

PIPS

Tutorial to use PIPS with Eclipse
Rapport technique E-334

Nelson LOSSING

January 10, 2014

Contents

| | | |
|-----------|---|-----------|
| 1 | Introduction | 3 |
| 2 | Prerequisites | 3 |
| 3 | Add the repositories of PIPS and their dependencies in Eclipse | 3 |
| 4 | Make the projects for PIPS | 5 |
| 5 | Configure your projects | 8 |
| 6 | How can we compile PIPS with Eclipse? | 12 |
| 7 | How can we debug PIPS with Eclipse? | 13 |
| 8 | Why can I only launch the debug for tpips only once? | 15 |
| 9 | Why debug step go randomly top and down? | 15 |
| 10 | How can I ask Eclipse to not recompile before launch debug? | 15 |
| 11 | How can I use SVN in Eclipse? | 16 |

1 Introduction

The goal of this tutorial is to explain how we can use Eclipse as IDE to work on PIPS. You can also read the PIPS developer_guide ¹ and the official web site of PIPS <http://pips4u.org> for more information.

This document will explain how we can configure the project linked with the SVN sources. But also how to compile and to launch PIPS with Eclipse and so use the debugger provided by Eclipse.

This tutorial and the screen shoots are done with Eclipse JUNO version.

2 Prerequisites

Your Eclipse has to support C/C++ project.

You need to install *subversive* in your Eclipse. You can do this in “*Help/Install New Software...*” and search svn to find subversive.

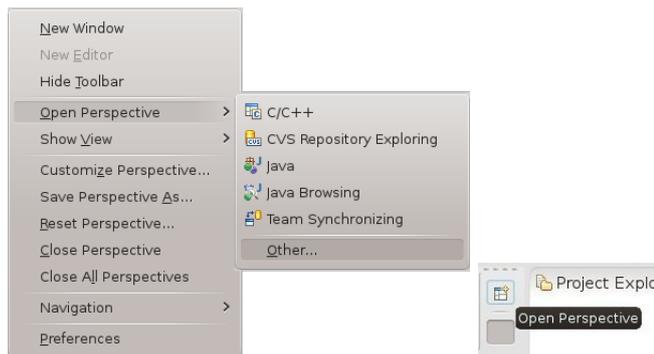
When you relaunch Eclipse after the first installation of *subversive*, it will probably ask you to install a connector. You can install the last version of SVNKit.

3 Add the repositories of PIPS and their dependencies in Eclipse

The first thing to do is to add the SVN repositories of PIPS and their dependencies.

1. Switch to the *SVN Repository Exploring*'s perspective.

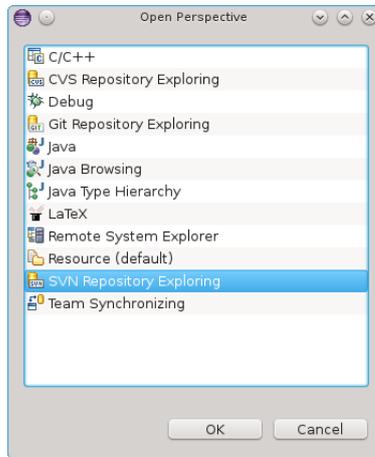
- (a) You can find this perspective in the perspective windows in menu “*Windows/Open Perspective/Other...*” (right picture) or button *Open Perspective* (left picture).



- (b) Choose *SVN Repository Exploring*. It will also add the shortcut in the *Perspective view*.

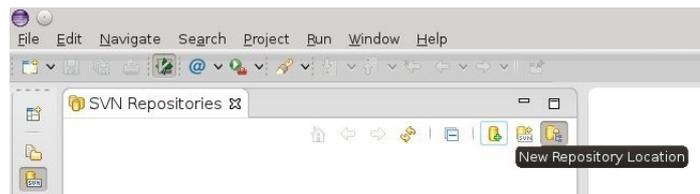
¹pdf version: http://www.cri.enscm.fr/pips/developer_guide.htdoc/developer_guide.pdf

HTML version: http://www.cri.enscm.fr/pips/developer_guide.htdoc

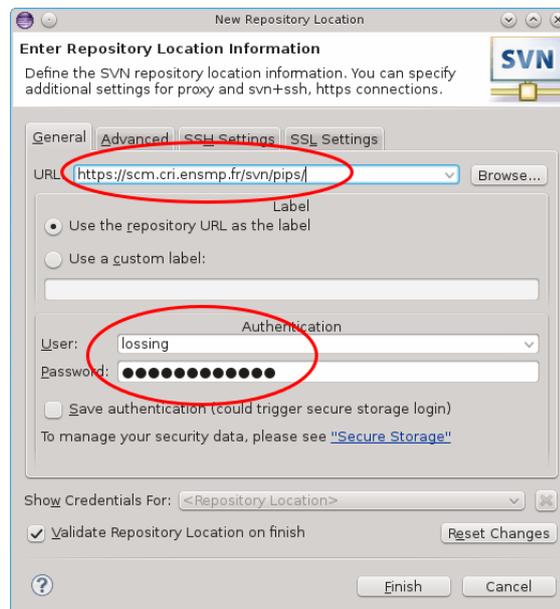


2. Add The repository location of PIPS
(<https://svn.cri.ensmp.fr/svn/pips/>).

(a) Open the *New Repository Location* wizard.



