

Thinkclipse / Nutpse

Configure the simple example with Eclipse's plugin :

Thinkclipse

Contents

1	Requirements	2
2	Installing thinkCclipse plugin	2
3	Setting up a new project	4
3.1	Creating the project	4
3.2	Copying files	5
4	Using the plugin	6
5	Compiling	8
5.1	Some modifications	8
5.2	Moving into action	9
6	Contacts	9

1 Requirements

You need :

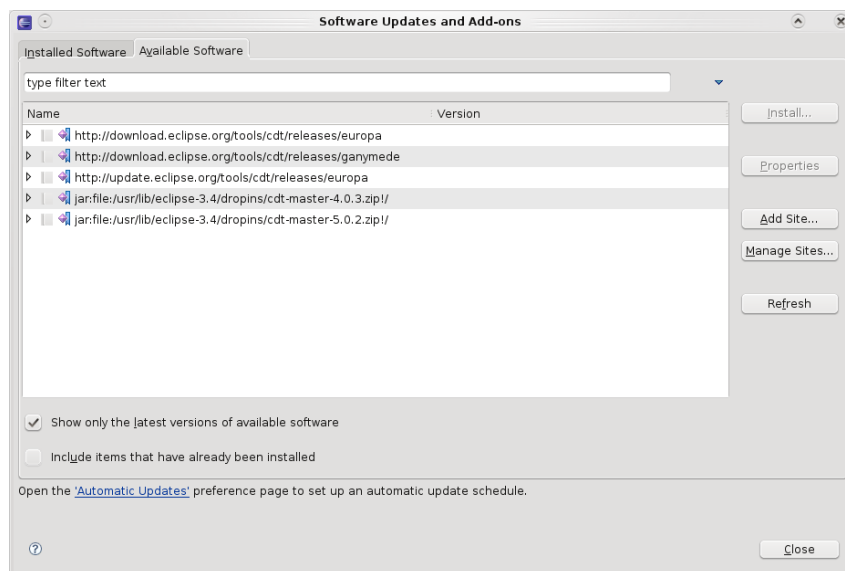
- * Eclipse¹
- * thinkClipse 2.0.3
- * nutpse unstable
- * kortex

2 Installing thinkClipse plugin

Open Eclipse and go to the menu :

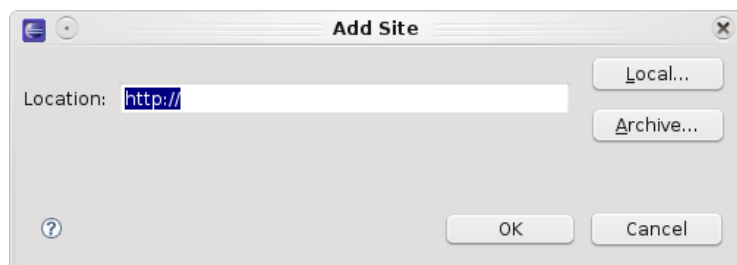
Help - Software updates... and select the second tab **Available Software**

Then you will see this window :



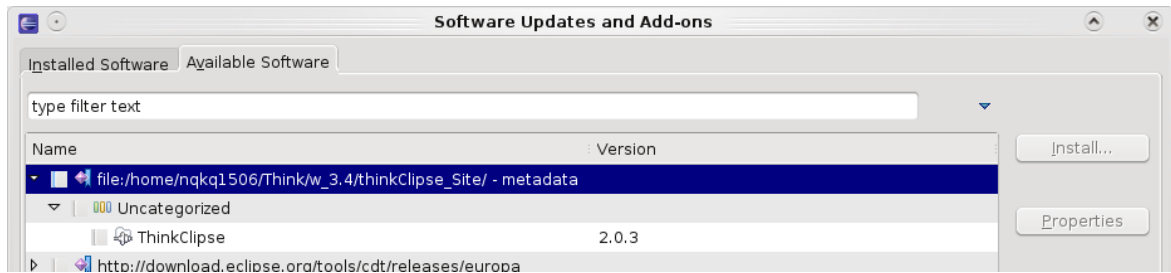
Click on the **Add site...** button, and type :

`http://think.ow2.org/thinkClipse/`



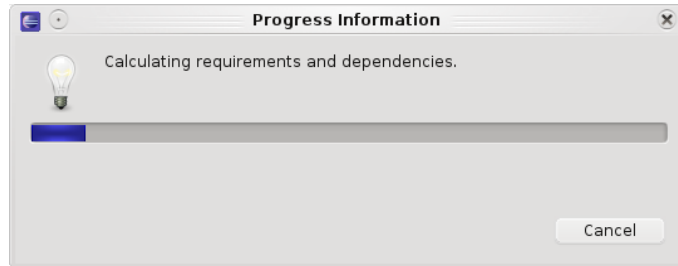
Validate and in the **Available Software** tab, you will see :

¹I have tested with Eclipse 3.3 and 3.4.



