Table of Contents

Table of Contents

YAIM 4 guide for sysadmins

Introduction

This document provides a description of YAIM version 4. It contains a modular structure consisting of a main yaim core module, yaim clients and yaim services.

For the general installation process check:

- gLite 3.2 Generic Installation & Configuration guide
- gLite 3.1 Generic Installation & Configuration guide

Known issues

Please, check this section if you are using one of the yaim versions listed here. It contains useful information about known bugs.

glite-yaim-amga

Bug	Description and Workaround	Fixed in
#48849	Manual edit of `mdclient.conf` and `amgad.config` is needed to enable Certificate Authentication.	No fix so far
#46162	Run `service postgresql initdb` before anything else and remove option `tcpip_socket` to be able to start the server.	No fix so far

glite-yaim-clients

Bug	Description and Workaround
#52825	SL5 TAR UI configuration problem due to missing `GLITE_EXTERNAL_ROOT/usr/lib64` in `LD_LIBRARY_PATH` manually.
#49850	TAR UI and TAR WN use a utility called `central_certs` distributed in yaim core. The command `find` is u of the passed parameters, the following warning message is printed: `find: warning: you have specified the -maxdepth option after a non-option argument -mtim but options are not positional (-maxdepth affects tests specified before it as well as t Please specify options before other arguments.` This can be ignored until it is fixed.
#47672	Due to a missing dependency, when configuring the VOBOX with yaim, the function `config_info_servic` won't find `glite-info-service-vobox.conf.template`. This file comes with the glite-info-provider-servic manually until this bug is fixed.
#47668	The information published by the VOBOX service provider is not complete. The Owner and ACBR informa your VOBOX to this version of yaim clients if it's important to publish this information in your site.
#48464	Problem in the 64bit TAR WN: there's a missing library in the `LD_LIBRARY_PATH` and lcg-* commands don't add manually `${GLITE_EXTERNAL_ROOT}/usr/lib64`
#40944	`/usr/lib64` should be exported in `LD_LIBRARY_PATH` in 64bit machines installing 64 binaries. YAIM fails t done manually until the bug is fixed.
#39930	`config_vomsdir` should not be used with the TAR WN and TAR UI because it doesn't work. It has been re

glite-yaim-core

Bug	Description and Workaround
#73080	root umask 022 needed to proper set file permission during yaim configuration. Workaround: `# (umask 022;/opt/glite/yaim/bin/yaim-c -s <location of site-info.def> -n <node-type-1>`
#61454	`GLOBUS_TCP_PORT_RANGE` is not properly defined in SL5 and SL4 TAR UI. The workaround is to replace in `${INSTALL_ROOT}/external/etc/profile.d/grid-env.sh` the line `gridenv_set "GLOBUS_TCP_PORT_RA` `gridenv_set "GLOBUS_TCP_PORT_RANGE" "20000,25000"`
#59862	`-v` functionality is not always supported. This is because yaim _check functions should be implemented with make `-v` option work. Functions will have to be reviewed.
#59017	The function `config_bdii_only` should check whether the bdii is actually installed and fail otherwise. The and this doesn't allow to detect problems.
#58089	YAIM needs to fix the issues described in this bug so that CERNVM can succesfully configure a glite UI. C manual workarounds until this bug is fixed.
#57884	YAIM doesn't support the vo aliasing in the vomses file. See the bug for more details
#56575	The `grid-env.sh` file is not properly recreated when running a partial configuration calling only one functio a copy of the file before launching single function configurations and always make sure the environment is p `grid-env.sh` afterwards.
#56573	Affecting the WMS when running a partial configuration with `config_vomsmap`. The ownership of the gridm running the function and should be set to `chown root:${GLITE_GROUP} ${GRIDMAPDIR}`
#56543	YAIM doesn't respect the case of VO names, VO names are transformed to lower case when YAIM is used groupmap and lsc files. This bug is not going to be fixed for the time being, since JSPG recommends that V case.
#56152	Pool accounts cannot have more than 3 digits. See the bug description for more details.
#53843	The whole VO is authorised even when groups.conf only defines specific VO groups and/or roles, i.e. when have access.
#53462	This bug appears in glite-yaim-core 4.0.8-7 when sys admins define groups.d/ and comment out GROUPS_C site-info.def. YAIM fails to export the GROUPS_CONF variable when groups.d/ are defined. This will be fi suggest to leave GROUPS_CONF defined in site-info.def poiting to a non existing file.
#53271	yaim should only start the BDII once at the end of a configuration. It may happen that the BDII is started bef information is added. This is a problem when configuring a CE + Batch system. You need to restart the BDI is also configured, then this is not a problem. Only some errors are printed in the `/var/log/bdii/bdii-upd`
#52885	Missing closing " in the config_gip_vo_tag function. Please, add them at the end of the affected lines as a w
#50878	`config_mkgridmap` creates a wrong `/etc/cron.d/lcg-expiregridmapdir`, because it uses `${INSTALL_ROOT}/edg/sbin/lcg-expiregridmapdir.pl` (no more present) instead of `${INSTALL_ROOT}/glite/sbin/lcg-expiregridmapdir.pl`
#50872	The rgma client is no longer configured in gLite 3.2 (SL5). YAIM detects the OS where we are running a co `/etc/issue.net/`. If YAIM detects we are in an SL5 machine, the rgma client is not configured. We have d machines this file is empty and therefore YAIM fails to detect the platform. In SL5 tries to configure the rgm this case, please remove `config_rgma_client` from the function list.
#49850	Warning due to a wrong order in the parameters of the command `find`. It can be ignored. It will be fixed in u
#48991	There's a wrong comparison when trying to create the home directory when creating a user. It should be `!=` i more details.

#49831	After introducing the fix for Bug #45887, YAIM has stopped to create `/opt/edg/var/info` directories. Th version of lcg-tags that will be able to write in the new directory `/opt/glite/var/info/SubClusterUnique` released in Patch #2940 that hasn't been certified yet. Old directories need to be supported for a while. The `INSTALL_ROOT/glite/yaim/functions/cofig_gip_vo_tag` and add at the end (the old code to create the `/` directories):
	
	`for VO in $VOS; do`
	`  dir=${INSTALL_ROOT}/edg/var/info/$VO`

	`mkdir -p $dir` `f=$dir/$VO.list` `[ -f $f ] || touch $f` `# work out the sgm user for this VO` `sgmusers=`users_getspecialusers $VO sgm`` `sgmuser=`echo $sgmusers | cut -d " " -f 1`` `vogroup=`users_getvogroup ${VO}`` `sgmgroup=`users_getspecialgroup ${VO} sgm`` `sgmgroup=`id -g -n $sgmuser`` `chown -R ${sgmuser}:${sgmgroup} $dir` `yaimlog DEBUG "$vogroup, $sgmgroup"` `if [ "x$vogroup" = "x$sgmgroup" ]; then` `yaimlog DEBUG "Removing grop writeability of files in $dir,` `yaimlog DEBUG "sgm's primary group is equal to pool account` `chmod -R go-w $dir` `else` `yaimlog DEBUG "Adding grop writeability of files in $dir,"` `yaimlog DEBUG "sgm's primary group is different to pool acc` `chmod -R ug+rw,o-w $dir` `fi` `done`
#46882	Wrong implementation of the bug. gridmap file is not created instead of gridmap dir, when defining CONFI This has to be implemented properly so the bug is still valid.
#46299	Syntax error in config_mkgridmap causes wrong bash shell code in `lcgdm-mkgridmap.conf`. It affects DPM mentioned bug for the patched code.
#45852	PYTHONPATH paths for 64bit WN are incorrect. The g4bit paths should be defined first.
#45269	groups.conf per VO can't be used in the glexec WN, SCAS and SGE_server, CONDOR_server and LSF_ser installed in a separate host from the CE. If you are installing these nodes, please define groups.conf as usual.

