TUTORIAL

ON SETTING UP OPENERP 6.1 SOURCE CODE

IN

ECLIPSE CLASSIC VERSION: 3.7.2

UNDER WINDOWS PLATFORM

by Pir Khurram Rashdi

Web: ***http://www.linkedin.com/in/khurramrashdi*** ***Email:*** ***pkrashdi@gmail.com***

Written by Pir Khurram Rashdi Email: pkrashdi@gmail.com

TABLE OF CONTENTS

Written by Pir Khurram Rashdi Email: pkrashdi@gmail.com

TABLE OF FIGURES

Written by Pir Khurram Rashdi Email: pkrashdi@gmail.com

Written by Pir Khurram Rashdi Email: pkrashdi@gmail.com

1. INTRODUCTION

Nothing is impossible in this world of possibilities and as far as I've little experience of working for OpenSource, I was surprised to see that everything is possible on almost every platform when it comes to scalability of OpenSource software. Thanks goes to OpenSource communities including Developers and Sponsors who take interest in this world for making others life easier.

I will attempt in this tutorial to explain the procedure of setting up source code of OpenEclipse for customization under Eclipse on Windows platform. When I started it seemed little bit difficult but when I finished, it seemed much easier to repeat for me. So everybody who will follow this step-by-step procedure will get the source code in working state to start customizing while living on Windows platform. If anything is not clear please let me know through my email and I will update the contents for greater comprehension.

2. PROCEDURE

The procedure goes as follows:-

Step 1. JAVA IDE INSTALLATION

Install a Java-based version of Eclipse on Windows. In my case , I installed Eclipse SDK Indigo Version 3.7.2.

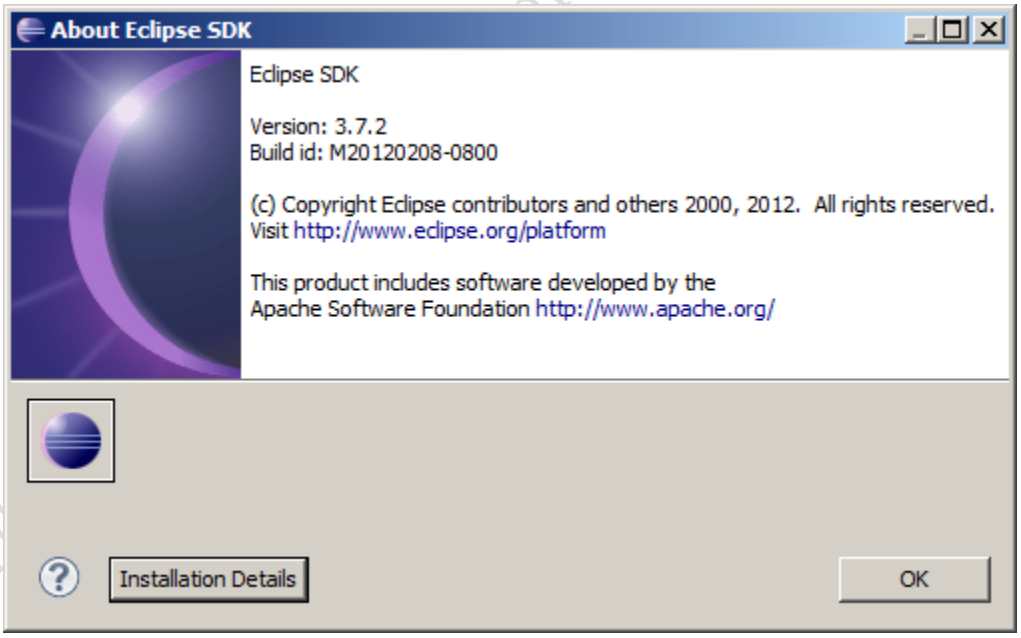

FIGURE 1 - ECLIPSE VERSION

Download link is:

For both 32-bit and 64 bit:-

Email: pkrashdi@gmail.com

http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.7.2-201202080800/eclipse-SDK-3.7.2-win32-x86_64.zip

After downloading this package, unzip it to the D:\ drive or root of any drive. It will create a folder named **Eclipse**, go to the folder and make a shortcut of **Eclipse.exe** to the desktop for easy access. When you will launch the Eclipse for the first time, it will require JRE (Java-Run time environment) so you'll need to download and install that too.

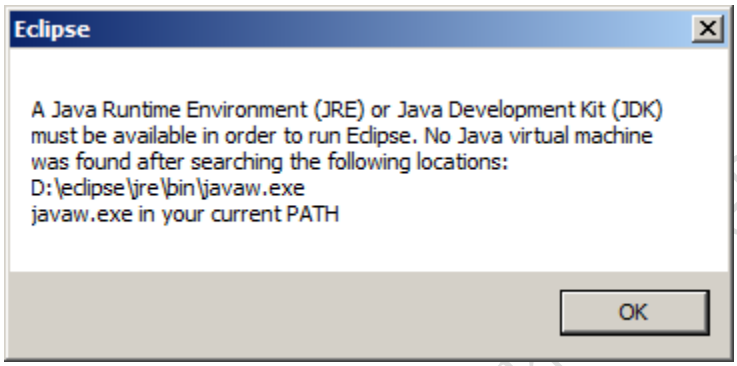

FIGURE 2 - JRE REQUIREMENTS NOTIFICATION

In my case, I downloaded **JRE (Java Runtime Environment)** from path:-

- http://download.oracle.com/otn-pub/java/jdk/7u3-b05/jre-7u3-windows-x64.exe (64-bit)
- http://download.oracle.com/otn-pub/java/jdk/7u3-b05/jre-7u3-windows-i586.exe (32-bit)

After downloading JRE, you need to run the .exe file (64-bit in my case) just downloaded, and it will proceed as follows:-

Written by Pir Khurram Rashdi Email: pkrashdi@gmail.com

FIGURE 3 -JRE INSTALLATION WIZARD

On clicking, the Install, the setup will begin and finish quickly as below:-

FIGURE 4 - JRE INSTALLATION FINISHED SUCCESSFULLY

Close this finished dialog. Then re-launch Eclipse by clicking shortcut on your desktop you created in previous step. Now when Eclipse is launched, it will detect JRE and will ask for Workspace folder for the first time, so you can create any folder. In my case, I created **D:\EclipseWorkspace** and chose to use this folder always by default.

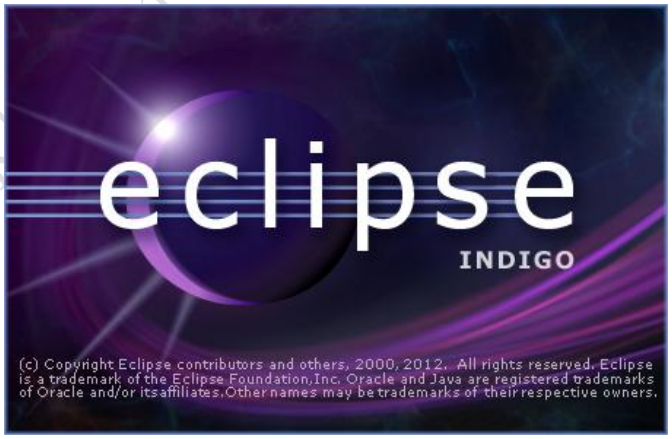

FIGURE 5 - ECLIPSE LAUNCH

Written by Pir Khurram Rashdi Email: pkrashdi@gmail.com

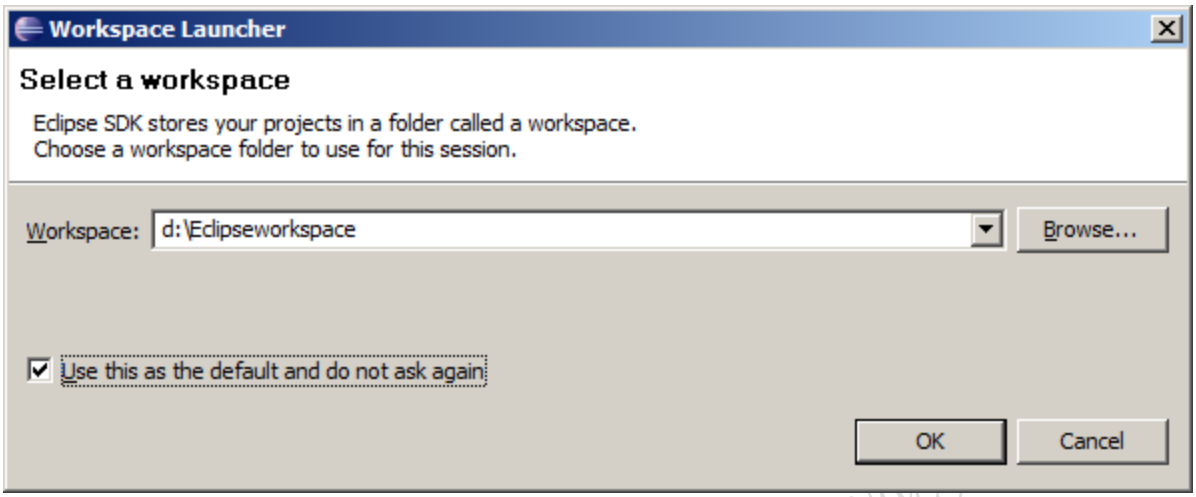

FIGURE 6 - NEW WORKSPACE LOCATION (WILL NEED TWO ONE FOR SERVER AND ONE FOR CLIENT)

Step 2. DOWNLOAD PYTHON LIBRARIES

Now we need to download the dependencies for OpenERP 6.1 to compile and run the source code for OpenERP. Please download the installers and libraries from the following path and sub-paths:-

Client: http://nightly.openerp.com/6.1/nightly/windows_manual_build/client/
Server: http://nightly.openerp.com/6.1/nightly/windows_manual_build/server/

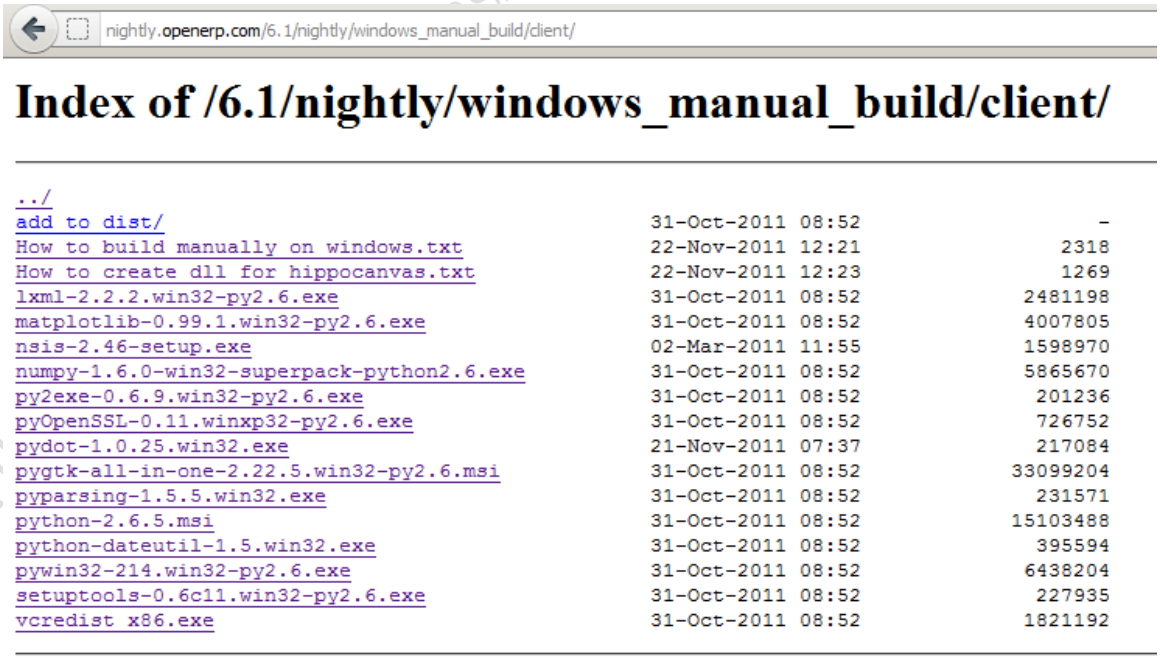

FIGURE 7 - CLIENT DEPENDENCIES OR PYTHON 2.6 LIBRARIES

Written by Pir Khurram Rashdi, Email: pkrashdi@gmail.com

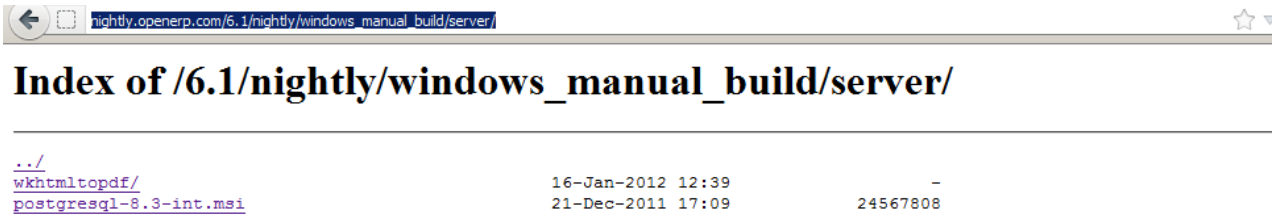

FIGURE 8 - SERVER DEPENDENCIES (JUST FOR INFO BUT WILL NOT NEED THESE ONES)

Step 3. PYTHON INTERPRETER INSTALLATION

Now we need to install Python interpreter which we just downloaded i.e.2.6.5. Just double click the installer i.e. **python-2.6.5.exe**, and it will start as below so choose a folder such as **Python26**. I installed the package to path **D:\Python26**.