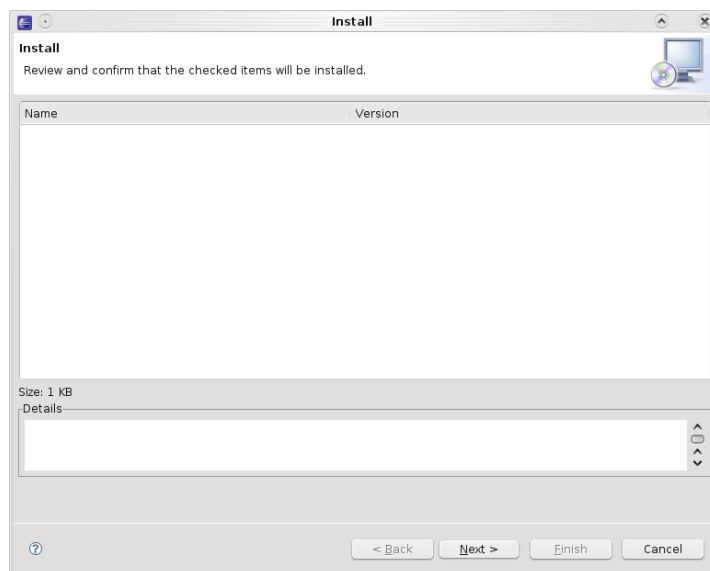
Then check ThinkClipse and click on **Install**.

You will see this window (be patient !!) :

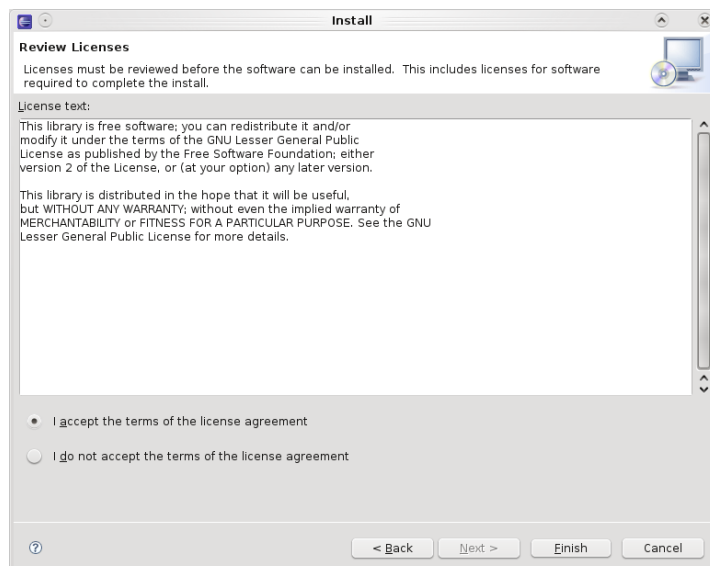


A new window appears, just click on **Next**.

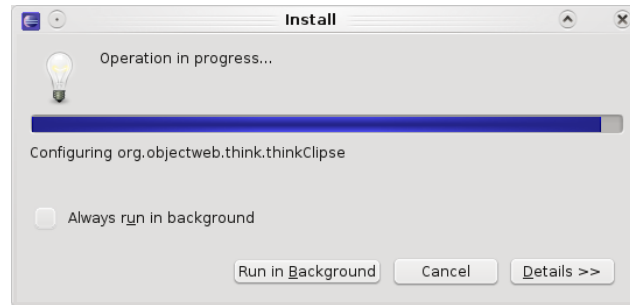
Don't worry about the size.