#44820	The TAR UI fails to detect the VDT installed version since YAIM relies on `rpm -q` and this doesn't obvious tarball. Due to this reason the `GLOBUS_TCP_PORT_RANGE` variable may be wrongly defined by YAIM. The wo `${GLITE_EXTERNAL_ROOT}/etc/profile.d/grid-env.(c)sh`:
	
	`Change:`
	`gridenv_set "GLOBUS_TCP_PORT_RANGE" "20000 25000"`
	`to:`
	`gridenv_set "GLOBUS_TCP_PORT_RANGE" "20000,25000"`

#43699	site-info.post variables should allow to use a user defined variable in site-info.def. See workaround describe savannah bug.
#43308	Ordinary pool accounts and static accounts can't have more than one group.
#43278	For fresh installations, if `CONFIG_USERS=no`, yaim doesn't create the cleanup-grid-accounts cron job in the C config_users functions manually to extract from the `if [ "x$CONFIG_USERS" = "xyes" ]...` the part cont `accounts` code
#40675	The following error is printed at configuration time due to the mentioned bug: `chown: cannot access `/o file or directory`. and `sed: can't read /opt/bdii/etc/schemas: No such file or directory`.
#40653	`JOB_MANAGER` is required in `config_globus_clients` and it's actually not used. This is in fact breaking som who shouldn't know about this variable. The workaround is to define in site-info.def something like `JOB_MAN` variable is removed from `requires`.
#40208 and #35495	`BDII_LIST` variable should not be used due to the mentioned bugs.
#40093	groups.conf per VO can't be used when the TORQUE server is installed in a separated host.
#40092	YAIM exits with an error if debug level 7 is used.
#39884	Affecting BDII. INFOSYS_GROUP and BDII_GROUP conflict since they have the same gid in edgusers.c
#39254	affecting WMS. The LCG VOMS host certificates rpm needs to be installed in the WMS. If you are planning you'll find the mentioned bug.
#39022	Affecting BDII. BDII_USER should always be the default 'edguser' for the time being until the mentioned b
#37621	grid-env.sh cannot be sourced with zsh. The workaround is to change in `INSTALL_ROOT/glite/yaim/etc/g` ocurrences of `mytmp="${!myvar}"` with `mytmp="`eval echo \\$$myvar`"`

#35373	Affecting config_vomsdir. Sites willing to configure the .lsc files can't use vo.d/ directory structure to define should define them in `site-info.def` until the bug is fixed.
#35307	Affecting the configuration directory permissions. When the permission of site-info.def parent directory is n that the permission of current directory is not correct. In reality it should complain about the permissions of This checking will disappear in the next release. If you want to get rid of it, please comment lines 211-217 a /opt/glite/yaim/bin/yaim.
#33314	Affecting the site BDII. When the BDII is installed together with the lcg CE, the `/opt/glite/libexec/gli` tries to create `/var/glite/tmp` and it doesn't have permissions. The workaround is to restart the site BDII. T reconfiguration.
#32786	Affecting config_lcgenv and the definition of PX_HOST in the environment
#32727	Affecting config_vomses and the removal of /opt/edg/etc/vomses.
#31288	Affecting the syntax checking of site-info.def. This is actually not done by yaim and incorrect site-info.def f detected.
#29878	`GLOBUS_TCP_PORT_RANGE` needs to be edited manually in the form of "num1,num2" for VDT versions 1.6, o versions < 1.6. This will be automatically done in the next YAIM release.
#29764	The configuration target `CE` is incorrect but YAIM doesn't complain. It should be used `lcg-CE`

glite-yaim-cream-ce

Bug	Description and Workaround	Fixed in version
#47397	CREAM doesn't allow for the publication of FQAN VOViews.	4.0.8-2
#45844	Yaim based CREAM conf. procedure doesn't call config_gip_vo_tag, which is needed to publish in the BDII the VO software tags.	4.0.7-3
#44712	Because of a problem in the lcmaps configuration, in some cases the same user is mapped to different local accounts by glexec and gridftpd. The workaround is to replace the file `/opt/glite/etc/lcmaps/lcmaps-suexec.db` with this one.	4.0.7-2
#43399	yaim-cream-ce renames everything in `/etc/grid-security/vomsdir/file` as file.pem, including directories.	4.0.7-2
#39684	Version wrongly reported in `/opt/glite/yaim/etc/versions/glite-yaim-cream-ce`	4.0.7-2

glite-yaim-hydra

Bug	Description and Workaround	Fixe in vers
#53856	tomcat is not properly started after configuring HYDRA with yaim. Please, edit `/etc/tomcat/tomcat5.conf` and uncomment the JAVA_HOME definition. This will be fixed when the yaim core function `config_secure_tomcat` is used	No so f
#43945	The following error appears the first time the configuration is run: `ERROR 1045 (28000): Access denied for user 'hydra1'@'localhost' (using password: YES)` . It will dissappear after running the configuration a second time.	No so f

glite-yaim-fts

Bug	Description and Workaround	Fixed in version
#53858	tomcat is not properly started after configuring FTS with yaim. Please, edit `/etc/tomcat/tomcat5.conf` and uncomment the JAVA_HOME definition. This will be fixed when the yaim core function `config_secure_tomcat` is used	No fix so far

glite-yaim-lb

Bug	Description and Workaround	Fixed in version
#36336	There are conflicts between different YAIM functions to configure `GLITE_LOCATION_VAR` causing that LB services can not be stopped and started by daemon scripts. A workaround is to define `GLITE_LOCATION_VAR=/var/glite` in the file `/opt/glite/yaim/defaults/glite-lb.pre` before configuring the LB service.	4.2.1-1

glite-yaim-lcg-ce

Bug	Description and Workaround	Fixed in version
None	glite-yaim-lcg-ce 4.0.4-2 depends on yaim core 4.0.4-1. If they are not installed together, the config_gip_ce_check function will fail. This error message dissapears when removing from the requires list the __GROUP_ENABLE variable in the config_gip_ce_check function.	4.0.5-6
#54530	SE_MOUNT_INFO_LIST is not properly implemented. It allows to define one mount point per SE for all the available queues.	No fix so far
#80537	If wanted, LCGCE_CLUSTER_MODE has to be set in the site-info.def file. Setting it in services/lcg-ce does not work properly.	No fix so far

glite-yaim-lsf-utils

Bug	Description and Workaround	Fixed in version
#49404	There's an extra space in function config_apel_lsf_check after "\". This has to be removed or the 'requires' won't control the variables appearing on the second line.	No fix so far

glite-yaim-myproxy

Bug	Description and Workaround	Fixed in version
#48440	Remove extra spaces in `INSTALL_ROOT/glite/yaim/examples/siteinfo/services/glite-px`.	4.0.4-1

glite-yaim-torque-utils

Bug	Description and Workaround	Fixed in version
#39014	There's an extra space in function config_apel_pbs_check after "\". This has to be removed or the 'requires' won't control the variables appearing on the second line.	4.0.3-1

glite-yaim-torque-server

Bug	Description and Workaround	Fixed in version
None	YAIM doesn't support multiple CEs configuration: if the same torque server is configured with several CEs, the `/etc/hosts.equiv` file has to be modified manually to include the hostnames of the other CEs.	No fix so far

glite-yaim-wms

Bug	Description and Workaround	Fixed in version
#58347	`LCG_GFAL_INFOSYS` is not defined in the WMS environment. As a workaround, please add the following line to the `grid-env.sh` file in the WMS: `gridenv_set LCG_GFAL_INFOSYS value`, changing `value` for something that make sense in your site (BDII or list of BDIIs)	4.0.7-1
#53297	In `/opt/glite/etc/glite_wms.conf` the "--ftpconn" values typically need to be increased from 30 e.g. to 300, to avoid the limiter refusing jobs too frequently.	No fix so far, but see Proposed workaround
#53294	The WMProxy logging is fairly useless at its default level, so the admin may want to increase it from 5 to 6.	No fix so far, but see Proposed workaround

