

# PyDev: Developing Python code in Eclipse

presented to the Python Interest Group

Pete R. Jemian, BCDA

401/B2100

2011-03-16

ICMS Content ID: **APS\_1419152**



<http://www.rovio.com>

# Abstract



Aptana PyDev (<http://pydev.org>) is a Python IDE for Eclipse which may be used in Python, Jython, and IronPython development. It can be used anywhere that Eclipse runs (including Windows, Mac, Linux). Features include syntax highlighting, code completion, source code debugging with breakpoints and inspection of variables, and refactoring.

Note that Python must be installed separately. It is possible to configure PyDev to use a specific Python if more than one Python is installed.

This presentation will focus on how to install, configure PyDev for routine use including systems with more than one Python installation.



# Topics

- Installation
  - Python
  - Java
  - Eclipse
  - PyDev
- Configuration
  - Setting up Python
- Routine Use
  - Workspace
  - Project
  - Package
  - Module
  - Debugger
- Systems with more than one Python



# Python Distributions

(various distributions add extra packages)

- Python
  - <http://python.org/download/>
- Enthought Python Distribution (Win, Mac, Linux)
  - <http://www.enthought.com/products/epd.php>
- Python X,Y (Win, Linux)
  - <http://www.pythonxy.com/>
- Active State Python, Community Edition (Win, Mac, Linux)
  - <http://www.activestate.com/activepython>

- Linux package installer (Linux)

**Note:**  
Install Python in the  
default location unless  
you know better



# Eclipse needs Java: <http://java.oracle.com> click on “Java SE”

Oracle Technology Network for Java Developers - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.oracle.com/technetwork/java/index.html

Most Visited BBC Bens fm /. TV TV ANL jobs APS GUP APS-U BCDA canSAS NeXus jemian SVN Safari hostauth GazeboNews

Oracle Technology Network for Java ...

ORACLE

Welcome Pete (Account | Help | Sign Out) United States Communities I am a... I want to... Secure Search

Products and Services Downloads Store Support Education Partners About Oracle Technology Network

Oracle Technology Network > Java

Java

RELEASE

**Java SE 7 Developer Preview Release Now Available**

The JDK7 Developer Preview Release (aka Milestone 12, aka Build b130) is now available. This build is feature complete, stable, and ready for vigorous community testing. Please download this build, test it and report bugs now.

Posted 2/24/11 // Tags: java, jdk7, release // [Headlines Archive](#)

Software Downloads [View All Downloads](#)

Top Downloads

- Java SE**
- Java EE and GlassFish
- Java FX
- Java ME
- JDeveloper 11g and ADF
- Enterprise Pack for Eclipse
- NetBeans IDE
- Pre-Built VM for Java Developers

Get Java

New Downloads

- GlassFish Server 3.1  
Released 2/28/11
- Java SE 7 Developer Preview  
Released 2/22/11
- NetBeans IDE 7.0 Beta 2  
Released 2/17/11
- Java SE 6 Update 24  
Released 2/15/11
- LWUIT 1.4  
Released 8/9/10

# Select the Java Standard Edition (Java SE), Development Kit (JDK)

Overview Downloads Documentation Community Technologies Training

## Java SE Downloads

Latest Release Next Release (Early Access) Embedded Use Real-Time Previous Releases

Download Download Download Download

Java Platform (JDK) JDK - JavaFX Bundle JDK - NetBeans Bundle JDK - Java EE Bundle

Here are the Java SE downloads in detail.

Java Platform, Standard Edition	
<b>Java SE 6 Update 24</b> This release includes security enhancements and bug fixes. <a href="#">Learn more</a>	
<a href="#">Download JDK</a>	<a href="#">Download JRE</a>
<b>JDK 6 Docs</b> <a href="#">Installation Instructions</a> <a href="#">ReadMe</a>	<b>JRE 6 Docs</b> <a href="#">Installation Instructions</a> <a href="#">ReadMe</a>

## Java SE Development Kit 6u24

Provide Information, then Continue to Download

Select Platform and Language for your download:

Platform: Windows x64  
Language: Multi-language

I agree to the [Java SE Development Kit 6u24 License Agreement](#).

[Continue »](#)

# Start the download, then install it in the default place

## Download Java SE Development Kit 6u24 for Windows x64, Multi-language

Download Information and Files

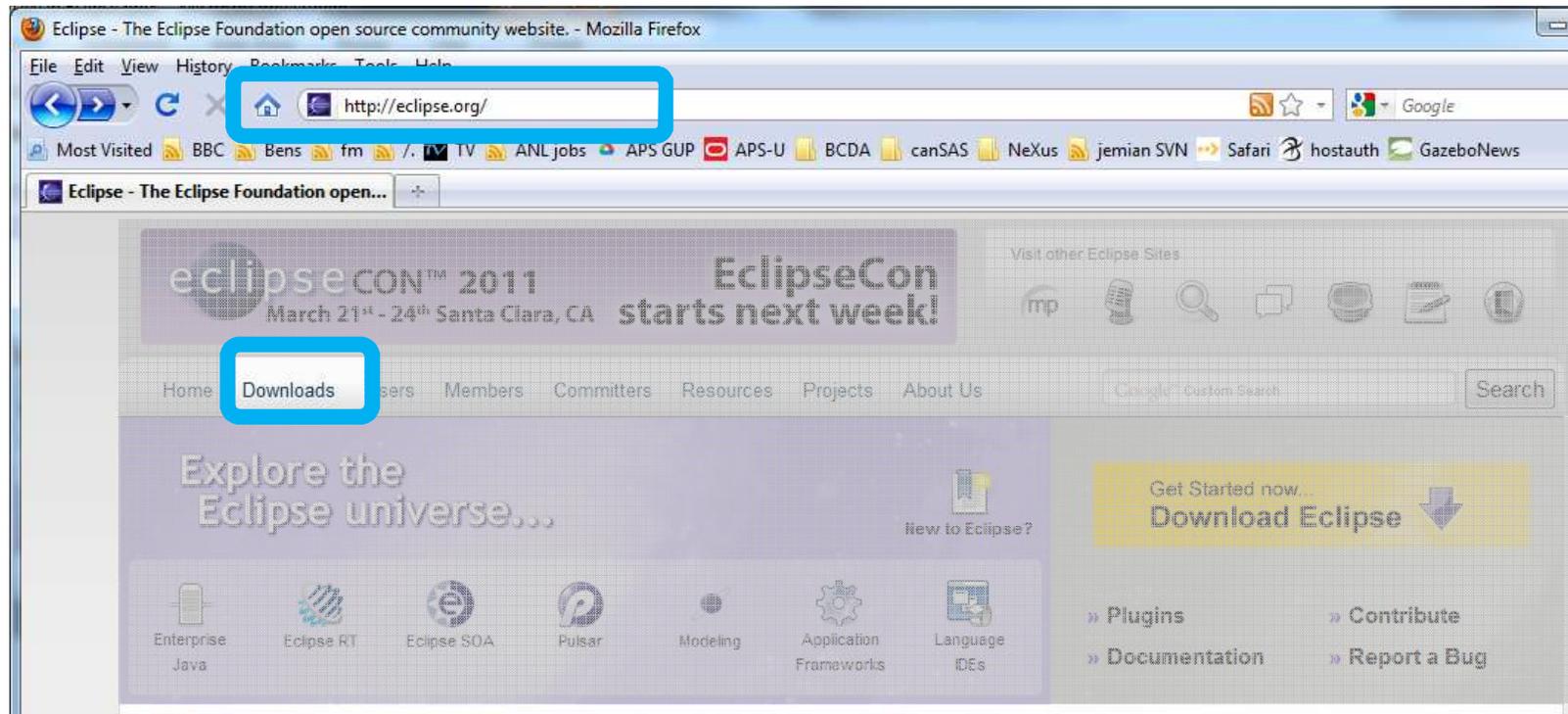
Instructions: Click the file name to start the download.

Available Files

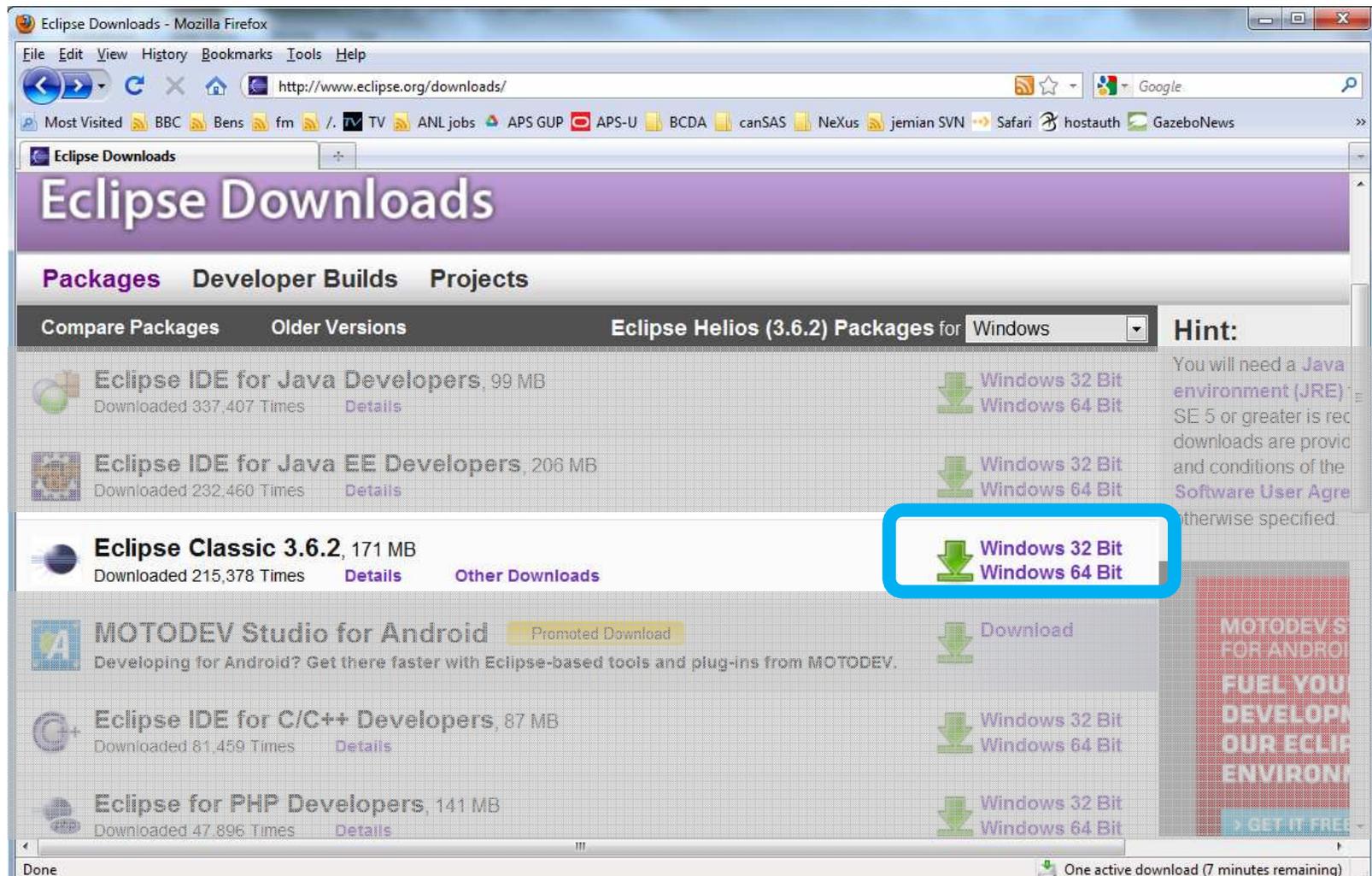
File Description and Name	Size
Java SE Development Kit 6u24 <a href="#">jdk-6u24-windows-x64.exe</a>	66.85 MB

Notes:

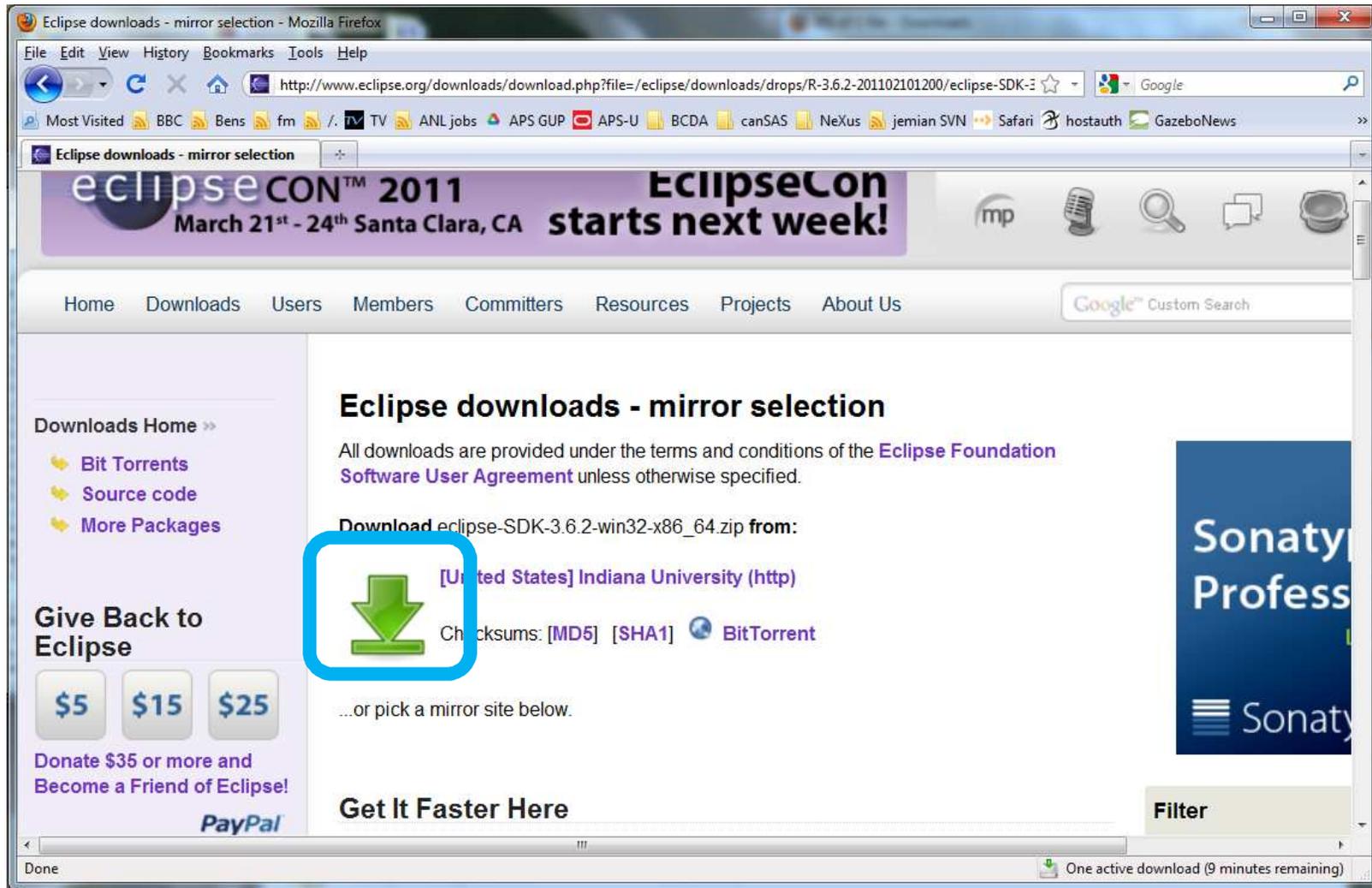
# Downloading Eclipse: <http://eclipse.org> Click on “Downloads”