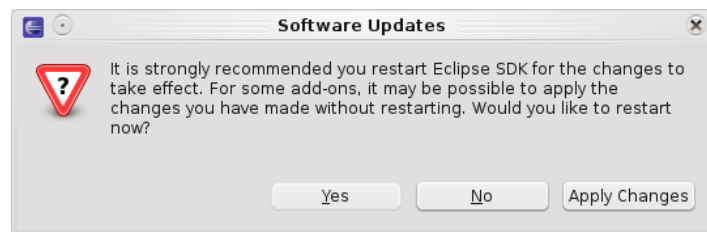
Select the option **I accept the terms of the license agreement** and click on **Finish**.



The installation will continue (faster this time) :



I recommend to restart Eclipse, using the **Yes** button, but it also works with **Apply changes**.

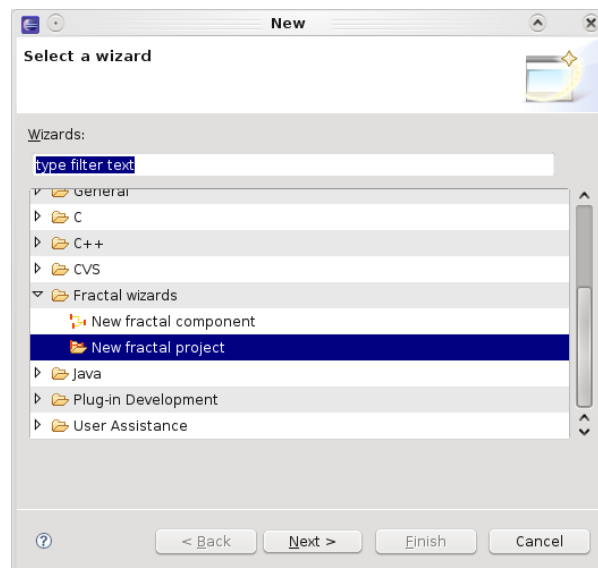


3 Setting up a new project

3.1 Creating the project

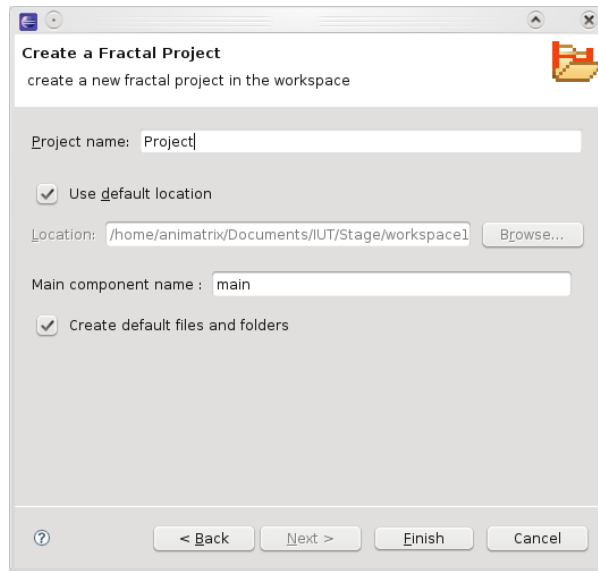
Go through the menu **File - New - Other....**

Open the **Fractal wizards** view and select **New Fractal project** and validate with **Next** button.

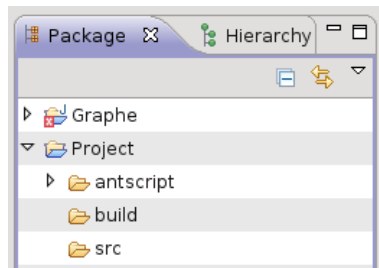


You just have to enter a *Project name* and click on **Finish** button.

You also need to set **helloWorld** as the **Main component name**.



Now, you can see on the Package view the project :



3.2 Copying files

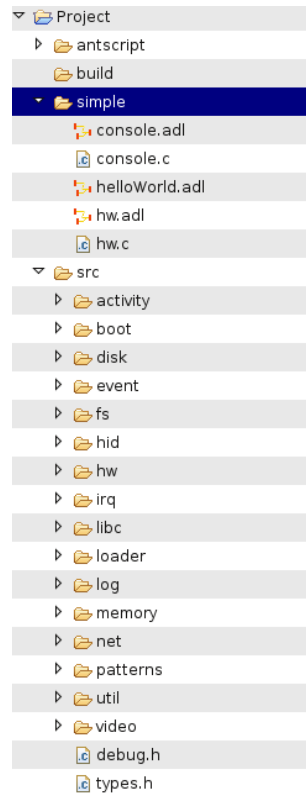
Extract the nuptse file and copy the content of **example/simple/src** in a new sub-directory of your project (example : simple).

