

# Table of Contents

<b>YAIM 3.1.1 guide for sysadmins.....</b>	<b>1</b>
Introduction.....	1
Basics.....	1
What is YAIM.....	1
Modular structure.....	1
The configuration variables.....	2
site-info.def.....	3
vo.d directory.....	8
services directory.....	10
nodes directory.....	10
users.conf.....	10
groups.conf.....	11
Running the configuration.....	11
The interface.....	11
Installing a node.....	12
IMPORTANT NOTE: YAIM doesn't support installation of gLite 3.1 nodes but just its configuration.....	12
Configuring a node.....	13
Configuring a Batch system.....	14
Partial configuration.....	15
YAIM log file.....	15
YAIM tool.....	16
What is different from yaim 3.0.1 and yaim 3.1.0?.....	16
Known issues.....	16
For further reading.....	17

# YAIM 3.1.1 guide for sysadmins

**NOTE:** This guide is not up to date. Better use the YAIM4 guide with the latest information

## Introduction

This document provides a description of YAIM version 3.1.1. This YAIM release is a merge of YAIM 3.0.1-22 and YAIM 3.1.0. It allows to configure all gLite 3.0 and gLite 3.1 services. It contains a modular structure consisting of a main yaim core module, yaim clients and yaim services. Not all the gLite services have a yaim module in this release. This will change in future releases. Please check the *Known Issues* section for more details.

For the general installation process check:

- gLite 3.1 Generic Installation & Configuration guide
- gLite 3.0 Generic Installation & Configuration guide

## Basics

### What is YAIM

The aim of YAIM (Yet Another Installation Manager) is to implement a configuration method for the LCG and 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 looks for errors. 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-core (it contains the remaining components that do not appear in the list)
- glite-yaim-clients (UI, WN and VOBOX)
- glite-yaim-wms (new WMS)
- glite-yaim-lb (new LB)
- glite-yaim-fts
- glite-yaim-dpm
- glite-yaim-lfc
- glite-yaim-dcache
- glite-yaim-myproxy

## 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:

- *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 distributes an example file in `/opt/glite/yaim/examples/siteinfo/services/glite-node-type`. It's not a mandatory folder.
- *vo.d*: it contains a file per VO with the format *vo\_name*. It's not a mandatory folder.
- *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. It's not a mandatory folder.

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/`. 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 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.

### site-info.def

**IMPORTANT:** YAIM 3.1.1 is backwards compatible with previous YAIM releases because it can use the same site-info.def file

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. In future YAIM releases, each yaim module will distribute a file containing the variables relevant to the associated node type. This file will be under /opt/glite/yaim/examples/siteinfo/services/glite-node-type. In that case, site-info.def can be kept to a minimum containing only the common variables to all node types. In the current YAIM version, the example site-info.def file in /opt/glite/yaim/examples/siteinfo/site-info.def contains a full list of all the variables used to configure any service.

site-info.def contains different configuration variables to be defined. Some of them are compulsory and no default value is assigned to them by the functions. Some of them can be left undefined and the functions will use the default values. In the following table all the variables used by this version of YAIM are listed: ( c = compulsory (if you going to configure that type of node) , o = optional )

(Click on the column name to sort on a column basis)

Variable name	Type	Description
APEL_DB_PASSWORD	C	Database password for APEL.
BATCH_BIN_DIR	C	The path of the lrms commands, e.g. /usr/pbs/b
BATCH_SPOOL_DIR	C	The batch system log directory. It will replace BATCH_LOG_DIR in 4.0.1-x release.
BATCH_VERSION	C	The version of the Local Resource Management S OpenPBS_2.3.
BATCH_SERVER	C	Set this to specify your torque server host. TORQUE_SERVER variable.
BDII_FCR	O	Set the URL of the Freedom of Choice for Reso
BDII_HOST	C	BDII hostname.
BDII_HTTP_URL	C	URL pointing to the BDII configuration file (
BDII_REGIONS	C	List of node types publishing information to each item listed in the BDII_REGIONS variable. create a set of new variables as follows:
BDII_RESOURCE_TIMEOUT	C	The timeout value to be used with a resource time the BDII will wait when running the GIP.
BDII_SITE_TIMEOUT	C	The timeout value to be used with a site BDII the BDII will wait when querying resource BDII
BDII_<NODE-TYPE>_URL	C	URL of the information producer (e.g. BDII_CI