FIGURE 9 - PYTHON 2.6.5 INTERPRETER INSTALLATION WIZARD

Written by Pir Khurram Rashdi, pkrashdi@gmail.com

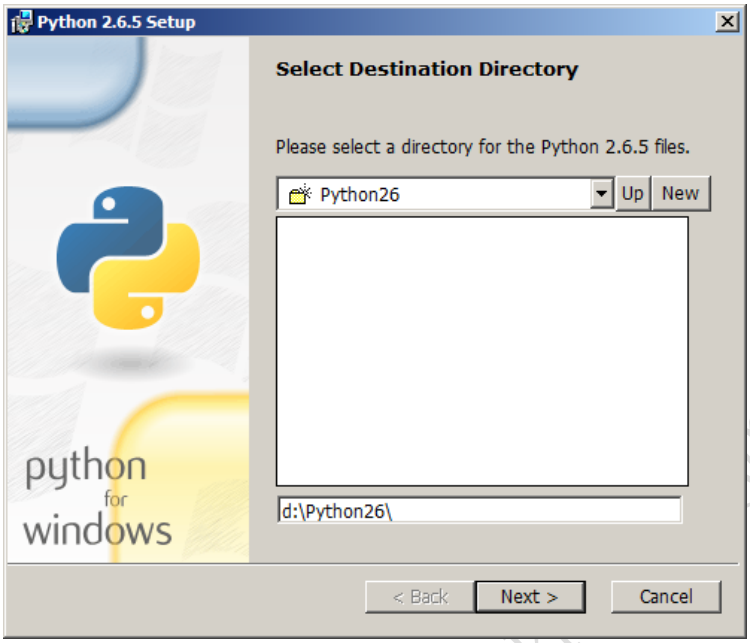

FIGURE 10 - PYTHON INSTALLATION PATH SELECTION

FIGURE 11 - PYTHON OPTIONAL COMPONENTS (ALL SELECTED)

Written by Pir Khurram Rashdi, Email: pkrashdi@gmail.com

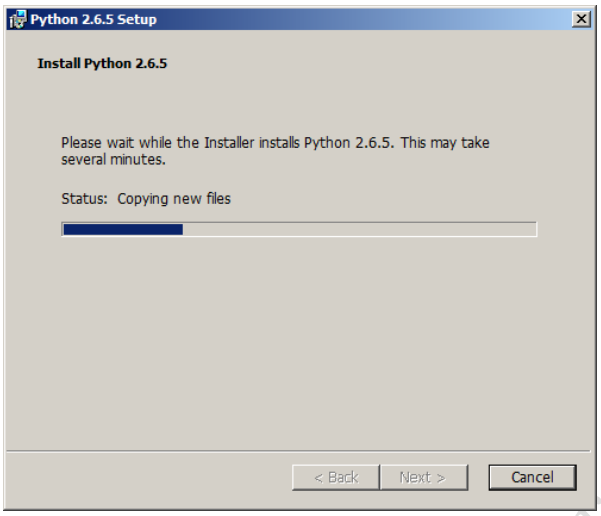

FIGURE 12 - PYTHON INSTALLATION IN PROGRESS

FIGURE 13 - PYTHON INSTALLATION FINISHED

Step 4. ENVIRONMENTAL VARIABLE SETUP FOR PYTHON

Add path of python to **PATH** variable concatenated by semicolon (;) to the existing string in Environmental variables, by right-clicking My computer icon then choosing **properties.** Rest is clear in screenshot below how we added the path:-

<existing path>;D:\Python26

Written by Pir ... shdi@gmail.com

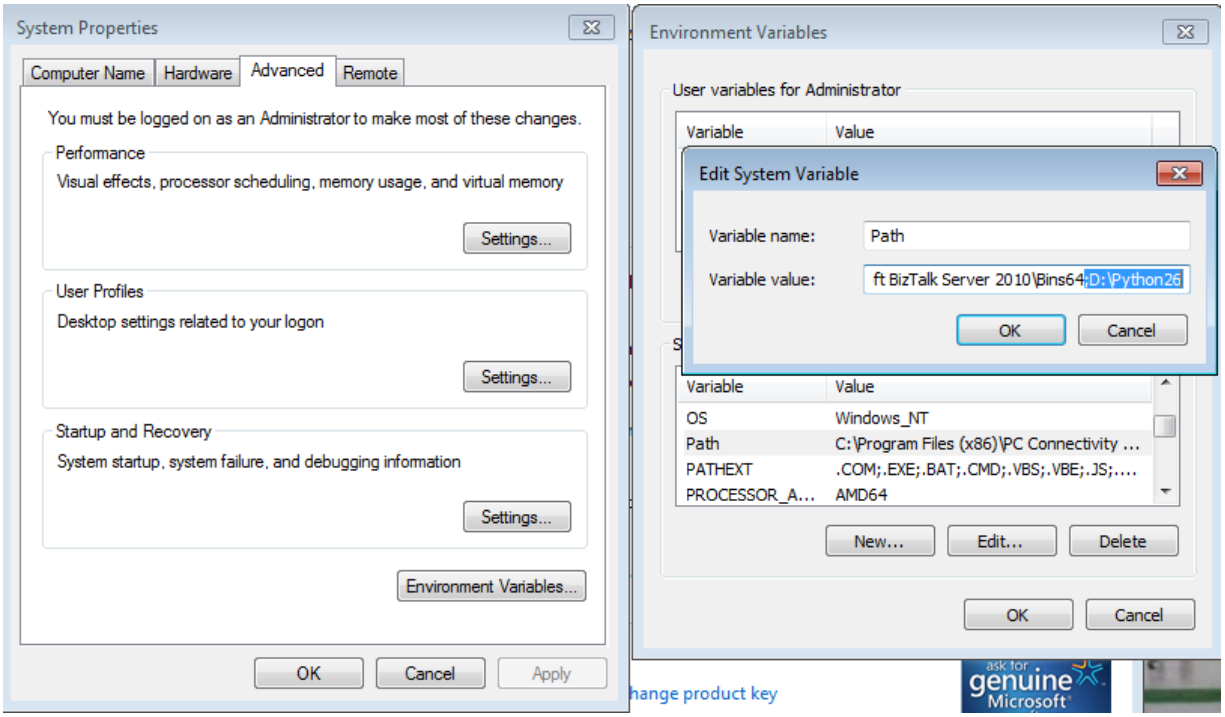

FIGURE 14 - ENVIRONMENTAL VARIABLES (PATH) SETTINGS FOR PYTHON INTERPRETER

Step 5. POSTGRESQL INSTALLATION

Install **PostgreSQL Ver 8.4** by downloading package **postgresql-8.4.8-1-windows.exe** and running setup. It will ask for program and data folders, so accept what is default path. It will utilize default port: 5432. It will ask password for user **postgres.** Enter some complex password and keep it somewhere in notes.

Download path:-
http://www.enterprisedb.com/products-services-training/pgdownload#windows

Step 6. INSTALLATION OF PYTHON LIBRARIES FOR OPENERP 6.1

Now install the dependencies in the form of installers other than Python interpreter (already installed in previous step), one-by-one. It includes all dependencies. Following screenshot shows majority of dependencies, but not all.

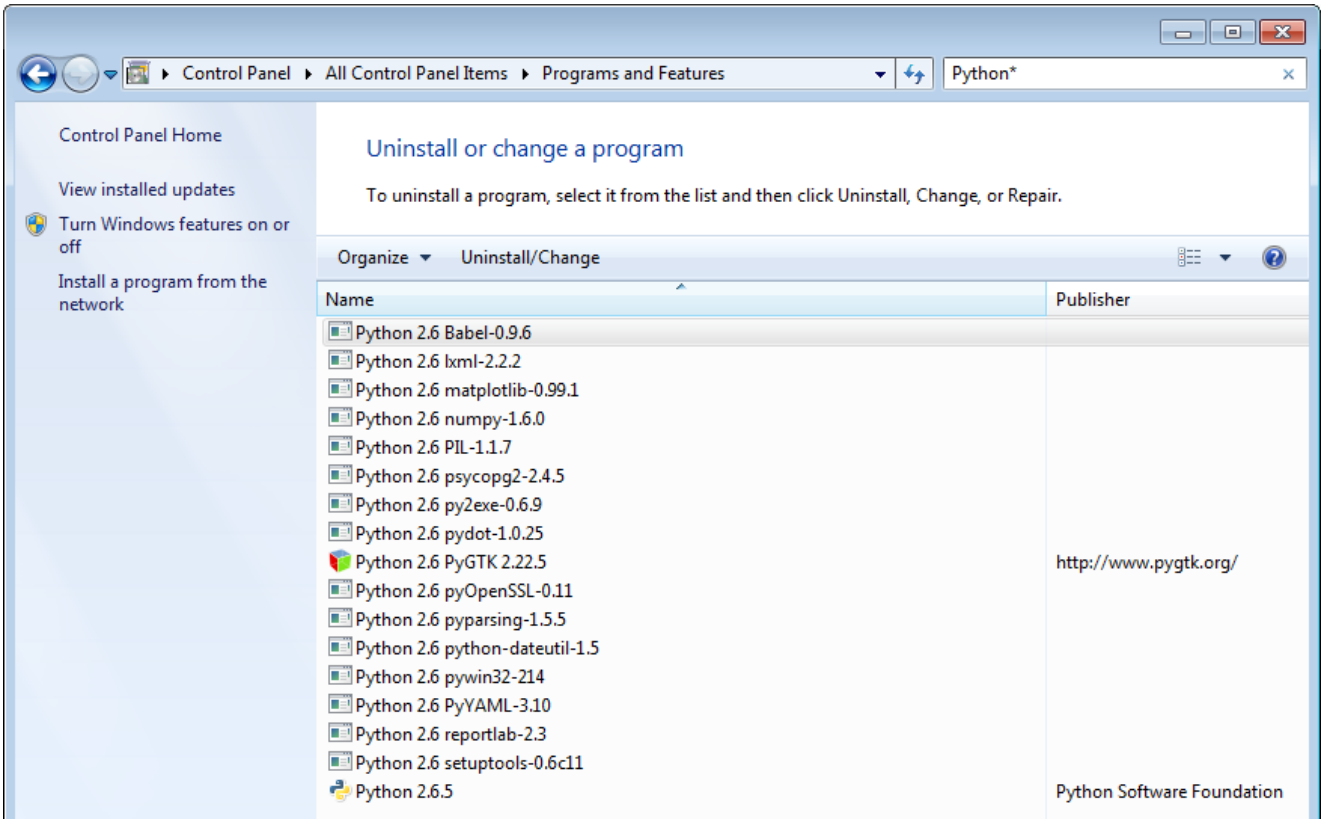

FIGURE 15 - PYTHON LIBRARIES VIA WINDOWS INSTALLERS INSTALLED (NOT A COMPLETE LIST)

Following are the rough steps for python libraries/ dependencies, but you need to directly proceed, almost all Windows installers will automatically detect Python and install in respective path of Python (**D:\Python26** in my case) so this will be quite straight-forward process:

a. Install all in one PyGTK package: **pygtk-all-in-one-2.22.5.win32-py2.6.msi** (default installation)

b. Install setuptools-0.6c11.win32-py2.6

c. Install python-dateutil-1.5.win32

d. Install pywin32-214.win32-py2.6

e. Install lxml-2.2.2.win32-py2.6

f. Install pydot-1.0.25.win32.exe.

g. Install matplotlib-0.99.1.win32-py2.6

h. Install numpy-1.6.0-win32-superpack-python2.6

i. Install pyparsing-1.5.5.win32

j. Install pyOpenSSL-0.11.winxp32-py2.6

k. Install py2exe-0.6.9.win32-py2.6

l. Install nsis-2.46-setup.exe for creating auto installer

Step 7. Download and Install Dependencies for Server

Dependencies for Server/Web client are as under:-

m. Download and install **psycopg2-2.4.5.win32-py2.6-pg9.1.3-release.exe** from URL:- http://www.stickpeople.com/projects/python/win-psycopg/psycopg2-2.4.5.win32-py2.6-pg9.1.3-release.exe

n. Download and install **simplejson-2.5.0.tar.gz** from URL:- http://pypi.python.org/packages/source/s/simplejson/simplejson-2.5.0.tar.gz

   Unzip (using WinRAR) to some folder and it will contain file **setup.py** python script which will be installed to **D:\Python26** install path using following command in **Command Prompt** window when current folder is the path where individual package is unzipped :-

   <Path where package is unzipped> **python setup.py install**

   Please also note that since **D:\Python26** where we installed the python interpreter is already included into environmental variable **PATH** that's why the above command will run.

o. Download and Install **PyYAML** for Python 2.6 from URL:- http://pyyaml.org/download/pyyaml/PyYAML-3.10.win32-py2.6.exe

p. Download and install **reportlab-2.3.win32-py2.6.exe** from link: http://www.reportlab.com/ftp/reportlab-2.3.win32-py2.6.exe.

