using the previous configuration file.

For more information on how to configure the service provider please check the YAIM Feature list -> `config_info_service_<node-type>` .

## **config\_info\_service\_px**

This function configures the service provider for Myproxy. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite-info-service-myproxy.conf` : configuration file containing the service information that needs to be published.
- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-info-provider-service-myproxy-wrapper` : wrapper that will contain the call to the `glite-info-service` function using the previous configuration file.

For more information on how to configure the service provider please check the YAIM Feature list -> `config_info_service_<node-type>` .

## **config\_info\_service\_wms**

This function configures the service provider for WMS. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/glite-info-service-wmproxy.conf` : configuration file containing the service information that needs to be published.
- `${INSTALL_ROOT}/glite/etc/gip/provider/glite-info-provider-service-wmproxy-wrapper` : wrapper that will contain the call to the `glite-info-service` function using the previous configuration file.

For more information on how to configure the service provider please check the YAIM Feature list -> `config_info_service_<node-type>` .

## **config\_jobmanager**

This function runs the setup configuration for a jobmanager under a PBS, LSF, SGE and Condor type batch system. The job manager is created by the gatekeeper and is responsible for starting the job on the local system. It creates the following files:

- `${GLOBUS_LOCATION}/etc/grid-services/jobmanager-${JOB_M}` : where `JOB_M` is either `lcppbs`, `lclfs`, `lcsge` or `lcgcondor`. Contains the global options for the job manager configuration.

- `${GLOBUS_LOCATION}/share/globus_gram_job_manager/${JOB_M}.rvf` : contains the list of queues in the lcg CE where the job manager would be able to submit queues.

## config\_jobmon

This function configures the job monitoring for the LB. It creates the following files:

- `/opt/lcg/etc/lcg-mon-job-status.conf` : configuration file for the job monitoring.

It starts the following daemons:

- `/opt/lcg/etc/init.d/lcg-mon-logfile-daemon` : it writes the log files under `/etc/logrotate.d/glite-lb-bkserver-rgma`

## config\_lcas\_lcmaps\_gt4

This function configures the lcas and lcmaps for gridftp and gatekeeper. It creates the following files:

- `${GLITE_LOCATION}/etc/lcas/lcas.db`: lcas configuration file.
- `${GLITE_LOCATION}/etc/lcmaps/lcmaps.db` : lcmaps configuration file. It supports plain and voms proxies.

## config\_lcgen

This function defines some environment variables. It adds some variable definitions to

`$INSTALL_ROOT/glite/etc/profile.d/grid-env.sh`:

- If no `VO_<vo-name>_DEFAULT_SE` variable is defined, it creates this variable per supported VO with the first value in the `SE_LIST`, if defined.
- It defines `LCG_GFAL_INFOSYS` to point to `BDII_LIST` or `BDII_HOST`, if not defined.
- It defines `MYPROXY_SERVER` if `PX_HOST` is defined in your `site-info.def`.
- It defines 64 bit library paths for `dcache`, `perl` and `python` if they exist.
- It defines `VO_<vo-name>_SW_DIR`
- It defines `GSSKLOG_SERVER` if `VOBOX_HOST` is defined in your `site-info.def` and `GSSKLOG` is yes.
- It defines `DPM_HOST` and `DPNS_HOST` if `DPM_HOST` is defined in your `site-info.def`.
- It defines `GLOBUS_TCP_PORT_RANGE`
- It defines `RFIO_PORT_RANGE`

## config\_lcgce\_dgas

This function enables the DGAS accounting system on the lcg CE. It creates the following files:

- `/etc/sysconfig/dgas-add-record.conf` : DGAS configuration file

## config\_ldconfig

This functions configures the dynamic libraries for gLite. It modifies the following files:

- `/etc/ld.so.conf` : YAIM adds several paths to libraries used by gLite.

## config\_lfc\_dli

This function configures the LFC dli. It creates the following files:

- `/etc/sysconfig/lfc-dli`

It starts the following daemons:

- `/etc/init.d/lfc-dli`

## config\_lfc\_mysql

This function configures mysql for LFC.

## config\_lfc\_oracle

This function configures oracle for LFC.