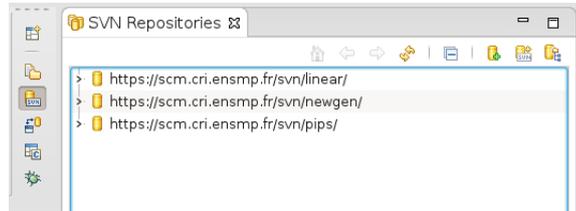
(b) Put the URL of PIPS <https://svn.cri.ensmp.fr/svn/pips/> and your authentication login of PIPS.



(c) click *Finish*.

3. Do the same for linear (<https://svn.cri.ensmp.fr/svn/linear/>) and Newgen (<https://svn.cri.ensmp.fr/svn/newgen/>).

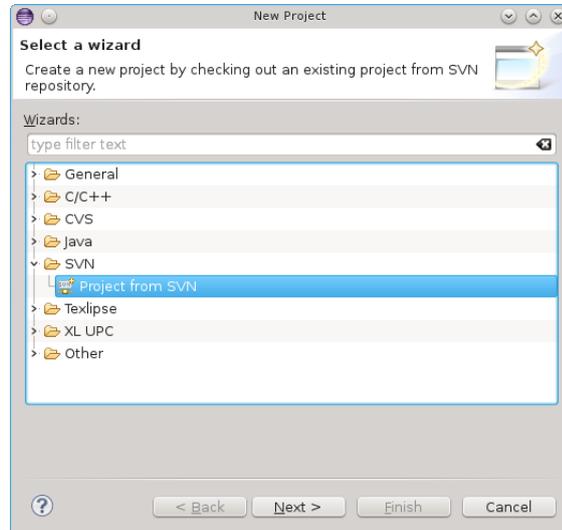
You will normally have something like that at the end:



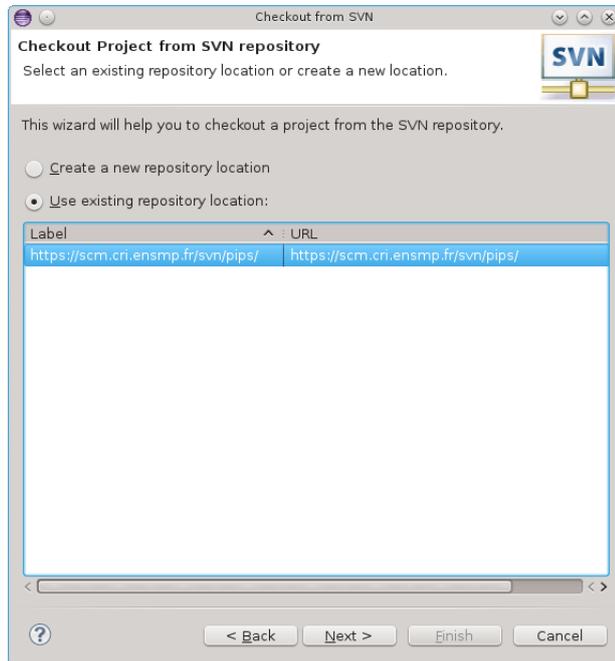
4 Make the projects for PIPS

You can now make a project that will be managed by Eclipse for PIPS under SVN.

1. Go back to the *Ressource* Perspective or to the *C/C++* Perspective (see Sec 3 step 1a to change the perspective).
2. Add a new SVN project for PIPS
 - (a) Create a new project (File/New/Project).
 - (b) Select a SVN project.

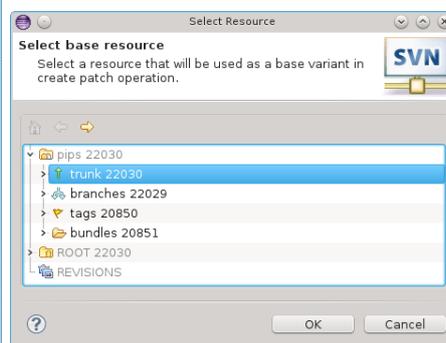
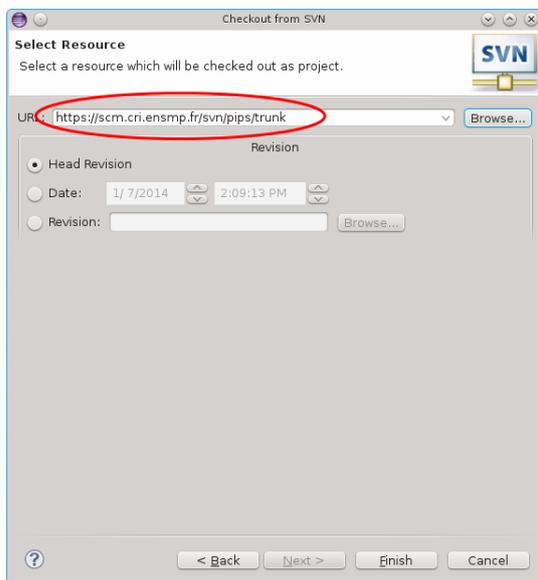


- (c) Select the PIPS URL.