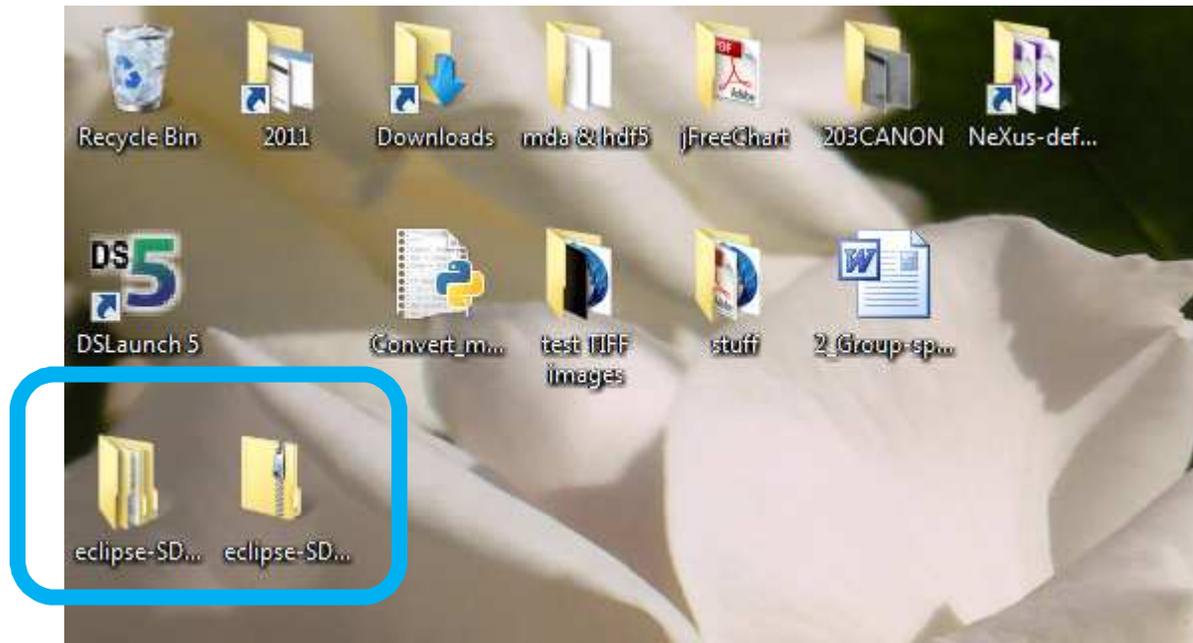
Pick a version: “Classic” provides many tools.  
You can always edit/remove other parts later



# Downloads from a mirror



# Unzip the eclipse download *anywhere that is convenient*

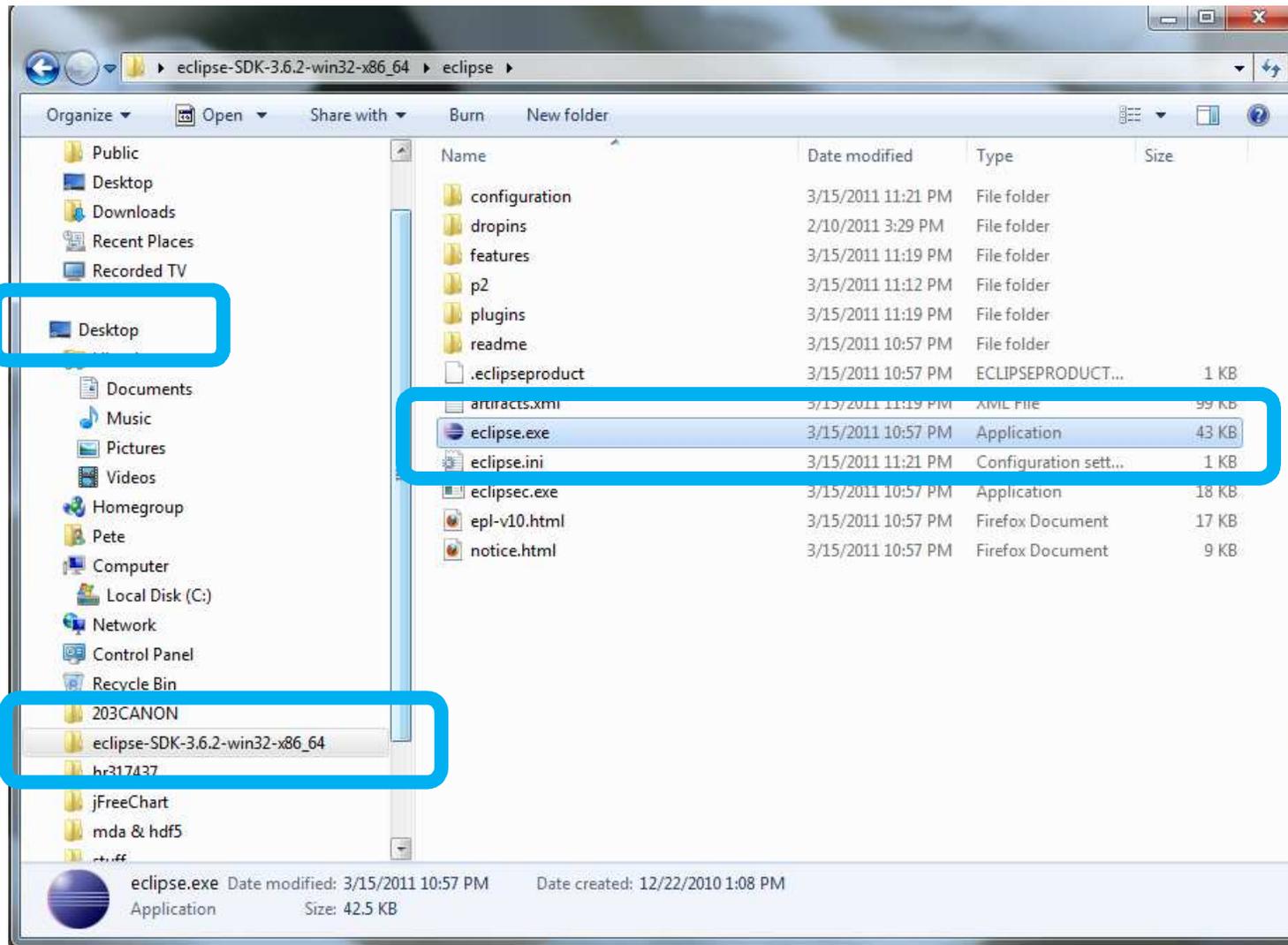


## Note:

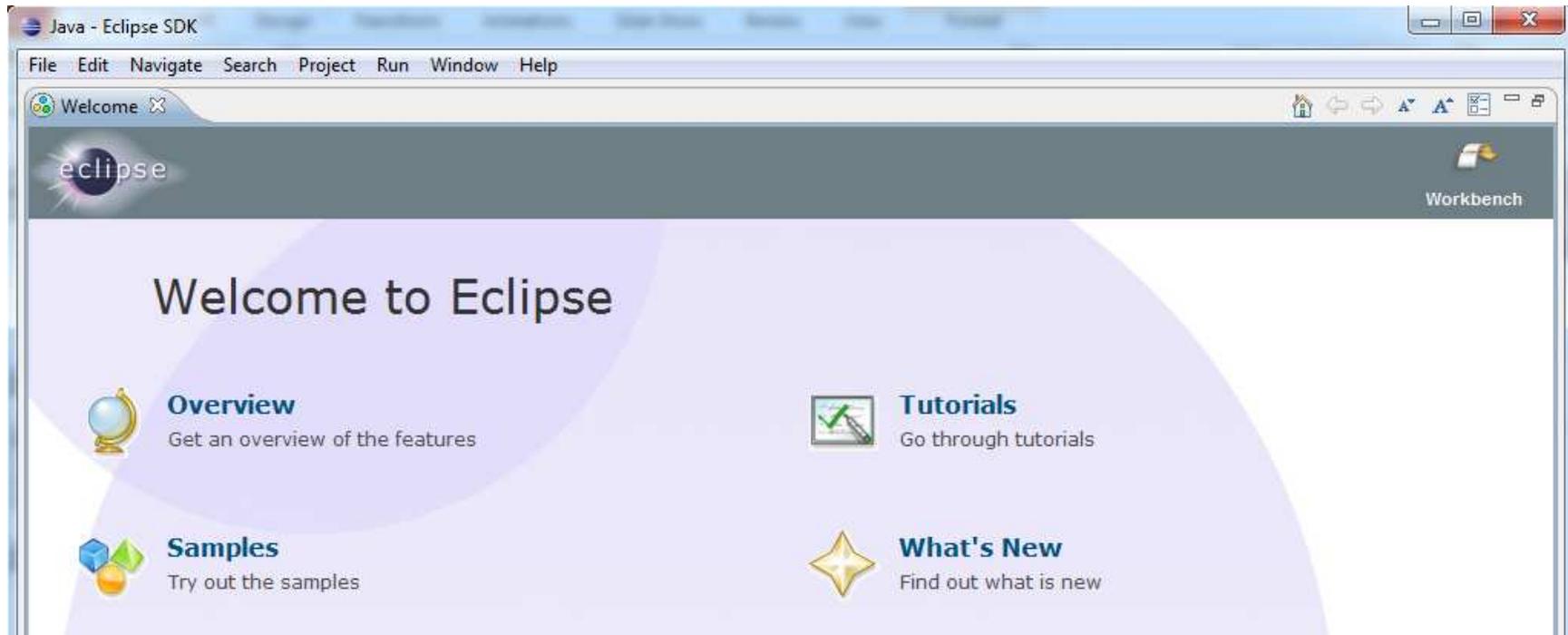
Here, the ZIP file is on the desktop and extracted to the desktop

# start the *eclipse* executable

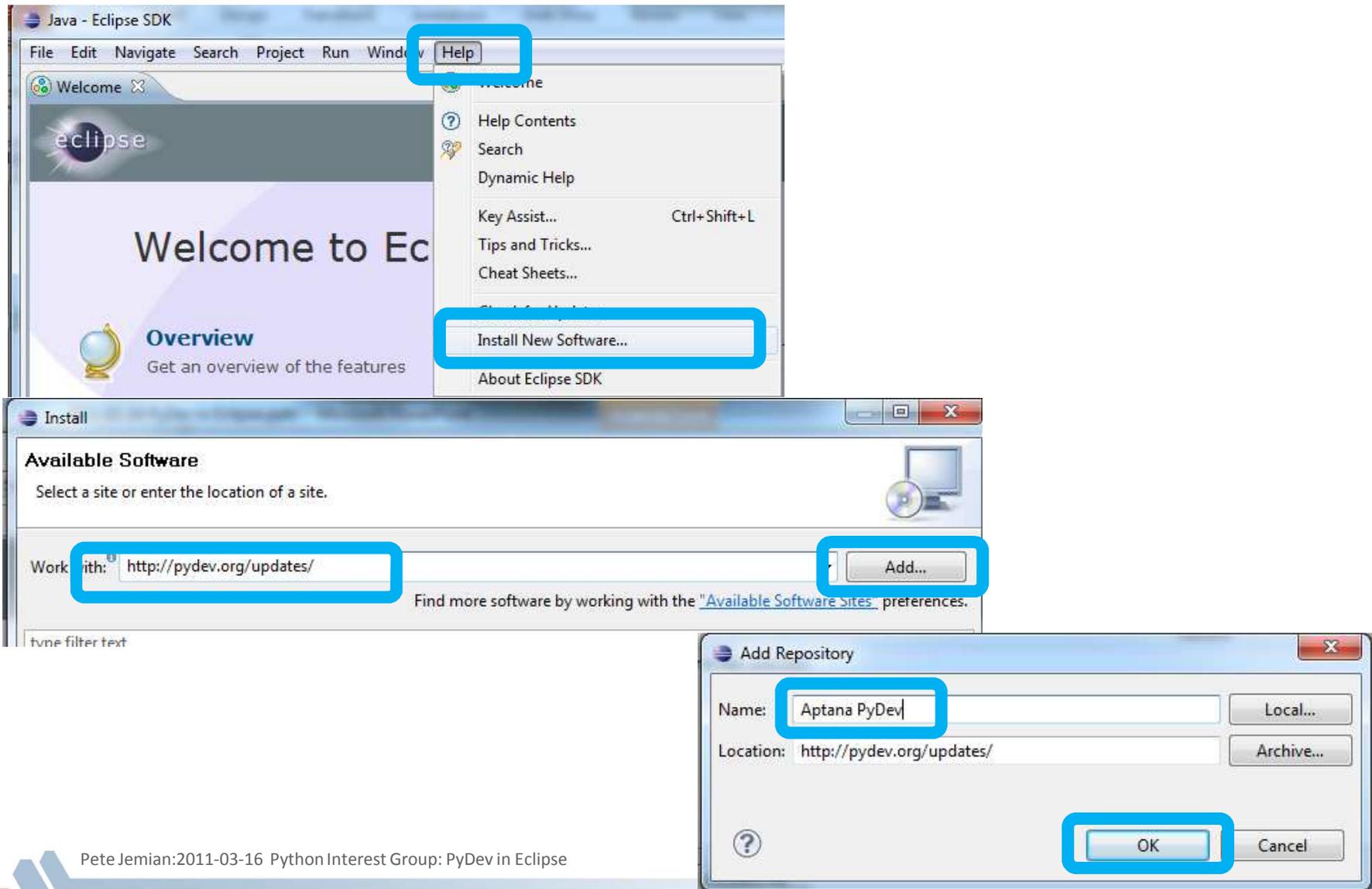
(look in <extracted\_folder>/eclipse/)