		CE information producer", BDII_SE_URL="URL of information producer").
CA_REPOSITORY	C	The repository with Certification Authorities (the example).
CE_BATCH_SYS	C	Implementation of site batch system Available: ``torque``, ``lsf``, ``pbs``, ``condor`` etc.
CE_CPU_MODEL	C	Model of the CPU used by the WN (WN specification parameter is a string whose domain is not defined in the GLUE EGEE Schema. The value used for Pentium is "pentium").
CE_CPU_SPEED	C	Clock frequency in Mhz (WN specification).
CE_CPU_VENDOR	C	Vendor of the CPU. used by the WN (WN specification parameter is a string whose domain is not defined in the GLUE EGEE Schema. The value used for Intel is "intel").
CE_HOST	C	Computing Element Hostname.
CE_INBOUNDIP	C	TRUE if inbound connectivity is enabled at your site otherwise (WN specification).
CE_MEMORYMEM	C	RAM size (Mbytes) (per WN and not per CPU) (WN specification).
CE_MEMORYMEM	C	Virtual Memory size in (Mbytes) (per WN and not per CPU) (WN specification).
CE_OS	C	Operating System name (WN specification) - see <a href="https://wiki.egi.eu/wiki/Operations/HOWTO05">https://wiki.egi.eu/wiki/Operations/HOWTO05</a>
CE_OS_RELEASE	C	CE Operating System release (WN specification) <a href="https://wiki.egi.eu/wiki/Operations/HOWTO05">https://wiki.egi.eu/wiki/Operations/HOWTO05</a>
CE_OS_VERSION	C	CE Operating System Version (WN specification) <a href="https://wiki.egi.eu/wiki/Operations/HOWTO05">https://wiki.egi.eu/wiki/Operations/HOWTO05</a>
CE_OS_ARCH	C	CE Operating System Architecture. Value of 'uname -m' (usually i386, i586, i686, or x86_64) executed on the WN (from the SubCluster represented by that CE).
CE_OUTBOUNDIP	C	TRUE if outbound connectivity is enabled at your site otherwise (WN specification).
CE_RUNTIMEENV	C	List of software tags supported by the site. It should include VO-specific software tags. In order to ensure backward compatibility it should include the name of the current middleware version and the list of supported middleware tags.
CE_SF00	C	Performance index of your fabric in SpecFloat 2000 (WN specification). For some examples of Spec values see <a href="http://www.specbench.org/osg/cpu2000/results/">http://www.specbench.org/osg/cpu2000/results/</a>
CE_SI00	C	Performance index of your fabric in SpecInt 2000 (WN specification). For some examples of Spec values see <a href="http://www.specbench.org/osg/cpu2000/results/">http://www.specbench.org/osg/cpu2000/results/</a>
CE_SMP_SIZE	C	Number of cpus in an SMP box (WN specification)
CLASSIC_HOST	C	The name of your SE_classic host.
CLASSIC_STORAGE_DIR	C	The root storage directory on CLASSIC_HOST.
CRON_DIR	C	Yaim writes all cron jobs to this directory. If you want to turn off Yaim's management of cron, set this variable to an empty string.
DCACHE_ADMIN	C	Host name of the server node which manages the dCache.
DCACHE_POOLS	C	List of pool nodes managed by the DCACHE_ADMIN.
DCACHE_PORT_RANGE	C	dCache Port Range. This variable is optional. The default value is "20000,25000".
DCACHE_DOOR_SRM	O	