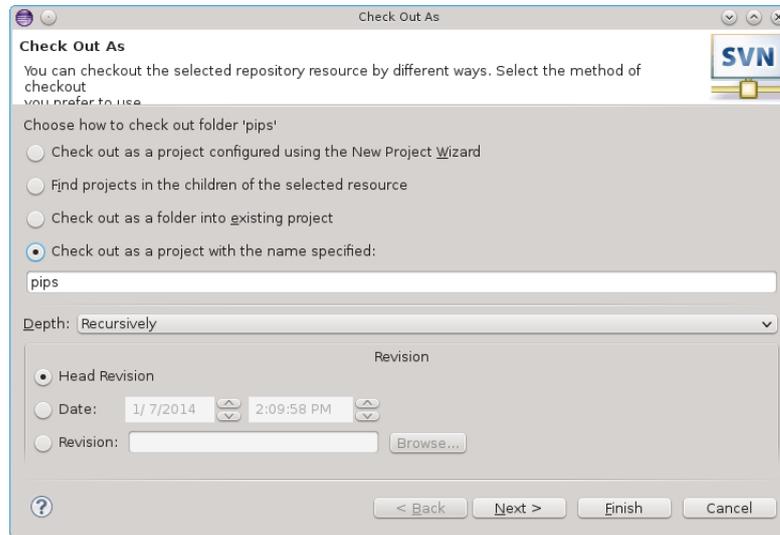


- (d) Select the trunk or your branch copy of the trunk. You can use “*Browser...*” to help you select it. You can also choose an older revision if you want.

WARNING: if you don't select the trunk or your specific branch, it will make a new project with all the depositories of PIPS (including trunk, all the branches and the tags).



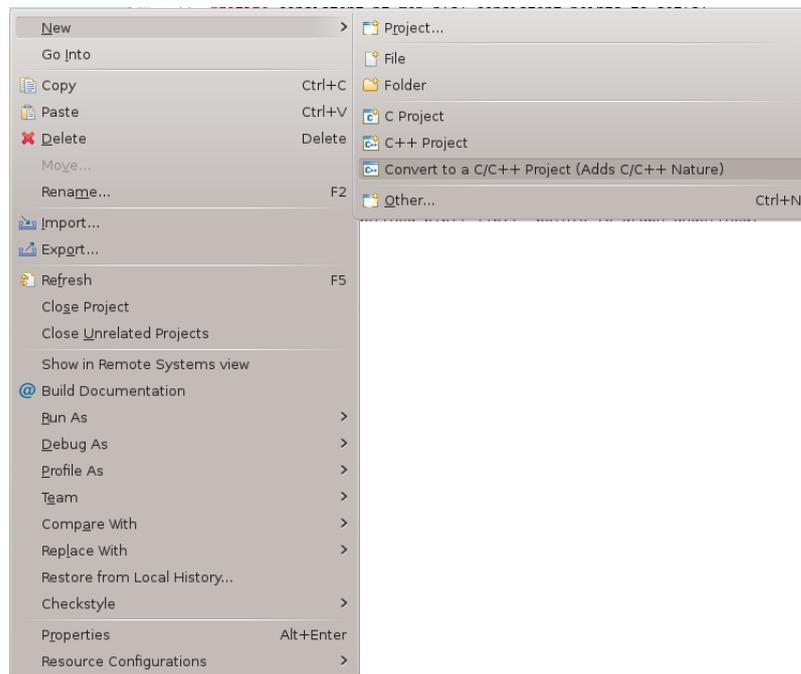
- (e) Click “*Finish*”.
- (f) A new window appears. Choose *Check out as a project with the name specified*.



(g) Click “*Finish*”.

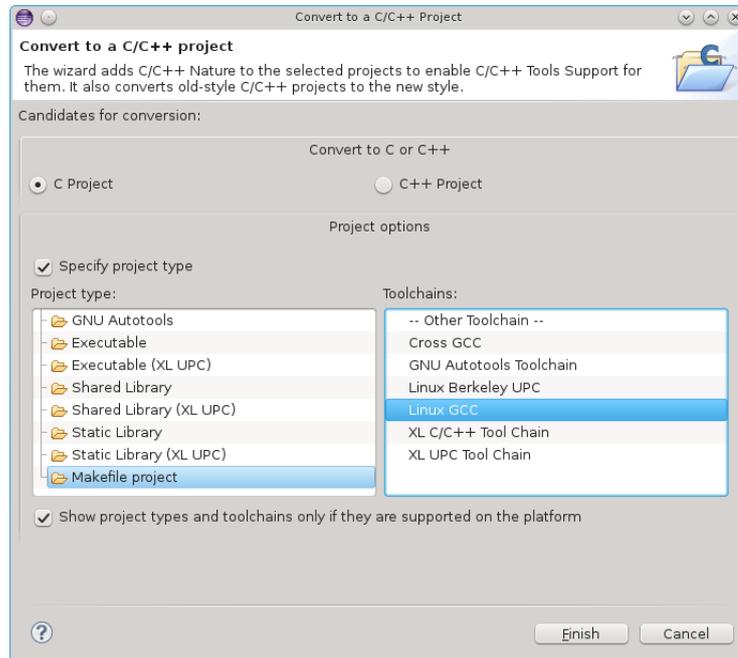
3. Convert this new project into a C project

(a) Right click on this new SVN project and choose “*New/Convert to C/C++ Project (Add C/C++ Nature)*”



(b) Choose to convert into a C project.