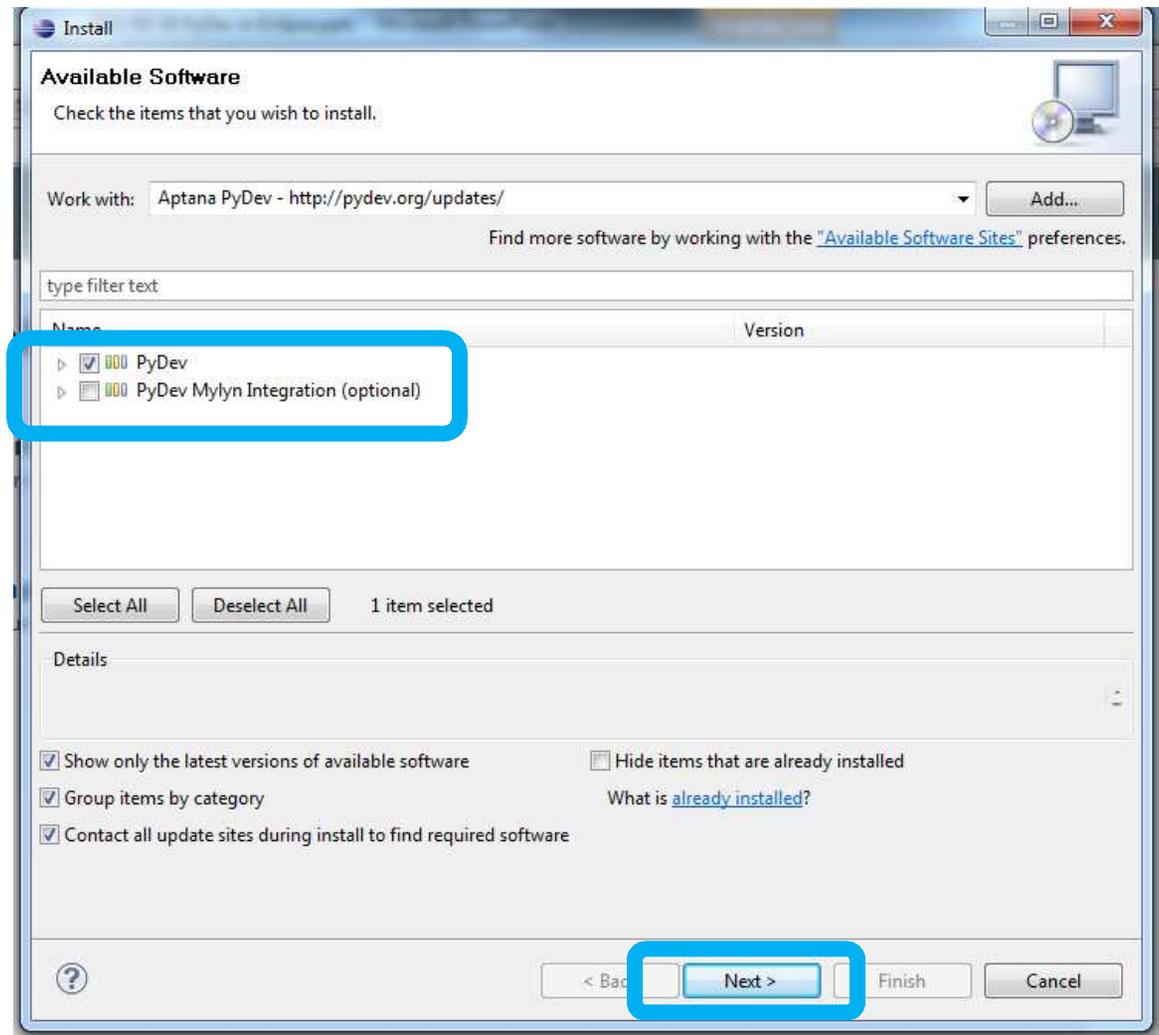
# Welcome to eclipse



# Installing PyDev: <http://pydev.org/updates/> (this URL is an eclipse update site - use eclipse to get it)



# Select PyDev



## Note:

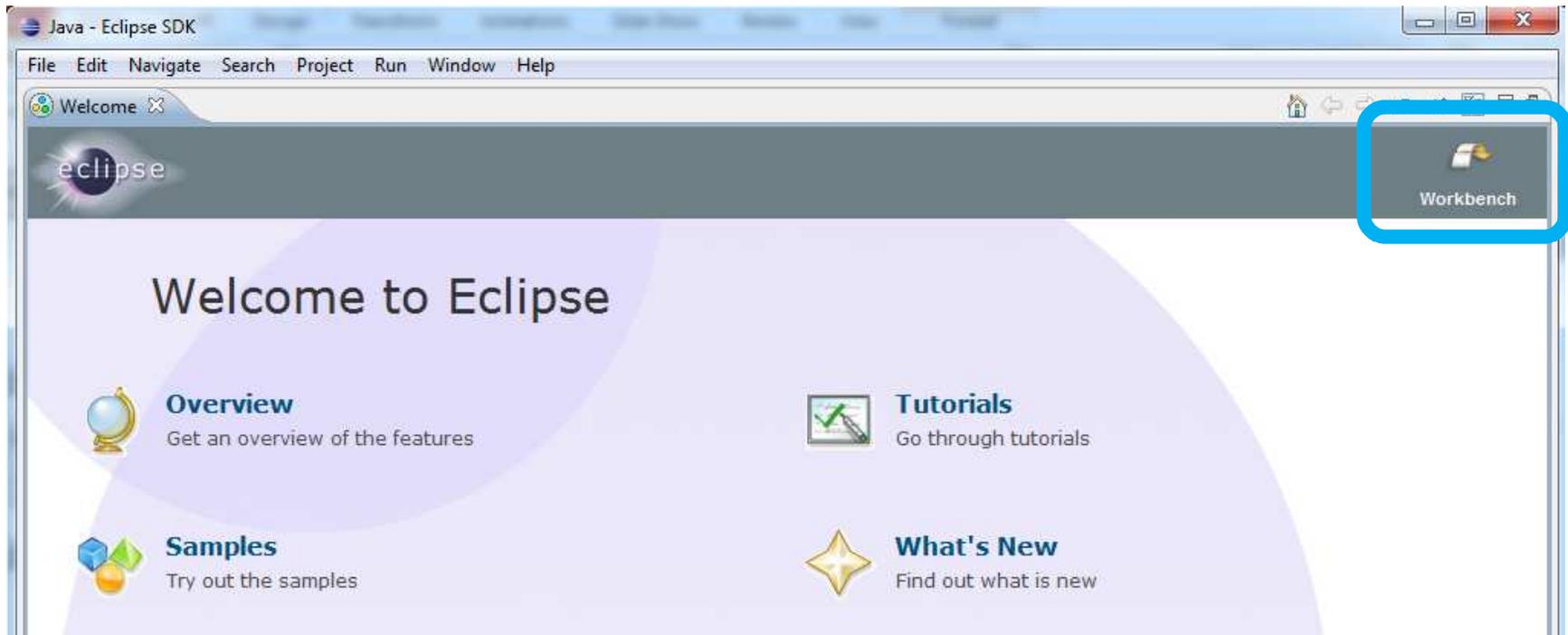
click [Next] and click through, including License Agreement and Certificate,  
Then, restart eclipse as requested



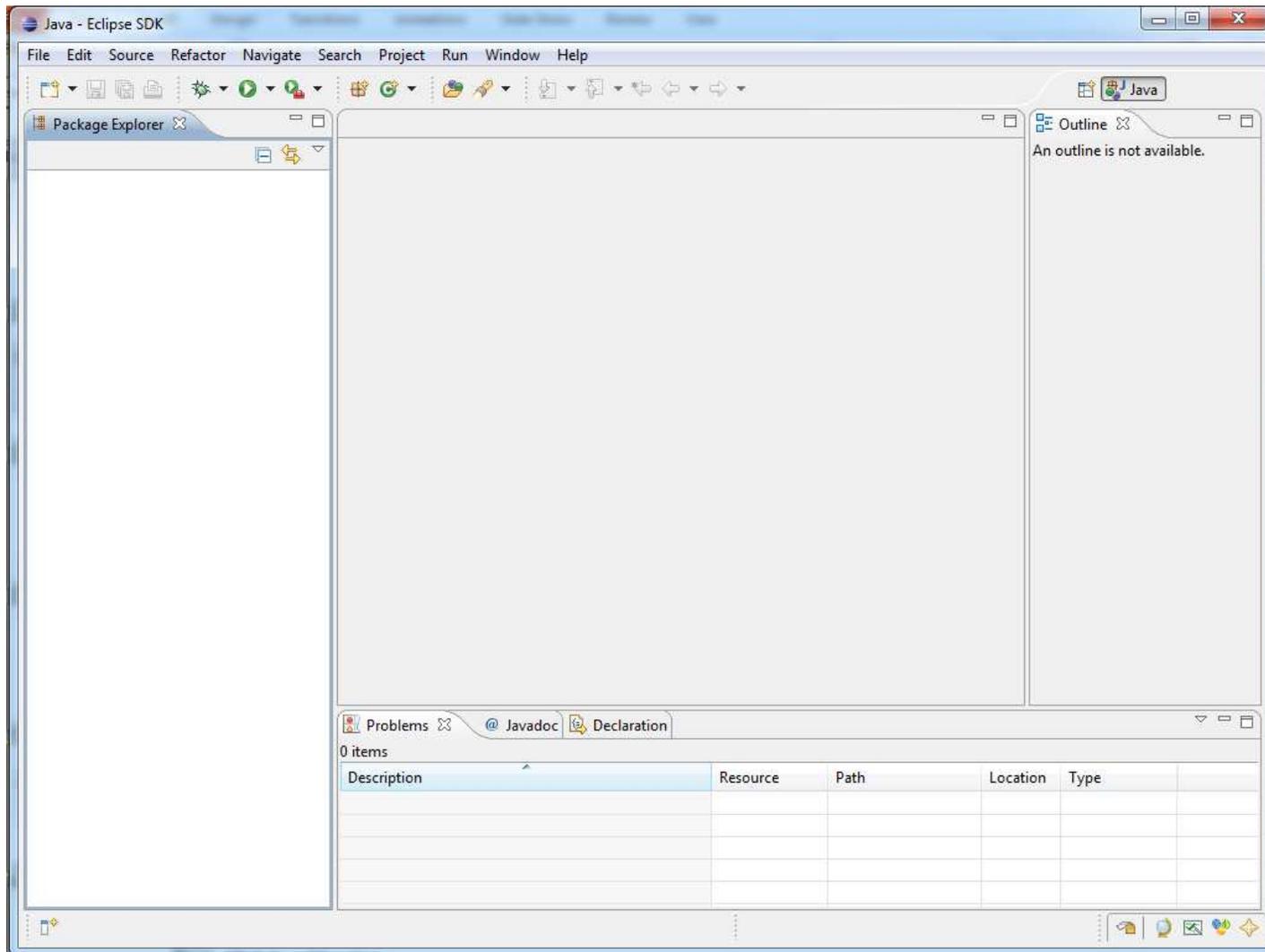
# Installation is complete, take a break



# Configure PyDev for the Python Interpreter

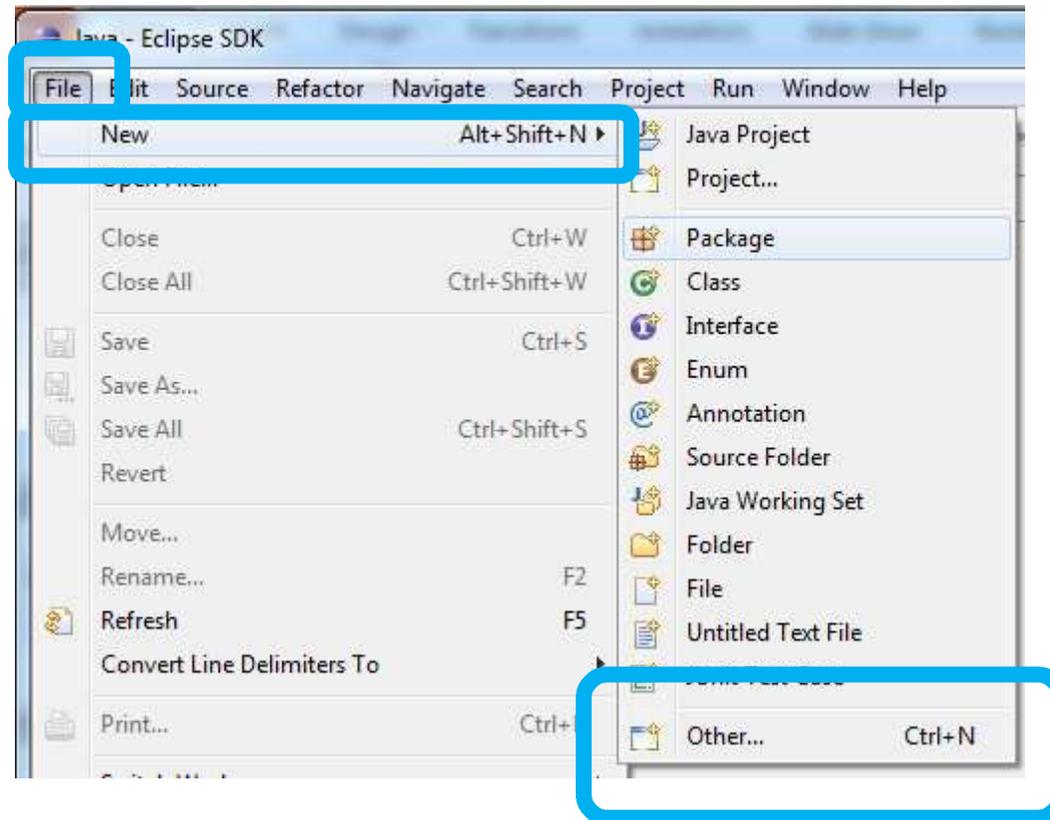


# Arrive at the empty workbench (progressing to configure PyDev for Python interpreter ...)



# New Python Project

(... still progressing to configure PyDev for Python interpreter ...)