Basics

What is YAIM

The aim of YAIM (Yet Another Installation Manager) is to implement a configuration method for the gLite software. YAIM is a set of bash scripts and functions. YAIM is distributed in rpm form and it usually resides in `/opt/glite/yaim`.

In order to configure a site, one or more configuration files are edited and the YAIM script is then executed. Since YAIM is mainly bash, all the configuration files have to follow the bash syntax. For example, no space between the equal sign and the key-value variables are allowed.

WRONG :

```
SITE_NAME = NorthPole
```

CORRECT:

```
SITE_NAME=NorthPole
```

A good syntax test for the `site-info.def` is to source it:

```
source ./site-info.def
```

and look for errors. YAIM will check this for you anyway. The configuration procedure is described in the following sections.

Modular structure

YAIM is distributed in several rpms. The `glite-yaim-core` contains common functions and definitions, while the other packages like `glite-yaim-clients` implement the functionality to configure specific node types. The appropriate yaim package will be installed with the service metapackage. The available yaim modules are:

- glite-yaim-amga
- glite-yaim-bdii

- glite-yaim-clients (UI, WN and VOBOX)
- glite-yaim-core
- glite-yaim-condor-utils
- glite-yaim-cream-ce
- glite-yaim-dcache
- glite-yaim-dpm
- glite-yaim-e2emonit
- glite-yaim-fts
- glite-yaim-hydra
- glite-yaim-lb
- glite-yaim-lcg-ce
- glite-yaim-lfc
- glite-yaim-lsf-utils
- glite-yaim-mon
- glite-yaim-mpi
- glite-yaim-myproxy
- glite-yaim-rb
- glite-yaim-se-classic
- glite-yaim-sge-utils
- glite-yaim-torque-client
- glite-yaim-torque-server
- glite-yaim-torque-utils
- glite-yaim-voms
- glite-yaim-wms

The configuration variables

Configuration files should be stored in a directory structure. All the involved files should be under the same folder, in a safer place which is not world readable. This folder should contain:

- *site-info.def*: It contains a list of configuration variables in the format of key-value pairs. It's a mandatory file and it's a parameter passed to the yaim command.

Optionally, the configuration folder can contain the following directories:

- *services*: it contains a file per node type with the format *glite-node-type*. The file contains a list of configuration variables specific to that node type. In the future, each yaim module will distribute an example file in `/opt/glite/yaim/examples/siteinfo/services/glite-node-type`.
- *vo.d*: it contains a file per VO with the format *vo_name*.
- *nodes*: it contains a file per host with the format *hostname.domain_name*. The file contains host specific variables that are different from one host to another in a certain site.
- *group.d*: it contains a file per VO with the format *groups-<vo-name>.conf*. The file contains VO specific groups and it replaces the former *groups.conf* file where all the VO groups were specified all together.

The optional folders are created to allow system administrators to organise their configurations in a more structured way. However, it's still possible to use only the site-info.def file and place all the necessary variables for one site there. In case the optional folders are used, this is the sourcing flow (where *siteinfo* refers to the path to the configuration folder):

- siteinfo/site-info.def
- siteinfo/services/*
- siteinfo/nodes/*
- siteinfo/vo.d/*

YAIM distributes an example of *site-info.def* and *services/* under `/opt/glite/yaim/examples/siteinfo/site-info.def`. In case the system administrator is interested in using these files, it should move them to a safer location.

YAIM also distributes an example of edgusers.conf, users.conf and groups.conf files under `/opt/glite/yaim/examples/`. These files can be placed at any location since their path is specified in the variables USERS_CONF and GROUPS_CONF in site-info.def. However, edgusers.conf location is defined by the variable `EDGUSERS` that defines the default path `/opt/glite/yaim/examples/edgusers.conf`. We recommend not to change this variable and use the default value.

site-info.def

This is the main configuration file of YAIM. It is installed by the glite-yaim-core rpm and it's located under `/opt/glite/yaim/examples/siteinfo/site-info.def`. `site-info.def` distributes only those variables common to the configuration of the different yaim modules that need to be defined by sys admins.

Variables that contain a meaningful default value are distributed under `/opt/glite/yaim/defaults/site-info.pre or post`. `/opt/glite/yaim/defaults/site-info.post` is for those variables that depend on `INSTALL_ROOT`.

Find a description of the different general variables in the site info configuration variables wiki:

- site-info.def configuration variables
    - VO related variables
- site-info.pre configuration variables
- site-info.post configuration variables

System administrators are free to choose the configuration structure they prefer. It's possible to keep all the configuration variables in big `site-info.def` or maintain a smaller `site-info.def` together with `services/`, `nodes/` and/or `vo.d/` directories.

services directory

This directory should be located under `siteinfodir/services`, being `siteinfodir` the directory where you store the YAIM configuration. The `services` directory is created to make easier the configuration of different node types present in one site. Each yaim module distribute a file containing node type specific variables. The file name is *glite-node-type* and like site-info.def, it contains a list of key-value pairs. The site-info.def example file distributed by yaim core contains only the variables that are used by several node types. It's up to the system administrator to decide whether to keep a modular configuration or keep using a single site-info.def file. For a list of service specific variables, please check: YAIM configuration variables wiki.

vo.d directory

This directory should be located under `siteinfodir/vo.d`, being `siteinfodir` the directory where you store the YAIM configuration. The `vo.d` directory was created to make the configuration of the DNS-like VOs easier. It contains a file name per VO whose name has to be the lower-cased version of the VO name. The matching file should contain the definitions for that VO. In case the VO is also defined in `site-info.def`, the vo.d file will overwrite the variables which are defined there. Again, bash syntax should be followed. The syntax is different compared to that of `site-info.def`. In vo.d files, the `VO_(VONAME)` prefix should be omitted. For example while in `site-info.def`:

```
VO_BIOMED_SW_DIR=$VO_SW_DIR/biomed
VO_BIOMED_DEFAULT_SE=$CLASSIC_HOST
VO_BIOMED_STORAGE_DIR=$CLASSIC_STORAGE_DIR/biomed
```

in `vo.d/biomed` file:

```
SW_DIR=$VO_SW_DIR/biomed
DEFAULT_SE=$CLASSIC_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/biomed
```