q. Download and extract the **Mako** package from the link:- http://pypi.python.org/packages/source/m/mako-version-tag/mako-version-tag-1.1.tar.gz#md5=d721b121ee42963a75920a6f793a36aa. Then run command **python setup.py install** as we did in previous step from the path where the package was extracted. This will also download the dependencies for Mako from Internet and process them, so will require some time. Screen at the end would look like as below:-

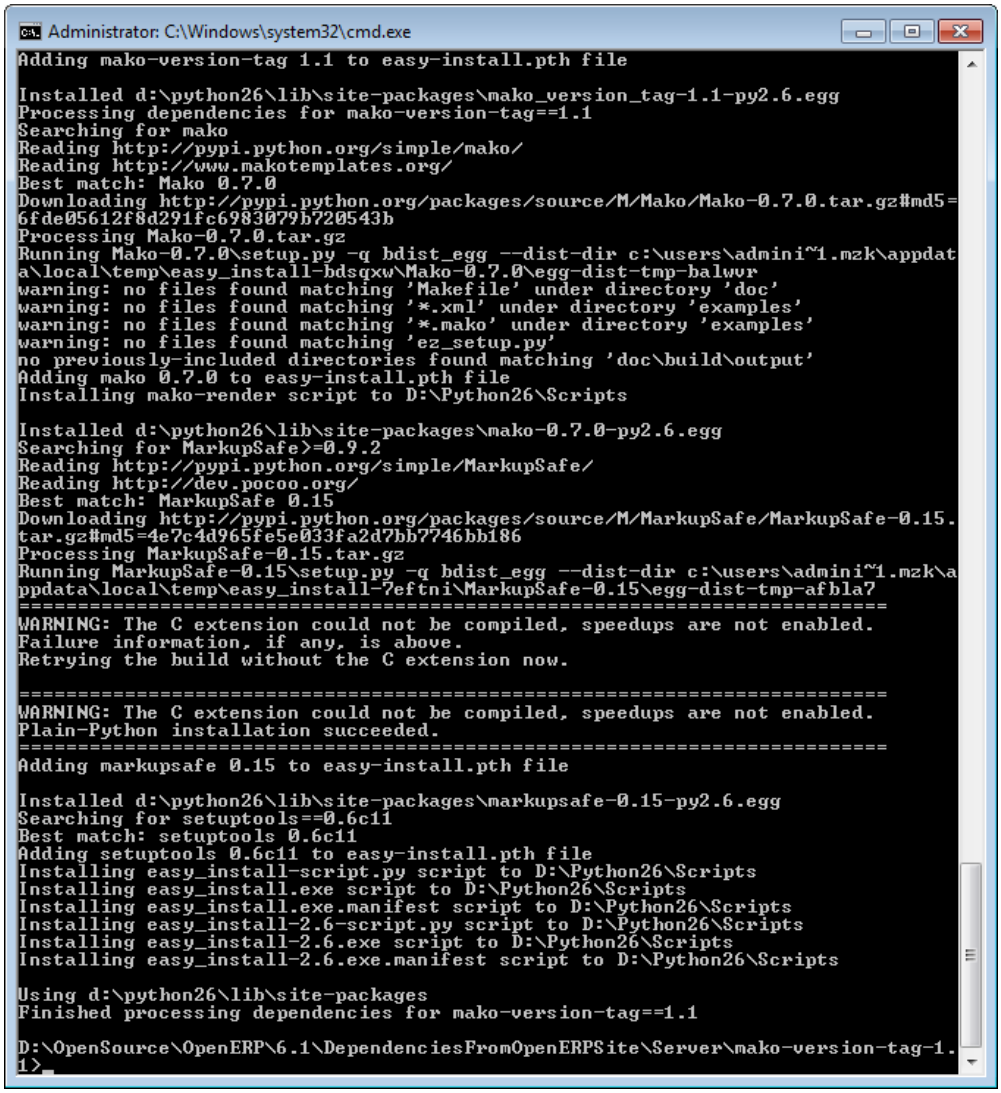

FIGURE 16 – MAKO PACKAGE INSTALLED VIA PYTHON

r. Download and extract **PyChart** i.e. file **PyChart-1.39.tar.gz** from link: http://download.gna.org/pychart/PyChart-1.39.tar.gz. Then run command **python setup.py install** as we did in previous step from the path where the package was extracted. It will install **PyChart** for Python.

s. Download and extract package **Werkzeug-0.8.3.tar.gz** from the link: http://pypi.python.org/packages/source/W/Werkzeug/ . Then run command **python setup.py install** as we did in previous step from the path where the package was extracted. It will install **Werkzeug** for Python.

Step 8. DOWNLOAD AND INSTALL DEPENDENCIES FOR WEB CLIENT

Dependencies for OpenERP webclient and their download links are as under:-

- Download and extract the package **simplejson-2.5.0.tar.gz** from the link: http://pypi.python.org/packages/source/s/simplejson/simplejson-2.0.9.tar.gz#md5=af5e67a39ca3408563411d357e6d5e47. Then run command **python setup.py install** as we did in previous steps from the path where the package was extracted. It will install **simplejson-2.5.0** for Python.

- **Babel-0.9.6** Windows installer, simply download run, it is straight-forward and install automatically to Python's path :- http://ftp.edgewall.com/pub/babel/Babel-0.9.6.win32.exe

- **Python-openid 2.2.5** Windows installer, simply download run, it is straight-forward and install automatically to Python's path: http://pypi.python.org/packages/source/p/python-openid/python-openid-2.2.5.tar.gz#md5=393f48b162ec29c3de9e2973548ea50d

- **PIL 1.1.7** Windows installer, simply download run, it is straight-forward and install automatically to Python's path: http://effbot.org/media/downloads/PIL-1.1.7.win32-py2.6.exe

Step 9. Installation of PyDev Add-on For Python in Eclipse

Then comes the role of open-source plug-in of PyDev for Eclipse which will integrate the Python interpreter, we installed in previous step, to Eclipse and will allow us to create a new project type of PyDev type (Python-based). Install PyDev via the Eclipse update manager (**Help menu > Install New Software**) by adding a new site http://pydev.org/updates and then running the update to install this plug-in.