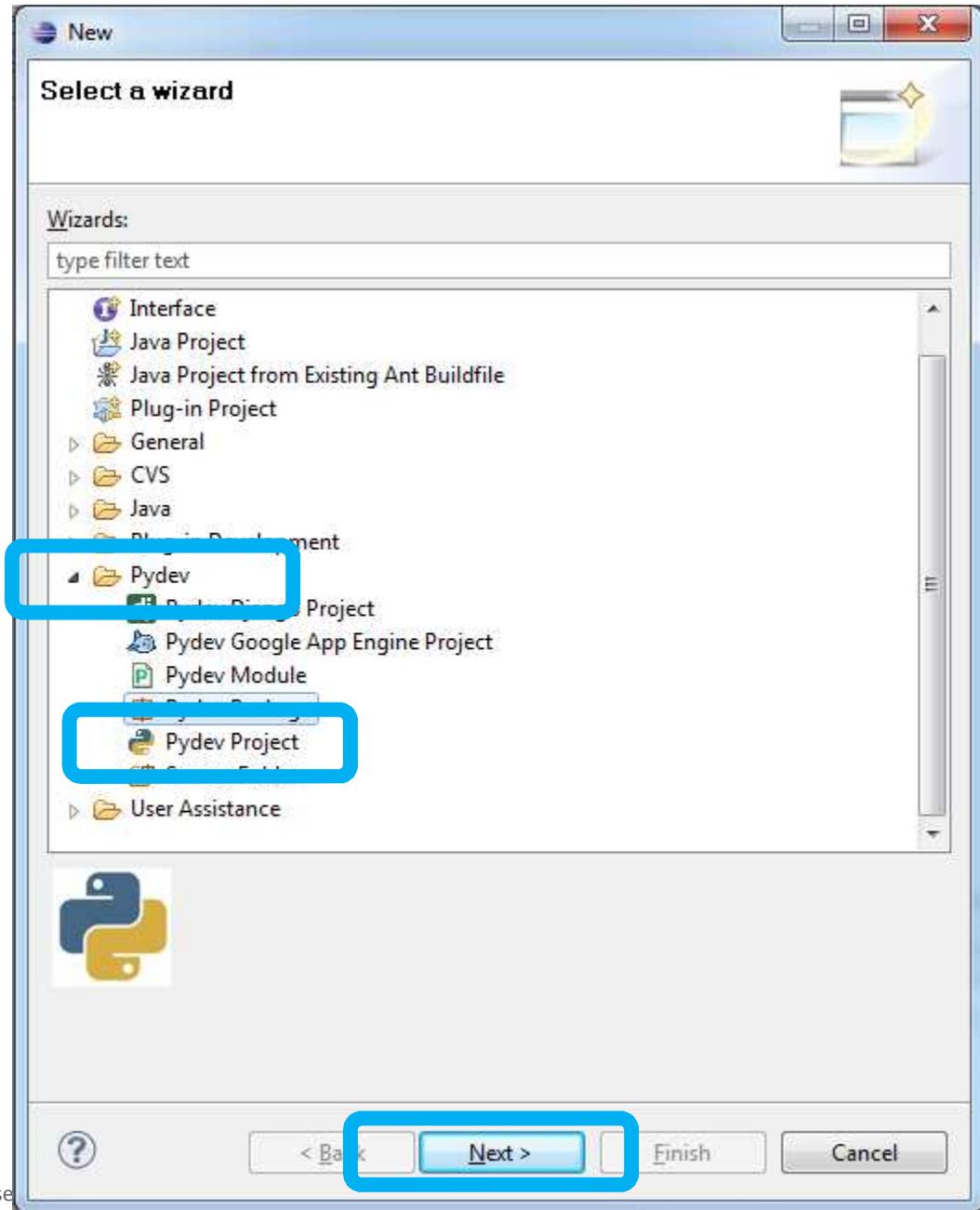
## Note:

There are other ways to get to the Properties dialog to configure PyDev for the Python interpreter. This one seems indirect at first but is very short.

Expand this entry →

Click this next →

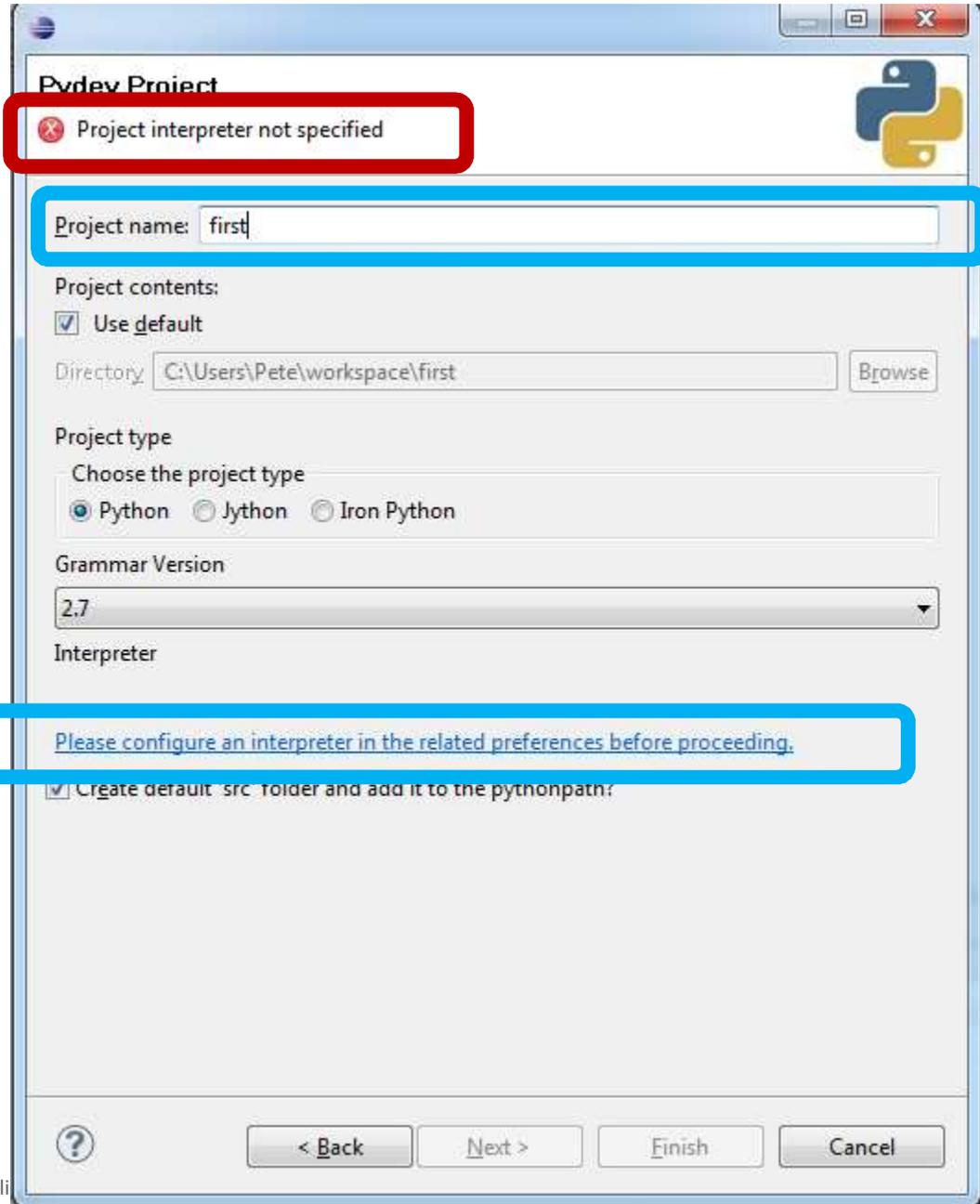
Click [Next >] →



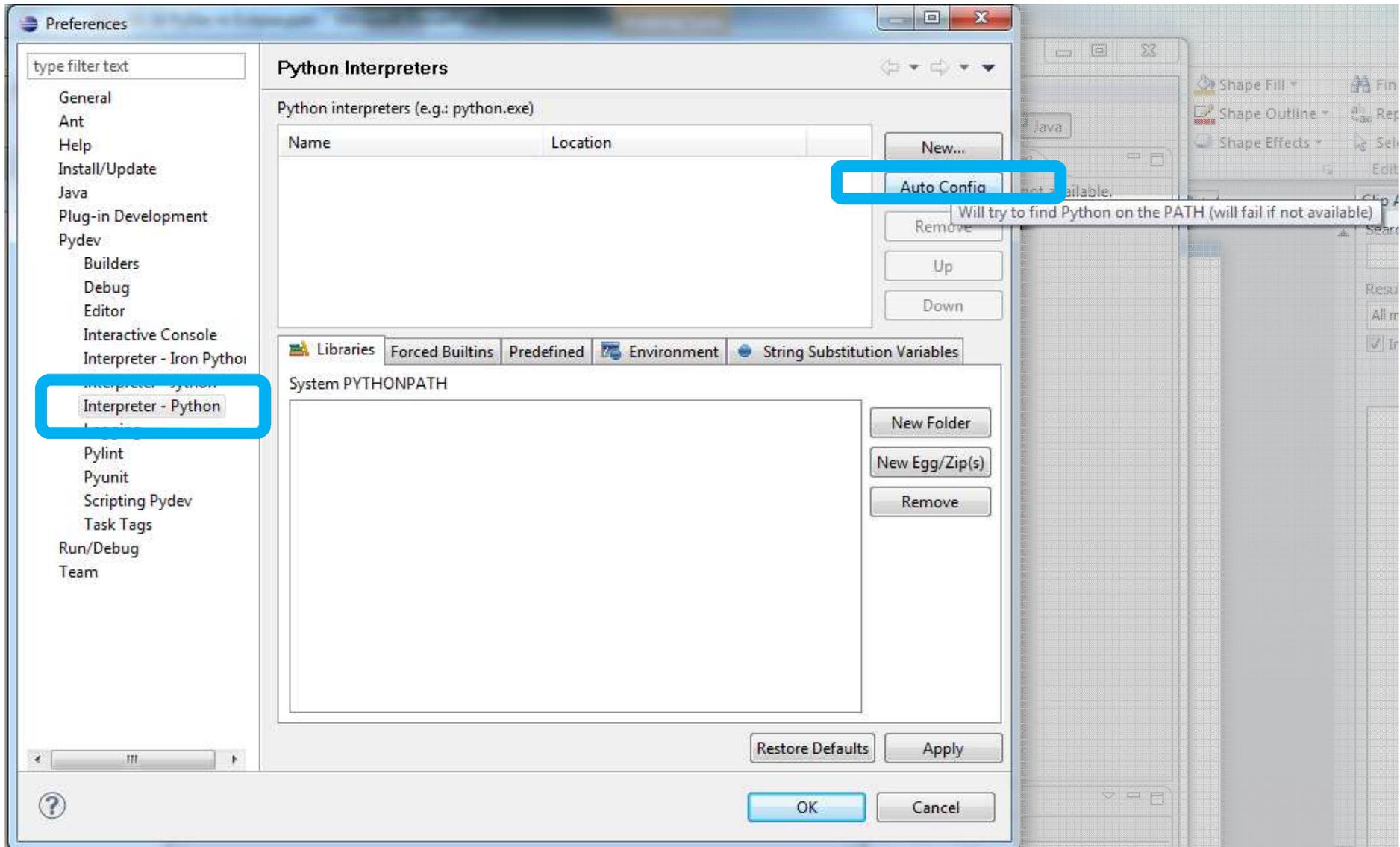
# Make Project & Configure

Give some name ("test") →

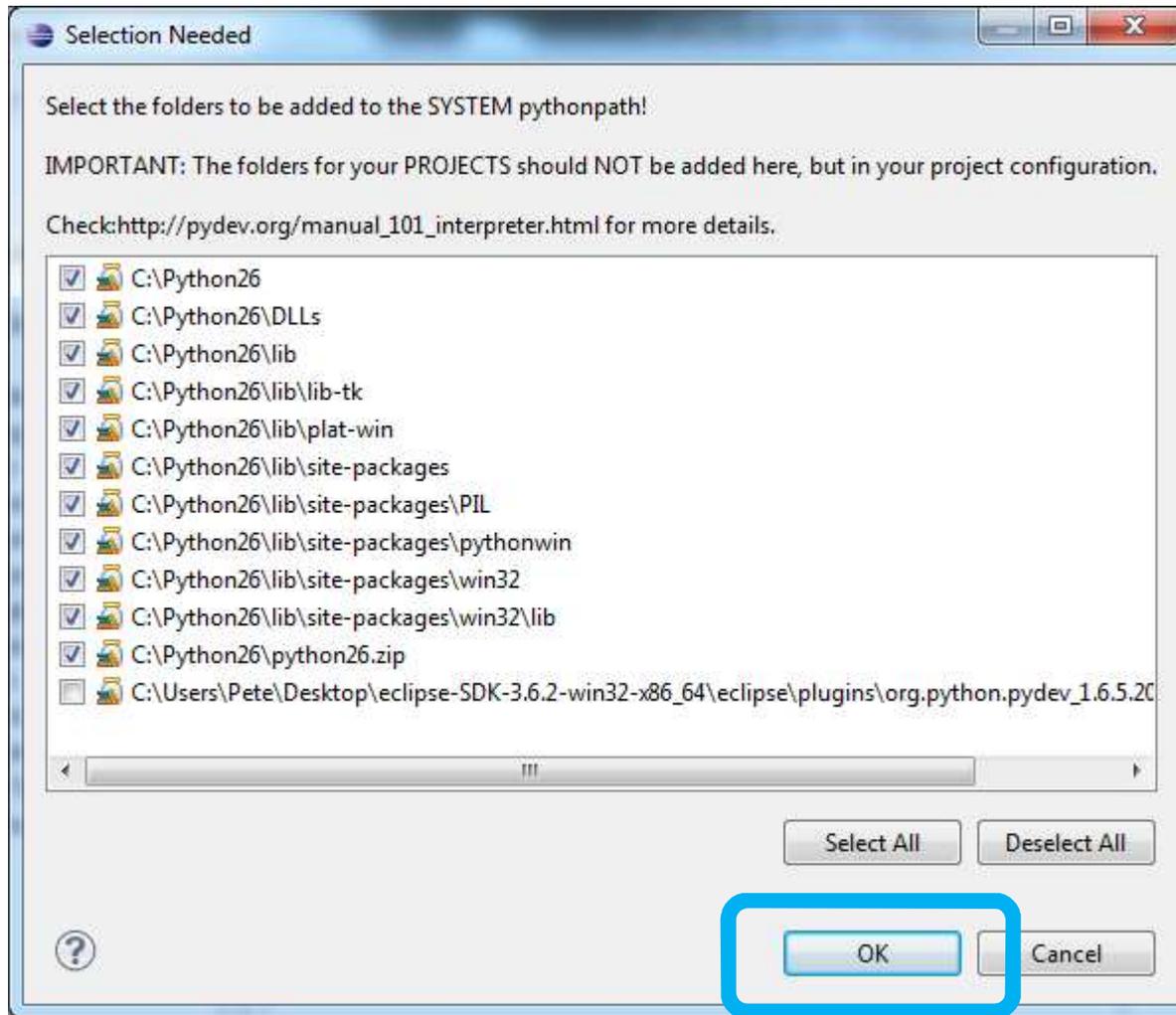
Click here →



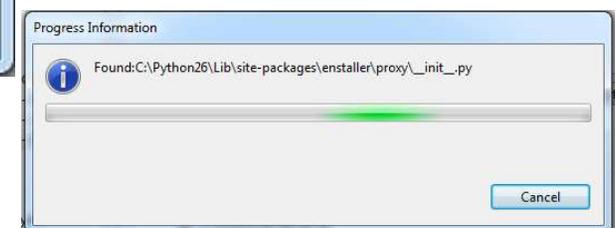
# Run “Auto Config”



# “Auto Config” found this ...



Click [OK] on next screen, also, then this step takes a few minutes ...



