Tutorial 1. Navigation and editing with the Web Application

Exercise 01. View and navigate metadata

Open Firefox and load the web application at http://grids17.eng.it/etics

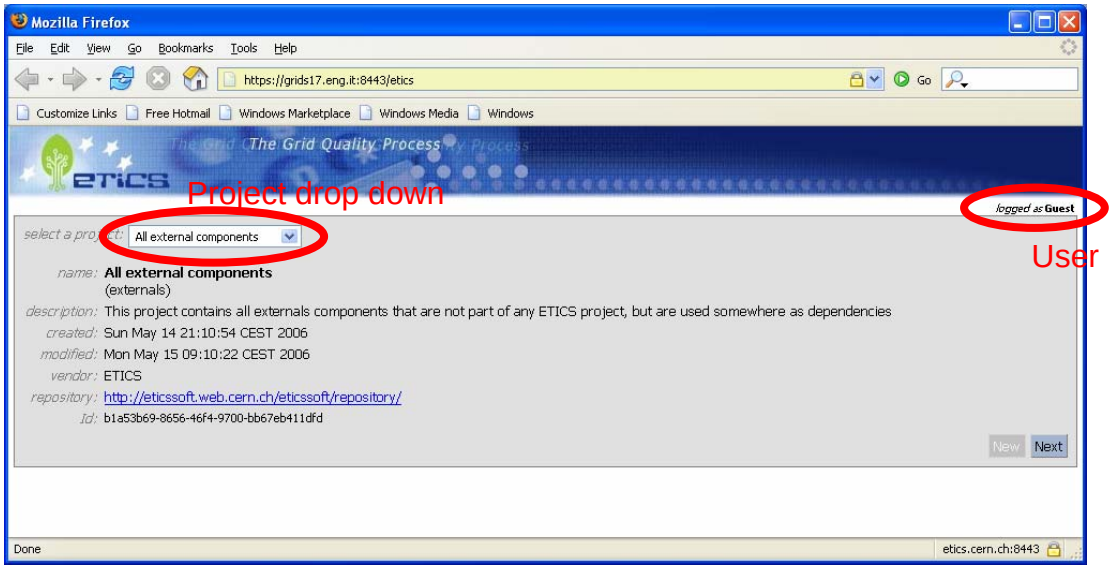

Figure 1: selecting a project

Check that you are "*logged in as Guest*" in the top-right side of the page (see Figure 1)

Selecting Project

Select *The ETICS System* from the project drop down list and click "*Next"*.

Browsing a Project

Browse the Project Tree and view the metadata of any module (i.e. project, subsystem or component), selecting it in the tree (see Figure 2)

Configurations defined for this module are listed in the '*Configuration'* tab at the bottom centre of the page.