Written by Pir Khurram Rashdi, Email: pkrashdi@gmail.com

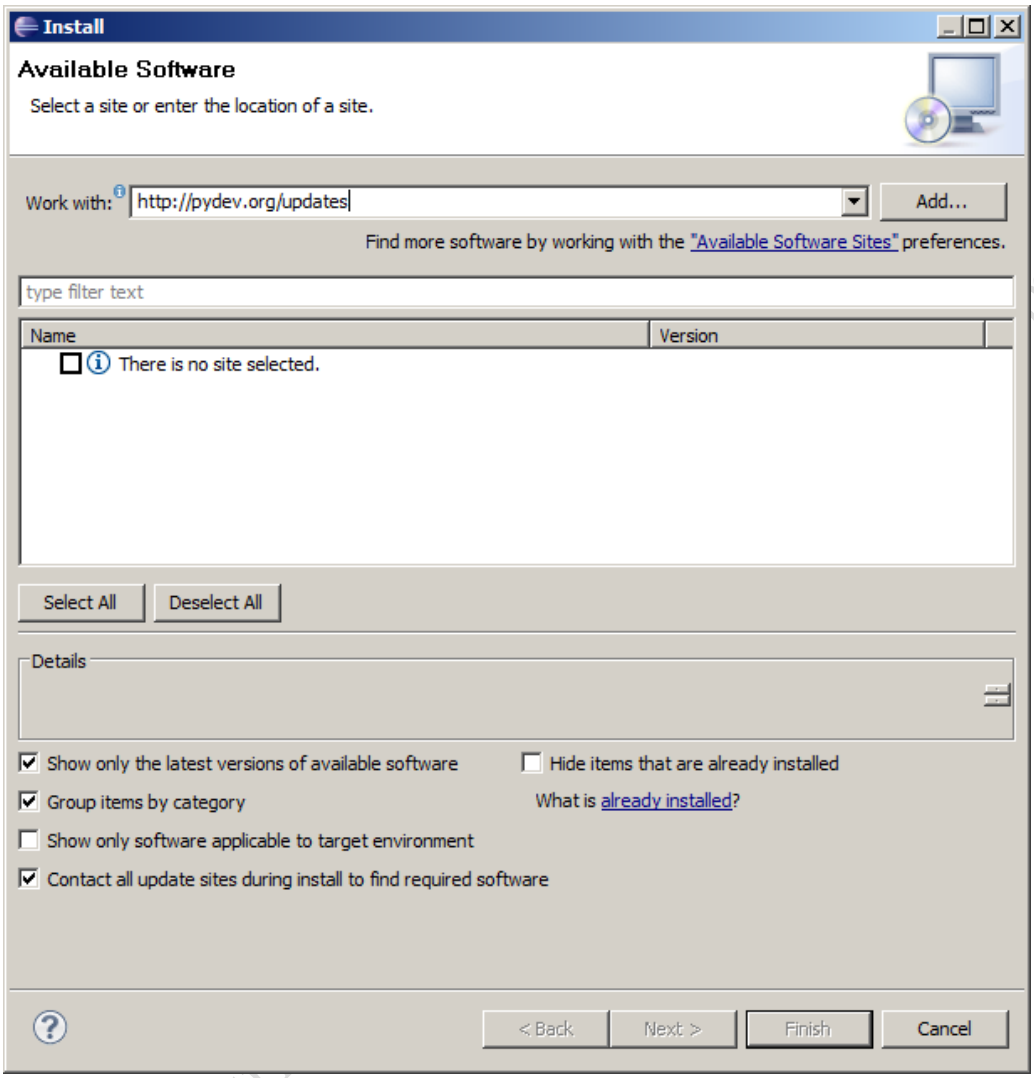

FIGURE 17 - ADDING PYDEV PLUG-IN SITE INFO FOR INSTALL/UPDATE

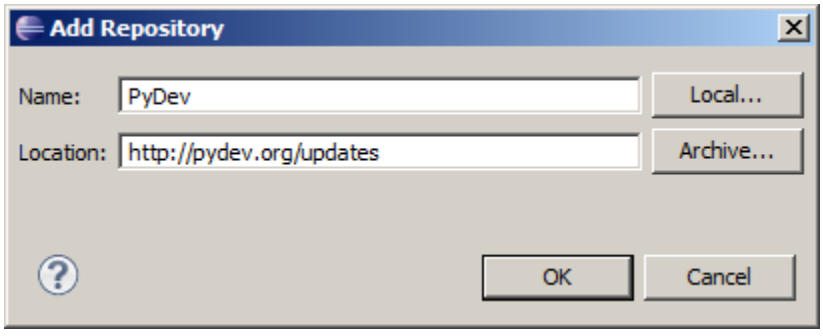

FIGURE 18 - PYDEV INSTALLATION URL

Written by Pir Khurram Rashdi

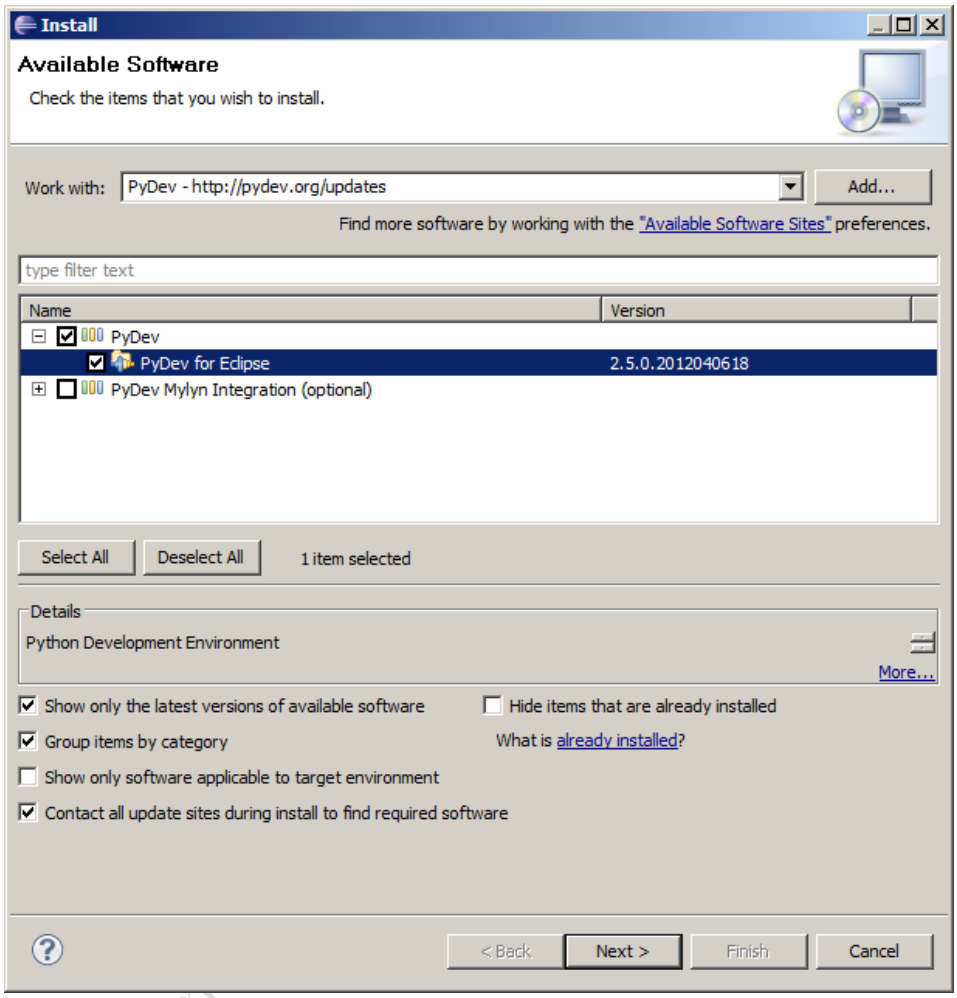

FIGURE 19 - PYDEV ADDON FOR ECLIPSE SELECTED FOR INSTALLATION WHILE ONLINE

Written by Pir Khurram

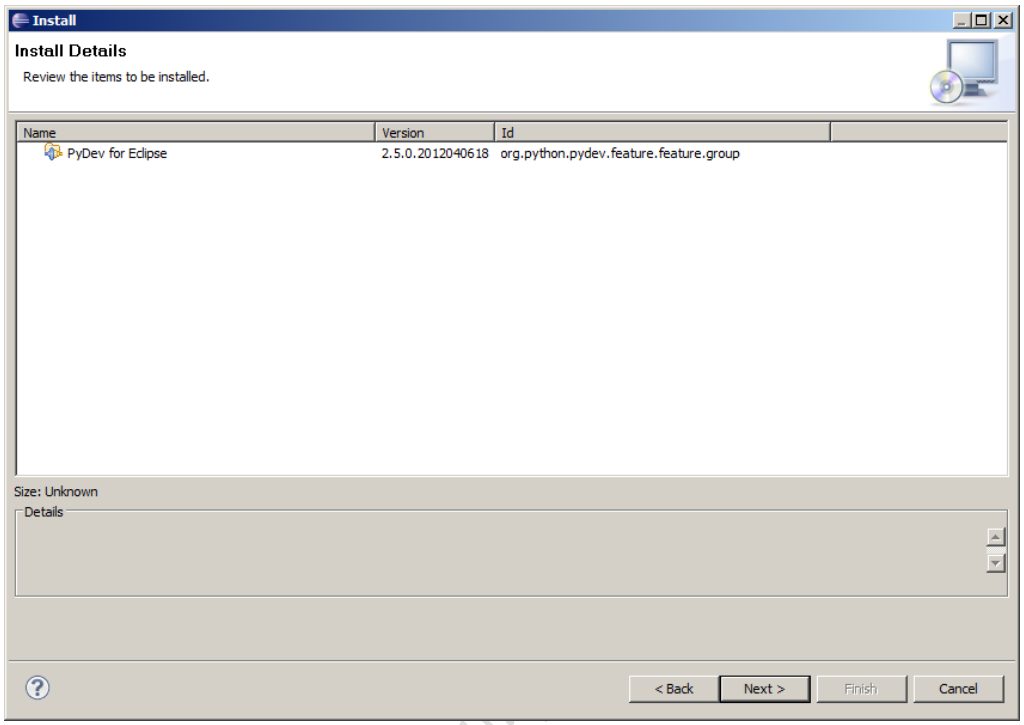

FIGURE 20 - PYDEV INSTALLATION WIZARD CONTINUES

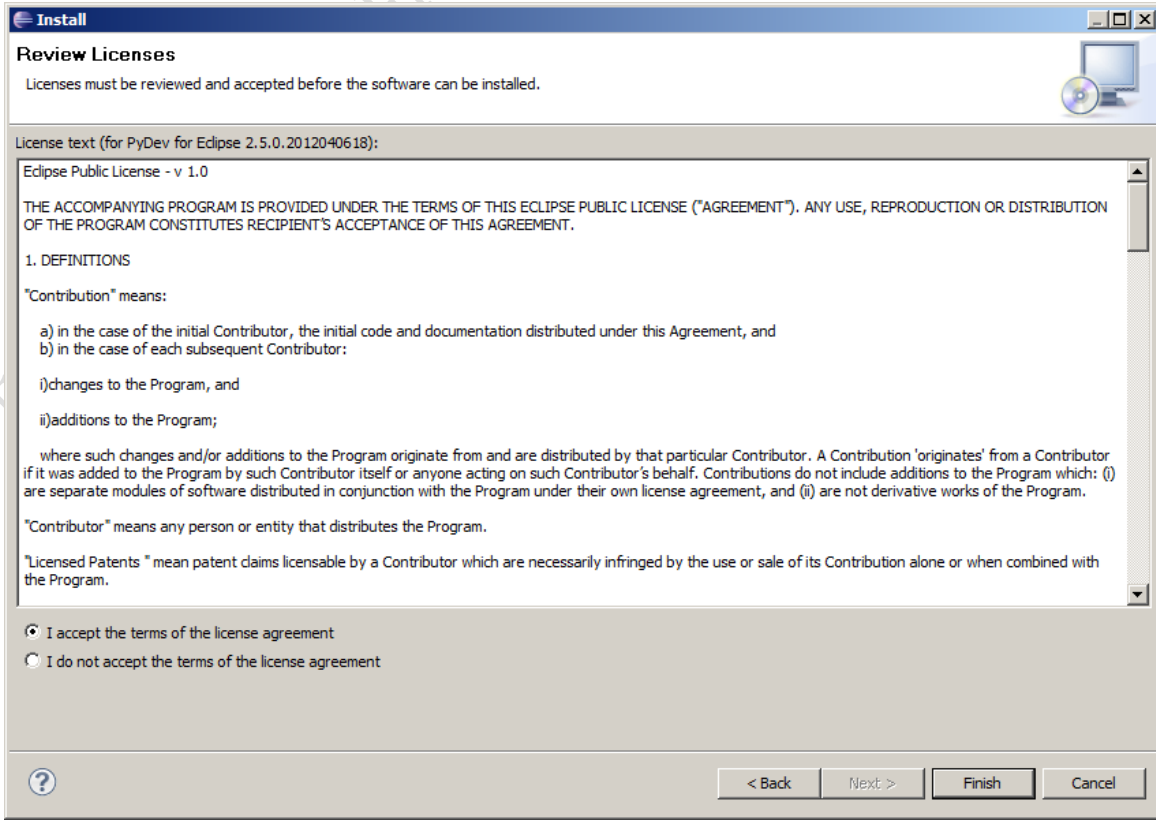

FIGURE 21 - ACCEPT LICENSE

Written by Pir Khurram Rashdi, Email: pkrashdi@gmail.com

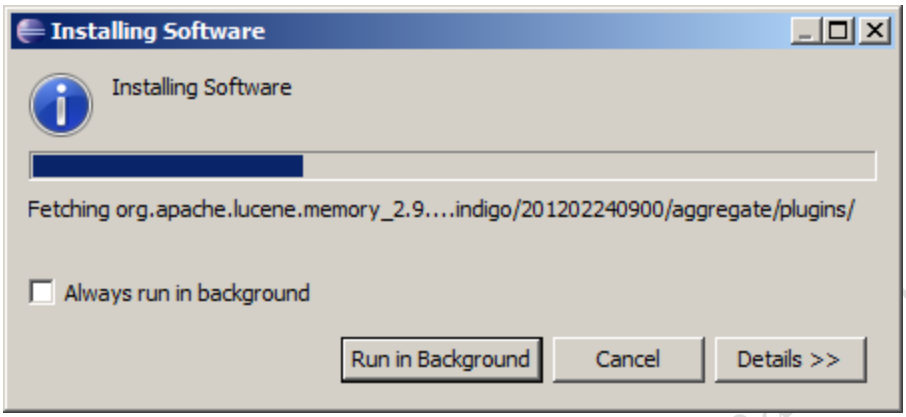

FIGURE 22 - INSTALLATION PROCEEDS

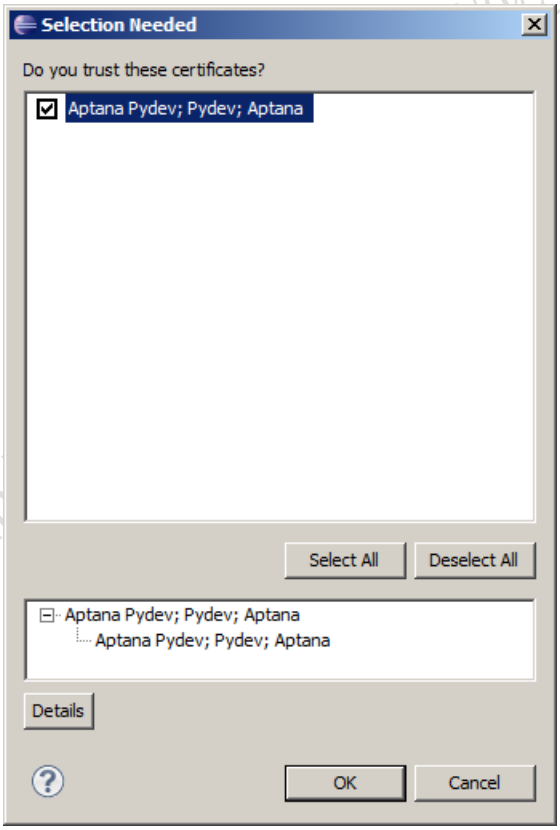

FIGURE 23 - PYDEV CERTIFICATE ACCEPTANCE

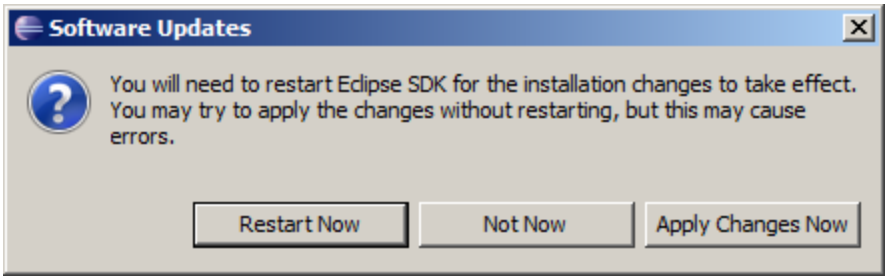

FIGURE 24 - RESTART ECLIPSE FOR PYDEV CHANGES TO TAKE PLACE

Written by Pir Khurram

Clicked Restart Now.

Step 10. Configuration of PyDev Add-On For Python in Eclipse

Now we have to configure the PyDev plug-in installed in Eclipse in previous step by providing the path of Python interpreter Ver 2.6.5 which we installed in previous steps. Open the menu **Window -> Preference** and select **Pydev-> Interpreter Python**. Then press New button in top pane and locate the python.exe in the installed path e.g. **D:\Python26\python.exe** via Browse button and also give the name to this interpreter as shown in snapshot.