# Finish making the project

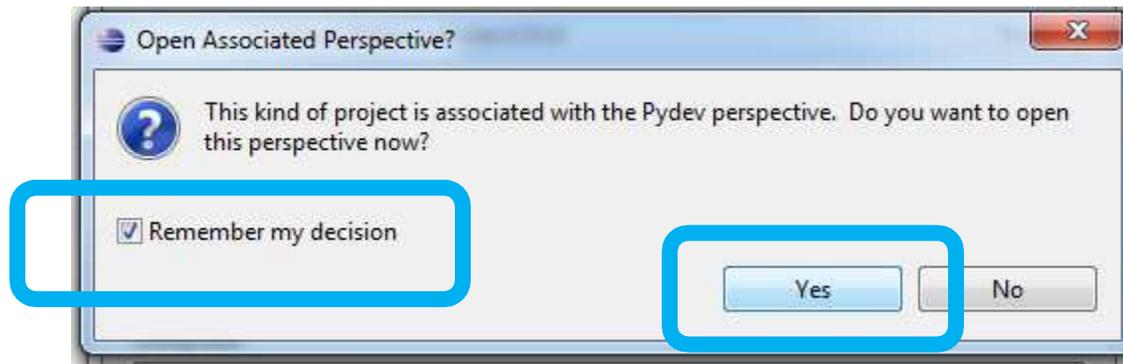
Make sure the “Grammar Version” your installed Python

The screenshot shows the 'Pydev Project' dialog box in Eclipse IDE. The dialog is titled 'Pydev Project' and contains the following fields and options:

- Project name:** first
- Project contents:**  Use default
- Directory:** C:\Users\Pete\workspace\first (with a 'Browse' button)
- Project type:** Choose the project type:  Python,  Jython,  Iron Python
- Grammar Version:** 2.6 (highlighted with a blue box)
- Interpreter:** Default (highlighted with a blue box)
- Create default 'src' folder and add it to the pythonpath?

At the bottom of the dialog, there are four buttons: '?', '< Back', 'Next >', and 'Finish' (highlighted with a blue box). The 'Cancel' button is also present.

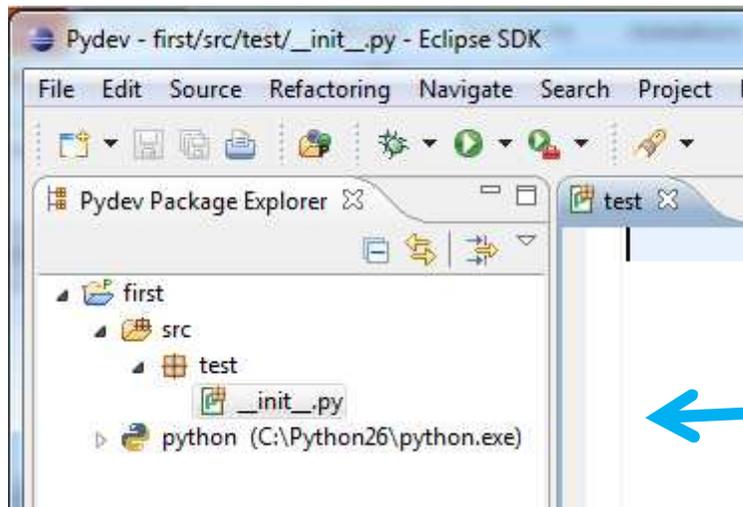
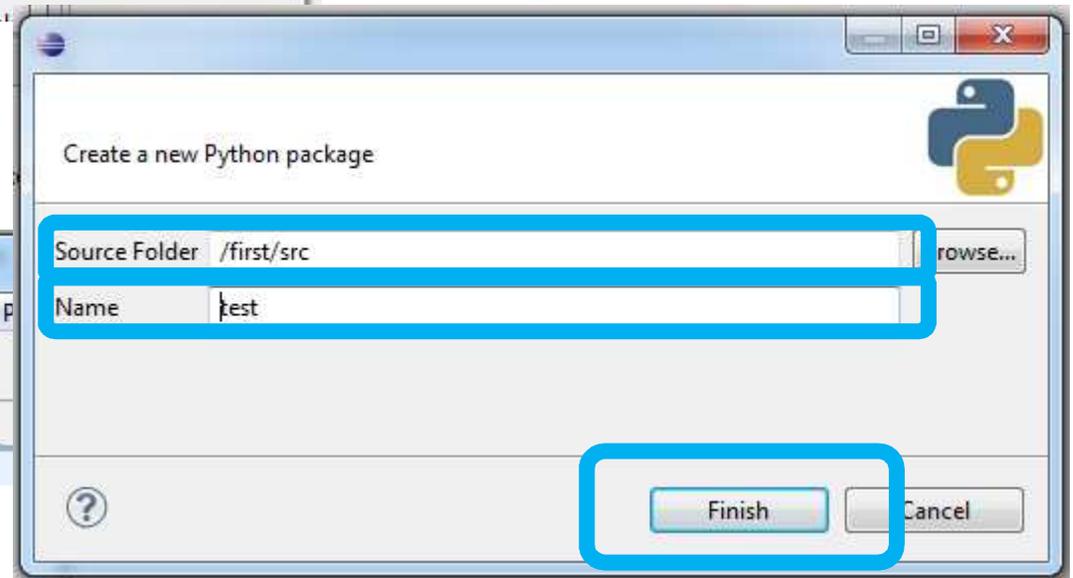
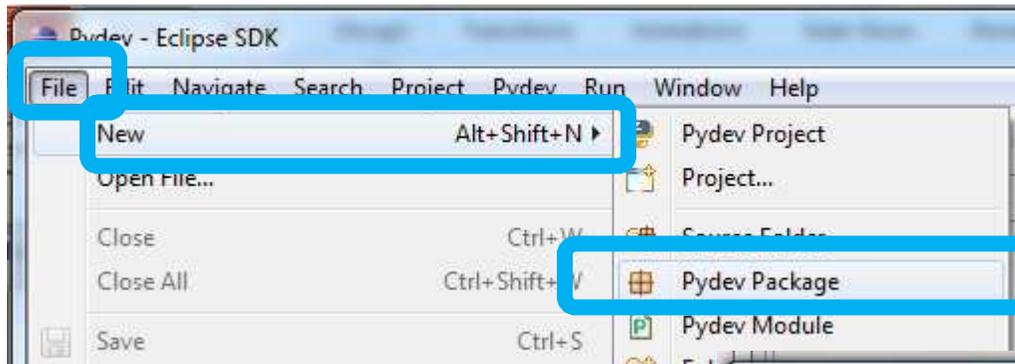
# Switch to the “Python perspective”



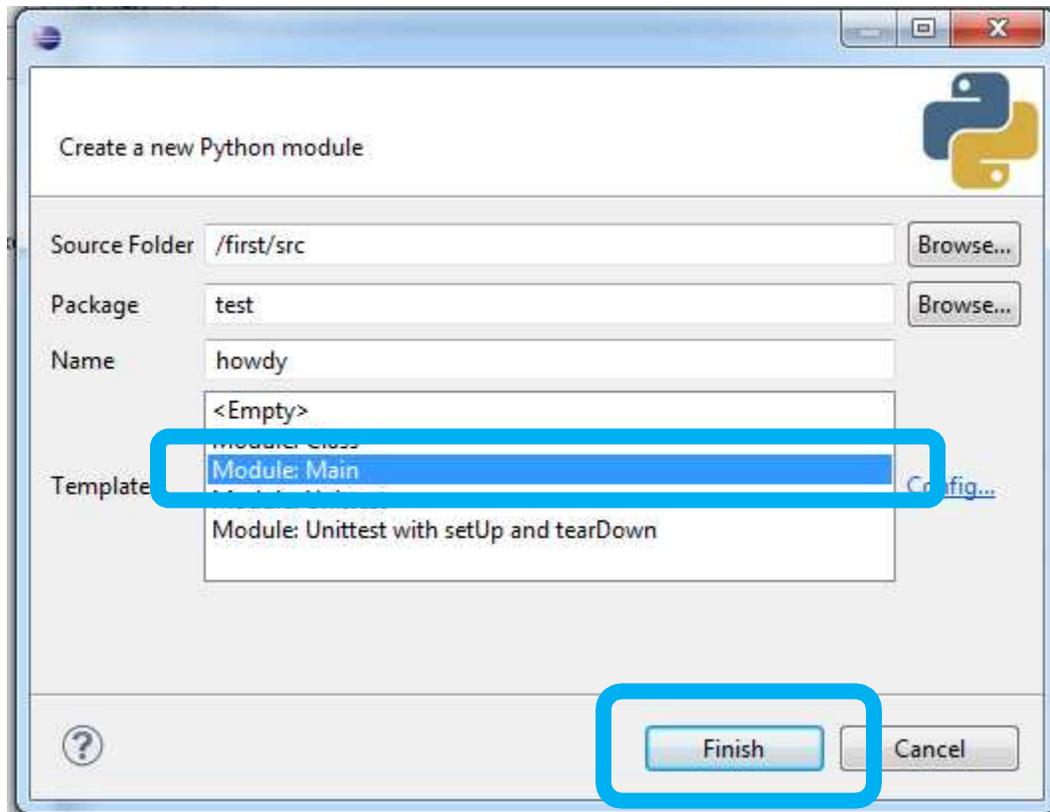
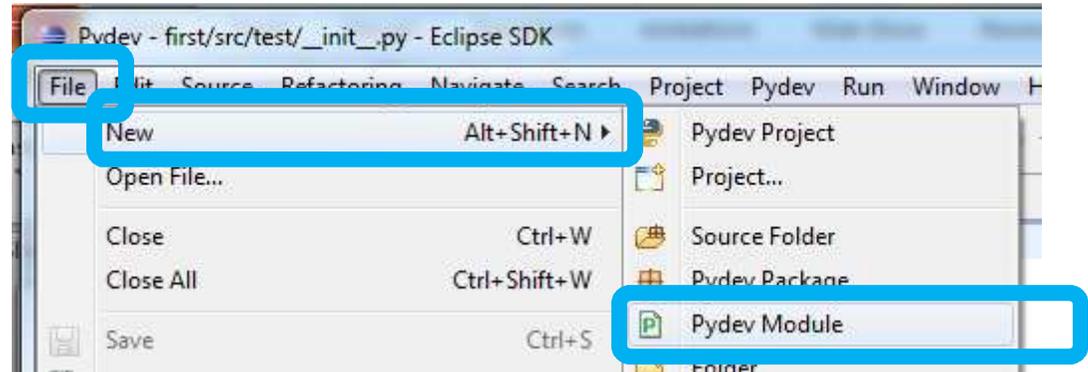
The “Python perspective” provides all the PyDev tools for developing Python code.

# Make a Package ...

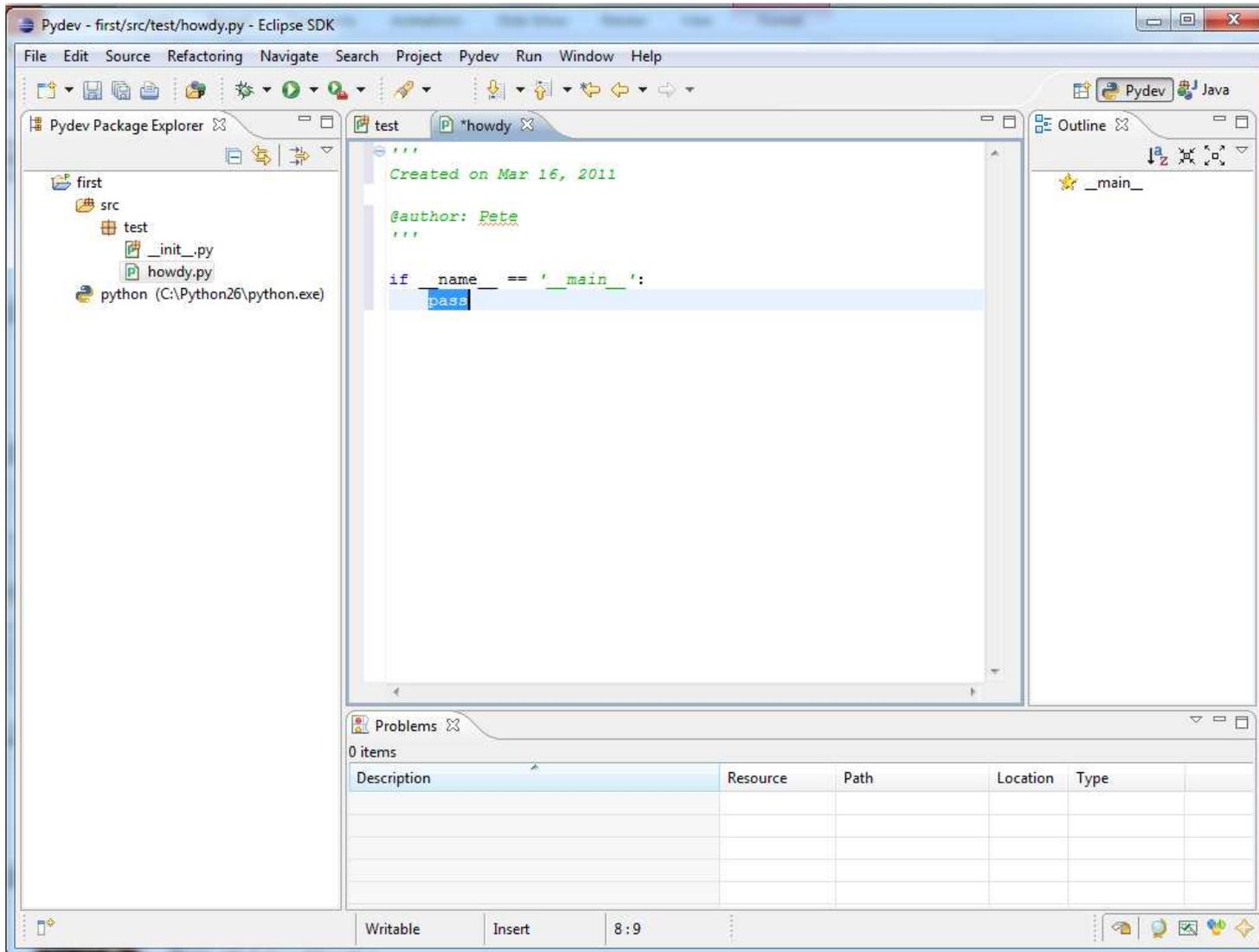
(be sure to select `src` first)