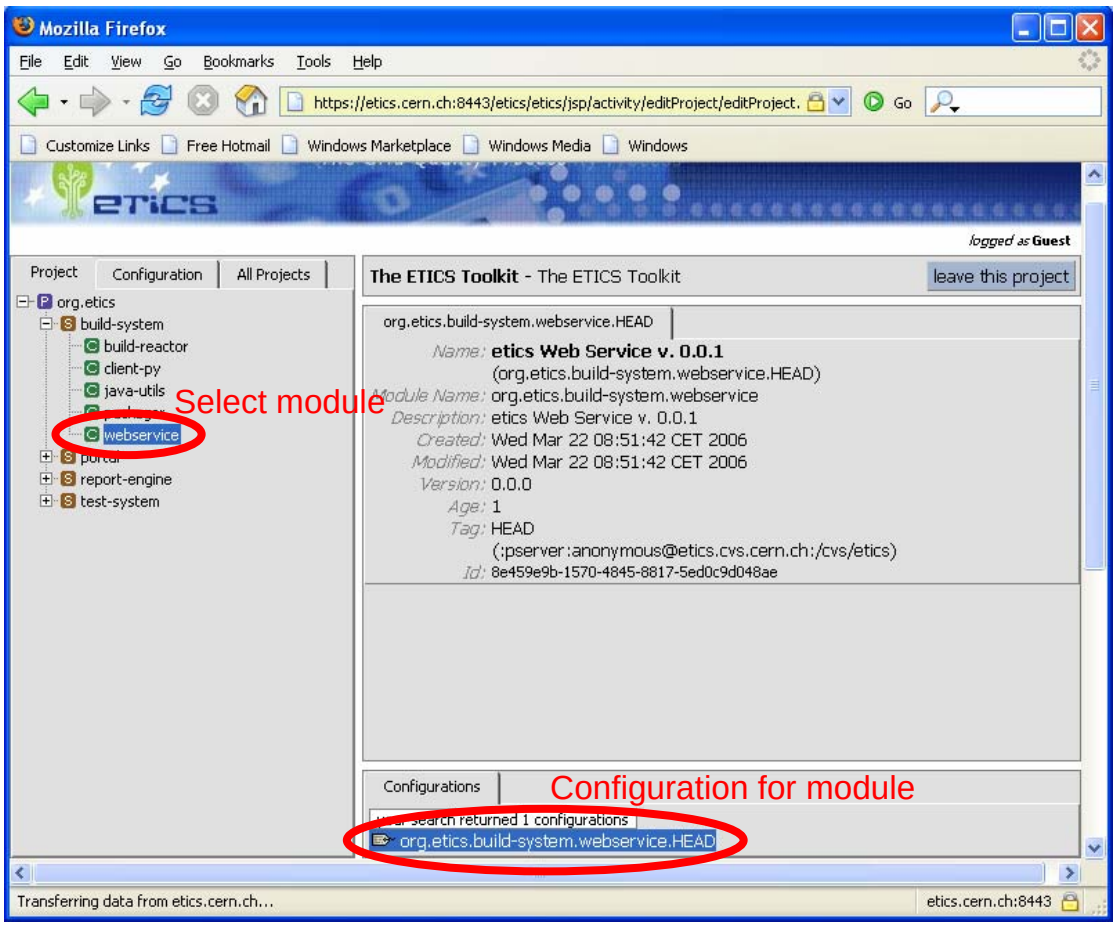

Figure 2: browsing a project

Browsing Configurations

To view the configuration summary just select it in the list.

Then, either right-click on the configuration name and select “Go to” or double-click it. This will display the ‘*Configuration Tree*’, from which you will be able to view the configuration’s metadata, as well as to browse the configuration structure.

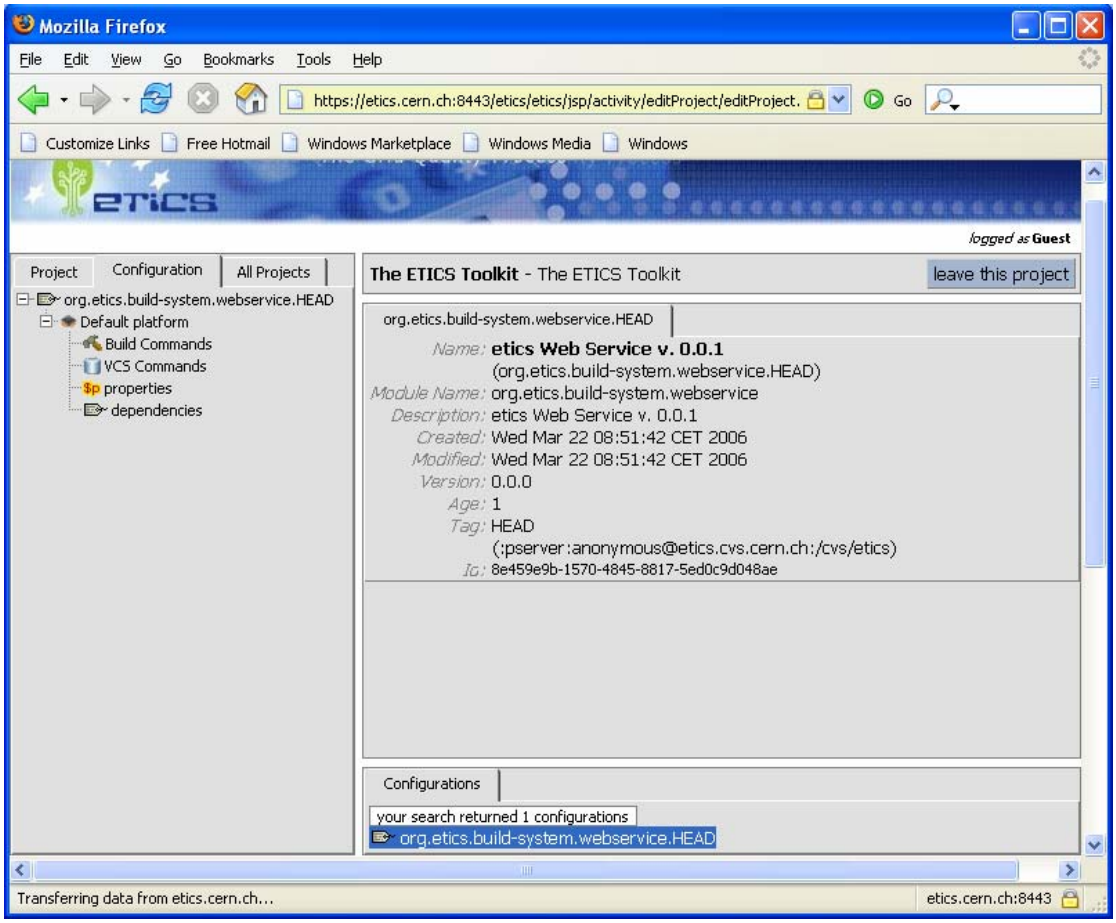

Figure 3: browsing configurations

If you expand a configuration (identified with a tag icon) in the configuration tree, you can see the platforms defined for the configuration.

Expanding a platform node the different constituents of the platform are shown. In the case of the *org.etics.build-system.webservice.HEAD*, the following are defined (see Figure 3):

- Build Commands
- VCS Commands
- Properties
- Dependencies

If you select any node in the configuration tree, as usual, you will see its details displayed on the right side of the page.

