

Table of Contents

Porting with ETICS.....	1
Repositories and Websites to Watch.....	1
Email Lists to Watch.....	1
Configuration of your Build Machine.....	1
Building Individual Modules.....	2
Building One Module.....	2
Building Multiple Modules together.....	2
Building an Entire Project.....	2
More Specialised OS Specific Builds.....	3
Mixed builds.....	3
Registering Packages.....	3
Getting Started.....	4

Porting with ETICS

The page serves as a way to get porting up and running at your site.

Repositories and Websites to Watch

1. <https://etics.cern.ch:8443/etics> this is the web interface where you can see your OS configurations for the gLite Middleware. Also you will need to become familiar with the *externals repository*.
2. <http://grid-deployment.web.cern.ch/grid-deployment/integration/status.html> This is the status page where you can see what is to be deployed for the reference platforms. The software to create a similar few and apt/yum repository can be obtained from:
<http://glite.cvs.cern.ch:8180/cgi-bin/glite.cgi/glite-release/>
3. <https://savannah.cern.ch/projects/etics> is where you should post your bug reports for ETICS related issues. However <https://savannah.cern.ch/projects/jra1mdw/> should be used for gLite middleware related issues and bugs.
4. <http://etics.cern.ch:8080/reportBrowser/> shows the results of certain builds under ETICS.
5. <https://twiki.cern.ch/twiki/bin/view/EGEE/GLiteBuildOverview> references the grid deployment dashboard and the full monolithic builds of org.gLite under SLC3 and SLC4.
6. <http://grid-deployment.web.cern.ch/grid-deployment/integration/status-porting.html> porting page where parallel builds and Unix builds will be reported.

Email Lists to Watch

List Name	E-mail list to Subscribe to
SA3	project-eu-egee-sa3@cernNOSPAMPLEASE.ch
Middleware Integration	project-eu-egee-middleware-integration@cernNOSPAMPLEASE.ch
EMT	project-eu-egee-middleware-emt@cernNOSPAMPLEASE.ch
RGMA	jra1-uk@physicsNOSPAMPLEASE.gla.ac.uk

Use Simba to subscribe to these lists: <http://listboxservices.web.cern.ch/listboxservices/>

Configuration of your Build Machine

There's little to do, just download the `etics-client-setup` and set up a few environment variables. However if you are using `http` and `cvs` proxies, then there are a few extra things to do. The total set of environment variables you will need to setup are:

```
export CVS_RSH=ssh
unset CVSROOT
export ETICS_HOME=/home/<username>/<your-etics-installation-directory>
export PATH=$ETICS_HOME/etics:$PATH
```

If you have `cvs` proxies and `http` proxies then set:

```
export http_proxy=http://yoursiteproxy:8080/
export https_proxy=http://yoursiteproxy:8443/
export cvs_proxy=cvs-proxy.yourdomain
```

In the case of `cvs_proxy` this will modify: `:pserver:anonymous@servername:/some/path` to `:pserver:anonymous%servername@cvs_proxy:/some/path`.

Building Individual Modules

It is suggested that you take the following issues into account then porting:

1. Building from source: If you don't build from source then you are attempting to build from binary. To build from binary assumes that you are going to look for the binary tarball in the AFS repository: <http://eticssoft.web.cern.ch/eticssoft/repository/org.glite> or in the case of external dependencies in: <http://eticssoft.web.cern.ch/eticssoft/repository/externals>
2. Building with verbosity: Its a good idea in the being at least to get used to building with verbosity so that you have very detailed logging information. This will help in the initial stages of getting your builds into a good state.
3. Continuing a build in the case of errors. This is always a good idea since you want to get as many modules to build as possible first go.
4. Force the checkout. The checkout needs to be forced if you are writing over a previous build. However in ETICS they would suggest that build in a clean workspace environment. However if for some reason (maybe during testing) you want to use a workspace with a set directory structure already checked out then you will need to include the --noask flag or use the deprecated command --forcecheckout.

Building One Module

By way of example here is how you would build one module with a given configuration assuming you want to continue on error with verbosity and forcing of the checkout.

```
cd some/directory
etics-workspace-setup
etics-get-project org.glite
etics-checkout --fromsource --forcecheckout --verbose --continueonerror --projectconfig gli
etics-build --verbose --continueonerror --config moduleConfiguration1 moduleName1
```

Building Multiple Modules together

The merge command is required for the 2nd and subsequent module when performing the checkout. The build can be performed in any order.

```
cd some/directory
etics-workspace-setup
etics-get-project org.glite
etics-checkout --fromsource --forcecheckout --verbose --continueonerror --projectconfig gli
etics-checkout --fromsource --forcecheckout --verbose --continueonerror --projectconfig gli
.....
etics-checkout --fromsource --forcecheckout --verbose --continueonerror --projectconfig gli
etics-build --verbose --continueonerror --config moduleConfiguration1 moduleName1
etics-build --verbose --continueonerror --config moduleConfiguration2 moduleName2
.....
etics-build --verbose --continueonerror --config moduleConfigurationN moduleNameN
```

Building an Entire Project

```
etics-workspace-setup
etics-get-project org.glite
etics-checkout --fromsource --forcecheckout --verbose --continueonerror --project-config gli
etics-build --verbose --continueonerror org.glite
```

This is what is being described in CERN as the monolithic build. It requires that you glean the RPMs/tarballs or debs packages to build a node.