## config\_lfc\_oracle\_upgrade

This function configures an upgrade of the oracle DB for LFC. The possibilities are:

- Upgrading database schema from 2.3.0 to 3.0.0
- Upgrading database schema from 2.2.0 to 3.0.0
- Upgrading database schema from 2.1.0 to 3.0.0

## config\_lfc\_server

This function configures the LFC server. It creates the following files:

- `/etc/sysconfig/lfcdaemon`

It starts the following daemons:

- `/etc/init.d/lfcdaemon`

## config\_lfc\_upgrade

This function implements the upgrade to a newer version of LFC.

## config\_lfc\_user

This function creates the lfcmgr user needed by the LFC.

## config\_marshall

This function runs the setup configuration for marshal packages. These packages improve the performance of the lcg CE modifying the behaviour of the globus job manager. The optimization is done in different ways:

- limitation of the number of concurrently running jobmanager queries, which decreases the load on the file system and batch system and allows the queries to run faster.
- conversion of some of the globus Perl scripts to memory-persistent daemons which eliminates the need of compiling heavy Perl code on every invocation. Globus jobmanagers communicate with Perl

daemons over UNIX domain sockets using tiny compiled clients written in C. These clients replace the original Globus Perl scripts, so the interface change is transparent for both Globus and user.

- replacement for multiple `grid_monitor_agent` processes submitted by Condor to CE. Initially they were introduced to reduce load from jobmanagers polling queued jobs, but in reality with pool accounts they hit the same pitfall - too many concurrent job polls.

The daemons that are started as root are:

- `globus-gass-cache-marshal`
- `globus-job-manager-marshal`
- `globus-gma`

The relevant configuration files can be found under `/opt/globus/etc`. The log file location is `/opt/globus/var/log`.

Daemons should be started from init scripts before `globus-gatekeeper` on boot and stopped after it on shutdown. They intercept HUP signal so it's possible to rotate log files and change running configuration without restart.

## config\_mai\_cfg

This function configures Maui. Maui can be used as an external scheduler for torque/pbs. In this configuration, torque/pbs manages the job queue and the compute resources while Maui queries the torque/pbs server and the pbs mom to obtain up to date job and node information. Using this information, Maui directs torque/pbs to manage jobs in accordance with specified Maui policies, priorities, and reservations. It creates the following files:

- `/etc/services` : Add the port and protocol for the `maui` service
- `/var/spool/maui/maui.cfg` : Maui configuration file.

The daemons that are started are:

- `/etc/rc.d/init.d/maui`

## config\_mkgridmap

This function creates the DN grid-map file. It creates the following configuration files for the `edg-mkgridmap` command:

- `$(INSTALL_ROOT)/edg/etc/edg-mkgridmap.conf` : for `lcg` CE.
- `$(INSTALL_ROOT)/lcg/etc/lcgdm-mkgridmap.conf` : for `DPM` and `LFC`.
- `$(INSTALL_ROOT)/glite/etc/submit-mkgridmap.conf` : for `FTS`.

The following grid-map files are created:

- `{GLITE_EXTERNAL_ROOT}/etc/grid-security/grid-mapfile` : for `DPM` and `lcg` CE.
- `$(INSTALL_ROOT)/lcg/etc/lcgdm-mapfile` : for `DPM` and `LFC`.
- `$(INSTALL_ROOT)/glite/etc/glite-data-transfer-submit-mapfile` : for `FTS`.

It also creates the `gridmap` directory and its pool account files:

- `/etc/grid-security/gridmapdir` : this is not created for `LFC` and `FTS`.

The cron jobs that are created to clean the `gridmapdir` are:

- `lcg-expiregridmapdir`

The cron job that are created to create the gridmap files regularly are:

- `edg-mkgridmap` : for lcg CE and DPM.
- `lcgdm-mkgridmap` : for DPM and LFC.
- `glite-data-transfer-submit-mkgridmap` : for FTS.

## config\_proxy\_server

It configures the Myproxy server. It creates the following files:

- `${INSTALL_ROOT}/glite/etc/myproxy-server.conf` : Describes who can actually use the service.
  - ◆ All accepted credentials are allowed. They are checked against the installed CAs.
  - ◆ If `GRID_TRUSTED_BROKERS` is defined, it's added to the config file but a warning message is displayed since it's deprecated.
  - ◆ If `GRID_AUTHORIZED_RENEWERS`, `GRID_DEFAULT_RENEWERS`, `GRID_AUTHORIZED_RETRIEVERS`, `GRID_DEFAULT_RETRIEVERS`, `GRID_AUTHORIZED_KEY_RETRIEVERS`, `GRID_DEFAULT_KEY_RETRIEVERS`, `GRID_TRUSTED_RETRIEVERS`, `GRID_DEFAULT_TRUSTED_RETRIEVERS` is defined, it's added to the config file.