In case you are declaring a DNS-like VO name in site-info.def, remember to change the "." or "-" with "_", like in the example:

```
VO_VO_TEST_DOMAIN_ORG_SW_DIR=$VO_SW_DIR/test
VO_VO_TEST_DOMAIN_ORG_DEFAULT_SE=$CLASSIC_HOST
VO_VO_TEST_DOMAIN_ORG_STORAGE_DIR=$CLASSIC_STORAGE_DIR/test
VO_VO_TEST_DOMAIN_ORG_VOMS_SERVERS="vomss://voms.domain.org:8443/voms/vo.test.domain.org"
VO_VO_TEST_DOMAIN_ORG_VOMSES="vo.test.domain.org voms.domain.org 15001 /DC=ORG/O=DOMAIN/O=Hosts/C
```

to include multiple voms servers for a VO, the syntax is:

```
VO_VO_TEST_DOMAIN_ORG_VOMS_SERVERS="'vomss://voms.domain.org:8443/voms/vo.test.domain.org' 'vomss
```

ie, server URL enclosed by single quotes separated by spaces, and enclosed within double quotes

After the experience of AEGIS01-PHY-SCL and EGEE SEE ROC, who have been deploying DNS-like VO names, the following recipe has been prepared with their contribution. **Note that this has proven to work for the mentioned ROCs but this is not the standard approach and the YAIM team hasn't certified this solution**:

If you are deploying a VO with DNS-like name (example: vo.test.domain.org), and you want to declare the VO related variables in site-info.def, you would first need to choose a shorter name for the VO (example: test). Then the short name can be used instead of the DNS-like name. However there are exceptions and still in some of them, the full VO DNS-like name has to be used.

- site-info.def contains several relevant variables:
  - ♦ VOS should contain full VO names; example:

```
VOS="atlas cms alice lhcb dteam ops vo.test.domain.org"
```

  - ♦ QUEUES should contain short VO names, in the same order as entered in $VOS (note that the default is QUEUES=${VOS}, which now cannot be used); example:

```
QUEUES="atlas cms alice lhcb dteam ops test"
```

  - ♦ If you are configuring LFC and setting LFC_LOCAL and LFC_CENTRAL variables to select for which VOs your LFC will be local and for which central, those variables should contain full VO names; example:

```
LFC_LOCAL="atlas cms alice lhcb dteam ops"
LFC_CENTRAL="vo.test.domain.org"
```

  - ♦ <QUEUE-NAME>_GROUP_ENABLE variable should be named after short VO name (in capital letters), but should contain full VO name, as well as all roles and groups for that VO defined in groups.conf (except for the root); example

```
TEST_GROUP_ENABLE="vo.test.domain.org
/vo.test.domain.org/ROLE=lcgadmin
/vo.test.domain.org/ROLE=production"
```

- If you choose to put VO-specific variables in the file in vo.d named after the full VO name (example:

vo.d/vo.test.domain.org), it should contain at least the following variables: SW_DIR, DEFAULT_SE, STORAGE_DIR, VOMS_SERVERS, VOMSES:

```
SW_DIR=$VO_SW_DIR/test
DEFAULT_SE=$CLASSIC_HOST
STORAGE_DIR=$CLASSIC_STORAGE_DIR/test
VOMS_SERVERS="vomss://voms.domain.org:8443/voms/vo.test.domain.org"
VOMSES="vo.test.domain.org voms.domain.org 15001 /DC=ORG/O=DOMAIN/O=Hosts/CN=host/voms.doma
```

- Such setup is tested to work; however, note that still some bugs exist and that some manual steps may be needed:Bug #27817.
- Other VO-related variables (RBS, VOMS_POOL_PATH) can be also defined if needed.
- users.conf file entries should be based on short VO name (example: test)
- groups.conf should contain full VO name; example:

```
"/vo.test.domain.org/ROLE=lcgadmin":::sgm:
"/vo.test.domain.org/ROLE=production":::prd:
"/vo.test.domain.org"::::
```

nodes directory

This directory should be located under `siteinfodir/nodes`, being `siteinfodir` the directory where you store the YAIM configuration. The `nodes` directory is created to make easier the configuration of variables that have different values depending on the host in the site. The file name is *hostname.domain_name* and like site-info.def, it contains a list of key-value pairs.

Example of two hosts which support different VOs:

```
lxb1430.cern.ch specific parameters
VOS=dteam

# lxb1431.cern.ch specific parameters
VOS="atlas alice"
```

defaults directory

This directory is located under `/opt/glite/yaim/defaults`. It contains variables with a meaningful default value provided by YAIM and that don't need to be changed unless you are an advanced user and you know what you are doing. The files are:

- site-info.pre
- site-info.post
- node-type.pre
- node-type.post

In case you really need to change these variables, you don't need to modify the value in these files if you don't want to edit them. You can just add the same variable in site-info.def since this will overwrite the variables declared in these files. See the configuration flow in YAIM in the next section.

Configuration flow in YAIM

This is the order in which the different configuration files are sourced:

1. /opt/glite/yaim/defaults/site-info.pre
2. /opt/glite/yaim/defaults/glite-node-type.pre
3. siteinfo_dir/site-info.def
4. siteinfo_dir/services/glite-node-type

5. siteinfo_dir/nodes/machine.domain
6. /opt/glite/yaim/defaults/site-info.post
7. /opt/glite/yaim/defaults/glite-node-type.post
8. siteinfo_dir/vo.d/vo_name
9. /opt/glite/yaim/node-info.d/glite-node-type

User configuration in YAIM

Two types of users are needed by the middleware and created by YAIM: users defined in `users.conf` and users defined in `edgusers.conf`. User creation can be disabled in YAIM if you prefer to do this on your own. In order to do that you just need to define the variable:

```
CONFIG_USERS=no
```

And then make sure you create `users.conf` and `edgusers.conf` list of users in your system. Moreover, you would need to provide a proper `users.conf` reflecting the users you've defined in your system. This file is used by many functions so it should be defined and should be coherent with your system.

users.conf

This file defines the UNIX users to be created on the service nodes that need them (mainly CE and WNs). The format is as follows (fields must not have any white space):

```
UID:LOGIN:GID1[,GID2,...]:GROUP1[,GROUP2,...]:VO:FLAG:
```

- UID = user ID. This must be a valid uid. Make sure the number you choose is not assigned to another user.
- LOGIN = login name
- GID1 = primary group ID. This must be a valid gid. Make sure the number you choose is not assigned to another group.
- GID2 = secondary group ID.
- GROUP1 = primary group
- GROUP2 = secondary group
- VO = virtual organization
- FLAG = string to identify special users, further described below

You can customise this file to your site needs. YAIM only provides an **example file**.

Ordinary pool accounts

Pool accounts enable the dynamic allocation of local UNIX user names to grid users. For each supported VO, a set of ordinary pool accounts should be defined.

- Pool account identifiers must end in digits following a dedicated common base string, like `ops001`.
- Ordinary pool accounts have an empty FLAG.
- Ordinary pool accounts should belong to only one group. This is a known issue: bug #43308. It will be fixed in glite-yaim-core 4.0.6-x.

Examples:

```
45001:ops001:45000:ops:ops::
45002:ops002:45000:ops:ops::
45003:ops003:45000:ops:ops::
```

Special pool accounts

Subsets of users in a VO may be mapped to dedicated sets of accounts, e.g. to receive a higher priority in the batch system or to have access to some dedicated queue. For each such category the site admin can define a `FLAG` to identify the corresponding accounts. In groups.conf the same `FLAG` has to be used to mark the VOMS attributes corresponding to the subset of users. See the groups.conf section for further details.