For example, you can select '*dependencies'* and view which dependencies this configuration requires, as well as what type of dependencies they are (fields are grayed-out if you do not have editing permissions).

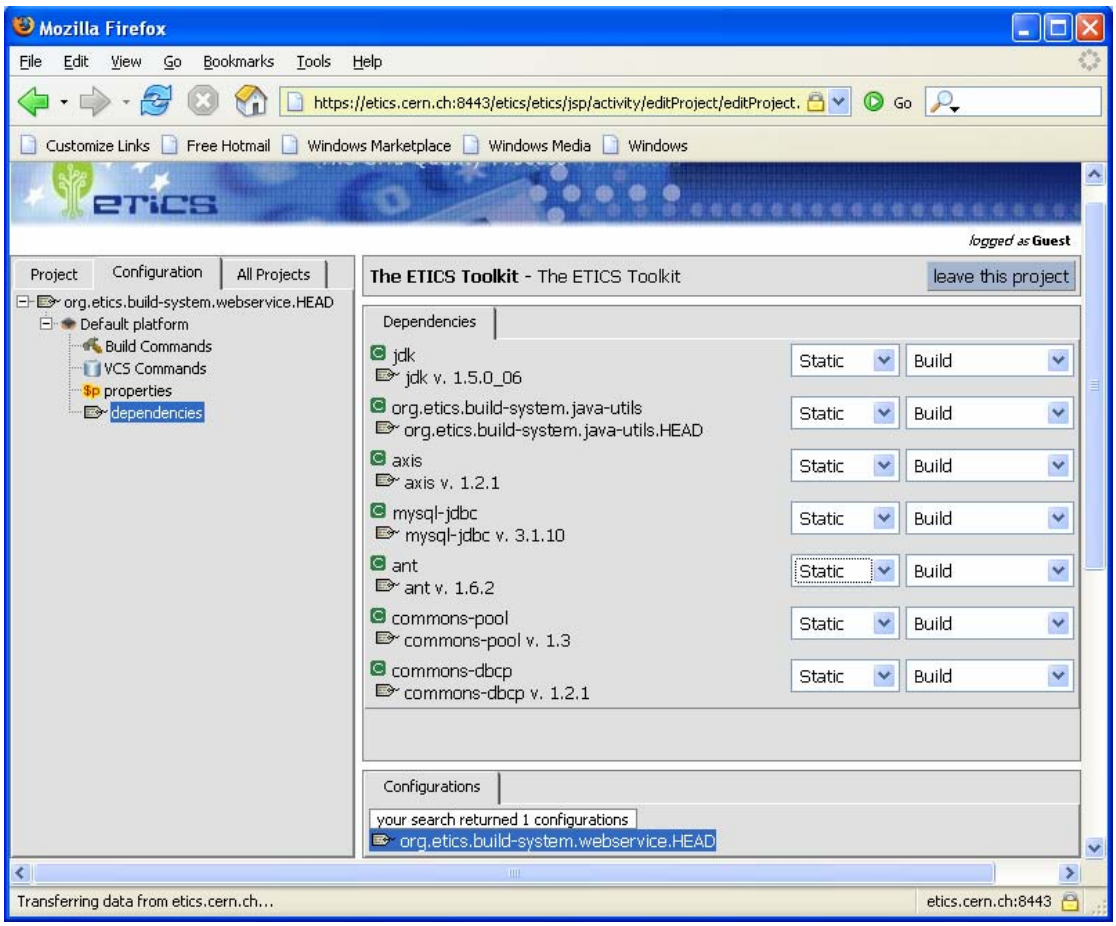

Figure 4: viewing dependecies

To leave the project and go back to the application portal, simply click on "*leave this project"* in the top right corner of the page.

Exercise 02. Load a tutorial certificate in Firefox

In order to be authenticated by the ETICS Web Application (and the ETICS Service), you need to load a valid certificate in your browser. For this tutorial, we have prepared a dedicated certificate, which can be retrieved at:

http://grids17.eng.it/engrepository/etics-training/etics-guest-user-cert.p12

The simplest way to load a certificate in Firefox is to:

a. Linux: go to *Edit -> Preferences -> Advanced -> Security*
b. Windows: go to *Tools -> Options -> Advanced*.

Under the *Certificates (or Security)* section, click on *Manage (or View) certificates.*
Under the tab *Your Certificates* click *Import*.
Select the *p12* tutorial certificate downloaded. When prompted for the private key password, enter 'guest'.

Note: if you already have a user certificate loaded in Firefox, it's a good idea to delete it (to save existing certificates, you can expert them to your local disk) such that Firefox doesn't get confused with which certificate to use when contacting the ETICS web application.