		Set up srm server on dCache pool nodes (ex door_node2[:port])
DCACHE_DOOR_GSI FTP	O	Set up srm gsift on dCache pool nodes (ex door_node2[:port])
DCACHE_DOOR_GSI DCAP	O	Set up gsidcap server on dCache pool nodes (ex door_node2[:port])
DCACHE_DOOR_DCAP	O	Set up dcap server on dCache pool nodes (ex door_node2[:port])
DCACHE_DOOR_XROOTD	O	Set up xrootd server on dCache pool nodes (ex door_node2[:port])
DCACHE_DOOR_LDAP	O	Set up ldap server on dCache admin_node (ex door_node2[:port])
DCACHE_PNFS_SERVER	O	For medium size sites where dCache can be overburden of running name servers and the other. If a reasonable sized dCache cluster is set up, recommended to dedicate a single host for PNFS for all central services and a single server variable is not set, the PNFS services will running on the admin node, just as the SRM default on the admin node.
DPMFSIZE	C	The default disk space allocated per file (ex DPM node).
DPM_DB_USER	C	The db user account for the DPM
DPMPOOL	C	Name of the Pool (per default Permanent).
DPM_FILESYSTEMS	C	Space separated list of DPM pool hostname:/pa
DPM_DB_PASSWORD	C	Password of the db user account.
DPM_DB_HOST	C	Set this if your DPM server uses a db on a se Defaults to localhost.
DPM_HOST	C	Host name of the DPM host, used also as a def lcg-stdout-mon.
DPM_DB	C	The dpm database name (default is dpm_db)
DPNS_DB	C	The dns database name (default is dns_db)
DPM_INFO_USER	C	The DPM database info user.
DPM_INFO_PASS	C	The DPM database info user's password.
E2EMONIT_LOCATION	O	e2emonit config download site.
E2EMONIT_SCRIPTS	O	Used when generating e2emonit cron scripts.
EDG_WL_SCRATCH	O	Optional scratch directory for jobs.
FTS_HOST	C	The hostname of your FTS server - use this on an FTS server.
FTS_SERVER_URL	C	The URL of the File Transfer EGEE Service ser
FUNCTIONS_DIR	C	The directory where YAIM will find its functi
GLP_CACHE_TTL	C	How long information in the cache is valid.
GLP_FRESHNESS	C	If the information from the plug-ins is withi timelimit, the dynamics plug-ins will not be
GLP_RESPONSE	C	How long the GLP will wait for dynamic plug-i reading the information from the cache.
GLP_TIMEOUT	C	The timeout value to be used with dynamic plu
GLOBUS_TCP_PORT_RANGE	C	Port range for Globus IO. It should be specif "num1,num2". YAIM automatically handles the s variable depending on the version of VDT. If leaves "num1,num2". If it's a version < VDT 1 "num1 num2".

GRI D_I C E_ S E R V E R_ H O S T	O	GridIce server host name (usually run on the
GRI D_ T R U S T E D_ B R O K E R S	C	List of the DNs of the Resource Brokers host which are trusted by the Proxy node. (ex: / O=Grid/ O=CERN/ OU=cern.ch/ CN=host/testbed013.
G R O U P S_ C O N F	C	Path to the groups.conf file which contains i mapping VOMS groups and roles to local groups this configuration file is given in /opt/gLite/yaim/examples/groups.conf.
G S S K L O G	C	yes or no, indicating whether the site provide authentication server which maps gsi credentials to Kerberos tokens.
G S S K L O G_ S E R V E R	C	If GSSKLOG is yes, the name of the AFS authentication host.
I N S T A L L_ R O O T	C	Installation root - change if using the re-local distribution.
J A V A_ L O C A T I O N	C	Path to Java VM installation. It can be used a different version of java installed locally
J O B_ M A N A G E R	C	The name of the job manager used by the gatekeeper
L C G_ R E P O S I T O R Y	C	APT repository for the EGEE middleware. This is gLite 3.0 that uses apt. For gLite 3.1, please link.
L F C_ C E N T R A L	C	A list of VOs for which the LFC should be connected to central catalogue.
L F C_ D B	C	db name for LFC server.
L F C_ D B_ H O S T	C	db hostname for the LFC server. Set to LFC server
L F C_ D B_ P A S S W O R D	C	db password for LFC user.
L F C_ H O S T	C	Set this if you are building an LFC_HOST, not using clients.
L F C_ H O S T_ A L I A S	C	Alias of the LFC host
L F C_ L O C A L	C	Normally the LFC will support all VOs in the you want to limit this list, add the ones you LFC_LOCAL.
		--- For each item listed in the VOS variable create a set of new variables as follows:
V O_ < V O_ N A M E >_ D E F A U L T_ S E	C	Default SE used by the VO. WARNING: VO-NAME must be in capital cases.
V O_ < V O_ N A M E >_ S T O R A G E_ D I R	C	Path to the storage area for the VO on an SE. WARNING: VO-NAME must be in capital cases.
V O_ < V O_ N A M E >_ S W_ D I R	C	Area on the WN for the installation of the experimental software. If on the WNs a predefined shared area is mounted where VO managers can pre-install software, these variable should point to this area. If not a shared area and each job must install software, then this variables should contain a dot ( . ) for mounting of shared areas, as well as the local path of VO software is not managed by yaim and should be installed locally by Site Administrators. WARNING: VO-NAME must be in capital cases.
V O_ < V O_ N A M E >_ V O M S E S	C	List of entries for the vomses files for this VO. These values can be given if enclosed in single quotes.
V O_ < V O_ N A M E >_ V O M S_ E X T R A_ M A P S	C	To add any further arbitrary maps you need in the edg-nkgridmap.conf.
V O_ < V O_ N A M E >_ V O M S_ P O O L_ P A T H	C	