# Make a Module



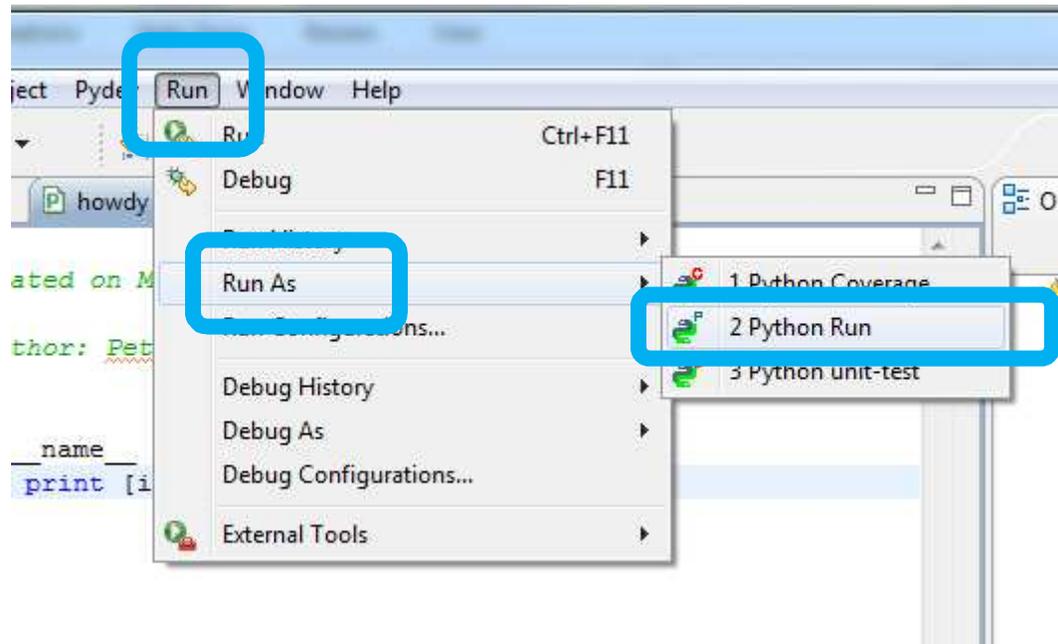
# Your module in the PyDev eclipse editor



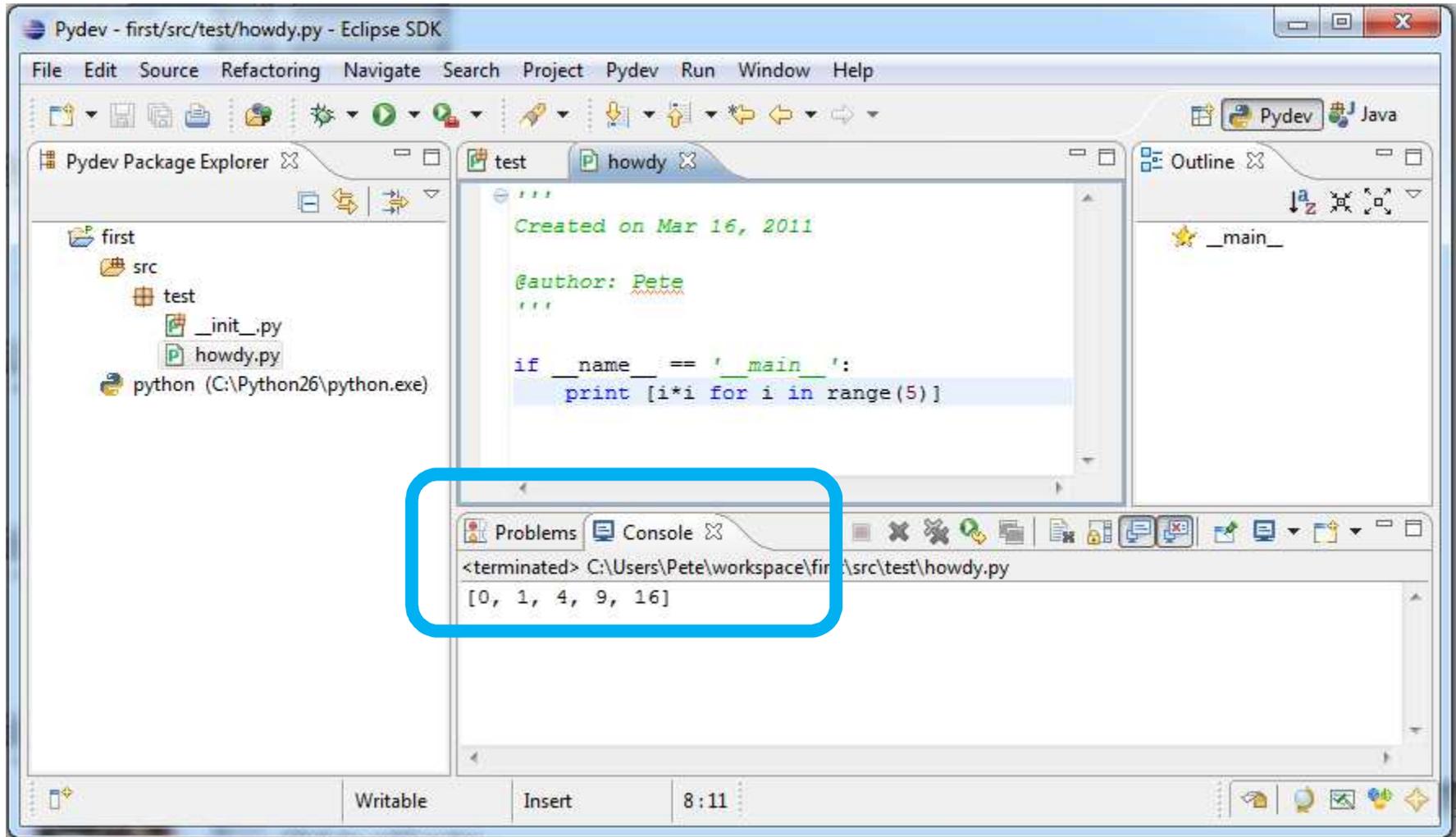
## Add some test code

```
'''  
Created on Mar 16, 2011  
  
@author: Pete  
'''  
  
if __name__ == '__main__':  
    print [i*i for i in range(5)]
```

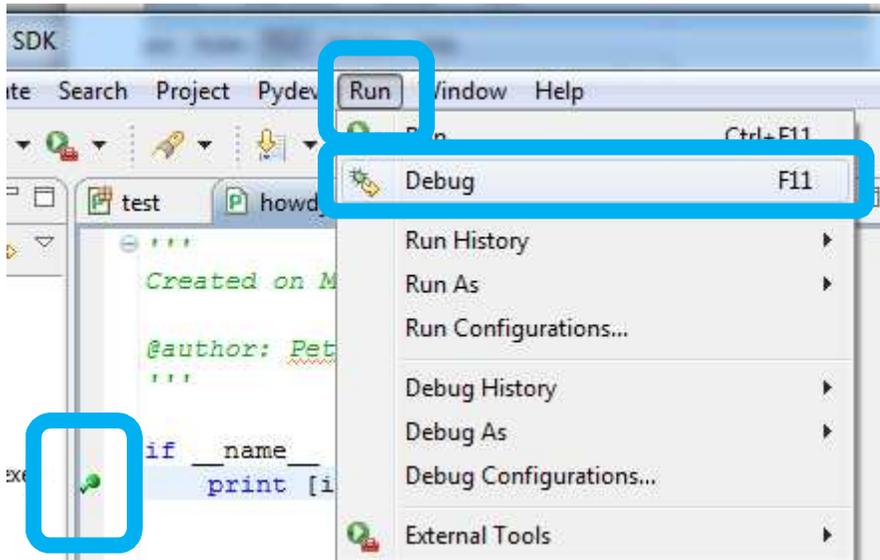
# Select this once for each .py to be run



# Results



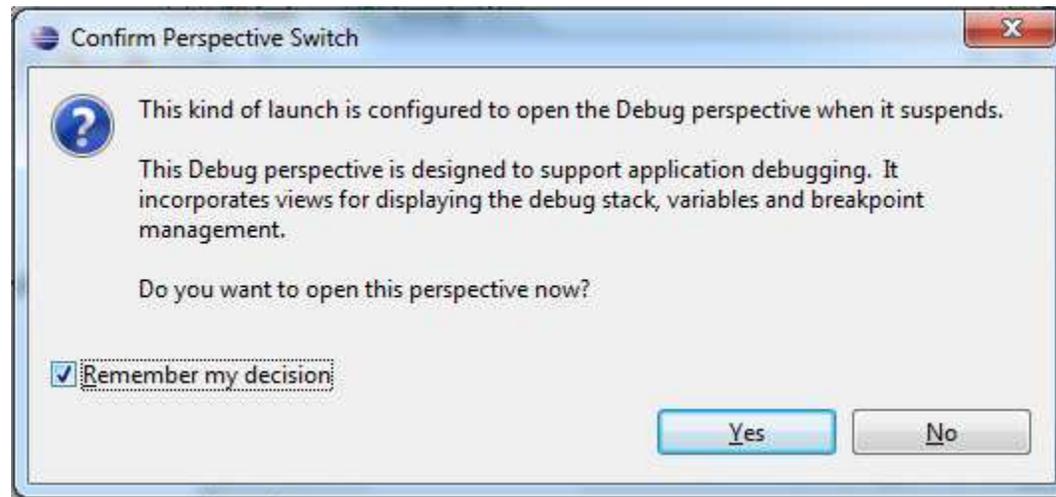
# Source Code Debugger



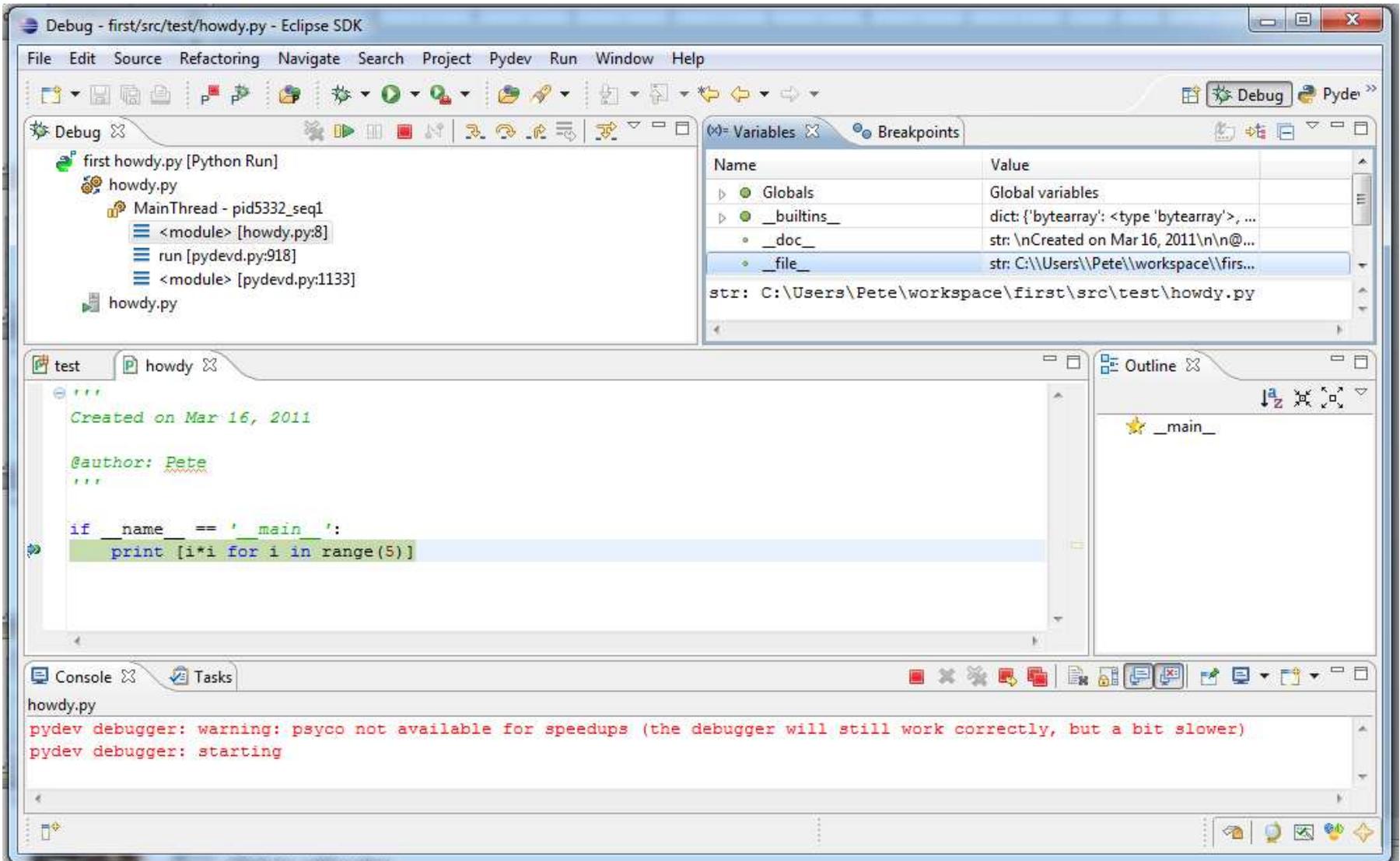
**Note:**  
*Set at least one breakpoint or the debugger will not stop*



# Debugging happens in the “Debug perspective”



# Paused at first breakpoint



# Example of debugging another project ...

The screenshot displays the Eclipse IDE interface during a Python debugging session. The main editor window shows the source code of `EpicsDatabase.py`, with a breakpoint set at line 135 in the `parse` method. The code is as follows:

```
result[k][field] = utilities.replaceMacros( v, macros )
return result

def _parse(self):
    """
    Interpret the contents of the .db file
    into the internal memory structure.
    """
    pvDict = {}
    tkn = self.tokenLog.nextActionable()
    while tkn is not None:
        if tkn['tokName'] == 'NAME' and tkn['tokStr'] in ('record', 'grecord'):
            # start of record declaration
            tkn = self.tokenLog.nextActionable() # token with "(" character
            rtyp, name, tkn = self.getTwoItems(tkn)
            fieldDict = {
                'RTYP': rtyp, # record type
                'NAME': name # record name
            }
            tkn = self.tokenLog.nextActionable() # load the next token
            while tkn['tokStr'] != "):":
                if tkn['tokName'] == 'NAME' and tkn['tokStr'] in ('field'):
                    tkn = self.tokenLog.nextActionable() # "(" character
                    s = tkn['tokLine'].strip()[len('field'):]
                    args = utilities.strip_outer_pair(s, '(', ')').split(",")
                    print len(args), args
                    # TODO need to change the classDB base
```

The Variables window on the right shows the state of the `tkn` variable, which is a dictionary:

Name	Value
Globals	Global variables
pvDict	dict: {}
self	Db: <_main__Db instance at 0x024CEF80>
tkn	dict: {'tokName': 'OP', 'end': (4, 8), 'tokType': 51, 'start': (...)
_len_	int: 6
end (30593632)	tuple: (4, 8)
start (30470624)	tuple: (4, 7)
tokLine (38573312)	str: grecord(scaler,"\$(P)\$(S)")\n
tokName (38572928)	str: OP
tokStr (38573024)	str: (
tokType (38605984)	int: 51

The Console window shows the PyDev debugger starting:

```
EpicsDatabase.py
pydev debugger: warning: psyco not available for speedups (the debugger will still work correctly, but a bit slow)
pydev debugger: starting
```

The Debug console shows the call stack:

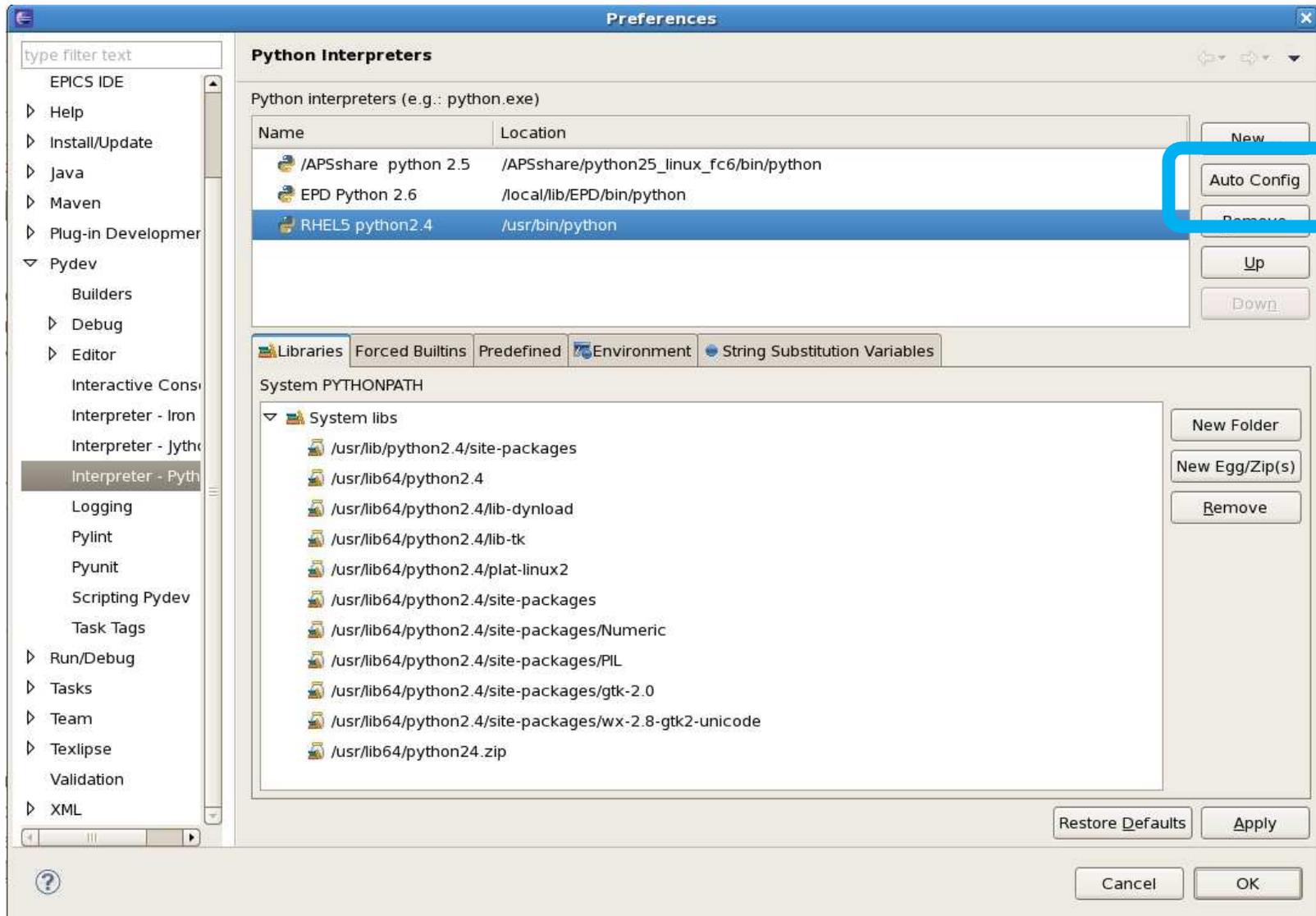
```
topdoc EpicsDatabase.py [Python Run]
EpicsDatabase.py
  MainThread - pid5804_seq1
    _parse [EpicsDatabase.py:135]
    _init_ [EpicsDatabase.py:56]
    <module> [EpicsDatabase.py:234]
    run [pydevd.py:918]
    <module> [pydevd.py:1133]
EpicsDatabase.py
```

# Systems with more than one Python distribution

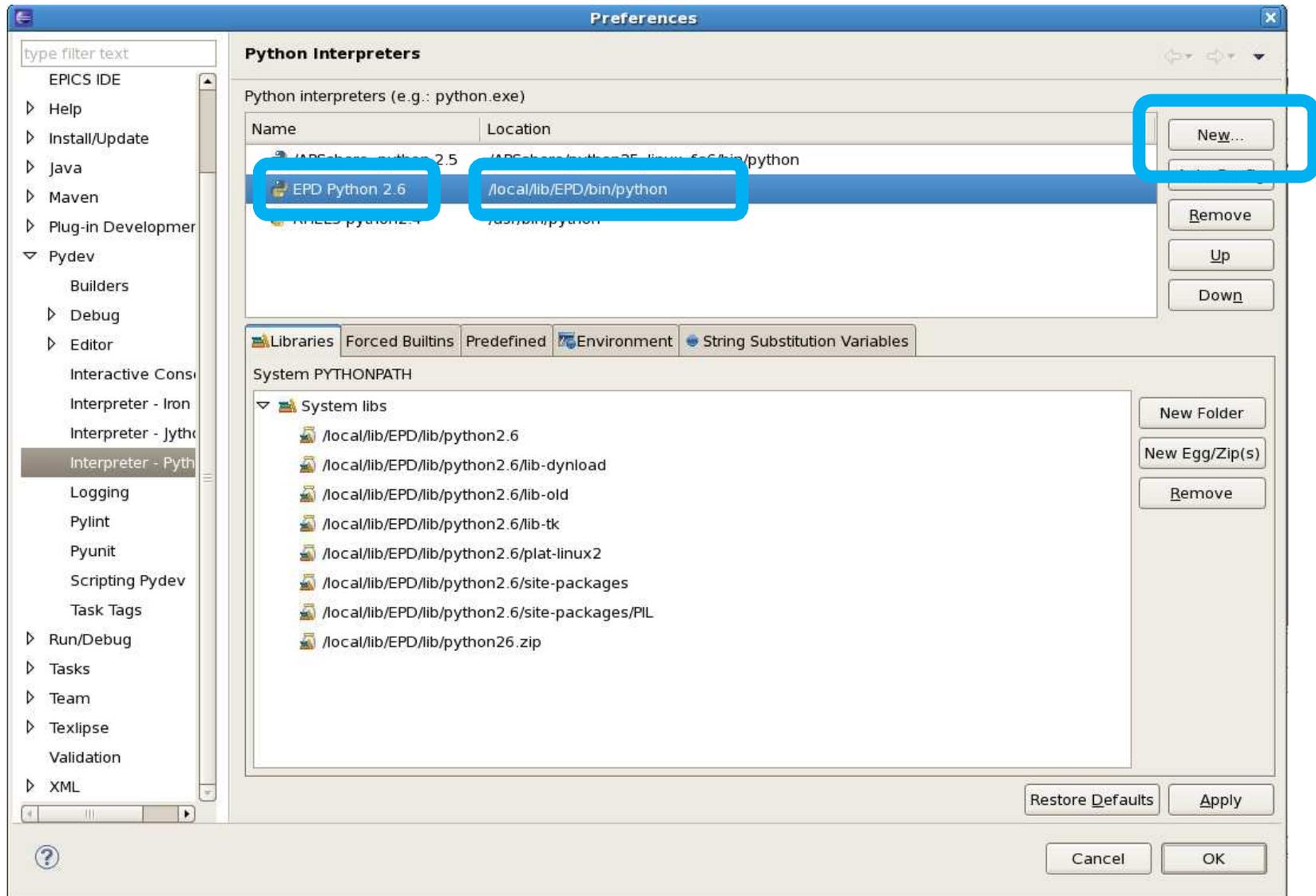
- Example Linux system has 3 Pythons to choose:
- RedHat Enterprise License v5 server
  - Python 2.4 supplied by RedHat
  - Found by “Auto Config” in PyDev
- User-installed Enthought Python Distribution in local directory
  - Python 2.6
  - Manually configured in PyDev
- BCDA Python on /APShare
  - Python 2.5 with CaChannel, CaPython, and ca\_utils
  - Manually configured in PyDev