Extract the kortex file and copy the content of **src/generic** in the **/src** folder of your workspace.

Now, go back to Eclipse.

In the Package view, Refresh the view, using the key **F5**.

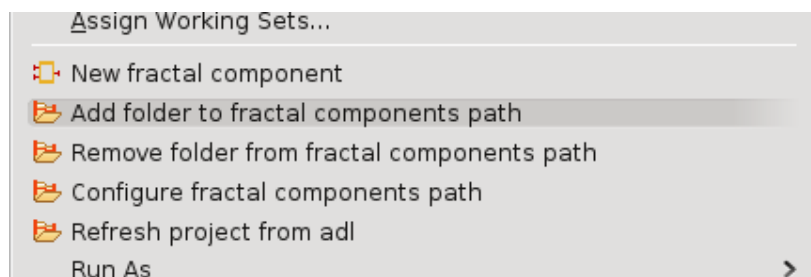
You should see :



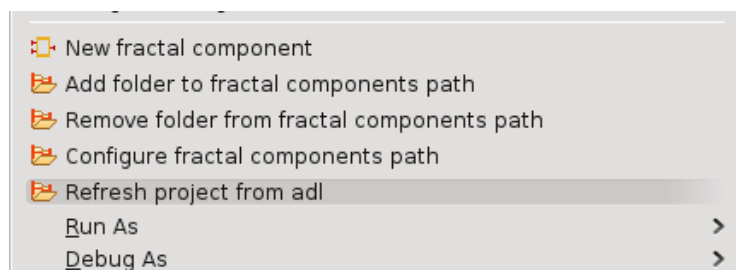
4 Using the plugin

Because your components are in the simple folder, you need to add them.

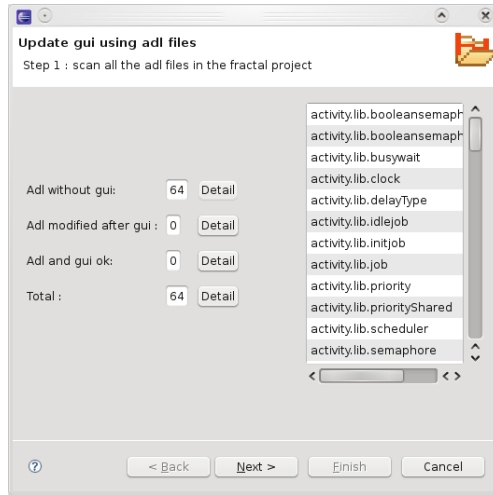
Right click on the simple folder, in the package view and select **Add folder to fractal components path**.



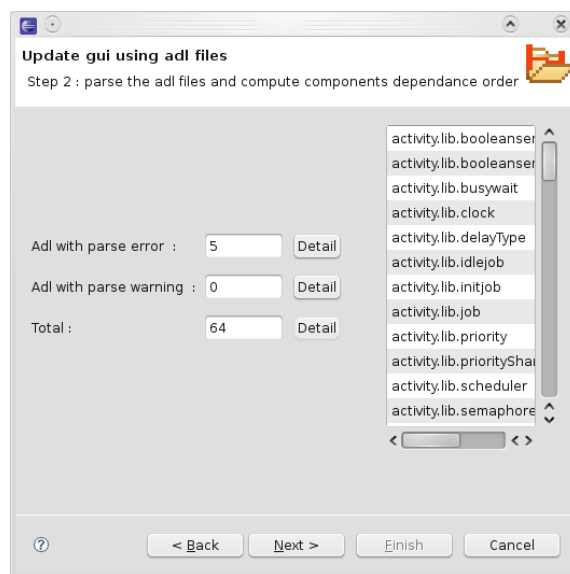
In order to generate graphs, right click on a file or folder of the project and select **Refresh project from adl**.