		If necessary, append this to the VOMS_SERVER account list.
VO_<VO- NAME>_VOMS_SERVERS	C	A list of VOMS servers for the VO.
VO_<VO- NAME>_RBS	C	A list of RBs which support this VO.
		--- End of VOs variable listing.
MON_HOST	C	MON Box hostname.
MYSQL_PASSWORD	C	The mysql root password.
MY_DOMAIN	C	The site's domain name.
OUTPUT_STORAGE	C	Default Output directory for the jobs.
PX_HOST	C	PX hostname.
QUEUES	C	The name of the queues for the CE. These are as the VO names.
<QUEUE- NAME>_GROUP_ENABLE	C	Space separated list of VO names and VOMS FQDNs allowed to access the queue. It will be translated to the group_enable parameter.
RB_HOST	C	Resource Broker Hostname.
REG_HOST	C	RGMA Registry hostname.
REPOSITORY_TYPE	C	It can only be apt. Moreover, this is only valid for 3.0. For gLite 3.1 please visit this link <a href="#">http://www.lcgfr.org/wiki/GettingStartedWithGlide31</a>
RESET_DCACHE_CONFIGURATION	O	Set this to yes if you want YAIM to configure dCache. If unset (or 'no') yaim will only configure the front-end to dCache.
RESET_DCACHE_PNFS	O	yes or no DO NOT set these values to yes on experimental production services, dCache internal databases will be deleted.
RESET_DCACHE_RDBMS	O	yes or no DO NOT set these values to yes on experimental production services, dCache internal databases will be deleted.
RFIO_PORT_RANGE	O	Optional variable for the port range with default "20000,25000".
SE_ARCH	C	=defaults to multidisk - "disk, tape, multidisk" populates GlueSEArchitecture.
SE_LIST	C	A list of hostnames of the SEs available at the site.
SITE_EMAIL	C	The site contact e-mail address as published in the information system
SITE_SUPPORT_EMAIL	C	The site's user support e-mail address as published in the information system
SITE_HTTP_PROXY	O	If you have a http proxy, set this variable (the same as that of the http_proxy environment variable) and use it in config_crl and used by the cron jobs in order to reduce the load on the CA host.
SITE_LAT	C	Site latitude.
SITE_LOC	C	"City, Country".
SITE_LONG	C	Site longitude.
SITE_NAME	C	Your GIS.
SITE_SUPPORT_SITE	C	Support entry point ; Unique Id for the site as published in the information system
SITE_TIER	C	Site tier.
SITE_WEB	C	Site site.
USERS_CONF	C	Path to the file containing a list of Linux users (and their accounts) to be created. This file should be owned by the Site Administrator, which contains a plain list of

		and IDs. An example of this configuration file is <code>/opt/glite/yaim/examples/users.conf</code> .
VOBOX_HOST	C	VOBOX hostname.
VOBOX_PORT	C	The port the VOBOX gsisshd listens on.
VOS	C	List of supported VOs.
VO_SW_DIR	C	Directory for installation of experiment software.
WMS_HOST	C	Hostname of the gLite WMS/LB server.
WN_LIST	C	Path to the list of Worker Nodes. The list of nodes is a file to be created by the Site Administrator. It contains a plain list of the batch nodes. An example configuration file is given in <code>/opt/glite/yaim/examples/wn-list.conf</code> .
YAI_M_LOGGING_LEVEL	C	The logging level to print debugging information. Possible values are NONE, ABORT, ERROR, WARNING, INFO, DEBUG.