Once the certificate is loaded, close and restart Firefox. To verify that you've loaded the certificate successfully, connect to the ETICS Web Application again, now via TLS:

https://grids17.eng.it:8443/etics

This time, you should see that you are authenticated with the tutorial certificate as indicated in the top-right corner of the window (see Figure 5)

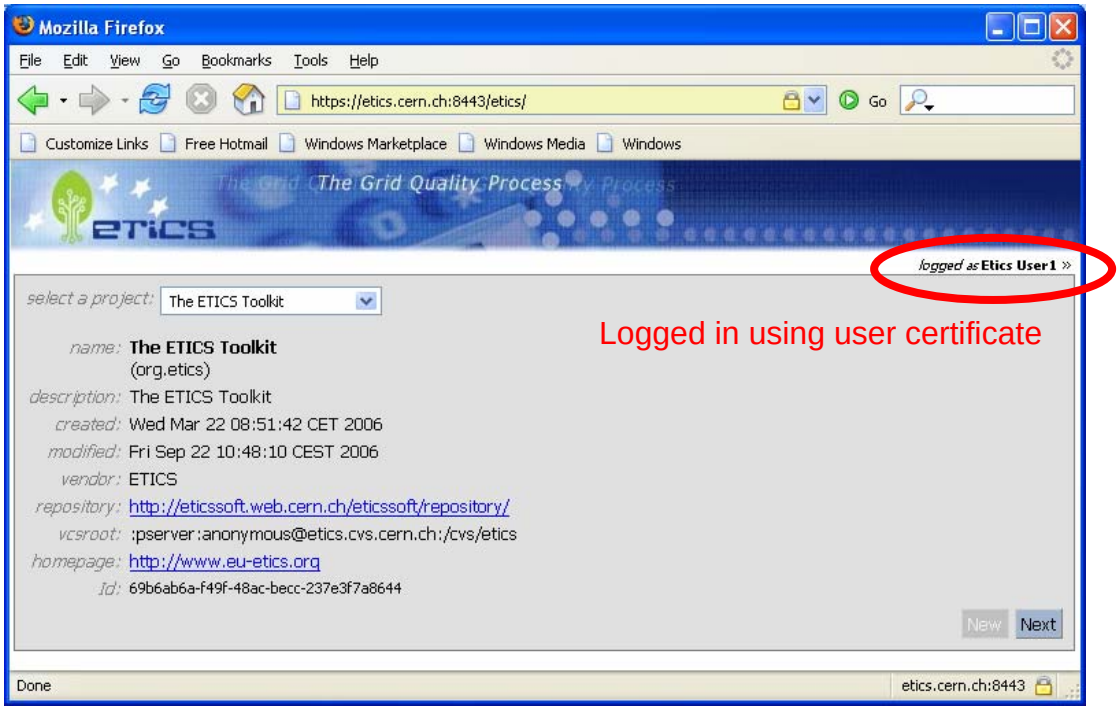

Figure 5: checking your logged identity

Exercise 03. Create a new subsystem and a component

Important: In order to avoid name clashing, you will have been provided with a *tutorial student number*. Please append all the subsystems, components and configurations names with this number.

From the application start page (as done in Exercise 01), select the '*ETICS-Training*' project from the list and click '*Next*'.

Select the project node in the tree, right-click on it and select '*New > Subsystem'.*

Note: if the '*New'* and '*Edit'* buttons on the pop-up menus are grayed-out, it means that the certificate (if any) loaded in the browser doesn't have write privileges on the module you're trying to edit.