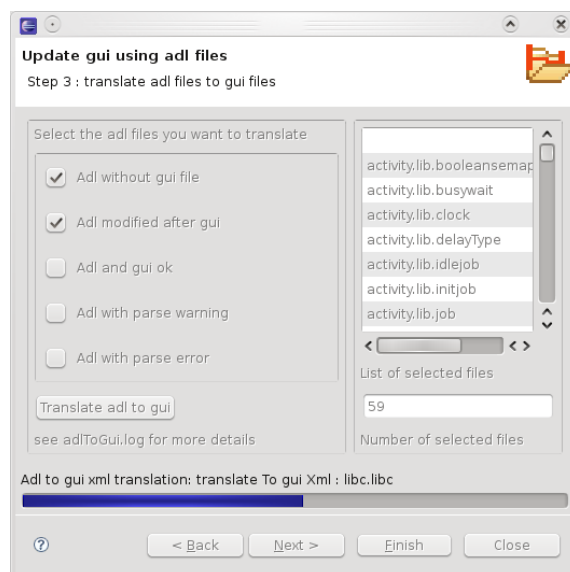
You have 64 graphs to make. So click on **Next** button.



There are 5 errors (from kortex), but continue, using the **Next** button.

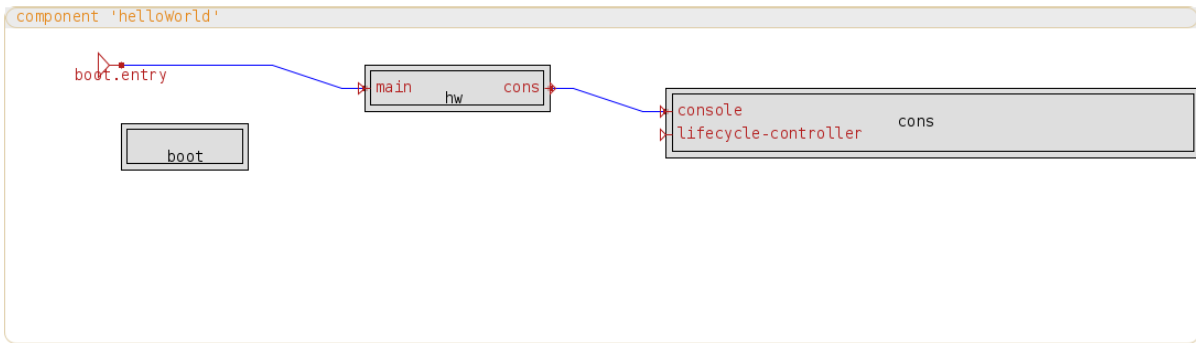


Just click on **Translate adl to gui** and wait !



That's all for graphs. For each *.adl file (eg. helloWorld.adl), you have several tabs :

Architecture :



Properties :

General	
Id	Value
Project	Project
Package	
Component	helloWorld
Location	/home/animatrix/Documents/IUT/Stage/workspace1/Project/.gui
Path	.gui/helloWorld.gui
Abstract	No
Parameters	
Extends	Edit Browse
Extends parameters	
Extension properties	
Singleton	No
NoLifecycle	No
Controller	
CFlags	
LdFlags	
Interfaces	
Attributes	

Adl description :

You can easily edit the file, with colored syntax enabled.

```

29 component helloWorld {
30
31     // Contains the base sub-component named 'hw' and which definition is given
32     // in file hw.adl
33     contains hw = hw
34
35     // Contains a boot and a cons components.
36     contains boot = boot.lib.boot
37     contains cons = console
38
39     // Interface client 'entry' of component 'boot' is bound to server interface
40     // 'main' of component 'hw'.
41     binds boot.entry to hw.main
42
43     // Interface client 'cons' of component 'hw' is bound to server interface
44     // 'console' of component 'console'.
45     binds hw.cons to cons.console
46
47     // attribute of sub components can be assigned here (this attribute is defined in
48     // hw.adl)
49     // assigns hw.positionX = 4
50 }

```

5 Compiling

5.1 Some modifications

In the package tree, you can see the folder **antscript**.

All files have been auto-generated.

But you need to edit **compiler.properties**, using the **Properties File Editor** :

```
nuptse.install /Path/To/nuptse  
thinkadl.root /Path/To/nuptse/thinkadl
```

The last thing to do is to copy **build.xml**, from **template/** to **thinkadl/** (in the nuptse folder).

5.2 Moving into action

To compile the project, just right click on **build.xml** and select **Run as - Ant build**.

If you see **BUILD SUCCESS**, everything is fine !

6 Contacts

Edouard Marquez : edouard.marquez@iut-valence.fr

Sébastien Jean : sebastien.jean@iut-valence.fr