It starts the following daemons:

- `/etc/init.d/myproxy`

## config\_rfio

This function starts the rfio service. It modifies the following files:

- `/etc/services` : Add the port and protocol for the rfio service.

The following daemons are started:

- `/etc/init.d/rfioid`

## config\_rgma\_client

This function configures the RGMA client. It runs the following script to configure the RGMA client:

```
${RGMA_HOME}/share/rgma/scripts/rgma-setup.py
```

## config\_rgma\_gin

This function configures the GIN for the GIP or FMON. It modifies the following files:

- `${INSTALL_ROOT}/glite/etc/rgma/ClientAuthentication.props`

It starts the following daemons:

- `/etc/init.d/rgma-gin`

## config\_rgma\_server

This function configures the RGMA server. It creates/modifies the following files:

- `/${RGMA_HOME}/etc/rgma-server/ServletAuthentication.props`
- `/etc/tomcat5/tomcat5.conf`

It starts the following daemons:

- `/etc/init.d/mysqld`
- `/etc/init.d/tomcat5`
- `/etc/init.d/rgma-servicetool`

### **config\_root\_email**

This function configures the root `.forward` file if the `ROOT_EMAIL_FORWARD` variable is set in `site-info.def`. It creates the following file:

- `/root/.forward`

### **config\_sd2cache**

This function configures the `sd2cache` tool that creates a cron job to maintain the `/opt/glite/etc/services.xml` file. It creates the following files:

- `/${GLITE_LOCATION}/etc/glite-sd2cache-cron.conf`

It starts the following daemons:

- `/etc/init.d/glite-sd2cache`

### **config\_sw\_dir**

This function creates the software area directory specified per VO and gives the correct permissions and ownerships. It only does this if the directory doesn't exist, otherwise it prints a warning reminding which are the correct ownerships and permissions:

- If the VO has a single sgm account : `chmod -R u+rw,go-w sw_dir`
- If the VO has multiple sgm accounts : `chmod -R ug+rw,o-w sw_dir`

### **config\_sysconfig\_edg**

This function configures the edg sysconfig. It creates the file `/${GLITE_EXTERNAL_ROOT}/etc/sysconfig/edg`.

### **config\_sysconfig\_globus**

This function configures the globus sysconfig. It creates the file `/${GLITE_EXTERNAL_ROOT}/etc/sysconfig/globus`.

### **config\_sysconfig\_lcg**

This function configures the lcg sysconfig. It creates the file `/${GLITE_EXTERNAL_ROOT}/etc/sysconfig/lcg`.

### **config\_tomcat**

This function configures a secure tomcat for FTS.

## config\_torque\_client

This function configures the torque client. It creates or modifies the following files:

- `/etc/services` : Add the port and protocol for the services `pbs`, `pbs_mom` and `pbs_resmon`
- `/etc/ssh/ssh_config` : ssh client configuration
- `/opt/edg/etc/edg-pbs-knownhosts.conf`
- `/etc/ssh/ssh_known_hosts`
- `/var/spool/pbs/mom_priv/config`

The daemons that are started are:

- `/etc/rc.d/init.d/pbs_mom`

The cron jobs that are created are:

- `edg-pbs-knownhosts`
- `mom_logs` : it zips the log files under `/var/spool/pbs/mom_logs`.

## config\_torque\_server

This function configures a torque/pbs server. It basically configures `qmgr` with the supported queue names, torque server configuration options and queue ACLs. The function modifies or creates the following files:

- `/etc/services` : Add the port and protocol for the services `pbs`, `pbs_mom` and `pbs_resmon`
- `/var/spool/pbs/server_name` : to define the batch server hostname in
- `/var/spool/pbs/server_priv/nodes` : create the worker node list including the number of CPUs/cores per worker node.

The daemons that are started are:

- `/etc/rc.d/init.d/pbs_server`

The cron jobs that are created are:

- `server_logs` : it regularly zips the server log files under `/var/spool/pbs/server_logs`.

When the torque server and the CE are in different hosts, the `/etc/hosts.equiv` has to be modified so that the batch server can communicate with the CE.

