

# Table of Contents

<b>How to develop a YAIM module called TURBO.....</b>	<b>1</b>
Contact the YAIM team.....	1
Define the configuration in CVS.....	1
Makefile and spec file.....	1
functions/.....	1
node-info.d/.....	3
Comments and YAIM log.....	4
services/.....	4
defaults/.....	5
Configuration flow in YAIM.....	6
Basic tests of the YAIM module.....	6
Build the YAIM rpm in ETICS.....	6
Include the YAIM rpm in the gLite release.....	7
Certify your YAIM module.....	7
Your YAIM module in production.....	7
YAIM module example.....	7

# How to develop a YAIM module called TURBO

This is a tutorial for new YAIM developers who want to be responsible for a YAIM module.

First of all, let's imagine that you want to develop a new YAIM module for a component called TURBO.

## Contact the YAIM team

Send a mail to `yaim-contact@cernNOSPAMPLEASE.ch` in order to announce your contribution. The YAIM team will create the TURBO CVS module and the TURBO component in ETICS. Afterwards, the YAIM team will contact you to know your afs login to give you the necessary rights to access the CVS module and to give you the necessary rights to work in ETICS.

- If you don't have an afs account yet, you need one. Check here to know what to do to get one.
- You also need to be registered in ETICS. Check here what to do to be registered if you haven't done it yet.

## Define the configuration in CVS

Checkout your module from CVS:

```
> export CVSROOT=:ext:afs_login@glite.cvs.cern.ch:/cvs/glite
> export CVS_RSH=ssh
> cvs co org.glite.yaim.turbo
```

Have a look at the module directory structure where you will find the following files:

- Makefile
- glite-yaim-turbo.spec
- LICENSE
- config/
  - ◆ node-info.d/
    - ◇ glite-turbo
  - ◆ functions/
    - ◇ config\_turbo
  - ◆ services/
    - ◇ glite-turbo
  - ◆ defaults/
    - ◇ glite-turbo.pre
    - ◇ glite-turbo.post

## Makefile and spec file

The `Makefile` and the `glite-yaim-turbo.spec` are provided by the YAIM team and do not need to be modified (unless new directories are created and we want them to be part of the rpm). These two files will be used to build the rpm. You can check how they look like here.

## functions/

In `functions/` you have a template to write a YAIM function:

```
#####
# Copyright (c) Members of the EGEE Collaboration. 2004.
```

## YAIMmoduleHOWTO < EGEE < TWiki

```
# See http://www.eu-egee.org/partners/ for details on the copyright
# holders.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#####
#
# NAME :          config_turbo
#
# DESCRIPTION :  This function configures TURBO.
#
# AUTHORS :      author_name@domain
#
# NOTES :       --
#
# YAIM MODULE:   glite-yaim-turbo
#
#####

config_turbo_check () {

# Include in this function the name of the variables that should be declared
# to configure TURBO.
#
# Example:
# requires $1 CE_HOST
#
# $1 is needed in case the --verify option of the yaim command is called
# or just "Currently this function doesn't require any variables."

}

config_turbo_setenv () {

# Declare in this function the environment variables that are needed to
# run TURBO.
#
# The following commands can be used:
# yaimgridenv_set      - sets an environment variable
# yaimgridenv_setind  - sets an environment variable if it is not already defined
# yaimgridenv_delete  - delete a value from an environment variable.
#                       if the value becomes null string then it unsets the environment variable
# yaimgridpath_prepend - prepend a value to a path-like environment variable
#                       - puts a ':' to the end of the beginning of the variable
#                       - removes the value from the string and removes duplicate ':'
#                       - prepend the value to the beginning of the variable
#                       - removes duplicate '::' and removes ':' from the beginning
#                       - export the variable
# yaimgridpath_append - the same as prepend but it appends the value to the end of the variable
#
# Example:
# yaimgridpath_append LD_LIBRARY_PATH "${GLOBUS_LOCATION}/lib"
# or just
# yaimlog DEBUG "Currently this function doesn't set any environment variables."
```

```

}

config_turbo () {

# Include here the code necessary to configure your service.

}

```

In each YAIM function, the following sub-functions should be defined:

- `config_turbo_check` : it will contain a list of variables that should be declared in order to configure TURBO. If no variable is needed, just include this line:

```
yaimlog DEBUG "This function doesn't currently require any variables."
```

- `config_turbo_setenv`: it will add to the file `grid-env.(c)sh` the declared variables. These are the variables that are needed in the environment for TURBO to run. If no declaration is needed, just include this line:

```
yaimlog DEBUG "This function doesn't currently set any environment variables."
```

- `config_turbo`: it will contain the main code for TURBO configuration.