Note: you can convert many projects in one step, if your SVN project is added as Candidate for conversion (empty in my screen shot)



(c) Click “*Finish*”.

4. Do the same for linear and newgen and your personal version of PIPS (in your branch). (Sec 4 step 2.)

5 Configure your projects

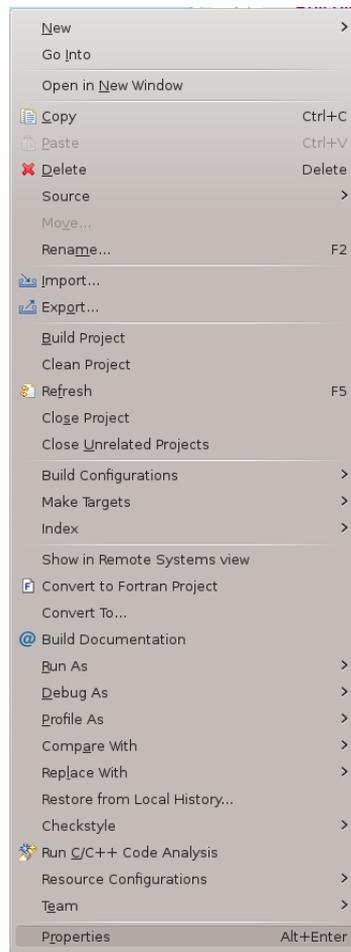
1. Add the *include* path for the different dependencies.

For this purpose you can see the *developer_guide* part **Section 3.2.2 Missing includes**. In summary, these configurations are done in the property project, menu “C/C++ General”, submenu “Paths and Symbols”, “Include” tab.

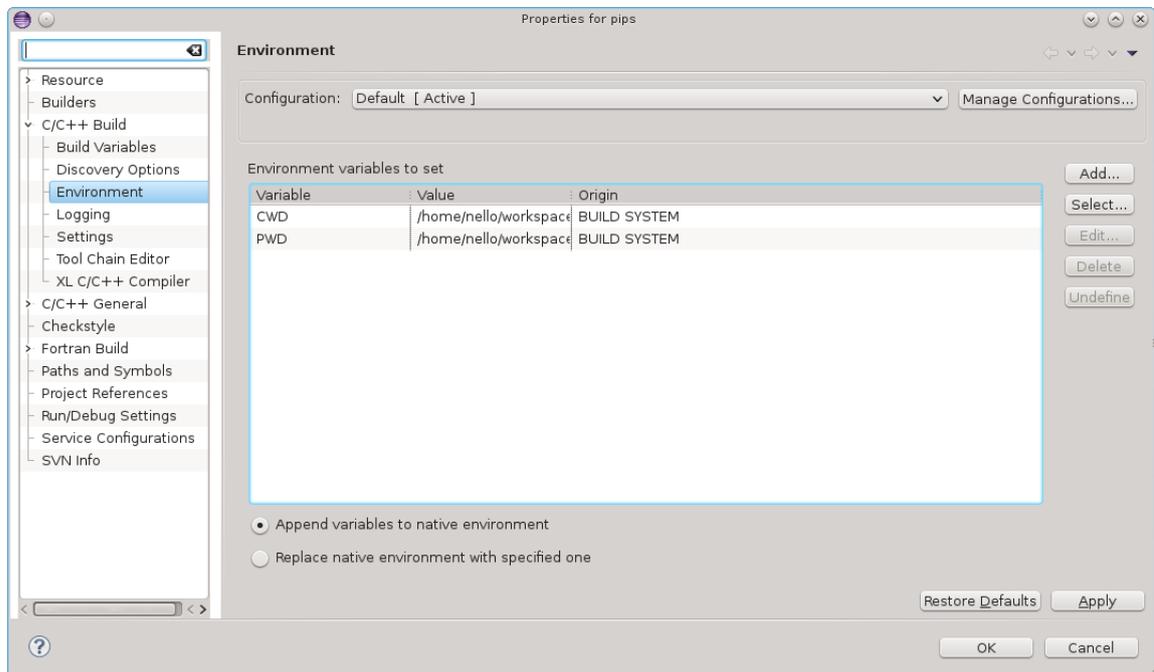
Without these includes, you won’t be able to use the power of Eclipse.

List of includes for the different projects:

- (a) **newgen** needs: `/newgen/include`
 - (b) **linear** needs: `/linear/inlcude` AND include from polylib
 - (c) **PIPS** needs: `/newgen/include`, `/linear/inlcude` and `/pips/include`
2. Modify the *Path* Environnement to add pipsrc. It corresponds to the path added with pipsrc.sh.
 - (a) Open the properties of your pips project
Alt+Enter or right-click on your pips project or *File/Properties*.