More Specialised OS Specific Builds

Sometimes, well quite often in fact you need to get your OS in line with the monolithic build coming out of CERN. To do this you can set properties that ensure you get the correct versions of modules to be built. For instance you might want to use the *boost v. 1.32.0-1.rhel4* instead of *boost v. 1.32.0-6* for newer OSes. Or you might want to build the source version of boost using the configuration *boost v. 1.32.0-src*. The point is, your OS may not have these definitions defined in CERN. So what do you do?

Either:

1. Contact etics-support@cern.ch and say that you want your OS to have a particular configuration added to a particular module. For instance: SLC4 has *boost-1.32.0-1.rhel4* attached to its structure by default, while SLC3 has *boost-1.32.0-6* attached by default.
2. Again contact etics-support@cern.ch to include your configuration but test your assumption adding a property to the project build as follows:

```
etics-checkout --fromsource --forcecheckout --verbose --continueonerror --property moduleName
```

Example: In the case of CentOS 4.4 we needed the same version of boost as used by SLC4 but under SuSE 9.3 we need the source tarball version.

1. We informed [etics-support](mailto:etics-support@cern.ch) about our requirement and looked at <https://etics.cern.ch:8383/etics> to see that our changes were included.
2. In the meantime we performed under CentOS 4.4:

```
etics-checkout --fromsource --forcecheckout --verbose --continueonerror --property boost.DE
```

Under SuSE 9.3 we performed:

```
etics-checkout --fromsource --forcecheckout --verbose --continueonerror --property boost.DE
```

The overrides were automatically incorporated into the build.

Mixed builds

Its possible to mix project builds together, so that you can build from *org.glite* but also include a module from externals. By way of example:

```
etics-checkout --verbose --continueonerror --forcecheckout --fromsource --property dcap.DEF
etics-checkout --verbose --continueonerror --forcecheckout --fromsource --project "external
```

This example is building a particular version of *GFAL* that requires *dcap* as a dependency. Note that you need to specify that *dcap* belongs to the externals project so that it will be included correctly.

Registering Packages

To register modules you need to include *--register* in the *etics-build* command. If you have your DN registered with [etics-support](mailto:etics-support@cern.ch) you will then be allow to register your modules back into CERN. So for instance if you look at: <http://eticsoft.web.cern.ch/eticsoft/repository/org.glite/lcg-mon-tools/1.0.5/> you will see the following directories:

```
[DIR] slc3_ia32_gcc323/                21-Mar-2007 15:29  -
[DIR] slc4_ia32_gcc346/                21-Mar-2007 15:34  -
[DIR] slc4_x86_64_gcc346/              17-Feb-2007 02:10  -
```

However if you perform a project build with `--register` on a new OS such as Debian you might see afterwards:

```
[DIR] deb_x86_64_gcc412/          05-Apr-2007 12:07  -
[DIR] slc3_ia32_gcc323/          21-Mar-2007 15:29  -
[DIR] slc4_x86_64_gcc346/        17-Feb-2007 02:10  -
[DIR] slc4_ia32_gcc346/          21-Mar-2007 15:34  -
```

If your site is <http://grid-deployment.web.cern.ch/grid-deployment/integration/status-porting.html> you will then be able to see modules turn green as you build more and more modules correctly and have them registered into the AFS in CERN. Note that <http://eticssoft.web.cern.ch/eticssoft/repository/> maps to </afs/cern.ch/project/etics/repository>.

Getting Started

This section is at the end because its good to read the rest of the page first. To get started you should do the following:

- Read <https://twiki.cern.ch/twiki/bin/view/ETICS/WebHome> but in particular the quick start manual: <https://twiki.cern.ch/twiki/bin/view/ETICS/EticsQuickStart>
- Register with the e-mail list stated above.
- Keep an eye on the dashboard and porting status page
- Register your DN with `etics-support@cernNOSPAMPLEASE.ch` so that you can register RPMs/tarballs or debs packages back into AFS.
- Register with `etics-support@cernNOSPAMPLEASE.ch` if you need to be able to edit and include new configurations of modules in the "External Dependencies" repository or in the "gLite Middleware" repository. This has to be acknowledged by the correct member of the SA3 team (its not automatic or guaranteed).
- download the `etics-client-setup` and attempt a few project builds, single module builds and multiple module builds to get used to using the build system.
- if you have your build nodes registered with NMI then you can simply use `--remote-project-config` instead of `--project-config` and `--remote-platforms` with a list of OS platform types you want to build on. See `etics-list-platform` for a complete list of platforms.

-- Main.ekenny - 05 Apr 2007

This topic: EGEE > PortingWithEtics

Topic revision: r2 - 2007-04-05 - EamonnKenny



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