## node-info.d/

In `node-info.d/` you have a template to write a function list:

```

#####
# Copyright (c) Members of the EGEE Collaboration. 2004.
# See http://www.eu-egee.org/partners/ for details on the copyright
# holders.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#   http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#####
#
# NAME :          glite_turbo
#
# DESCRIPTION :  This file contains the function list to configure TURBO.
#
# AUTHORS :      author_name@domain
#
# NOTES :        -
#
# YAIM MODULE:   glite-yaim-turbo
#
#####

TURBO_FUNCTIONS="
config_turbo
"

```

In the function list, the list of functions needed to configure TURBO will be listed.

## Comments and YAIM log

Using the given templates, you can write the necessary functions for the TURBO configuration. You then realise you need to implement `config_turbo`, but also you need to write two new functions `config_turbo_db` and `config_turbo_service`. You use the given template to do this and then you also add these two extra functions to the function list. So in the end, your `config/functions` directory contains:

- `config_turbo`
- `config_turbo_db`
- `config_turbo_service`

Remember to include the `-e` symbols to be able to use the `--explain` option of the `yaim` command: `####@` In this way, you can comment your code and allow `yaim` users to print a description of what the function does in case they want to use the `-e` option.

If you also want to print information of what's going on in the YAIM log file, you should use the `yaimlog` command.

And your function list looks like:

```
TURBO_FUNCTIONS="
config_turbo
config_turbo_db
config_turbo_service
"
```

## services/

You also realise that you need to declare some variables that affect only TURBO. These variables are:

```
TURBO_PORT=1234
TURBO_DB=myturbo
TURBO_BIN_DIR=/usr/bin/myturbo
```

Since these variables affect only TURBO, you should put them under the `services` directory where you will create `glite-turbo` and it will contain the variable list:

```
#####
# Copyright (c) Members of the EGEE Collaboration. 2004.
# See http://www.eu-egee.org/partners/ for details on the copyright
# holders.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#####
#
# NAME : glite_turbo
```

## YAIMmoduleHOWTO < EGEE < TWiki

```
#
# DESCRIPTION : This file contains the variable list to configure TURBO.
#
# AUTHORS :      author_name@domain
#
# NOTES :        -
#
# YAIM MODULE:  glite-yaim-turbo
#
#####

# This is the port where TURBO listens to
TURBO_PORT=1234

# This is the db name for TURBO
TURBO_DB=myturbo

# This is the bin directory for TURBO
TURBO_BIN_DIR=/usr/bin/myturbo
```

### defaults/

Finally, you also realise that there are some variables you would like to use, but that the sys admin do not need to configure, with some default values. To do that, you should use the `defaults` directory, where you create a `glite-turbo.pre` file where you will define those variables:

```
#####
# Copyright (c) Members of the EGEE Collaboration. 2004.
# See http://www.eu-egee.org/partners/ for details on the copyright
# holders.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#####
#
# NAME :          glite_turbo.pre
#
# DESCRIPTION :  This file contains the variable list with default values to configure TURBO.
#
# AUTHORS :      author_name@domain
#
# NOTES :        -
#
# YAIM MODULE:  glite-yaim-turbo
#
#####

X509_VOMS_DIR="/etc/grid-security/vomsdir/"
X509_CERT_DIR="/etc/grid-security/certificates/"
X509_HOST_CERT="/etc/grid-security/hostcert.pem"
X509_HOST_KEY="/etc/grid-security/hostkey.pem"
```

When you are happy enough with your code and you plan to include it in a YAIM release, you should create a CVS tag for it with the following format `glite-yaim-turbo_R_4_0_1_1`, being `4_0_1_1` an example of

version and release numbers.

## Configuration flow in YAIM

Once you've created all this files, you would like to know in which order YAIM is going to source these files:

1. /opt/glite/yaim/defaults/site-info.pre
2. /opt/glite/yaim/defaults/glite-turbo.pre
3. /root/siteinfo/site-info.def
4. /root/siteinfo/services/glite-turbo
5. /opt/glite/yaim/defaults/site-info.post
6. /opt/glite/yaim/defaults/glite-turbo.post
7. /root/siteinfo/nodes/lxb001.cern.ch (defined by the sys admin)
8. /root/siteinfo/vo.d/atlas (defined by the sys admin)
9. /opt/glite/yaim/node-info.d/glite-turbo

## Basic tests of the YAIM module

Once your YAIM module is ready, you would like to do some tests. At this stage, you can build your rpm locally by giving a release and a version number, for example 1-1, in the spec file and running:

```
> make rpm
```

You will have an rpm for testing called `glite-yaim-turbo-1.1.noarch.rpm`. Then you should probably need to install the related middleware to test your yaim rpm. Read these instructions to know more about how to install the middleware.