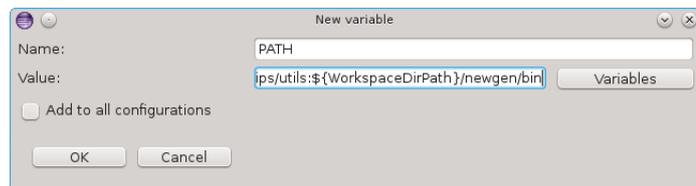


(b) Select *C/C++ Build/Environment*. Click Add.

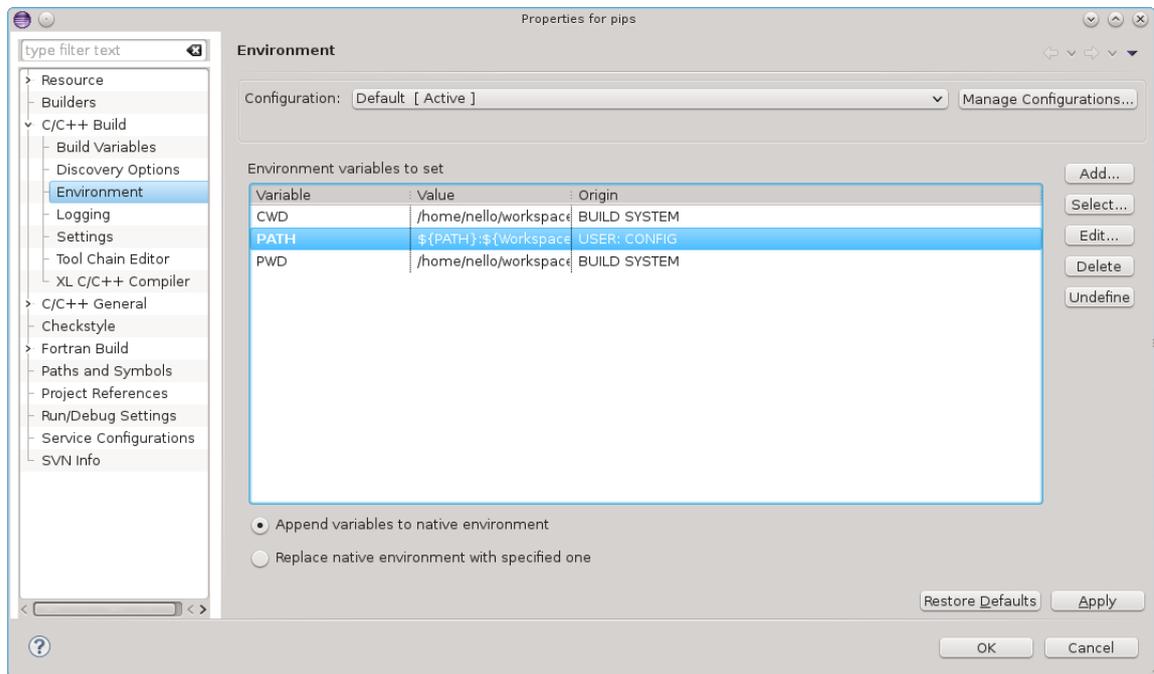


- (c) Add the variable ***PATH*** with the value
PATH:
{ WorkspaceDirPath }/pips/bin:
{ WorkspaceDirPath }/pips/utills:
{ WorkspaceDirPath }/newgen/bin

Replace *pips* by the name of your pips project if it needs.



- (d) *Apply* your modification and click *OK*.

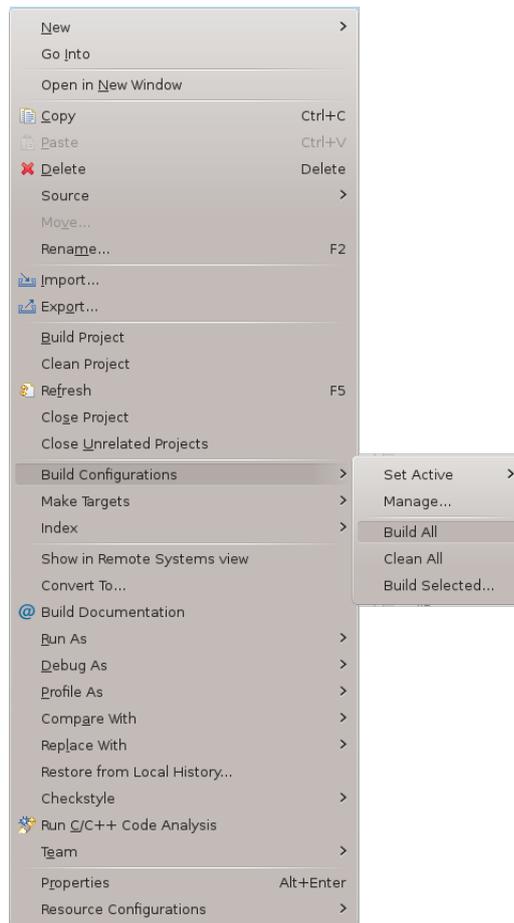


After this configuration has been done, you can use PIPS in Eclipse.