- Special pool account identifiers must end in digits following their own common base string, like `sgmops01`.
- Special pool accounts have a `FLAG` identifying the type of special user, like `sgm`.
- Special pool accounts should belong to more than one group.

In the case of the LHC experiment VOs, three special cases are identified:

- `sgm` - sgm users (with write permission on the shared software area)
- `prd` - prd users (with production manager privileges, if needed)
- `PILOT_JOB_FLAG` - pilot users allowed to run the `glexec` command. This variable is part of the glexec WN configuration, not yet released to production. If you are running pps tests you can direclty choose any identifier. Check in the VO-ID card which is the value given by the VO.

Special pool accounts normally have their own group as primary group and the group of the whole VO as a secondary group. For example, this allows the shared software area to be made group-writable for the `sgm` users and world-readable for the rest of the VO:

```
60701:sgmops01:46001,45000:opssgm,ops:ops:sgm:
60702:sgmops02:46001,45000:opssgm,ops:ops:sgm:
60703:sgmops03:46001,45000:opssgm,ops:ops:sgm:
```

YAIM complains if the special pool accounts do not have multiple groups. If a different primary group is *not* desired for such accounts, the VO group could be specified twice as a work-around. Examples:

```
60701:sgmops01:45000,45000:ops,ops:ops:sgm:
60702:sgmops02:45000,45000:ops,ops:ops:sgm:
60703:sgmops03:45000,45000:ops,ops:ops:sgm:
```

The following syntax is more appropriate to achieve the same result (available since glite-yaim-core >= 4.0.4-x):

```
60701:sgmops01:45000,-:ops,-:ops:sgm:
60702:sgmops02:45000,-:ops,-:ops:sgm:
60703:sgmops03:45000,-:ops,-:ops:sgm:
```

This syntax explicitly signals that no secondary group is wanted.

Static accounts

Special users may use static accounts instead of pool accounts. Although this is not recommended (*), YAIM supports it.

- Static account identifiers do not have digits at the end, like `opssgm`.
- Static accounts have a FLAG identifying the type of special user, like `sgm`.
- Static pool accounts should belong to only one group. This is a known issue: bug #43308. It will be fixed in glite-yaim-core 4.0.6-x.

Example:

```
65000:opssgm:45000:ops:ops:sgm:
```

(*) static accounts are *not* recommended. Pool accounts have better audit trails and allow batch systems to apply fair shares to all users, since each account is associated with a unique proxy. For more details on sgm/prd pool accounts please visit: How to switch to pool accounts for sgm/prd users

edgusers.conf

This file defines the users to be created on the service nodes to run the relevant daemons and processes. The format is as follows:

```
UID:LOGIN:GID:GROUP:DESCRIPTION:HOME_DIRECTORY
```

- UID = user ID. This must be a valid uid. Make sure the number you choose is not assigned to an existing user.
- LOGIN = login name
- GID = group ID. This must be a valid gid. Make sure the number you choose is not assigned to an existing group.
- GROUP = group name
- DESCRIPTION = user description
- HOME_DIRECTORY = optional variable to specify a home directory

You can customise this file to your site needs. YAIM only provides an **example file**.

The necessary grid system users and groups identifiers are defined through a set of variables in `${INSTALL_ROOT}/glite/yaim/defaults/site-info.pre`, for more information you can check the site-info.pre variables wiki. If you prefer to choose other identifiers, you can redefine the corresponding variables in your site-info.def changing the default name. But we recommend you don't change this unless you know very well what you are doing.

The list of current users is:

- DPM manager. Used by the DPM.
- LFC manager. Used by the LFC.
- RGMA user. Used by MON box.
- BDII user. Used by BDIIs (resource, site and top).
- edguser user. Currently hardcoded and used in DPM and FTA.
- edginfo user. Currently hardcoded and used in DPM.
- glite user. Currently used in WMS and LB.

The list of current groups is:

- DPM manager group. Used by the DPM.
- LFC manager group. Used by the LFC.
- RGMA group. Used by MON box.
- BDII group. Used by BDIIs (resource, site and top).
- edguser group. Currently hardcoded and used in DPM and FTA.
- edginfo group. Currently hardcoded and used in DPM.
- infosys group. Currently hardcoded and used in DPM. Also used by the GIP (this affects almost all the services).
- glite group. Currently used in WMS and LB.

Bear in mind that all these users are created in your system. It may turn out that depending on the service you are configuring, you don't need all of these users. You can then remove the ones you don't need from edgusers.conf.

Group configuration in YAIM

groups.conf

groups.conf defines the user categories that must be accepted by the grid services provided by a site. It indicates for each category to which kind of local accounts the user should be mapped, where applicable. The file has the following format:

```
"VOMS_FQAN":GROUP:GID:FLAG:[VO]
```

- VOMS_FQAN = VOMS proxy fully qualified attribute name
- GROUP = UNIX group
- GID = UNIX GID
- FLAG = string to identify special users, further described below
- VO = virtual organization (optional. It allows the VO to be specified explicitly, otherwise it will be derived from the VOMS FQAN

The groups.conf distributed by YAIM is only an example. You can remove the lines that doesn't apply to your site or VO and add new lines if needed. Example:

```
"/dteam/ROLE=lcgadmin":::sgm:
"/dteam/ROLE=production":::prd:
"/dteam"::::
```

The groups.conf file lists the VOMS proxy *primary* FQANs that are accepted.

If a proxy has a *secondary* FQAN that matches one of the FQANs listed, the mapped account may receive an extra secondary GID corresponding to the matched FQAN. That GID normally is derived from the corresponding accounts in the users.conf file. If there are no accounts dedicated to that FQAN, the desired extra GID (if any) and GROUP name must be given in groups.conf.

Note that it is normal for the second and third fields to be empty, as shown in the example.

Note that the account corresponding to the primary FQAN does *not* have to belong to any secondary group: the LCMAPS library can set secondary groups independently of what is in /etc/group.

Note that the order of the lines in groups.conf is important: for any FQAN only the first match is taken.

The **FLAG** selects a set of special accounts to be used for the mapping, namely those accounts in users.conf that have the same flag. By default, when the flag is empty, the ordinary pool accounts will be used.

In the case of the LHC experiment VOs, three special cases are identified:

- `sgm` - sgm users (with write permission on the shared software area)
- `prd` - prd users (with production manager privileges, if needed)
- `PILOT_JOB_FLAG` - pilot users allowed to run the `glexec` command. This variable is part of the glexec WN configuration, not yet released to production. If you are running pps tests you can direclty choose any identifier. Check in the VO-ID card which is the value given by the VO.

A set of accounts (and hence a flag) may be used for multiple FQANs.

Different VOs can use the same flag names independently.

Beware that wildcards may have unexpected side effects, in particular on gLite 3.0 (SL3):

- bug #29866
- bug #26990

YAIM offers the possibility of configuring wildcards by defining the variable `VO_<VO-NAME>_MAP_WILDCARDS=yes`. A new line containing wildcards will be added per defined FQAN.

group.d directory

It's possible to maintain a list of *groups.conf* file per supported VO under the *group.d* directory. This file should be located in the configuration directory as explained in the configuration variables section. The name of the file should be of the format *groups-<vo-name>.conf* and it contains the same syntax as explained above.

This directory would replace the *groups.conf* file, so sys admin should choose the format that better fits their needs.