## Build the YAIM rpm in ETICS

Once your rpm has been tested by you, and you are satisfied with the results, the next step will be to include the rpm into the gLite release. In order to do that, you need to have the rpm in ETICS. In order to build an rpm in ETICS, you need a CVS tag and a new configuration for your component.

The YAIM team will create the necessary TURBO component in ETICS under the project `gLite middleware` and the subsystem `yaim`. We do this by default for all the new yaim modules. You can access the component list by surfing in the project directory tree. The TURBO component would contain two configurations:

- `org.glite.yaim.turbo.HEAD`
- `glite-yaim-turbo_R_x_y_z_r`

Everytime you want to create a new configuration, follow these steps:

- right-click on the `glite-yaim-turbo_R_x_y_z_r` configuration, on the Configurations menu and select *clone*.
- A new configuration will be created with a similar name.
- right-click on the new configuration and open it in the workspace. Change the name, display name, version, age and tag (substitutue `x_y_z_r` with the version and release numbers of your CVS tag). In our example we need to create `glite-yaim-turbo_R_4_0_1_1`

Then, you can just follow the instructions described here to launch your remote build.

## Include the YAIM rpm in the gLite release

If you are working in a new YAIM node type it is probably due to the fact that you are working in a new node type for the middleware. This means that there is no metapackage created for it in the repositories. In this case you should work with the Integration team in order to define the metapackage and its dependencies. For more details, please, visit this link. In the case of TURBO, we need to ask to the integration team for a new metapackage called glite-TURBO and provide them with the list of rpms and dependencies that are needed to run TURBO. The yaim rpm will depend on the metapackage glite-TURBO. Normally yaim core should be part of the dependencies as well since it contains configuration affecting all node types.

If the node type you are working in, already exists, you don't need to do anything else special.

In both cases of a new or an old node type, you need to create a new patch in Savannah once all the necessary rpms are ready. Fill in the RPM name, Affected metapackages, Metapackage changes, External packages, Metapackages to be reconfigured, Metapackages to be restarted, Documentation, Configuration changes and Release Notes.

At this moment, your job has finished and you have to wait for the patch to be certified.

## Certify your YAIM module

There's nothing to do here. The certification team is responsible for this. In case of problems, the certification team might be in contact with you. Otherwise, you just need to wait for the patch to be certified.

If there are big bugs in your rpm, the patch might be rejected and that means you would need to fix those bugs first and restart the development process.

## Your YAIM module in production

Once the Savannah patch containing your YAIM rpm has been certified, it's pps and production services managed by SA1, who decide when the patch will be moved from certification to pps, and from pps to production. You can always track the status of the patch in Savannah.

## YAIM module example

We have created an example module called lemon that you can use to play with YAIM:

1. wget  
`http://grid-deployment.web.cern.ch/grid-deployment/yaim/examples/org.glite.yaim.lemon.tar.gz`
2. `gzip -d org.glite.yaim.lemon.tar.gz`
3. `tar -xvf org.glite.yaim.lemon.tar`
4. `cd org.glite.yaim.lemon`
5. Modify the spec file to define the release and version number
6. Create the rpm by executing `make rpm`
7. Install the created rpm

To run the configuration:

1. Install yaim core. Check which is the latest production version in the YAIM status page.
2. You can directly install the yaim core rpm from the ETICS repository in  
`http://eticssoft.web.cern.ch/eticssoft/repository/org.glite/org.glite.yaim.core`
3. Install the glite-version rpm from the ETICS repository in  
`http://eticssoft.web.cern.ch/eticssoft/repository/org.glite/glite-version`

4. Create your site-info.def in a safe location like /root (for lemon you just need a site-info.def containing `INSTALL_ROOT=/opt` for yaim core < 4.0.5-x)
5. If you want to maintain the lemon service specific configuration separately, copy `/opt/glite/yaim/examples/siteinfo/services/glite-lemon_server` under `/root/services/glite-lemon_server`; or copy the variables defined in `/opt/glite/yaim/examples/siteinfo/services/glite-lemon_server` in your site-info.def.
6. Edit the variables and give them a proper value.
7. Then run:

```
./yaim -c -s /root/site-info.def -n glite-LEMON_server
```

Now you can add more functions, variables or configuration targets following the similar structure.

If you delete `LEMON_HOST` from the configuration files and re-run the configuration, what happens?

---

This topic: EGEE > YAIMmoduleHOWTO

Topic revision: r18 - 2008-08-05 - MariaALANDESPRADILLO



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

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