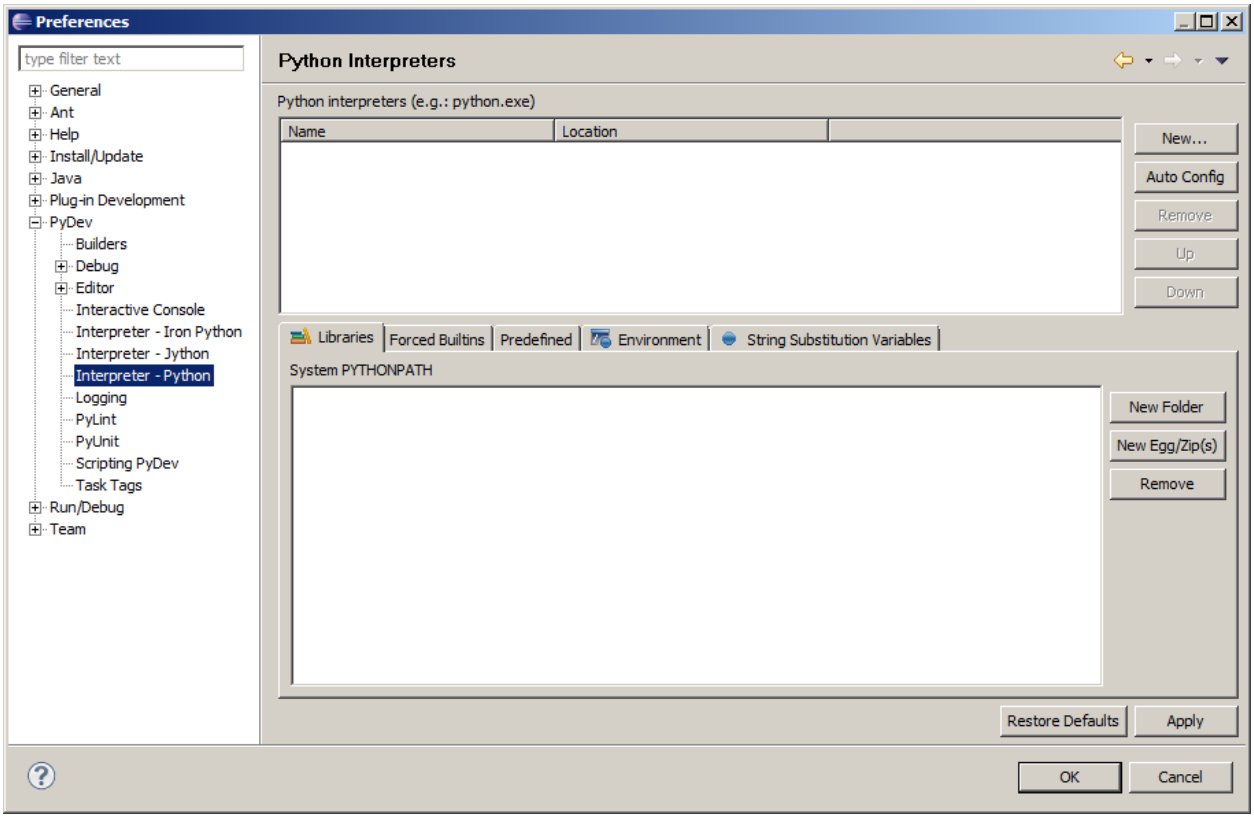

Figure 25 - Python interpreter configuration for PyDev in Eclipse

Written

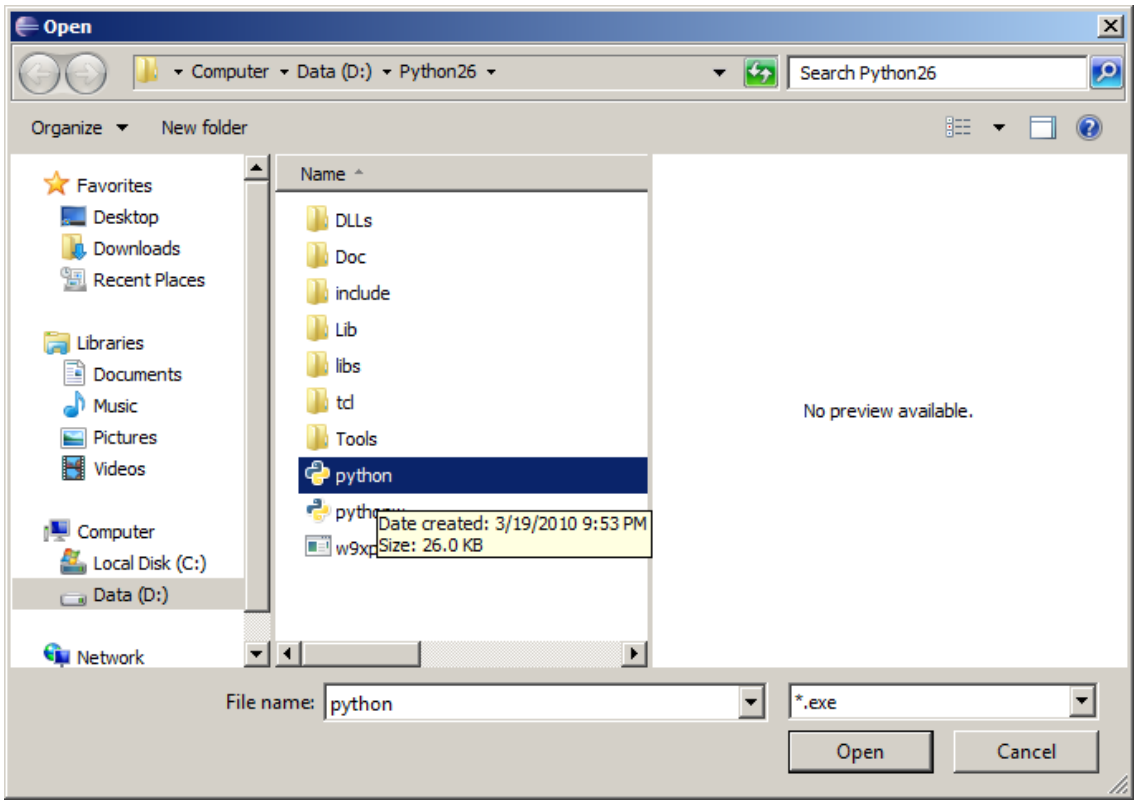

FIGURE 26 - DEFINING PYTHON 2.6 INTERPRETER PATH TO ECLIPSE

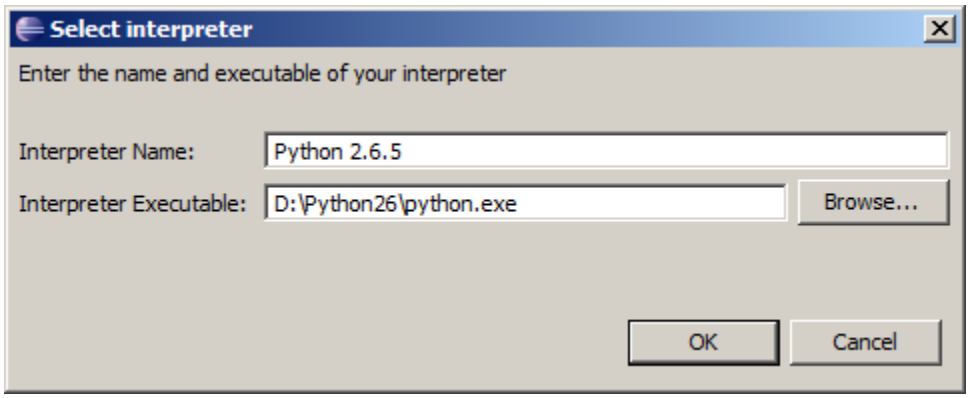

FIGURE 27 – NAME THE PYTHON INTERPRETER ONCE PATH IS SELECTED

It will also import all libraries installed to Python automatically in below pane and should something like the snapshot shown below.

Written by Pi

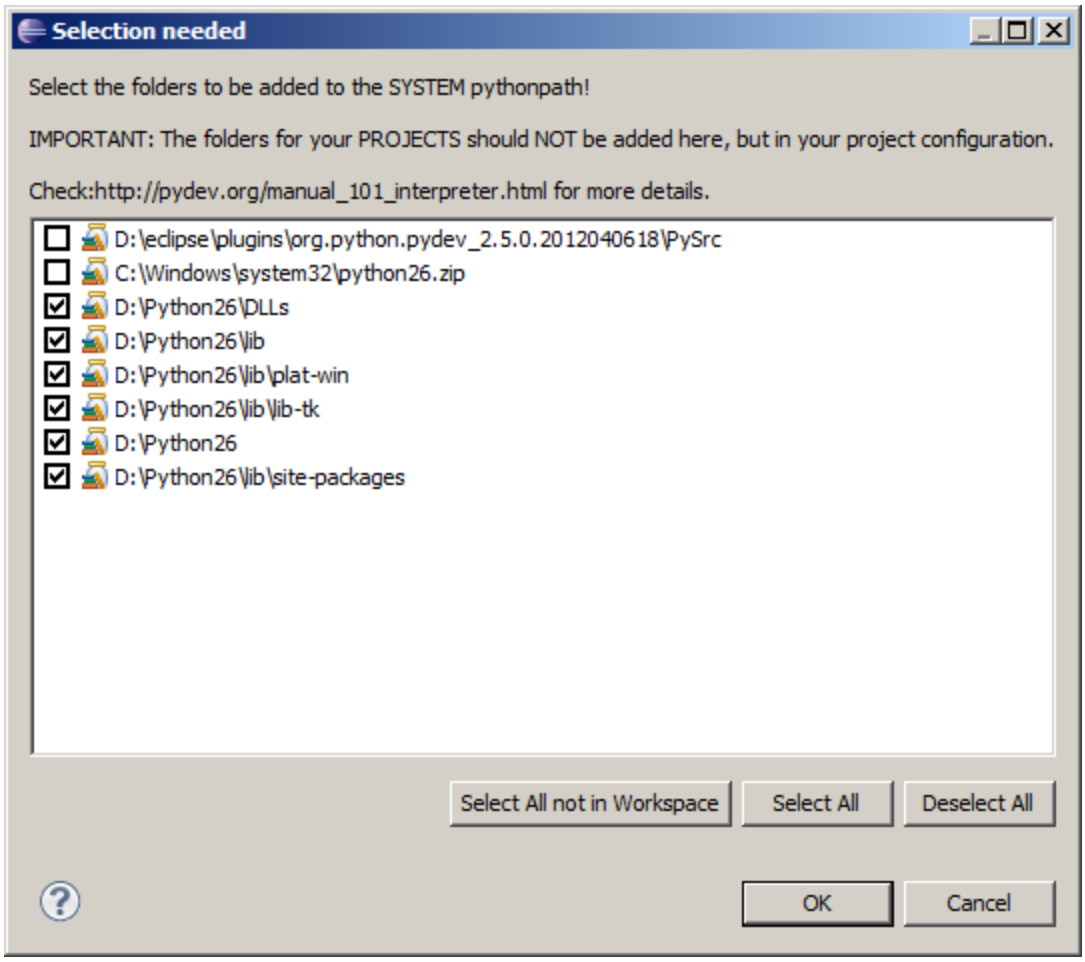

FIGURE 28 – SAMPLE PYTHON LIBRARIES TO BE IMPORTED (YOUR CASE MIGHT BE WITH MORE ITEMS)

2. That's it, press Ok and exit from here.

Written by Pir Khurram Rashdi

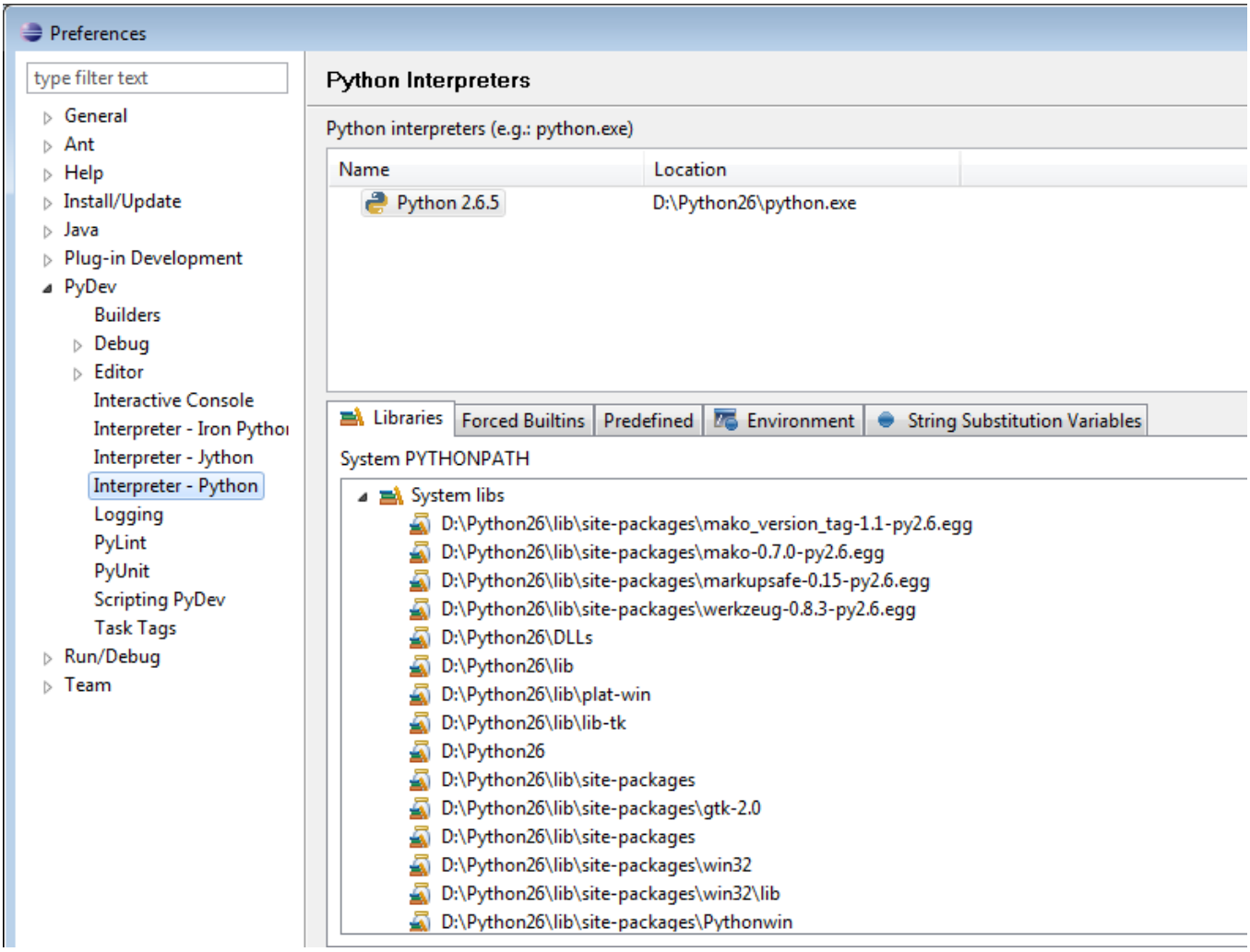

FIGURE 29 – AFTER PYTHON 2.6 GETS DEFINED, LIBRARIES ARE ALSO IMPORTED

Written by Pir Khurram Rashdi

Step 11. DOWNLOAD SOURCE CODE FOR OPENERP SERVER AND CLIENT

Then download and unzip tarballs (zipped form of source code) for OpenERP 6.1 server, Client and Web-client using the following paths. You can use WinRAR to unzip all these packages to separate folders on same path like Client, All-in-One (Server + Webclient) folders in D:\OpenERP path:-

- http://nightly.openerp.com/6.1/releases/openerp-6.1-1.tar.gz (All-in-one)
- http://nightly.openerp.com/6.1/releases/openerp-client-6.1-1.tar.gz (Rich client)

Step 12. PYDEV PROJECTS CREATION FOR OPENERP SERVER AND CLIENT

Now you need to create PyDev projects in Eclipse for two projects i.e. All-in-one and Client. You have to name the PyDev project e.g. **OpenERPClient** and **OpenERPServer** and for each you've to uncheck the option **Use Default** and browse for the path where you've unzipped each of the packages. You've to choose **project type as Python** and its **grammar version as 2.6** and **Interpreter as Python 2.6.5**. Also, uncheck the last option of creating src folder, we don't need that. Take help from snapshot below.