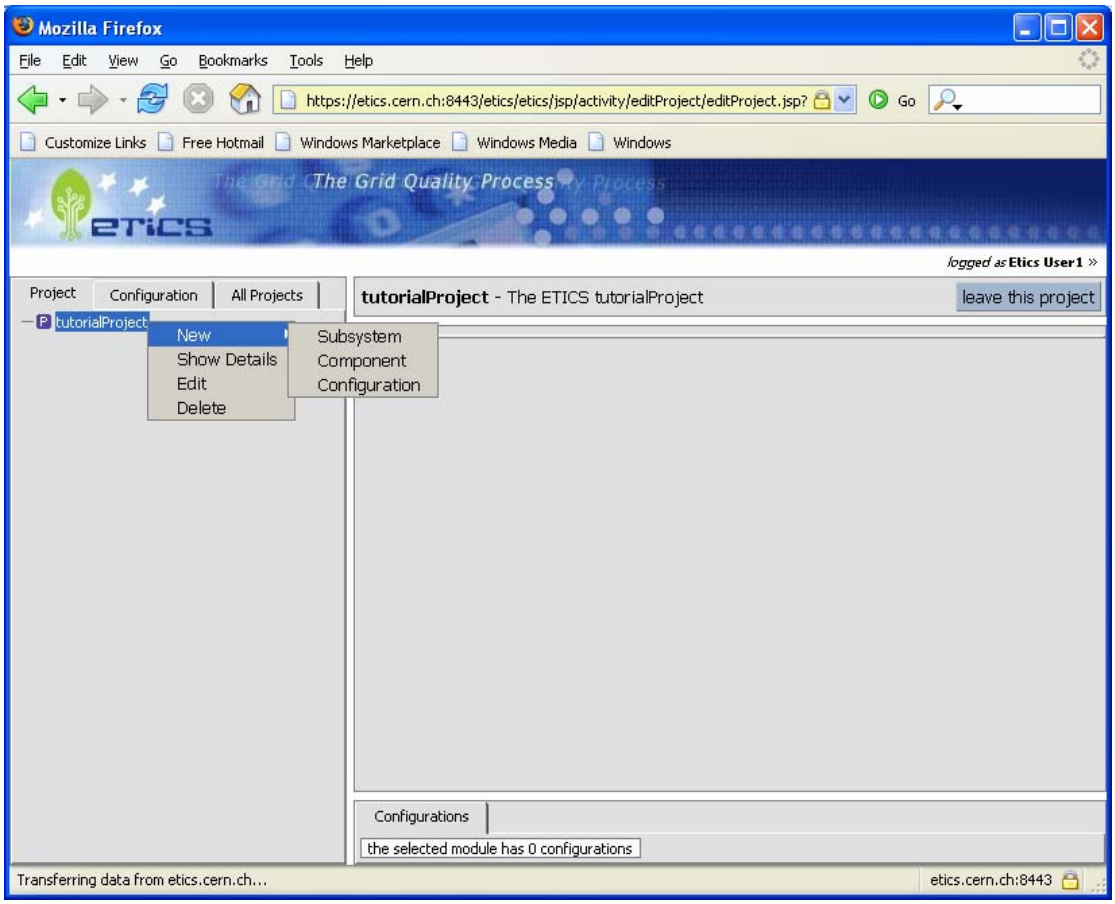

Figure 6: creating modules

Fill the form making sure you choose a unique name for the subsystem (your name or any fantasy name, appended with your tutorial student number). Only use alphanumeric characters for the name field.
Save the form.

Verify that the subsystem has been correctly created in the project tree. Also check that a default '*<your-subsystem>.HEAD*' configuration has been created.

Exercise 04. add and configure Components and their Configurations

In this exercise, you will create and configure a component pointing at an existing training module on a remote CVS. The objective of this exercise is to show you how to model an existing module.

Select the **subsystem** created in the previous exercise, right-click on it and select '*New > Component'*. Fill the form making sure you set, at least, the *name* field:

name = component<N>

where <N> is your assigned tutorial student number.

Fill the other fields to the values you want.
Save the form.

Now select the **component**, double-click on its configuration and open the edit form by right-clicking on the configuration name in the configuration tree and selecting *"Edit"*. Fill the form with the following entries:

version = <any version number>
tag = HEAD

Save the form.

In order to checkout, build and test configurations, more information is needed. We first need to choose a platform for the build. To do that, right-click on the configuration in the tree and select '*Attach platform > Default Platform*'

A '*Default platform*' entry will be created below the configuration.

We now need to tell the system where to find the source code. In ETICS, this is done using **VCS Commands**. Right-click on the '*Default platform*' entry and select '*Add VCS Commands*'

The '*VCS Commands*' entry is created below the '*Default platform'*. A default checkout command is already set to the appropriate value for CVS (see Figure 7)