Until bug 53462 is fixed, sys admins using groups.d/, should also define `GROUPS_CONF` pointing to a non existing file.

local groups.conf

A file with local groups supported by a site can also be defined. In order to do that you have to define the variable `LOCAL_GROUPS_CONF` which will point to the local_ groups.conf_ file that should follow the syntax explained above.

WN list

There's a variable in YAIM called `WN_LIST` that specifies the location of a file. This file contains the list of WN hostnames (FQDN) in the site. The syntax has been improved to deal with the configuration of the `glite-wn-info` utility.

The `glite-wn-info` utility is designed to be executed on the WN by a job submitter. It returns information about that worker node in a grid context. Initially only the following information is supported:

```
$ glite-wn-info -n GlueCEUniqueClusterId
mysubcluster-id
```

See PATCH:2114 for more details.

The possible syntax for `WN_LIST` are:

The classic one:

```
wn-hostname1
wn-hostname2
...
wn-hostname3
```

The one that is introduced for `glite-wn-info`:

```
wn-hostname1:subcluster1-id
wn-hostname2:subcluster2-id
...
wn-hostname3:subclusterX-id
```

Where `subclusterX-id` is the unique identifier of the subcluster the WN belongs to. Subcluster identifiers are for the time being the lcg CE host where the subcluster is published.

Both syntax are supported. If the classic syntax is used, the subcluster identifier will be deduced from `${CE_HOST}`. Since the `CE_HOST` variable is not a mandatory variable in the WN configuration, make sure you define it or yaim will complain.

If you use the new syntax, you need to manually specify the lcg CE host name as the subcluster identifier. You can't use the variable `${CE_HOST}`. That is, write something like:

```
wn-hostname1:my-ce-host.cern.ch
```

And **NOT**:

```
wn-hostname1:${CE_HOST}
```

In order to be backwards compatible, only the classic syntax is currently supported if you want to define only one `WN_LIST` variable for the whole site (classic syntax is required by TORQUE and SGE batch system configuration). But if you want to benefit from the improved syntax in your WNs configuration, you can define a `yoursiteinfodir/services/glite-wn` configuration file defining a `WN_LIST` that points to a different file with the improved syntax.

For tarball WN installations, a new feature has been included in yaim core >= 4.0.10-1. There will be a new tag: `tarball-installations`, that can be used in `WN_LIST` as follows:

```
tarball-installations:subcluster-id
```

This allows to assign one subcluster id for a set of TAR WNs, using the same `WN_LIST` file for the installation of all these TAR WNs. The use of this tag is optional though:

If a TAR WN is being installed:

1. and the tag `tarball-installations` is present in `WN_LIST`, then the subcluster-id will be retrieved from there.
2. and the tag `tarball-installations` is not present, the subcluster-id will be also deduced from `CE_HOST`.

For versions of yaim core < 4.0.10-1, tarball installations should temporarily define the hostname of the machine where they are running the configuration in `WN_LIST`, otherwise YAIM will fail.

Running the configuration

The interface

YAIM comes with a script in `/opt/glite/yaim/bin/yaim`. This script should be used to perform the different configuration steps.

```
Usage: /opt/glite/yaim/bin/yaim <action>  <parameters>
Actions:
       -c | --configure   : Configure already installed services.
                            Compulsory parameters: -s, -n

       -r | --runfunction : Execute a configuration function.
                            Compulsory parameters: -s, -f
                            Optional parameters  : -n

       -v | --verify      : Goes through on all the functions and checks that
                            the necessary variables required for a given
                            configuration target are all defined in site-info.def.
                            Compulsory parameters: -s -n
```

```
       -d | --debug       : Define a loglevel, which overwrites the value of
                            YAIM_LOGGING_LEVEL defined in site-info.def.
                            Values: 1-7

       -e | --explain     : Doesn't perform configuration but explains what the
                            functions are doing by printing out the comments
                            found inside them.
                            Compulsory parameters: -s -n

       -a | --available   : Prints out the available configuration targets.
                            Compulsory parameter:  -s

       -p | --package     : Creates an rpm package from the configuration directory structure,
                            that can be installed on other nodes. It installs site-info.def and
                            if they exist: vo.d, group.d, nodes and local functions under:
                            /opt/glite/yaim/examples/siteinfo/mysiteinfo-${SITE_INFO_VERSION}

                            Compulsory parameter:  -s

       -h | --help        : Prints out this help.

Specify only one action at a time !

Parameters:
       -s | --siteinfo:   : Location of the site-info.def file
       -n | --nodetype    : Name of the node type(s) to configure
       -f | --function    : Name of the functions(s) to execute

Examples:

       Check what can you configure:
         /opt/glite/yaim/bin/yaim -a -s /etc/yaim/site-info.def

       Configuration:
         /opt/glite/yaim/bin/yaim -c -s /etc/yaim/site-info.def -n SE_dpm_mysql

       Running a function:
         /opt/glite/yaim/bin/yaim -r -d 6 -s /etc/yaim/site-info.def -n SE_dpm_mysql -f config_mk

       Verify your site-info.def:
         /opt/glite/yaim/bin/yaim -v -s /etc/yaim/site-info.def -n SE_dpm_mysql

       Create an rpm package from site-info.def:
         /opt/glite/yaim/bin/yaim -p -s /etc/yaim/site-info.def
```

When configuring multiple node types, they have to be defined all together on the same yaim command, for example:

```
./yaim -c -s /root/siteinfo/site-info.def -n glite-SE_dpm_mysql -n glite-BDII
```

It's very important to do this since otherwise the grid environment won't be properly defined.

Installing a node

IMPORTANT NOTE: YAIM doesn't support installation of gLite 3.1 and gLite 3.2 nodes but just its configuration

For installation of gLite 3.1, please have a look at the install section of the 3.1 Generic Install Guide.

For installation of gLite 3.2, please have a look at the install section of the 3.2 Generic Install Guide.

Configuring a node

If the installation was successful one should run the configuration:

```
./yaim -c -s <location of site-info.def> -n <node-type-1> -n <node-type-2> ...
```

Each node type is a configuration target and **if there is more than one installed and to be configured on a host then the configuration must be run together and not separately**.

IMPORTANT NOTE: It is becoming more common for root to have umask 077 by default, which may cause any files and directories created by YAIM to be readable only for root, whereas almost all of them need to be world-readable and the few that should be restricted have explicit chown/chgrp/chmod commands in their corresponding YAIM functions. Until bug #73080 is solved, before running the configuration, you have to set the root umask to "022", by running something like:

```
(umask 022;/opt/glite/yaim/bin/yaim -c -s <location of site-info.def> -n <node-type-1> -
```

Configuration target list

The available configuration targets are listed below (The prefix 'glite-' can be optionally added, except for the lcg CE and the cream CE):

Node Type	Configuration target (node type)	Description
APEL	glite-APEL	APEL box
Argus	glite-ARGUS_server	Argus server
AMGA Oracle	AMGA_oracle	AMGA server with Oracle db backend
AMGA Postgres	AMGA_postgres	AMGA server with Postgres db backend
site BDII	BDII_site	A site BDII
top level BDII for 3.0	BDII	A top level BDII
top level BDII for 3.1	BDII_top	A top level BDII
cream CE	creamCE	The cream Computing Element
lcg CE	lcg-CE	The LCG Computing Element
e2emonit	E2EMONIT	RGMA-based monitoring system collector server
glexec WN	GLEXEC_wn	glexec support for the WN
FTS	FTS2	gLite File Transfer Server
FTA	FTA2	gLite File Transfer Agent
FTM	FTM2	glite File Transfer Monitor
gLite LB	LB	LB node
LCG File Catalog server with mysql	LFC_mysql	Set up a mysql based LFC server
LCG File Catalog server with oracle	LFC_oracle	Set up a oracle based LFC server
MON-Box	MON	RGMA-based monitoring system collector server
MPI for CE	MPI_CE	MPI configuration for the computing element
MPI for WN	MPI_WN	MPI configuration for the worker node
Proxy	PX	Proxy Server
Resource Broker	RB	Resource Broker
SCAS	SCAS	SCAS server
Classic Storage Element	SE_classic	Storage Element on local disk
Disk Pool Manager (mysql)	SE_dpm_mysql	