## vo.d directory

The `vo.d` directory was created to make the configuration of the DNS-like VOs easier. Each file name in this directory has to be the lower-cased version of the VO name defined in `site-info.def`. The matching file should contain the definitions for that VO and will overwrite the ones which are defined in `site-info.def`. 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 can 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
```

After the experience of AEGIS01-PHY-SCL, EGEE SEE ROC who have been deploying DNS-like VO names, the following recipe has been prepared with their contribution:

In you are deploying VO with DNS-like name (example: `vo.test.domain.org`), due to the limitations of the shell and other languages used (e.g. "." is often not a valid character for a variable name) and length limitations, you would first need to choose shorter name for such a VO (example: `test`). Now the new VO can be added in all conf files - the only issue is where to use full VO DNS-like name, and where to use short name.

- `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=vo.test.domain.org/GROUP=/vo.test.domain.org/ROLE=lcgadmin
/VO=vo.test.domain.org/GROUP=/vo.test.domain.org/ROLE=production"
```

- The preferred way to define other VO-related variables is to put them in vo.d directory in a file named after a full VO name (example: vo.d/vo.test.domain.org); even if your VOs don't have DNS-like names. All such files should contain at least the following variables: SW\_DIR, DEFAULT\_SE, STORAGE\_DIR, VOMS\_SERVERS, VOMSES.
- If these variables are instead put in site-info.def directly, each variable name must be prepended with VO\_<VO-NAME>\_, where <VO-NAME> is full VO name in capital letters. In case of DNS-like VO names dots in VO name must be additionally changed to e.g. underscores. That can be quite tricky and is not suggested (nor YAIM is full-proofed to work correctly with this). For your reference, here is an example based on the VO name vo.test.domain.org (and its non-existing VOMS server voms.domain.org with the certificate DN /DC=ORG/O=DOMAIN/O=Hosts/CN=host/voms.domain.org):

```
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/CN=host/voms.domain.org"
```

- If you choose to put all other 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. The example is similar as the one above (just without the prepended VO\_<VO-NAME>\_ part):

```
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.domain.org"
```

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

## services directory

The services directory is created to make easier the configuration of different node types present in one site. In future releases of yaim, each yaim module will 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 will then contain only the variables that are common to all the 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.

## nodes directory

The node 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=&#8221;atlas alice&#8221;
```

## users.conf

This file defines the unix users to be created on different service nodes (mainly on CE and WNs). The format is the following:

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

Each line defines one user with uid, login the user has primary group gid1 and additional secondary, tertiary,... groups gid2, gid3,... which corresponds to the groups group1, group2, group3, is member of the VO vo, and has a special role flag, which will be associated with the corresponding line of groups.conf. Whitespaces and blank lines are not allowed. This file will be processed by one of the YAIM functions. The function will create the unix users if they don't exist already in the service node.

YAIM distributes an example users.conf file in /opt/glite/yaim/examples. Below you can find some examples as well:

```
25001:dteam001:2500:dteam:dteam::
25002:dteam002:2500:dteam:dteam::
25003:dteam003:2500:dteam:dteam::
```

```
25004:dteam004:2500:dteam:dteam::
...
25801:prddteam001:2580,2500:prddteam,dteam:dteam:prd:
25802:prddteam002:2580,2500:prddteam,dteam:dteam:prd:
...
25901:sgmdteam001:2590,2500:sgmdteam,dteam:dteam:sgm:
25902:sgmdteam002:2590,2500:sgmdteam,dteam:dteam:sgm:
```

Please, visit the [How to switch to pool accounts for sgm/prd users](#) for more details on sgm/prd pool accounts.

## groups.conf

The groups.conf file has the following format:

```
FQAN:group name:gid:users.conf flag:vo
```

Thus users with VOMS credential fqan will be mapped to group name, gid, and associated with the users having the same flag in users.conf. That means, if flag is given, the group name and gid are taken from there and do not need to be specified. If the last - optional - field is defined, then the group will be treated as the member of the vo instead of the one which is determined from the fqan.

YAIM distributes an example groups.conf file in /opt/glite/yaim/examples. Below you can find some examples as well:

```
"/VO=alice/GROUP=/alice/ROLE=lcgadmin":::sgm:
"/VO=alice/GROUP=/alice/ROLE=production":::prd:
"/VO=alice/GROUP=/alice":::
"/VO=atlas/GROUP=/atlas/ROLE=lcgadmin":::sgm:
"/VO=atlas/GROUP=/atlas/ROLE=production":::prd:
"/VO=atlas/GROUP=/atlas":::
"/VO=cms/GROUP=/cms/ROLE=lcgadmin":::sgm:
"/VO=cms/GROUP=/cms/ROLE=production":::prd:
```

## 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. For help just run:

```
Usage: ./yaim <action> <parameters>
```

Actions:

- i | --install : Install one or several meta package.  
Compulsory parameters: -s, -m
- 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 : Go through on all the function and checks that the necessary variables required for a given configuration target are all defined in site-info.def.  
Compulsory parameters: -s -n
- h | --help : This help

Specify only one action at a time !

Parameters:

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

Examples:

Installation:

```
./yaim -i -s /root/siteinfo/site-info.def -m glite-SE_dpm_mysql
```

Configuration:

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

Running a function:

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

Verify your site-info.def:

```
./yaim -v -s /root/siteinfo/site-info.def -n SE_dpm_mysql
```

When configuring multiple node types, they have to be defined on the command line, for example:

```
./yaim -i -s /root/siteinfo/site-info.def -m glite-SE_dpm_mysql -m glite-BDII
```

### Installing a node

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

For installation of gLite-3.1, please have a look at the rpm installation with YUM

YAIM can be used to install your gLite 3.0 nodes. After defining the appropriate variables in your site-info.def, run the following command:

```
./yaim -i -s <location of site-info.def> -m <meta-package-name>
```

The table below lists the available meta-packages for SL3 operating system

Node Type	meta-package name	meta-package description
WMS	glite-WMS	WMS node
LB	glite-LB	LB node
glite CE gliteCE	The gLite Computing Element	
FTS	glite-FTS	gLite File Transfer Server
FTA	glite-FTA	gLite File Transfer Agent
BDII	glite-BDII	BDII
LCG Computing Element (middleware only)	lcg-CE	It does not include any LRMS
LCG Computing Element (with Torque)	lcg-CE_torque	It includes the 'Torque' LRMS
	glite-LFC_mysql	LCG File Catalog

LCG File Catalog (mysql)		
LCG File Catalog (oracle)	glite-LFC_oracle	LCG File Catalog
MON-Box	glite-MON	RGMA-based monitoring system collector server
MON-Box	glite-MON_e2emnit	MON plus e2emnit
Proxy	glite-PX	Proxy Server
Resource Broker	l cg- RB	Resource Broker
Classic Storage Element	glite-SE_classic	Storage Element on local disk
dCache Storage Element	glite-SE_dcachepool	Storage Element interfaced to dCache for a pool node
dCache Storage Element	glite-SE_dcachepool_admin_postgres	Storage Element interfaced to dCache for an admin node pnfs
DPM Storage Element (mysql)	glite-SE_dpm_mysql	Storage Element with SRM interface
DPM Storage Element (Oracle)	glite-SE_dpm_oracle	Storage Element with SRM interface
DPM disk	glite-SE_dpm_disk	Disk server for a DPM SE
Dependencies for the re-locatable distribution	glite-TAR	This package can be used to satisfy the dependencies of the relocatable distro
User Interface	glite-UI	User Interface
VO agent box	glite-VOBOX	Agents and Daemons
Worker Node (middleware only)	glite-WN	It does not include any LRMS
Worker Node (with Torque)	glite-WN glite-torque-client-config	The gLite Computing Element with torque client

### 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> ...
```

IMPORTANT NOTE: YAIM doesn't support installation of gLite 3.1 nodes but just if

Each node type is a configuration target and \*if there is more than one installed and to be configure on a physical node then the configuration should be run together and not separately\*.