Note: You'll need to switch PyDev perspective when asked after you've created such project, and will also need to check the option to open this perspective whenever PyDev project is created or opened.

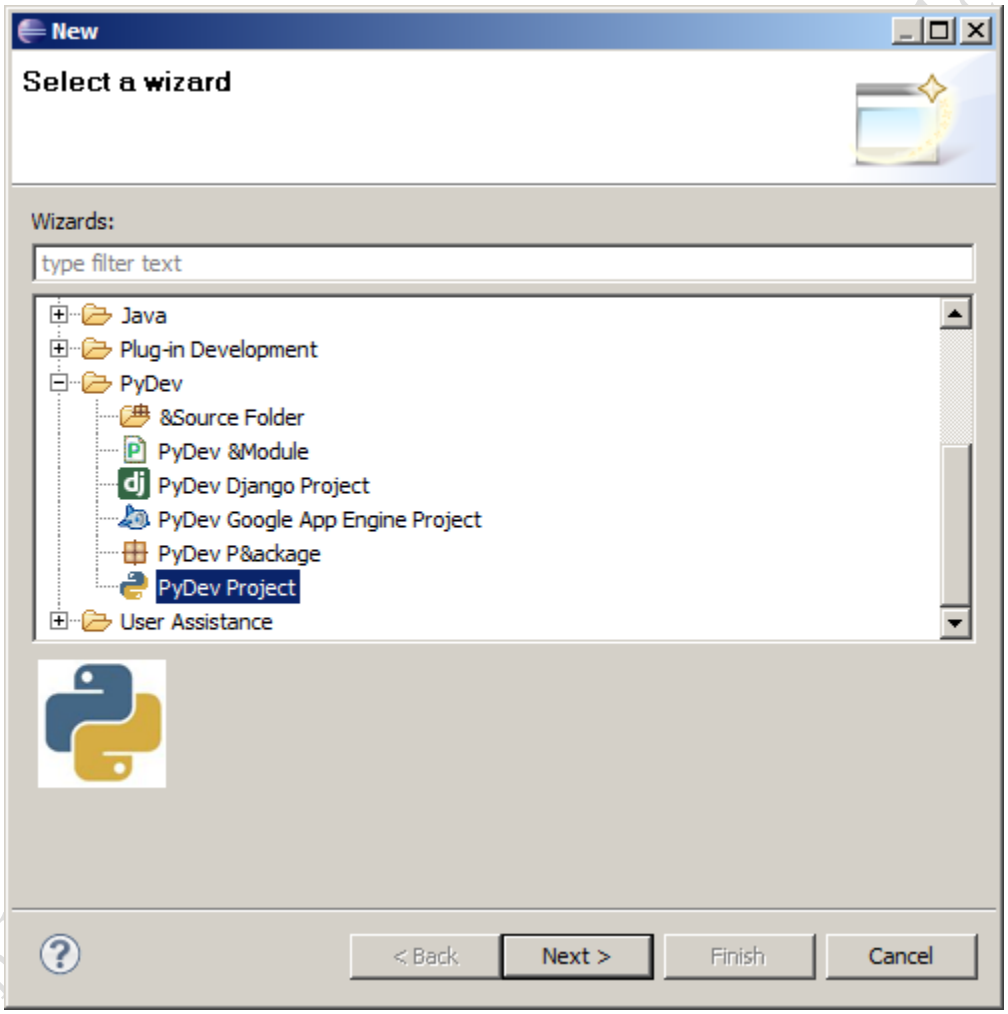

FIGURE 30 – CREATE NEW PYDEV PROJECT

Written ... .com

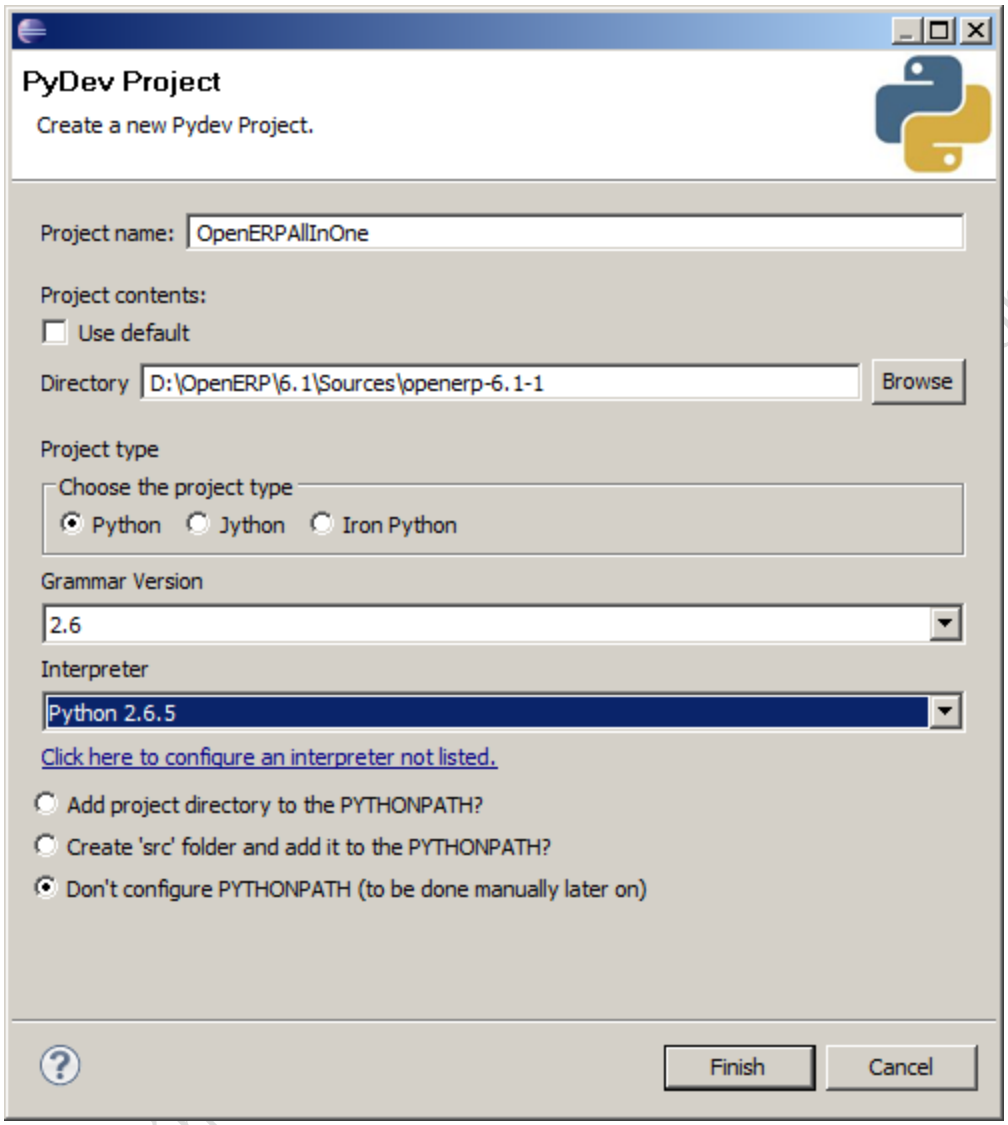

FIGURE 31 – PYDEV PROJECT CONFIGURATION

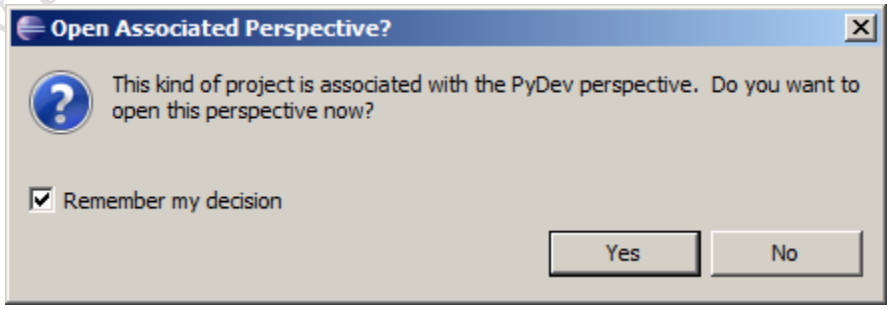

FIGURE 32 – SELECTION OF PYDEV PERSPECTIVE

Press finish immediately after this. You've to follow same procedure for all two projects. Finally, picture should resemble the snapshots as also shown below.