		Storage Element with SRM interface and mysql backend
Disk Pool Manager disk	SE_dpm_disk	Disk server for SE_dpm
dCache Storage Element	SE_dcache	Storage Element interfaced with dCache
Re-locatable distribution	glite 3.0: TAR_UI or TAR_WN	It can be used to set up a Worker Node or a UI
	glite 3.1: WN_TAR or UI_TAR	
User Interface	UI	User Interface
VO agent box	VOBOX	Machine to run VO agents
VOMS mysql (under development)	VOMS_mysql	VOMS server with MySQL DB backend
VOMS oracle (under development)	VOMS_oracle	VOMS server with Oracle DB backend
gLite WMS	WMS	WMS node
Worker Node (middleware only)	WN	It does not configure any LRMS

Notes on configuring AMGA

The glite-AMGA can be installed with the AMGA yaim module. This module can configure the AMGA server. In order to configure the node you have to include the AMGA postgres configuration variables. Note that *the amga yaim module inserts the test_user's DN and root's DN **ONLY** on the first successful configuration of the service*. If you want to include DN's of other users in order to give them access you have to insert their DN's "by hand" by executing something like the following command into an AMGA client(mdclient or mdcli):

```
> add_user_subject root 'C = CY, O = CyGrid?, O = UCY, CN = Name Surname'

> add_user_subject test_user 'C = CY, O = CyGrid?, O = UCY, CN = Another Name'
```

Something similar will have to be executed if you want to remove users.

Notes on configuring APEL

YAIM configures APEL when you also configure a batch system (by configuring TORQUE_utils, LSF_utils, SGE_utils or CONDOR_utils). In case that only a CE node type is configured, YAIM won't configure APEL and sys admins need to configure it manually.

Notes on configuring a Batch system

YAIM only provides configuration steps for the CE batch system interaction:

- Torque
- SGE
- Condor
- LSF (partial)

YAIM configures the CE batch system interface and just provides a very basic batch system configuration (for some of the batch systems) that can be enhanced by the site admin later on.

In order to configure the batch system, the following configuration targets can be chosen:

Node Type	Configuration target (node type)	Description

Torque server	TORQUE_server	It configures the 'Torque' LRMS server
Torque utils	TORQUE_utils	It configures utilities for the 'Torque' LRMS server
Torque client	TORQUE_client	It configures the 'Torque' LRMS client
LSF utils	LSF_utils	It configures utilities for the 'LSF' LRMS server
SGE server	SGE_server	It configures the 'SGE' LRMS server
SGE utils	SGE_utils	It configures utilities for the 'SGE' LRMS server
SGE client	SGE_client	It configures the 'SGE' LRMS client
Condor server	Condor_server	It configures the 'Condor' LRMS server
Condor utils	Condor_utils	It configures utilities for the 'Condor' LRMS server
Condor client	Condor_client	It configures the 'Condor' LRMS client

Torque

In case you want to install the TORQUE server together with the lcg CE, the following configuration targets must be used:

```
-n lcg-CE -n TORQUE_server -n TORQUE_utils
```

If the Torque server is running in a separate host:

```
-n TORQUE_server -n TORQUE_utils
```

Note that the proposed order needs to be respected.

Please, check some special cases due to missing features or bugs in previous versions of yaim:

- For glite-yaim-core 4.0.0-12:

```
./yaim -c -s site-info.def -n CE_torque
```

Note that

```
-n lcg-CE -n TORQUE_server
```

is not correct since some TORQUE utils that are needed won't be installed.

- For glite-yaim-core 4.0.1-6 + glite-yaim-lcg-ce 4.0.1-6:

```
-n lcg-CE -n TORQUE_server -n TORQUE_utils
```

If you install the TORQUE server in a separate node than the CE, please install TORQUE_utils both in the CE host and the TORQUE server host. In the case of the TORQUE server host, do the next manual steps if you are running glite yaim torque-utils 4.0.1-4:

- remove from the yaim torque utils function list in /opt/glite/yaim/node-info.d/glite-torque_utils:
  - config_gip_sched_plugin_pbs
  - config_torque_submitter_ssh

These steps will be fixed in yaim for the next release of Torque utils.

Then you have to run:

```
-n TORQUE_server -n TORQUE_utils
```

- For 3.0 WN:

♦ For glite-yaim-clients 4.0.0-5

```
yaim -c -s site-info.def -n WN_torque
```

or

```
yaim -c -s site-info.def -n WN -n TORQUE_client
```

- ♦ For glite-yaim-clients 4.0.1-1

```
yaim -c -s site-info.def -n WN -n TORQUE_client
```

- For 3.1 WN:

```
yaim -c -s site-info.def -n WN -n TORQUE_client
```

LSF

To configure lcg-CE with LSF submission capabilities, one should execute the following command:

```
yaim -c -s site-info.def -n lcg-CE -n glite-LSF_utils
```

Notes on configuring the BDII

The BDII is now configured in a difeferent way. Please, note that:

In 3.1 gLite:

- Top level BDII: BDII_top
- Site BDII: BDII_site

In 3.0 gLite, with glite-yaim-core 4.0.0-12

- Top level BDII: BDII
- Site BDII: BDII_site

In 3.0 gLite, with glite-yaim-core 4.0.1-6

- Top level BDII: BDII_top
- Site BDII: BDII_site

Notes on configuring CREAM

CREAM and batch servers

The cream CE is usually configured with a batch server. Check the examples below for possible scenarios:

- Torque (if the CREAM CE is Torque master)

```
/opt/glite/yaim/bin/yaim -c -s <site-info.def> -n creamCE -n TORQUE_server -n TORQUE_
```

- Torque (if the CREAM CE is NOT Torque master)

```
/opt/glite/yaim/bin/yaim -c -s <site-info.def> -n creamCE -n TORQUE_utils
```

- LSF

```
/opt/glite/yaim/bin/yaim -c -s <site-info.def> -n creamCE -n LSF_utils
```

BLparser configuration

For more information, please check also the BLparser configuration twiki maintained by the CREAM team.

BLparser **MUST** be configured after configuring CREAM. The BLparser must be installed on a machine where the batch system log files are available (let's call this host 'BLParser host'). So the blparser host can be the batch system master or a different machine where the log files are available (e.g. they have been exported via NFS). There are two possible layouts:

1. the blparser host (BLPARSER_HOST) is the CREAM CE host (CE_HOST)
2. the blparser host (BLPARSER_HOST) is different than the CREAM CE host

1. If your configuration satisfies the first layout, you need simply configure the blparser on your CREAM CE running:

```
/opt/glite/yaim/bin/yaim -r -s <site-info.def> -n creamCE -f config_cream_blparser
```

Beware: always specify **all** the node types that apply to the machine! For example:

```
/opt/glite/yaim/bin/yaim -r -s <site-info.def> -n creamCE -n TORQUE_utils -f config_cream_blpar
```

Otherwise certain configuration files may end up **incomplete**!