## config\_torque\_submitter\_ssh

This function configures ssh for the pbs/torque submitter. It creates the following files:

- `$(INSTALL_ROOT)/edg/etc/edg-pbs-shostsequiv.conf` :
- `$(INSTALL_ROOT)/edg/etc/edg-pbs-knownhosts.conf`

The daemons that are started are:

- `/etc/rc.d/init.d/sshd`

The cron jobs that are created are:

- `edg-pbs-shostsequiv`
- `edg-pbs-knownhosts`

## config\_ui\_tar

This function sets up the necessary environment variables used by the TAR UI.

## config\_users

This function configures a set of user pool accounts needed by most of the grid services if `CONFIG_USERS=yes`. It configures the users listed under `${INSTALL_ROOT}/glite/yaim/examples/users.conf`. For more information on user configuration, please check the YAIM guide -> User configuration.

It creates the following cron jobs:

- `cleanup-grid-accounts` : YAIM installs the cron only if the gLite node type is CE or WN. Moreover, the script only cleans local accounts, not remote ones like NFS mounted. The script removes all local account files, which are not modified for more than 15 days.

## config\_vobox

This function configures the VOBOX. The VOBOX can only be used with lcgadmin role. It configures only one sgm account per VO. It creates the following files:

- `${INSTALL_ROOT}/vobox/${vo-name}/start`
- `${INSTALL_ROOT}/vobox/${vo-name}/stop`
- `${INSTALL_ROOT}/vobox/${vo-name}/agents`
- `${INSTALL_ROOT}/vobox/${vo-name}/info-provider`
- `${INSTALL_ROOT}/vobox/${vo-name}/log`
- `${INSTALL_ROOT}/vobox/${vo-name}/proxy_repository`
- `/etc/rc.d/init.d/${vo-name}-box-proxyrenewal` : installs the proxy renewal service per VO.

The cron jobs that are created are:

- `<vo-name>-box-proxyrenewal`

## config\_vommdir

This function configures the lsc files. The lsc files will substitute the VOMS servers public key used to authenticate the VOMS proxy against a set of trusted VOMS servers. Now, there will be one lsc file per trusted VOMS server containing the VOMS server DN and its CA DN. It creates the following files:

- `${X509_VOMS_DIR}/<vo-name>/<voms-server-hostname>.lsc`

## config\_vomses

This function configures the vomses file to be able to contact the VOMS servers. It creates the following file for each VO `${INSTALL_ROOT}/glite/etc/vomses/<vo-name>-<voms-server-hostname>`.

## config\_vomsmap

This function creates the voms gridmap file and groupmap file containing the mappings between VOMS FQANs and user pool accounts and groups. It creates the following files:

- `${GLITE_EXTERNAL_ROOT}/etc/grid-security/grid-mapfile`
- `${GLITE_EXTERNAL_ROOT}/etc/grid-security/groupmapfile`



It also creates the gridmap directory and its pool account files:

- `/etc/grid-security/gridmapdir`

The cron jobs that are created to clean the gridmapdir are:

- `lcg-expiregridmapdir`

In the case of the lcg CE, it also creates a cron job that concatenates the DN gridmap file and VOMS map file in one single file:

- `lcg-ce-mkgridmap`

## config\_wn

This function configures the WN. It basically sets up a number of environment variables in `grid-env.(c)sh`.

## config\_wn\_tar

This function configures the TAR WN. It basically sets up a number of environment variables in `grid-env.(c)sh`.

## config\_workload\_manager\_client

This function configures the workload manager client. It creates the following files:

- `$(INSTALL_ROOT)/edg/etc/$(i)/edg_wl_ui.conf`
- `$(INSTALL_ROOT)/edg/etc/edg_wl_ui_cmd_var.conf`
- `$(INSTALL_ROOT)/edg/etc/edg_wl_ui_gui_var.conf`
- `$(INSTALL_ROOT)/edg/var/etc/profile.d`
  - ◆ `edg-wl-ui-env.csh`
  - ◆ `edg-wl-ui-env.sh`
  - ◆ `edg-wl-ui-gui-env.csh`
  - ◆ `edg-wl-ui-gui-env.sh`

If the `OUTPUT_STORAGE` variable is not defined, it uses the `/tmp/jobOutput` location if running the configuration as root, or the `$(HOME)/jobOutput` if running the configuration as a normal user.

-- MariaALANDESPRADILLO - 09 Jan 2008

---

This topic: EGEE > YAIMfunctions

Topic revision: r16 - 2008-08-29 - MariaALANDESPRADILLO



Copyright &© by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Ask a support question or Send feedback