The available configuration targets are listed below:

Node Type	Configuration target (node type)	Description
gLite WMS	glite-WMS	WMS node
gLite LB	glite-LB	LB node
glite CE	glite-CE	The gLite Computing Element
FTS	FTS	gLite File Transfer Server
FTA	FTA	gLite File Transfer Agent
top level BDI I	BDI I	A top level BDI I
site BDI I	BDI I_site	A site BDI I
Computing Element (middleware only)	CE	It does not configure any LRMS
LCG File Catalog server	LFC_mysql	Set up a mysql based LFC server
MON-Box	MON	RGMA-based monitoring system collector server
e2emonit	E2EMONIT	RGMA-based monitoring system collector server
Proxy	PX	Proxy Server
Resource Broker	RB	Resource Broker
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
Worker Node (middleware only)	WN	It does not configure any LRMS

### Configuring a Batch system

YAIM only provides configuration for the following batch systems:

- Torque
- SGE (under development)

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

Node Type	Configuration target	Description
-----------	----------------------	-------------

	(node type)	
Computing Element with Torque	CE_torque	It configures also the 'Torque' LRMS client and server
Worker Node with Torque	WN_torque	It configures the Torque client in the WN
Torque server	TORQUE_server	It configures the 'Torque' LRMS server
Torque client	TORQUE_client	It configures the 'Torque' LRMS client

**Please, note that YAIM doesn't configure LSF batch system**

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

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

Note that

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

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

On the other hand, in the case of TORQUE clients in the WN, both possibilities are in theory possible for 3.0 WN:

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

However, please check the Known issues section because there is a bug in this particular case.

For 3.1 WN, the configuration command is the following:

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

### 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
```

### 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.

## YAI M t ool

- There is a tool helping sysadmins to automatically create VO related YAIM configuration parameters. This is the YAIMtool [↗](#). Site administrators can use this utility to maintain configuration information for the VOs their site supports.

## What is different from yai m 3.0.1 and yai m 3.1.0?

YAIM 3.1.1 is different from YAIM 3.0.1 because YAIM 3.1.1 is now structured in a modular way and can be packaged and released independently for each different node type (not all the node types have a module yet). This doesn't affect the system administrators who can still use the same site-info.def file they used for YAIM 3.0.1.

YAIM 3.1.0 and YAIM 3.1.1 are the same. The only difference is that YAIM 3.1.1 also allows to configure glite 3.0 node types. Moreover, the same site-info.def file that was used with YAIM 3.1.0 can still be used with YAIM 3.1.1.

**TO TAKE INTO ACCOUNT:** Since now 3.1 and 3.0 UI and WN configuration is put together, in order to differentiate 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 ones, check whether you may have or not to add the '\_30' suffix.

## Known i ssues

- [#28789](#) [↗](#): Directory /tmp is world readable/writable
- [#27845](#) [↗](#) Due to this bug, the WN with Torque should be installed:
  - ◆ First calling the target name TORQUE\_client and then calling glite-WN
  - ◆ or if they are configured in one go with the same yaim command, in the following order: -n glite-WN -n TORQUE\_client
- [#27761](#) [↗](#) In all Computing Elements, the configuration does not release the shell at the end. Making ctrl+c to release it also kills the gatekeeper (bug [#27761](#) [↗](#)). The script has to be restarted, easy workarounds are "service gLite restart" for the gLite-CE and /etc/init.d/globus-gatekeeper restart on the lcg-CE.
- [#25567](#) [↗](#): In some of the nodes, the configuration does not release the shell either at the end but ctrl+c can be used without any problem
- Remember to stop using TORQUE\_SERVER variable and use instead BATCH\_SERVER variable.

The following list of known issues is inherited from YAIM 3.01 and YAIM 3.1.0:

- The site BDII is not configured on the lcg-CE automatically anymore. You have to set the configuration target explicitly, i.e. for example:

```
configure_node site-info.def CE_torque BDII_site
```

- There is a known installation conflict between the 'torque-clients' rpm and the 'postfix' mail server (Savannah. bug #5509). In order to workaroud the problem you can either uninstall postfix or remove the file /usr/share/man/man8/qmgr.8.gz from the target node.
- To configure correctly the glite-CE the following parameters MUST be set as follows:

```
JOB_MANAGER=pbs  
CE_BATCH_SYS=pbs
```

## For further reading

- The LCG Directory [↗](#)
- The LCG Troubleshooting Guide

-- MariaALANDESPRADI LLO - 27 Jun 2007

---

This topic: LCG > Yai nGuide311

Topic revision: r24 - 2011-06-22 - Andres Aeschli mann



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TW ki? Send feedback