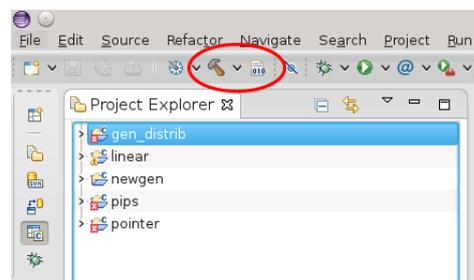
6 How can we compile PIPS with Eclipse?

Compilation corresponds to build your project in Eclipse.

1. Right click on the project you want to compile and choose build:



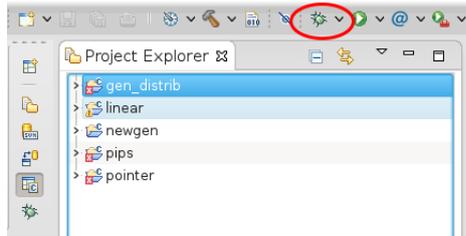
2. You can also select your project and use the shortcut in the C/C++ perspective:



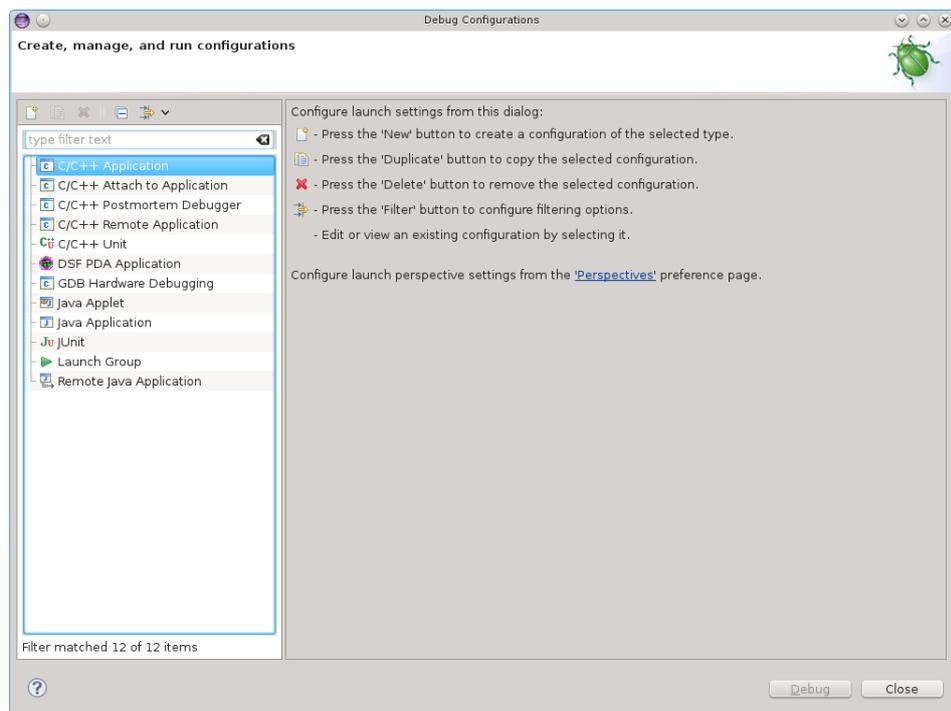
7 How can we debug PIPS with Eclipse?

For this purpose, you will need to set a debug configuration.

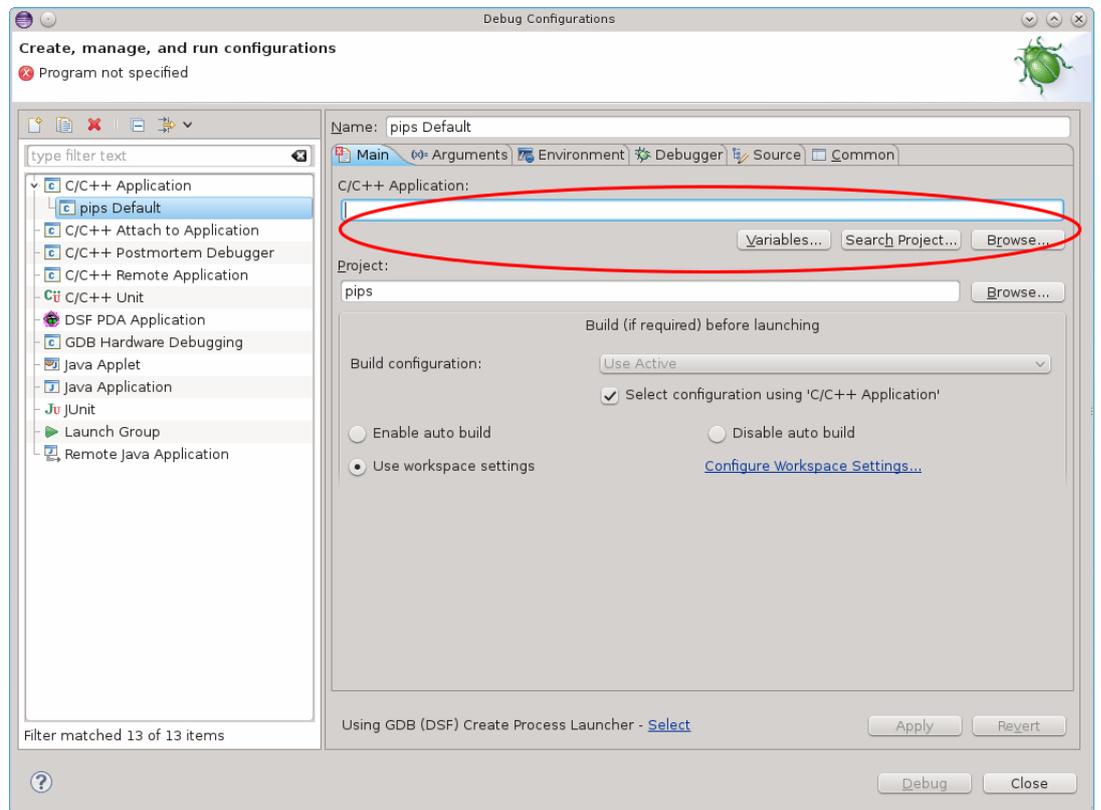
1. Right click on your PIPS project, and choose “*Debug As/Debug Configuration*”. In C/C++ perspective you can use the arrow near the shortcut.