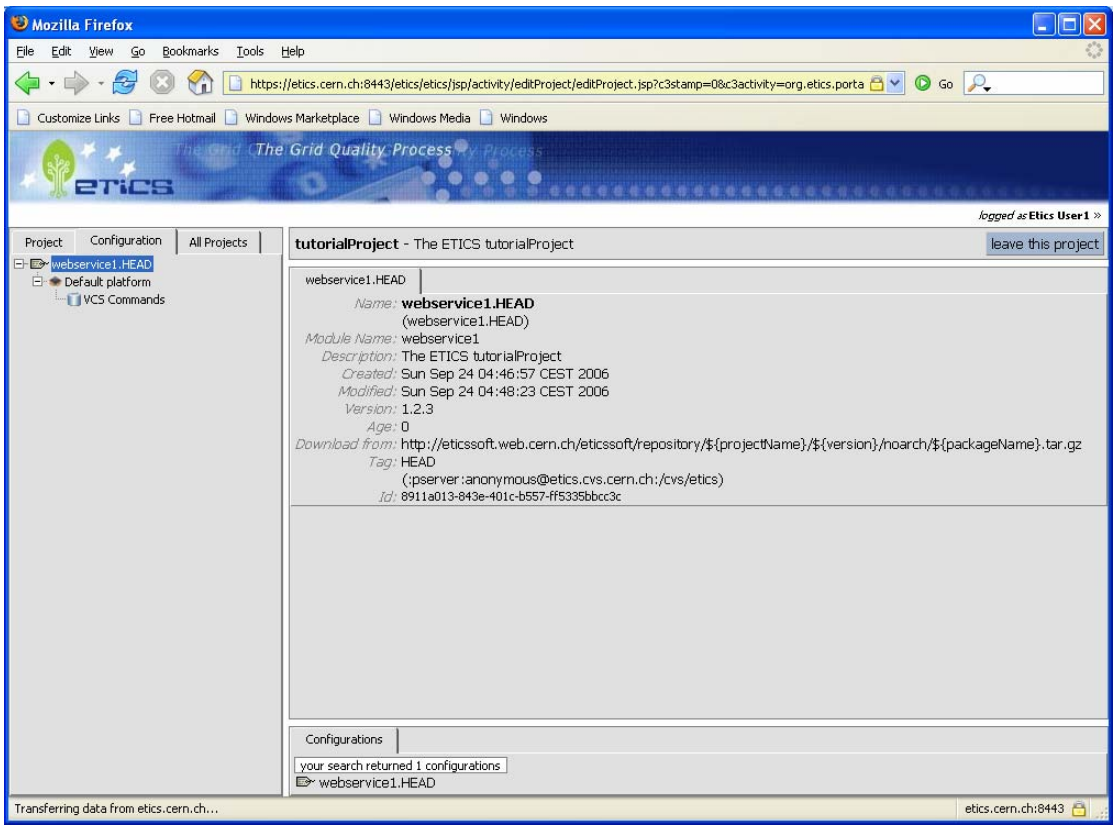

Figure 7: ??

The last step in this exercise is telling the system how to build your component.

In ETICS, this is done using **Build Commands**. A '*Build Commands'* node should already exist below the 'Default Platform' node.
If not, or you've removed it, right-click on the *'Default platform'* node and select '*Add Build Commands*'. A '*Build Commands*' node will appear.

Now, go to editing the Build Commands (right-click on the node, then '*edit*'). You can now edit each entry of the build Commands by either double-clicking on it or by clicking on the arrow on the right.

Enter the following values:

- *clean* = ant clean
- *init* = ant init
- *compile* = ant -Djaxb.location=${org.diligentproject.ext.jaxb.moduleDir} -Dhsqldb.location=${org.diligentproject.ext.hsqldb.moduleDir} jar
- *install* = mkdir -p ${prefix}/lib; cp lib/*.jar ${prefix}/lib

When finished, click '*Save*'.

Exercise 05. Assign dependencies to an existing configuration

In this exercise, you will set dependencies over existing software components to the configuration you've created in the previous exercise

Continue from where you left it in Exercise 04. Right-click on the '*Default platform*' entry and select '*Edit Dependencies*'. A '*dependencies*' node is created below '*Default platform*' and the tree refocuses on the '*All Projects*' tab on the left side of the page.
Your component requires the following dependencies:

jaxb
hsqldb

Expand the '*DILIGENT'* project in the tree, then expand the '*ext*' subsystem. Navigate to each of the required dependencies. For each of them, select it in the tree, then look at the available configurations (Figure 8)