Written by Pir Khurram Rashdi

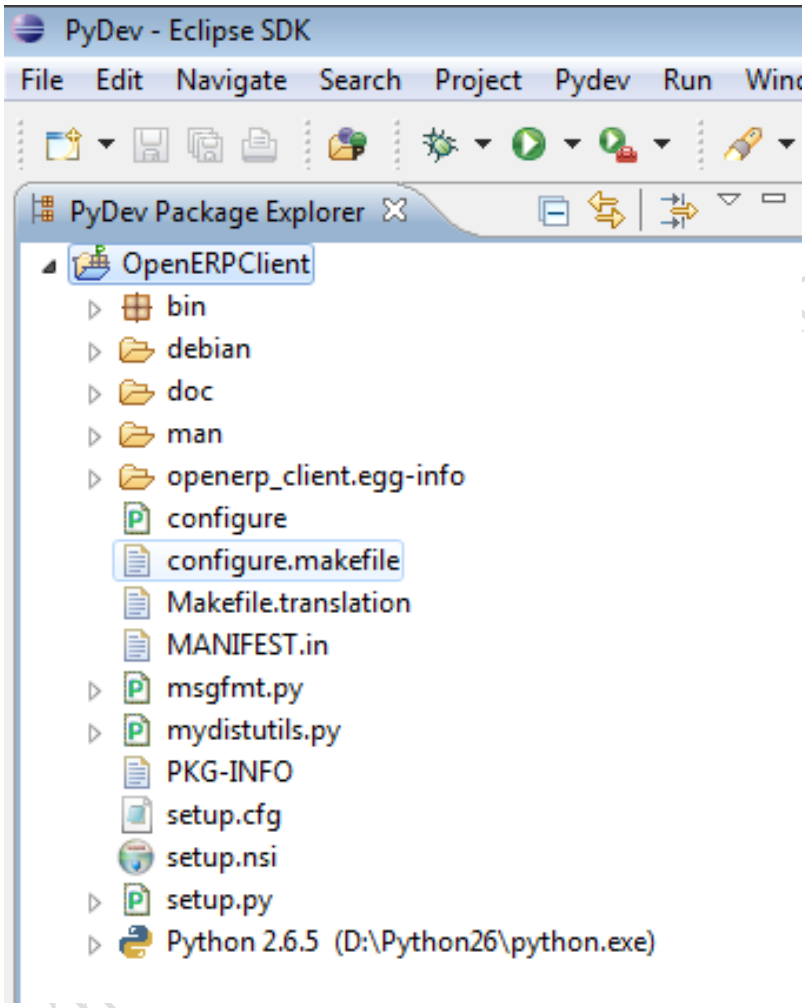

FIGURE 33 – PACKAGE EXPLORER VIEW SHOWING OPENERP CLIENT PROJECT

Written by Pir Kh ail.com

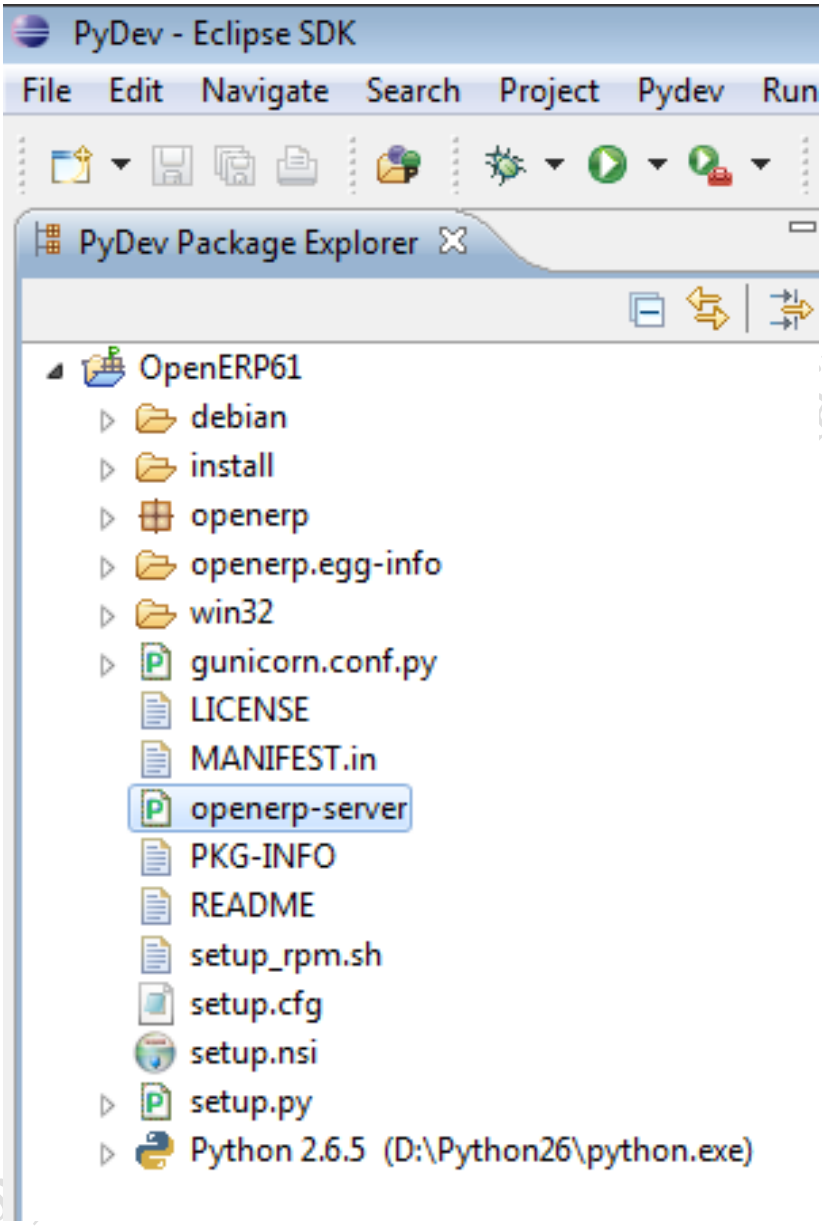

FIGURE 34 – PACKAGE EXPLORER SHOWING OPENERP SERVER (ALL-IN-ONE WITH SERVER + WEB) PROJECT

Step 13. DOWNLOAD AND INSTALL OPENERP ALL-IN-ONE FOR CONFIGURATION FILE

You will need to download and install OpenERP's All-in-one package for windows for the reason that we'll need a Database connection configuration over-rides setting file located in the path like:

C:\Program Files (x86)\OpenERP 6.1-1\server\openerp-server.conf

This file will need to be copied to the folder located in the path where we've unzipped server's source code e.g. **D:\OpenSource\OpenERP\6.1\Sources\openerp-6.1-1.**

The snapshot below will show an example **openerp-server.conf** configuration file. You can see that I've replaced all path constants to OpenERPServer's root path (source code one). You'll need same changes in your case too. The main items are **logfile**, **addons_path**, and **pg_path** to assign path strings to OpenERP server's source code path and PostgreSQL installed version's bin path.

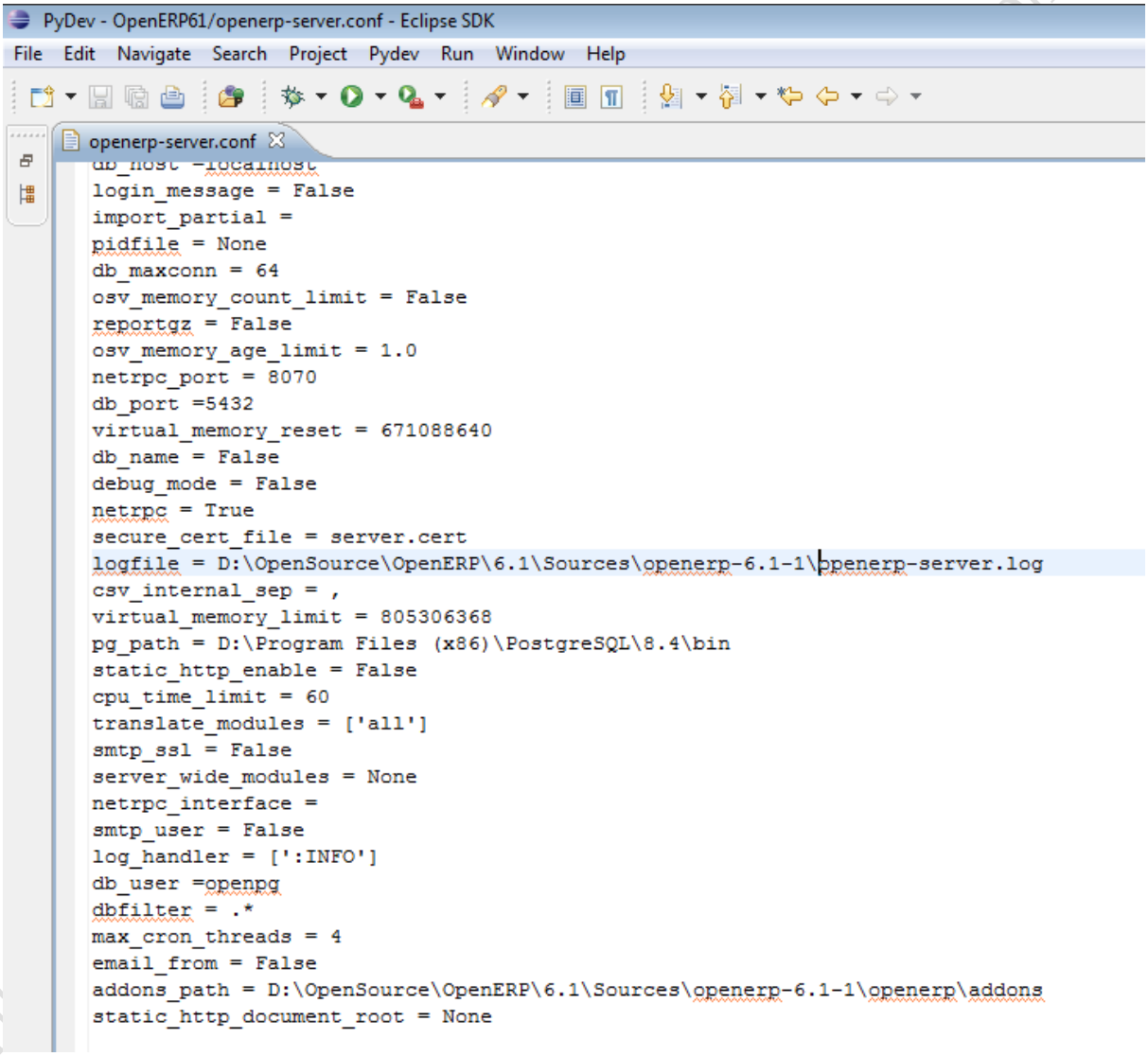

FIGURE 35 – VIEW OF OPENER-SERVER.CONF CONFIGURATION FILE ALL PATH CONSTANTS CHANGED

Download path to all-in-one windows installer is:-
http://nightly.openerp.com/6.1/releases/openerp-allinone-setup-6.1-1.exe

Note: When you've installed OpenERP all-in-one package, please go ahead and create a database with demonstration data loaded. This is required so that **opener-server.conf** get's written correctly with accurate **postgresql** login information. Plus, a database in **PostgreSQL** is created which can be tested later when we'll run the code through Eclipse. Don't worry, we'll stop both services i.e. OpenERP Server before running code through Eclipse so that these services plus desktop client should be running via Eclipse.

Step 14. Checking the PyDev Projects for OpenERP for Working

The last step is to check the two individual projects and included dependencies. If you face compilation errors, then go through once again to check for the missing steps; otherwise, write to me if anything is missed in this tutorial.

a. Check windows services using services.msc command in Run command and then stop service i.e. OpenERP server if it is **Started**.

b. First step is to start the Server using Eclipse so locate **openerp-server.py** (without .py file extension) file in main root folder while in Eclipse's package explorer, right-click when found and Run as Python Run. If it give compile time import error it means you've missed some dependency; otherwise, if success, it will give no message.

c. While All-in-one or OpenERP server project is running perfectly (without any messages in debugger in Eclipse), you'll have an additional step of launching web browser and giving the URL to access OpenERP web which is running in background now. In my case, the URL was http://localhost:8069 .