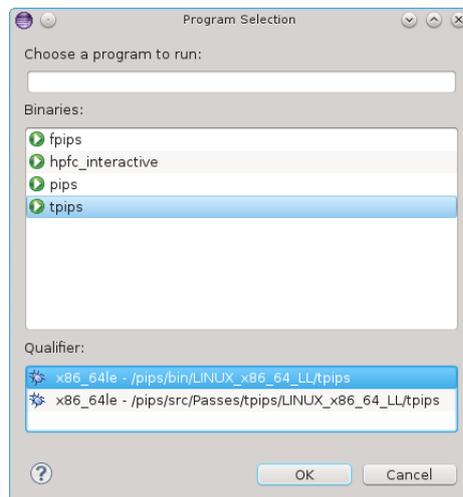
2. Select C/C++ Application and click on “*New*”.



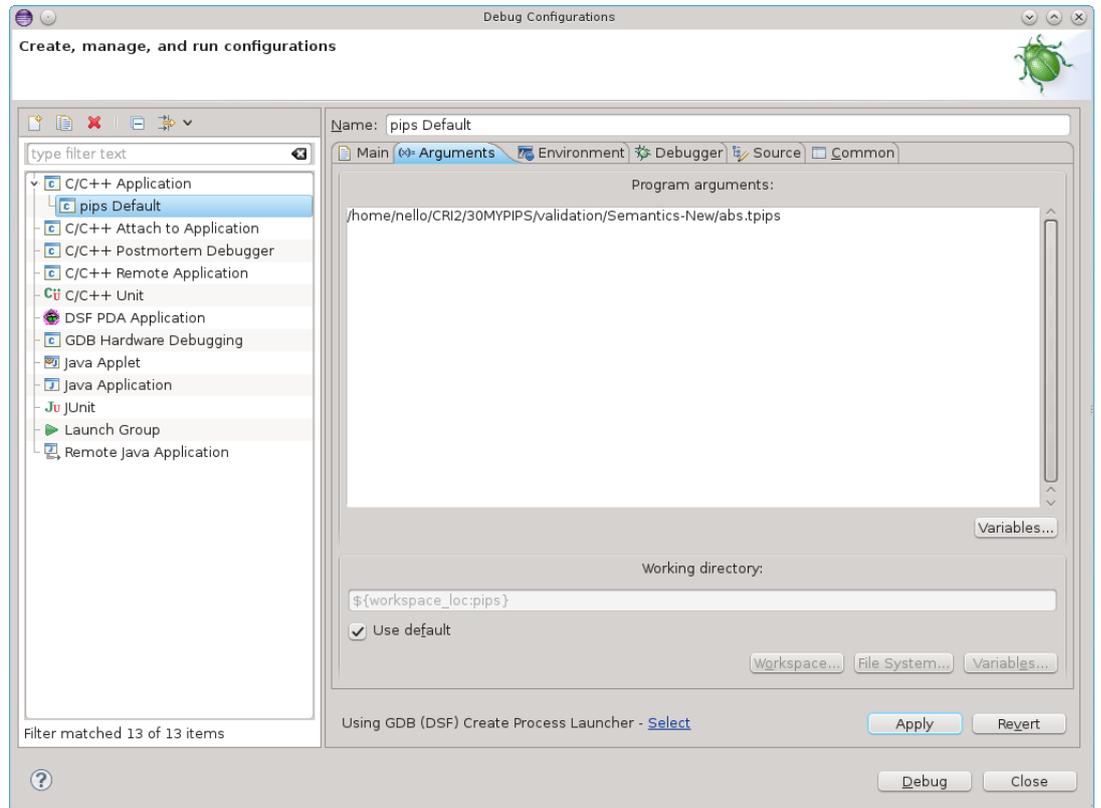
3. Click on “*Search Project*” to define what you want to launch.