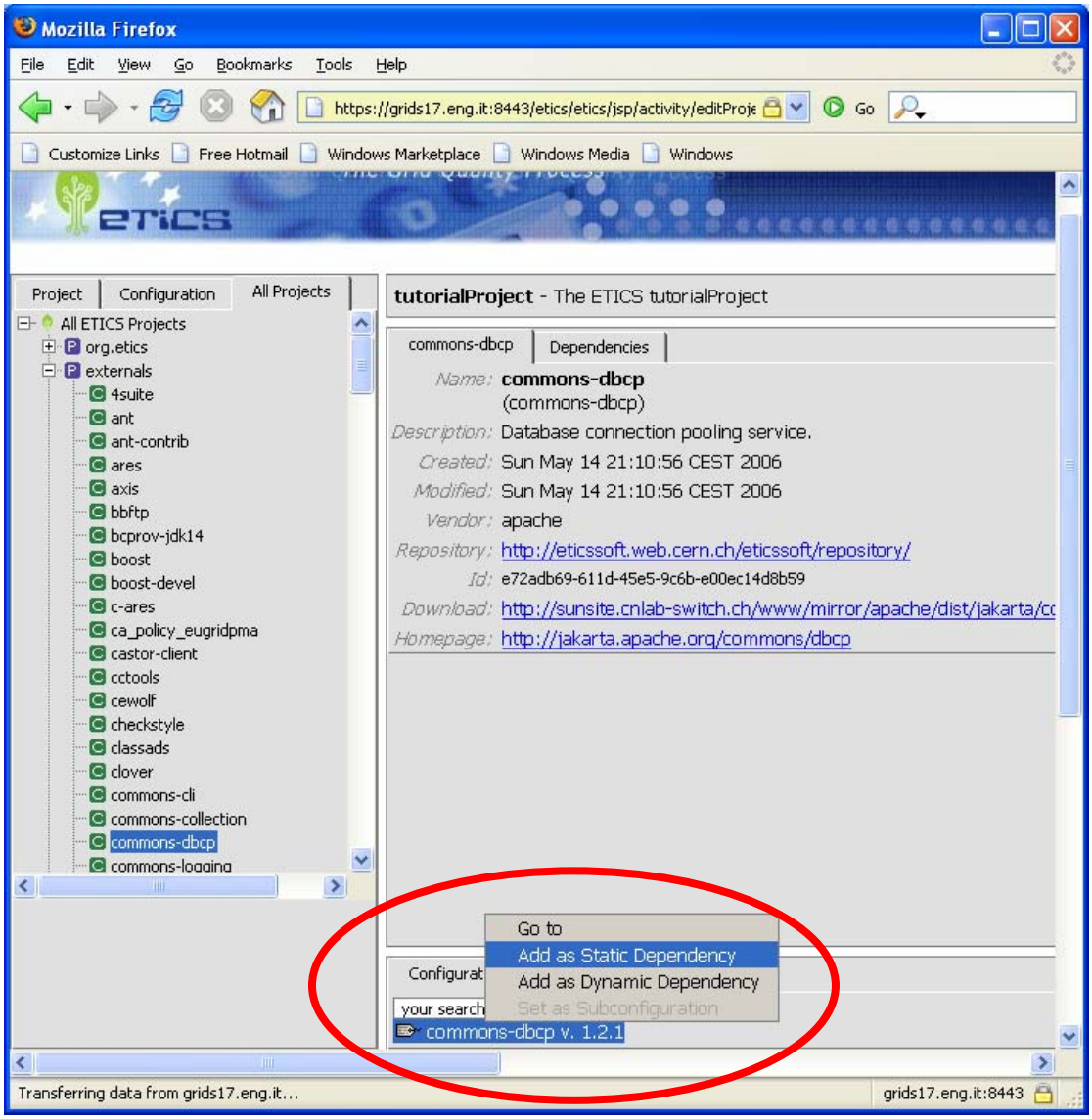

Figure 8: adding dependencies

Now, you can right-click on a configuration to select '*Add as static dependency*' or '*Add as dynamic dependency*'. Set both components as static dependencies.

For each dependency the scope has to be changed from '*Build & Runtime*' to '*Build*'. To do that, enter in edit mode by double-clicking on each dependency, change the value and click on '*Ok'* to save.

Once you have added the required dependencies, you should have something like the following:

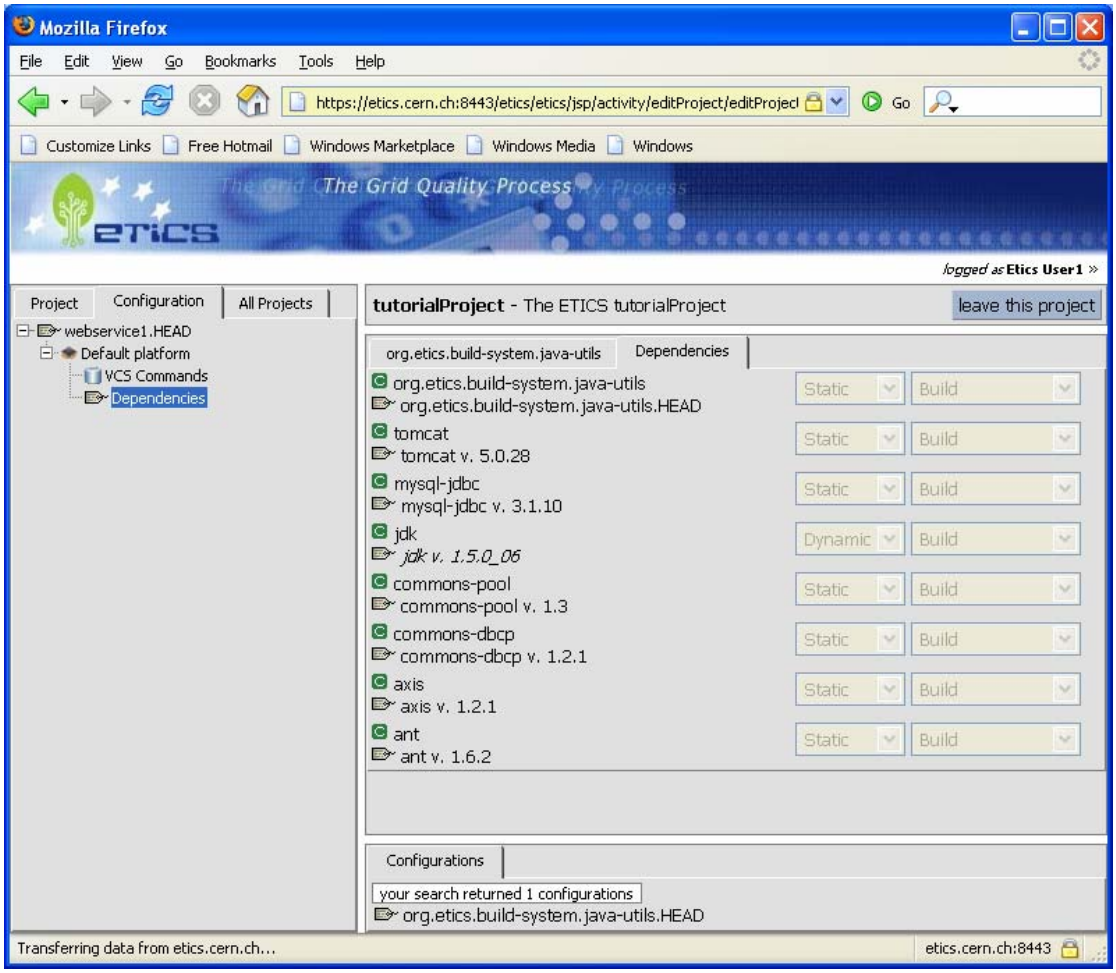

Figure 9: checking dependecies

Exercise 06. Link the Component Configuration to the Subsystem Configuration

Now that we've defined a component configuration, we can attach this configuration to its parent subsystem configuration.