Then restart tomcat: `service tomcat5 restart`

2. If instead your layout is the second one, you need to install glite-yaim-cream-ce, glite-yaim-core and glite-ce-blahp on your BLParser server and then run the following command to configure it:

```
/opt/glite/yaim/bin/yaim -r -s <site-info.def> -n creamCE -f config_cream_blparser
```

Then restart tomcat on the creamce node: `service tomcat5 restart`

Check your CREAM CE

See this page for some checks/tests that you can do on the new installed CREAM CE.

In particular you can perform the automatic checks reported in the CheckCreamConf page to control some parts of the configuration

Notes on configuring the dCache

dCache has taken the YAIM module within the dCache project and maintains the code as dCacheConfigure.sh. The YAIM model glite-yaim-dcache simple calls dCacheConfigure.sh and since the file formats for site-info.def will remain compatable but has been extended for dCache use.

Notes on configuring the DPM

When you install a DPM disk and a DPM mysql together is the same node, you don't need to configure DPM disk, it's enough to call -n SE_dpm_mysql.

Notes on configuring the FTS

When you run the configuration with yaim `/opt/glite/yaim/bin/yaim -c -s site-info.def -n "FTS2 FTA2"`, if the database is created from scratch the configuration aborts and you are invited to create the DB tables manually using some sql scripts.

Before submitting jobs, the configuration must be completed setting the vo shares with the command `/opt/glite/bin/glite-transfer-channel-setvoshare`, otherwise the job will fail because of missing authorisation.

Remember that in case of problems, you can control the agents daemons by using `service transfer-agents (start|stop|restart|status)`

For more details, please check the FTS server installation guide.

Notes on configuring the glite CE

The gLite CE is no longer configured by this version of YAIM. Please, remember that currently the glite-CE should be regarded as a proof of principle component that is not in a state to be deployed in production in a reasonable manner. In case you are still interested in installing a gLite CE, please use yaim core <= 4.0.0-13.

Notes on configuring MPI

Please, note that for configuring MPI with a computing element, a strict order in the list of configuration targets has to be followed. First MPI and then the computing element For example:

./yaim -c -s site-info.def -n glite-MPI_CE -n lcg-CE

For more information, please check the YAIM MPI wiki.

Notes on configuring the WMS

Until BUG:53297 and BUG:53297 are fixed, the WMS admin can create a file `/opt/glite/yaim/functions/post/config_glite_wms` with the following function to let YAIM adjust the parameters in its post-configuration step:

```
config_glite_wms_post()
{
    perl -i -pe '
   BEGIN {
       $flag = 0;
   }
   s/(--ftpconn) \d+/$1 300/;
   /^\s*WorkloadManagerProxy/ && ($flag = 1);
   $flag && s/(LogLevel *=) *\d+/$1  6/ && ($flag = 0);
    ' /opt/glite/etc/glite_wms.conf

    /opt/glite/etc/init.d/glite-wms-wmproxy restart
}
```

Partial configuration

It is possible to run only a configuration function. See the following example:

```
./yaim -r -s /root/siteinfo/site-info.def -n SE_dpm_mysql -f config_mkgridmap
```

Package your siteinfo directory

Option `-p | --package` of the YAIM command creates an rpm package called `/root/yaim-siteinfo-${SITE_NAME}-${SITE_INFO_VERSION}.noarch.rpm` that contains the configuration directory structure, that is:

- site-info.def
- vo.d/, if it exists
- group.d/, if it exists
- nodes/, if it exists
- /opt/glite/yaim/functions/local, if any local function is defined.

The package installs the configuration structure under `/opt/glite/yaim/examples/siteinfo/mysiteinfo-${SITE_INFO_VERSION}`. This allows to maintain a packaged site configuration and install it in the different machines of the site.

The variables `${SITE_NAME}` and ${SITE_INFO_VERSION} are mandatory and are expected to be defined under site-info.def and not in any other configuration file under the siteinfo directory.

YAIM log file

The output of a configuration is stored in `/opt/glite/yaim/log/yaimlog`. The amount of information that appears during the configuration can be selected defining the YAIM_LOGGING_LEVEL variable in site-info.def file. Possible values are:

- NONE
- ABORT
- ERROR
- WARNING
- INFO
- DEBUG

The default value is *INFO*. The logfile contains the timestamp, the functions that are executed and the output of the functions.

YAIM tool

There is a tool under development that will help sysadmins to automatically create VO related YAIM configuration parameters. This is the YAIM tool. Site administrators will be able to use this utility to maintain configuration information for the VOs their site supports.

Advanced configurations, customizing YAIM

This section describes how to adapt YAIM to your site's needs and can be skiped if you are reading this guide for the first time.

Use local functions

Sys admins have the possibility to customise existing yaim functions or perform some tasks before and after their execution by using the following directories:

- `/opt/glite/yaim/functions/local` : functions placed into this directory will be sourced by the `yaim` command. The file name has to be the same as the function defined inside it, and the same

function file has to exist in the `functions` directory. When the function is defined in `functions/local`, that version is executed instead of the one in the `functions` directory.
- `/opt/glite/yaim/functions/pre` the function name has to be the same as the file name but with an additional `_pre` suffix. For example, if the file name is `config_gip` the function inside should be `config_gip_pre`. These functions will be executed *before* the main function.
- `/opt/glite/yaim/functions/post` The same as the `pre` directory but with the `_post` suffix , and the functions will be executed after the main function.

Create your own module

Have a look at the YAIM module HOWTO where you can find an example to start creating your own YAIM module.

Changes respect to previous versions

What is different from yaim 3.1.1?

YAIM 4.0.0 is a release without any important functionality changes. It mainly fixes bugs from YAIM 3.1.1. The most relevant changes are:

- configuration of the new glite-info-generic rpm.
- new site-info.def variable SITE_BDII_HOST
- users.conf example files includes pool accounts for sgm and prd users.

For a list of bugs and patches fixed by YAIM 4.0.0, have a look at the Savannah patch 1238 for gLite 3.0 and patch 1239 for gLite 3.1.

TO TAKE INTO ACCOUNT: Since now 3.1 and 3.0 UI and WN configuration is put together, in order to differenciate which functions are only used in 3.0, we have used the suffix '_30'. In functions/ you may find config_function_name_30 and in node-info.d/ you may find glite-node-type_30. This means there are differences in the configuration between 3.0 and 3.1. In case you are using 3.0 local functions overriding the existing onces, check whether you may have or not to add the '_30' suffix.

What is different from previous versions after YAIM 3.1.1?

The main differences between YAIM versions can be obtained by reading the release notes in the Savannah patches used to release YAIM.

- glite-yaim-core 4.0.1-6 - patch 1413 for gLite 3.1 and patch 1419 for gLite 3.0.
- glite-yaim-core 4.0.2-1 - patch 1462 for gLite 3.1 and patch 1507 for gLite 3.0.
- glite-yaim-core 4.0.3-6 - patch 1516 for gLite 3.1 and patch 1517 for gLite 3.0.
- glite-yaim-core 4.0.3-13 - patch 1662 for gLite 3.1

For furher reading

- The LCG Directory
- The LCG Troublehooting Guide

This topic: LCG > YaimGuide400
Topic revision: r192 - 2011-12-27 - MaartenLitmaath