# RHEL5 configuration

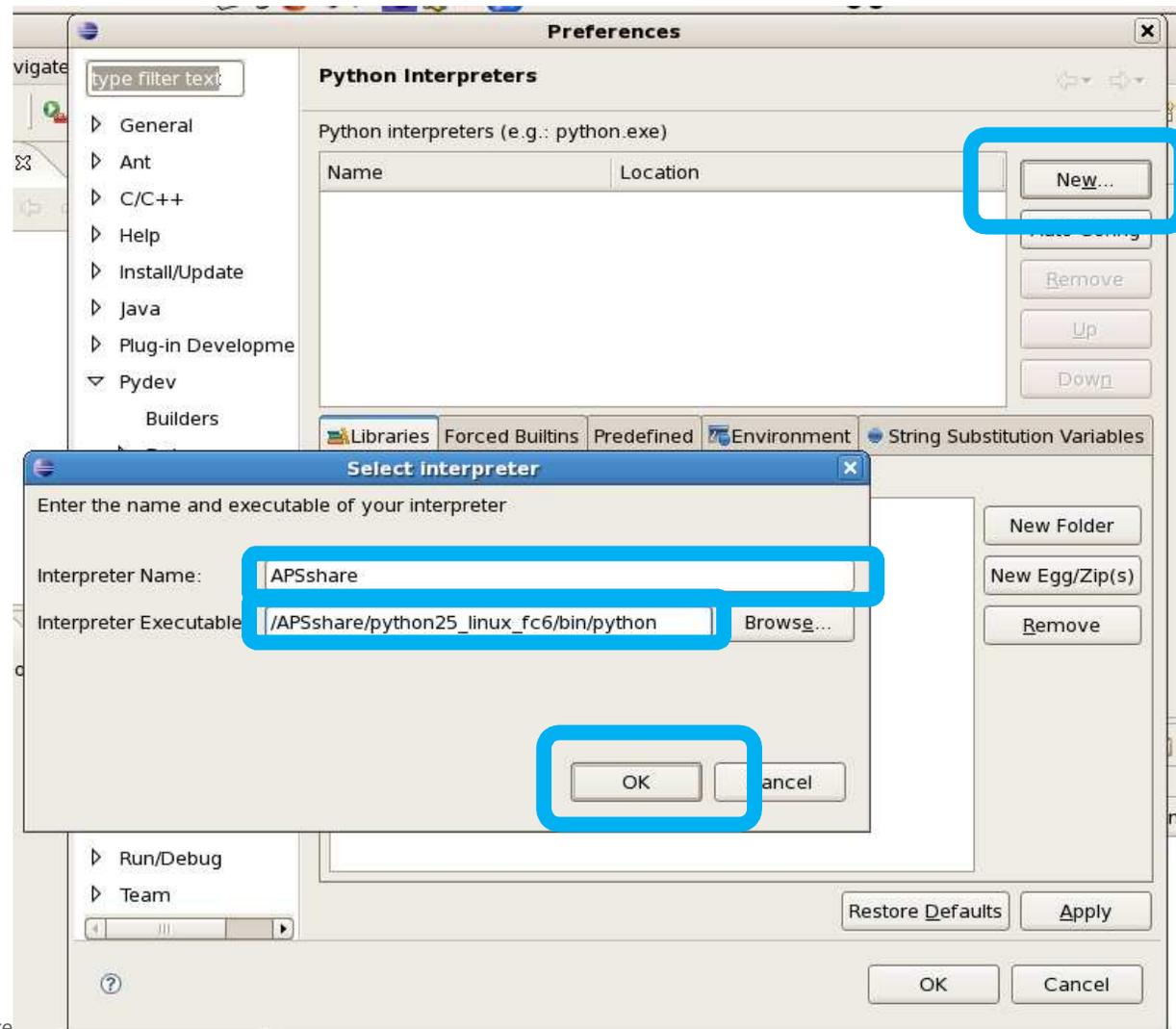


# local EPD Python configuration



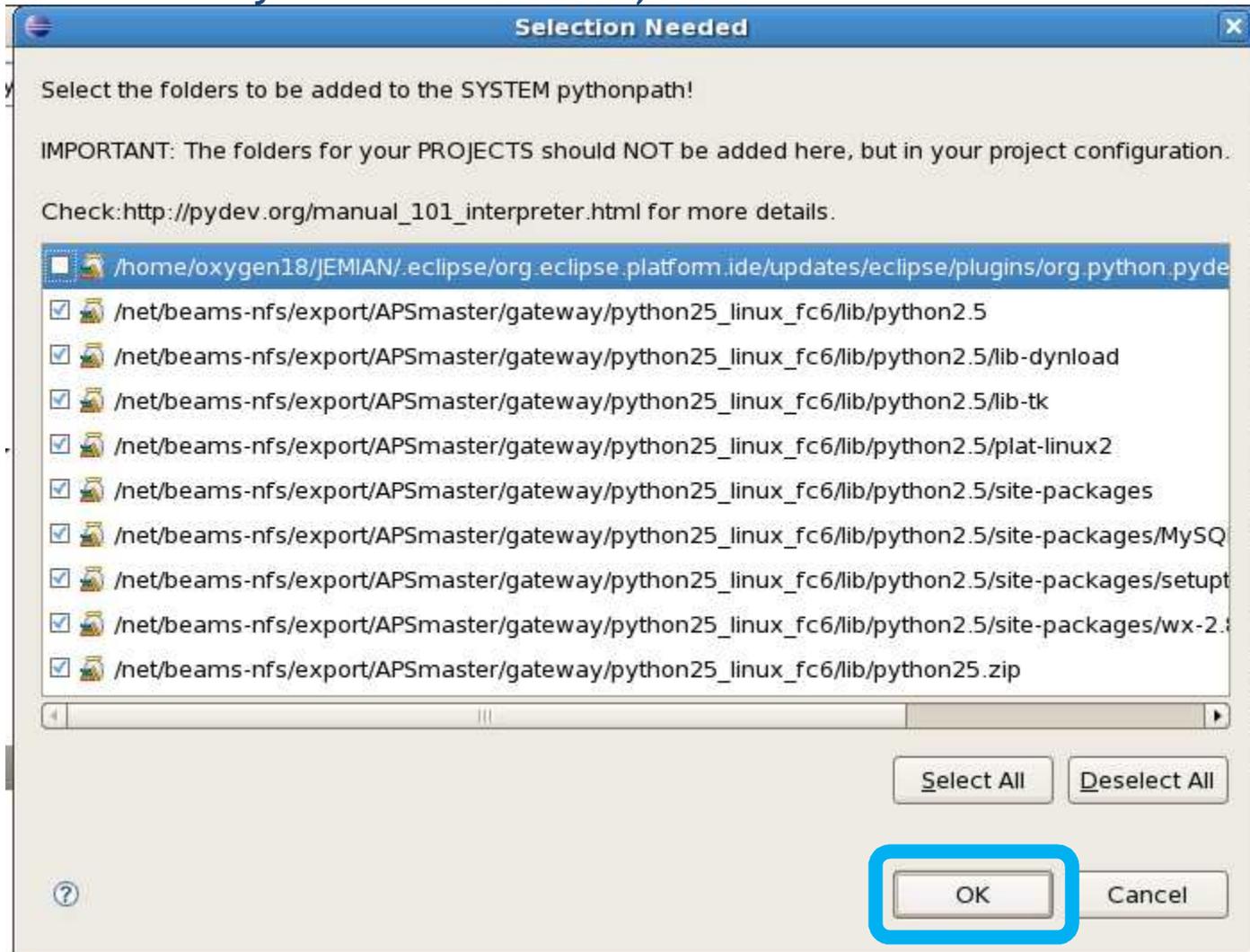
# /APSShare/bin/python configuration

executable: /APSShare/python25\_linux\_fc6/bin/python

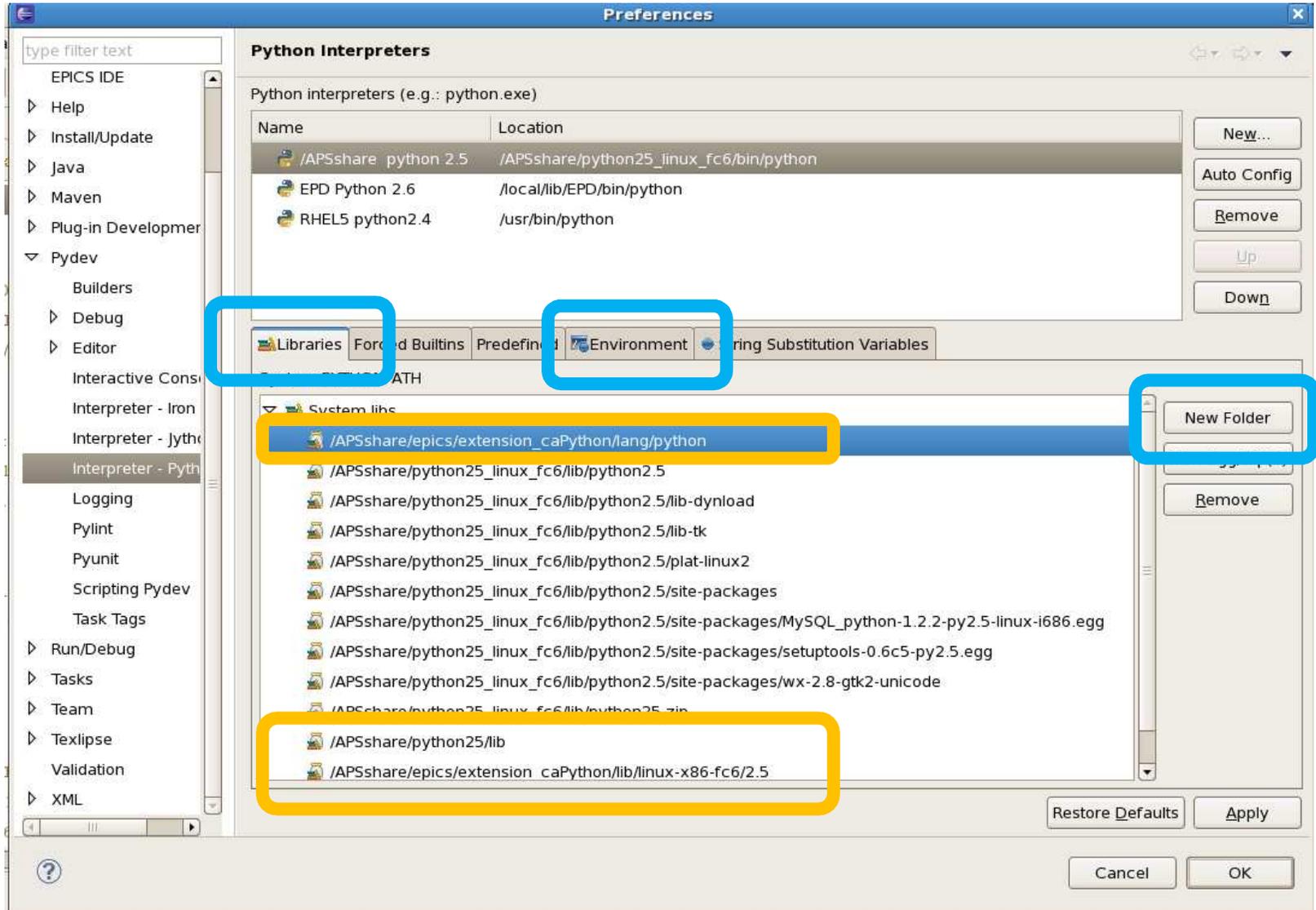


# Confirm this selection (probably all OK)

(note my paths are probably different than yours here, maybe shown like yours on next slide)

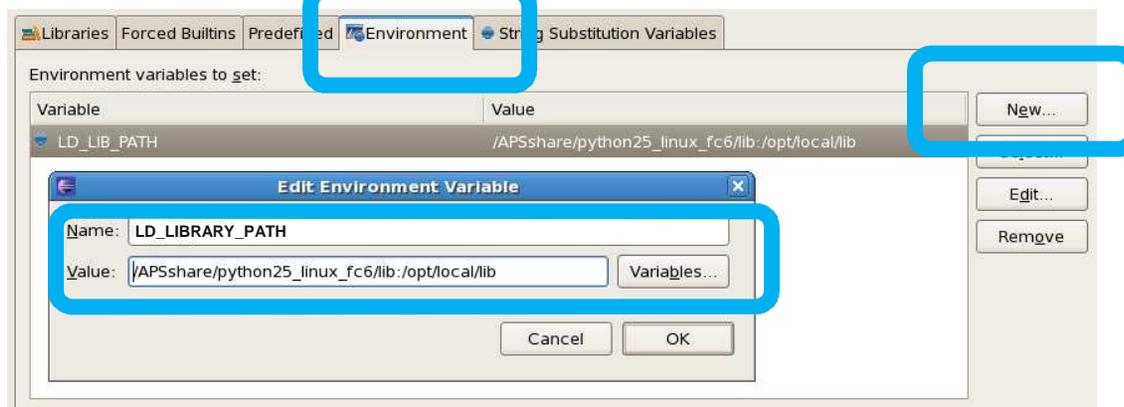


# Still more work to do ... (see next slide)



# Add to the APSshare Python

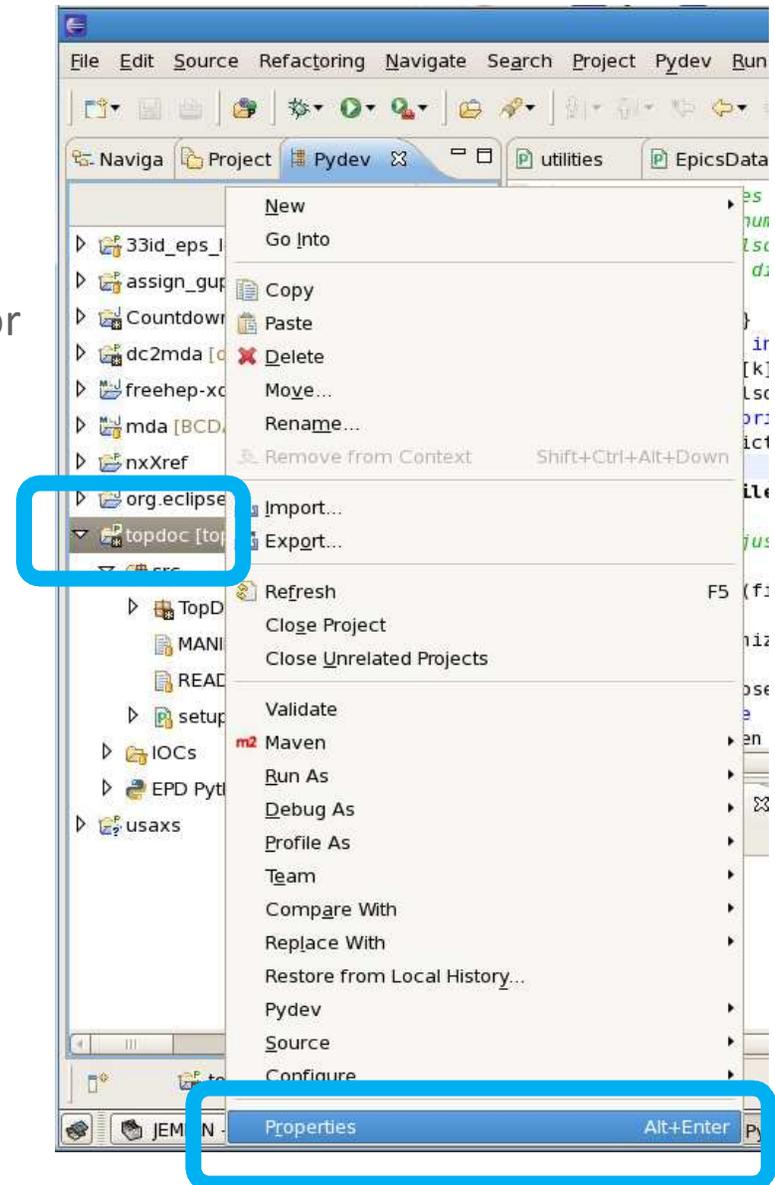
- Add these folders:
  - /APSshare/epics/extension\_caPython/lang/python
  - /APSshare/python25/lib
  - /APSshare/epics/extension\_caPython/lib/linux-x86-fc6/2.5
  - /APSshare/pythonlib
- Add this environment variable (ENVIRONMENT tab)
  - LD\_LIBRARY\_PATH /APSshare/python25\_linux\_fc6/lib:/opt/local/lib



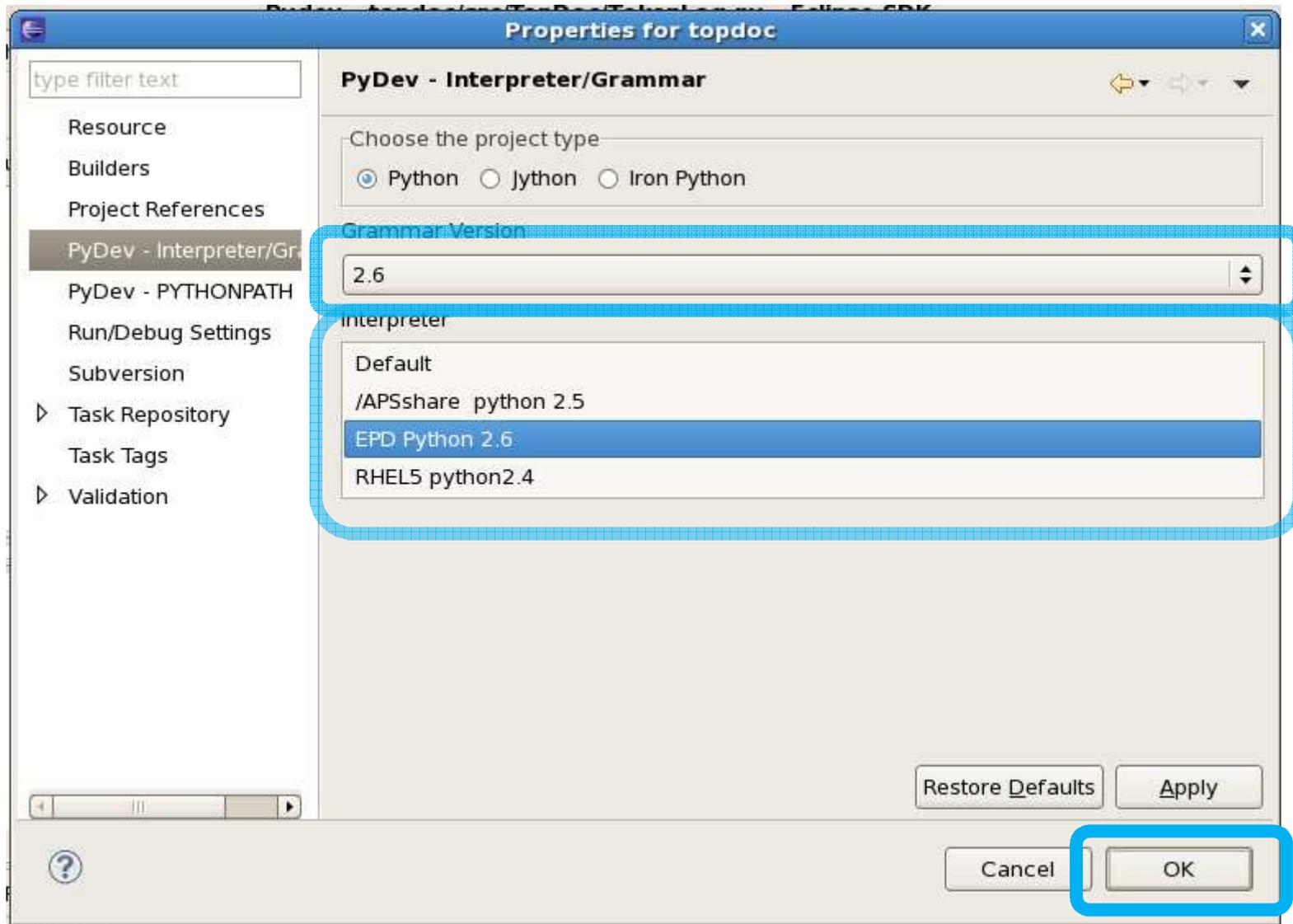
Click through the [Ok] buttons and accept the new configuration ...

# Getting to the Properties ...

- Select your project (“topdoc” here)
- Right click on the project name in the Navigator
- Pop-up menu appears
- Select “Properties” item at bottom of list
- On next screen ...
- Select “PyDev – Interpreter/Grammar” in tree pane (on left)
- Select per next slide



# Selection of Python configuration



Thank you for your attention