In the project tree, select the subsystem of the component you've being working with in the previous exercises.

Now double-click on its *<your-subsystem>.HEAD* configuration. In the configuration tree, right-click on the configuration and select '*Edit subconfigurations*'.

The list of all components (just one in this case) in the subsystem is displayed. If a configuration for a component has already been attached, it is displayed next to the component name.

Select the component name in the list. As usual, the available configurations for that component are displayed at the bottom of the page. Right-click on the desired configuration name and select '*Set as subconfiguration*'.

Tutorial 2. Navigation with the command-line client

Exercise 07. Setup

Connect, via ssh, to **grids16.eng.it**. Login using the following account:

User: etics_user_<your_student_number>
Password: etics_<your_student_number>

Exercise 08. Client installation

Choose/create an empty working directory in your system and change to it. Download the *etics-client-setup* script using the command:

```
wget "http://eticssoft.web.cern.ch/eticssoft/repository/etics-client-setup.py" -O etics-client-setup
```

Then, run

```
python etics-workspace-setup.
```

Executing this command will fetch and install the ETICS client in the current directory, as well as required dependencies, if required.

Note: Installation of the client requires *python-devel* and *openssl-devel* packages installed. We are working on providing pre-compiled packages.

Have a look at the current directory, some files and directory should have appeared. The most important of them is the `etics` directory where all the ETICS code is.

Set the `ETICS_HOME` and `PATH` environment variables:

```
export ETICS_HOME=<installation-directory>/etics
export PATH=$ETICS_HOME/bin:$PATH
```

For the moment check that the environment variables X509_USER_KEY and X509_USER_CERT are not set in your machine. If they are set, please unset them.

The read operations on the client can be done as GUEST. When you connect without certificates or with a certificate that isn't registered in the ETICS system, you are mapped as GUEST.

Exercise 09. Listing Projects and other Objects

All commands provided by the ETICS client start with '*etics-*'. The list-commands all start with '*etics-list-'.*
All ETICS commands support the '*–h/--help*" option that shows the usage information.

Try

```
etics-list-project
```

and

```
etics-list-platform
```

Now, you can start working on your own subsystem. First, do:

```
etics-get-project ETICS-Training
```

This command selects the project want to work on, and downloads some basic information about it.

Use `etics-list-configuration` to see the configuration associated with your modules. Run:

```
etics-list-configuration <subsystem-name>
```

where subsystem is the one you have created in the Web Application. You should see the *<subsystem-name>.HEAD* configuration.

Exercise 10. Checking-out a configuration

It's now time to checkout your configuration. Do that by using the `etics-checkout` command:

```
etics-checkout <subsystem-name>
```

If the configuration you want to checkout is not called '*<subsystem-name>.HEAD*', use the `–config` option to specify the name:

```
etics-checkout --config <config-name> <subsystem-name>
```

Exercise 11. Configure the Client to Use Authenticated Access (only required if editing through the CLI)

To edit the project metadata you can also use the edit commands, that require Authentication. For this you need to use the training user certificate. Download the files:

http://grids17.eng.it/engrepository/etics-training/guest_user_cert.pem
http://grids17.eng.it/engrepository/etics-training/guest_user_key.pem

Open the user configuration file in `~/.etics.conf`. Set the following variables in it:

`x509_user_cert:` <location of the public certificate file>
`x509_user_key:` <location of the private certificate file>

The first time you use the etics commands, you will need to provide the passphrase of your private key (in this case 'guest')

Tutorial 3. Building with the command-line client

Exercise 12. Building Components

Using the client, checkout your component '*component<N>*'. Then, build the with the following command:

```
etics-build component<N>
```

where N is your assigned student number.

If all commands, properties and dependencies are set correctly, the component will be built using the external dependencies checked out in the previous exercises and produce tarballs and RPMS in the '`dist`' directory of the workspace.

Exercise 13. Building Complete Subsystems

Repeat the steps of previous exercise to build entirely your subsystem instead of a single component.

Exercise 14. Building a DILIGENT Component/Subsystem

Repeat the steps you did in exercises 9 and 10 to checkout your own DILIGENT subsystem.

Note: you may need to specify a password to access your (or your dependencies) VCS. Passwords in VCS Commands have been parameterized; so you can give passwords as properties in the command line, as in the following example:

```
etics-checkout -p password_cnr=<<passCNR>> -p password_uoa=$<<passUOA>>
-c <<configurationName>> <<moduleName>>
```

This is the list of properties to be set. Values for them will be given:

```
password_cnr
password_uoa
password_fhg
```

Repeat the steps of exercise Exercise 12 to build a Component in your subsystem.