4. Select *tpips* and the binary corresponding.



5. Configure the arguments to send your *tpips* file to execute. Without them, you have to write yourself the different instructions.



6. Click on “Apply”, Click on “Debug”
7. Enjoy your debug with Eclipse.

8 Why can I only launch the debug for tpips only once?

You have to configure the PATH environment variable, see step 2 in part 5 Configure your project.

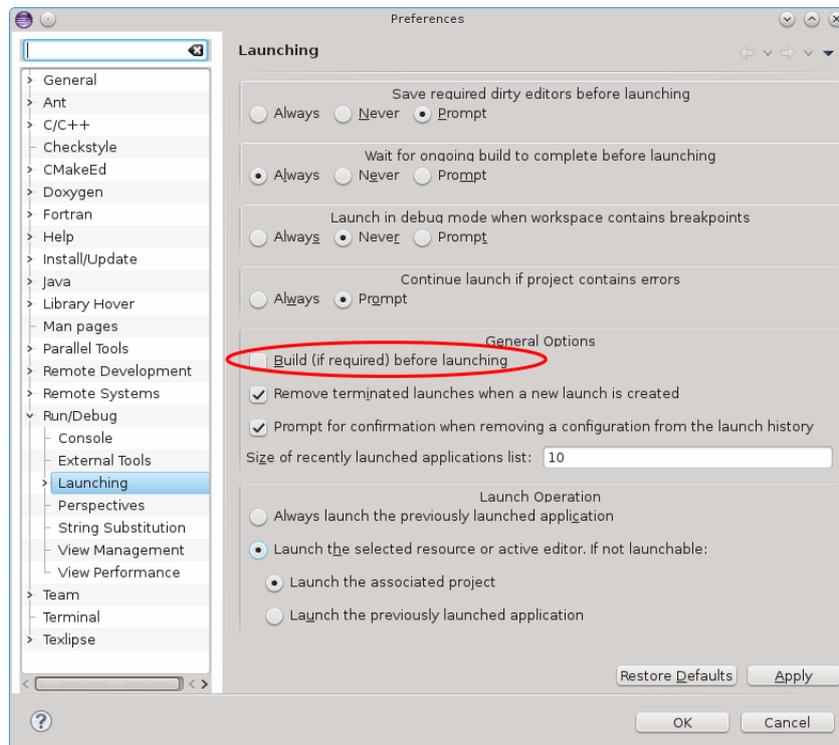
9 Why debug step go randomly top and down?

You have to compile without optimization options. The same thing will happens with gdb if the compilation is not configure.

In the file *makes/gnu-stuff.mk*, put *-O0* instead of *-O2*.

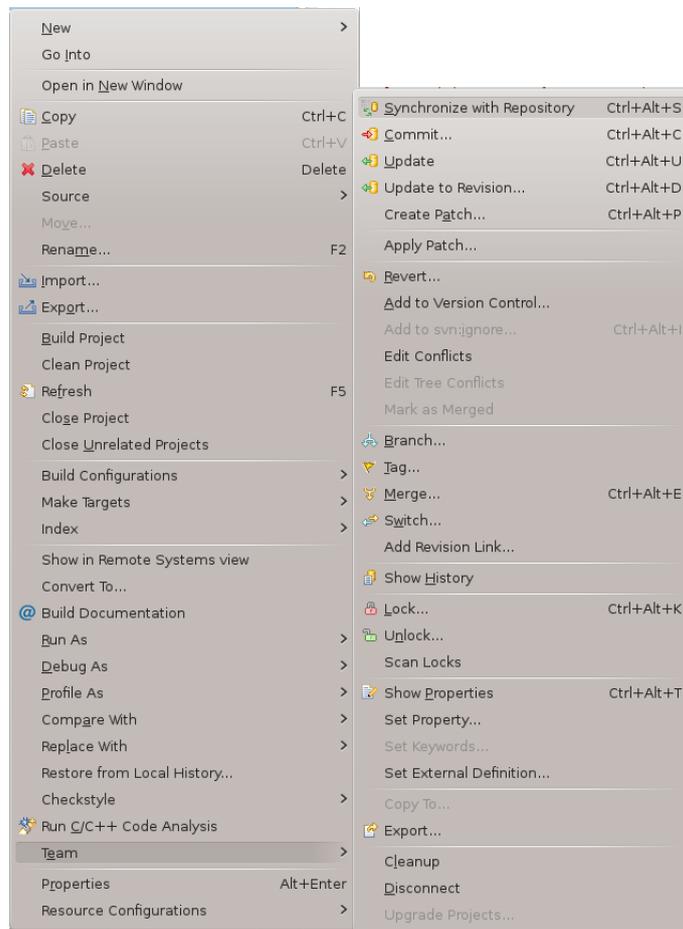
10 How can I ask Eclipse to not recompil before launch debug?

In *Windows/Preferences*, select *Run/Debug* menu, *Launching* submenu. Uncheck *Build (if required) before launching*.



11 How can I use SVN in Eclipse?

You can open the *Team Synchronizing* perspective for this purpose. You can also right click on a folder or file and look the different possibilities of *Team*.



WARNING: Take care to not commit the Eclipse settings or some files coming from compilation especially in the trunk.