d. Please consult the snapshots below for understanding:-

Written by Pir Khurram Rashdi Email: pkrashdi@gmail.com

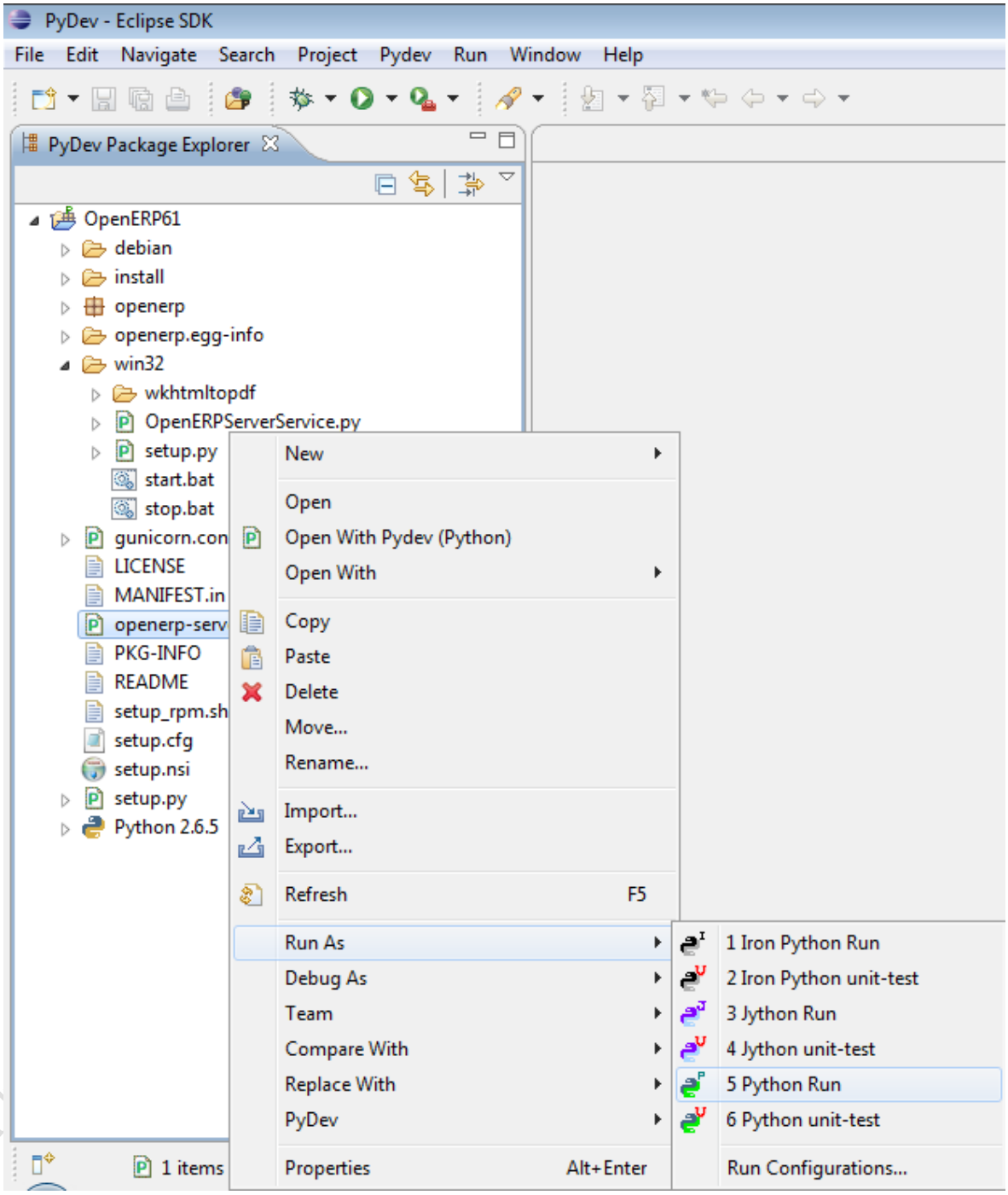

FIGURE 36 – RUNNING OPENERP SERVER (ALSO WEB) VIA ECLIPSE IDE AS PYTHON RUN

Written

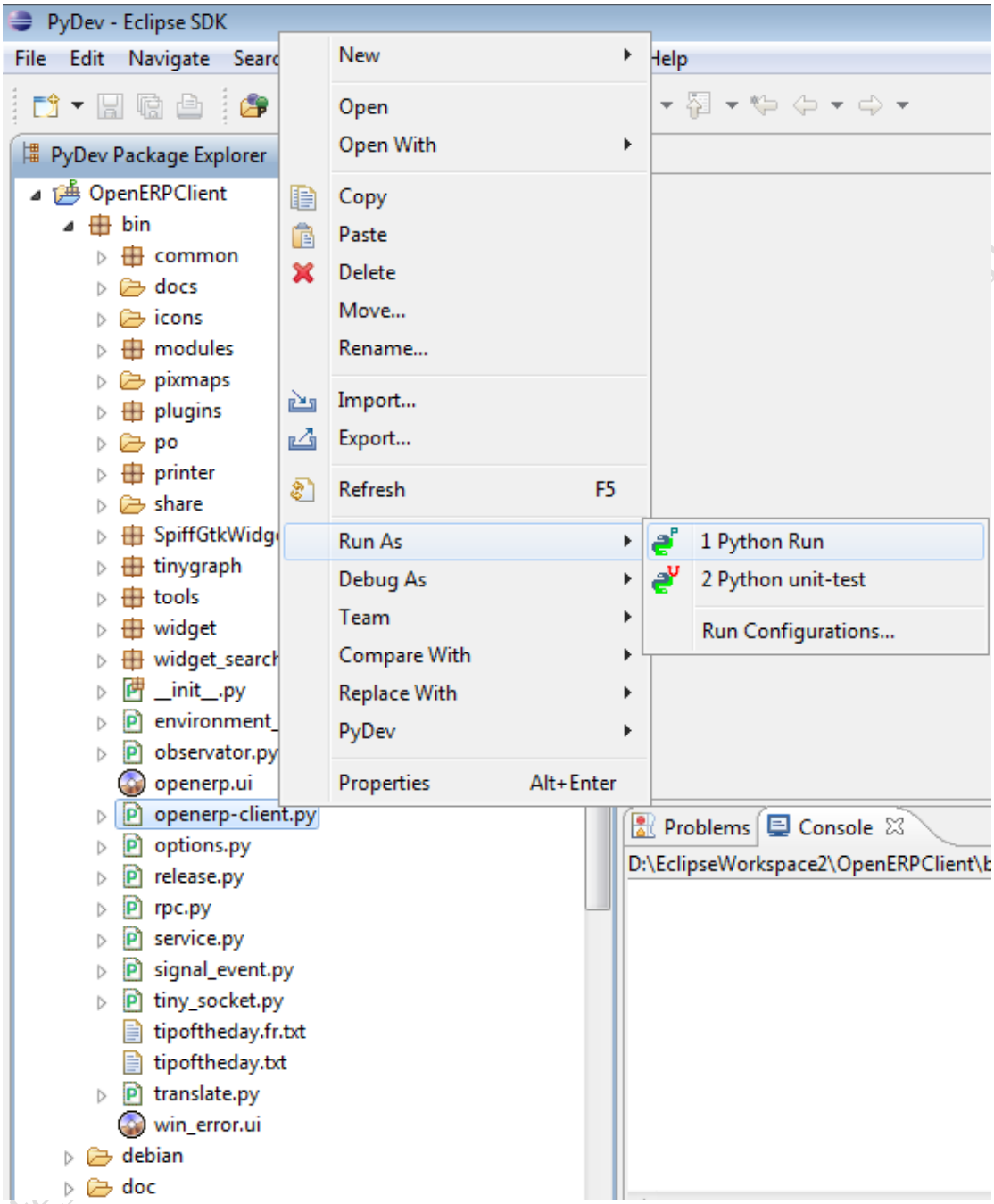

FIGURE 37 – RUNNING OPENERP 6.1 CLIENT VIA ECLIPSE IDE AS PYTHON RUN

Written

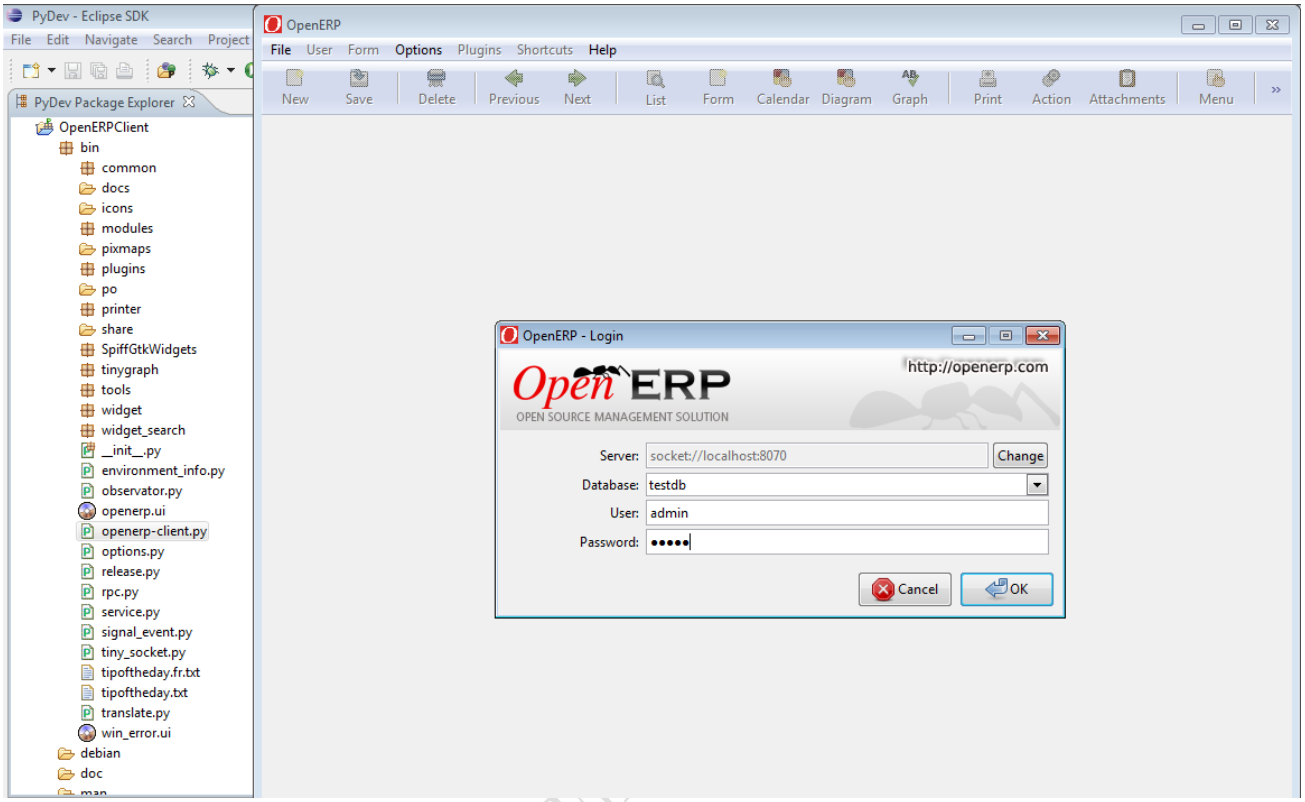

FIGURE 38 – OPENERP 6.1 CLIENT LAUNCHES VIA ECLIPSE

Written by Pir Khurram Rashdi

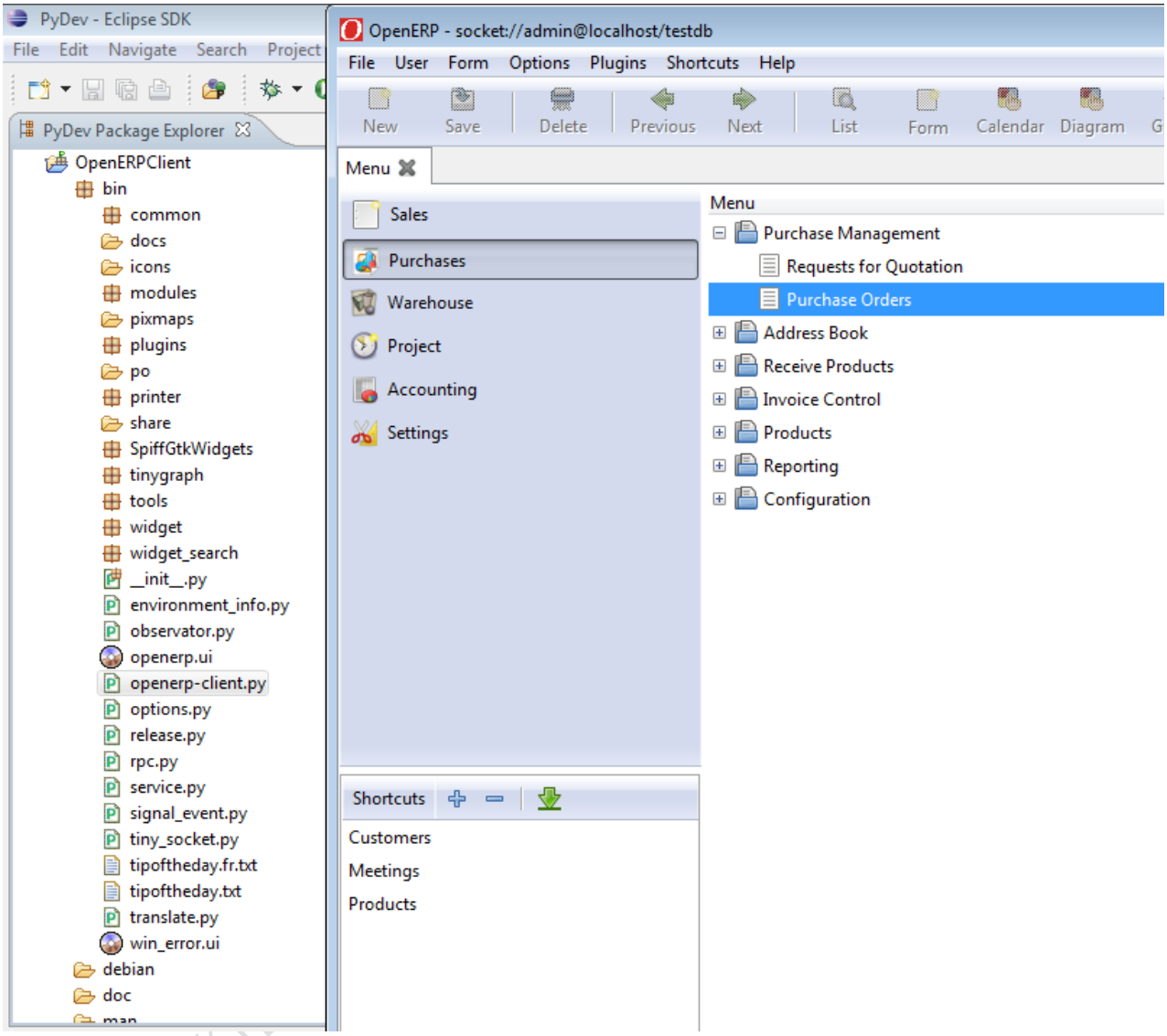

FIGURE 39 – OPENERP 6.1 CLIENT WHEN LOGGED-IN RUNNING VIA ECLIPSE IDE AND CONNECTED TO OPENERP SERVER ALSO RUNNING VIA ECLIPSE

Written by Pir Kh

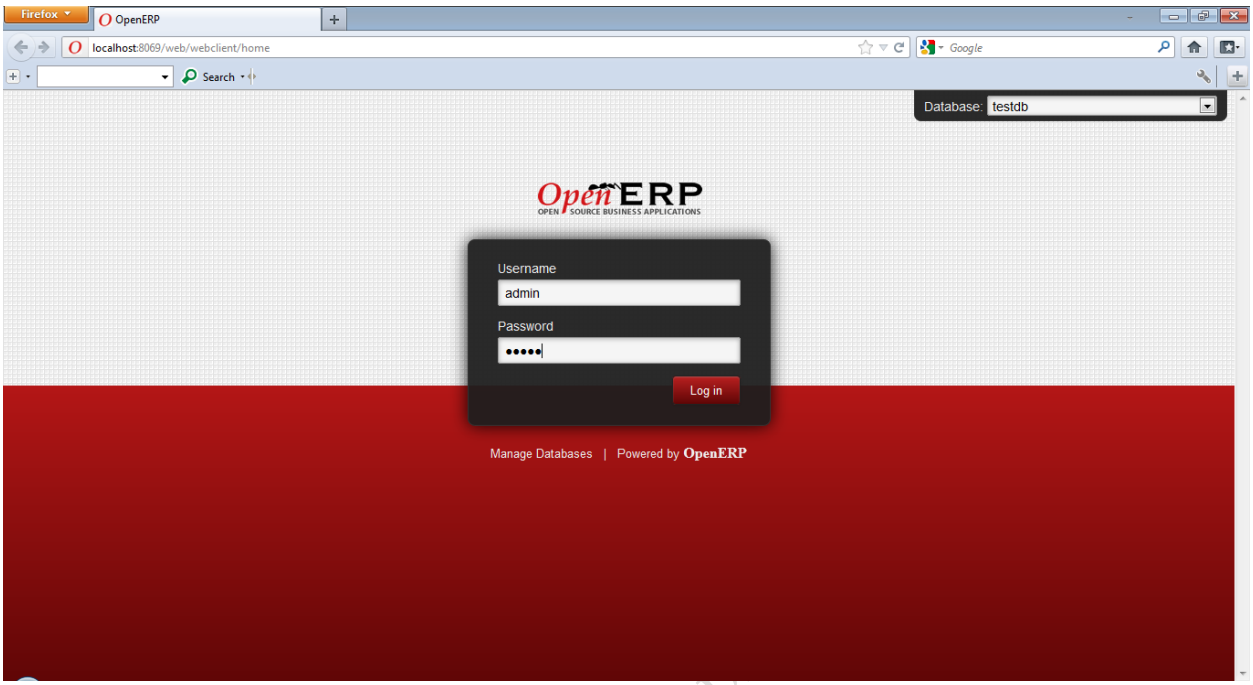

FIGURE 40 – CHECKING WEB CLIENT RUNNING VIA ECLIPSE THROUGH WEB BROWSER ON PORT 8069

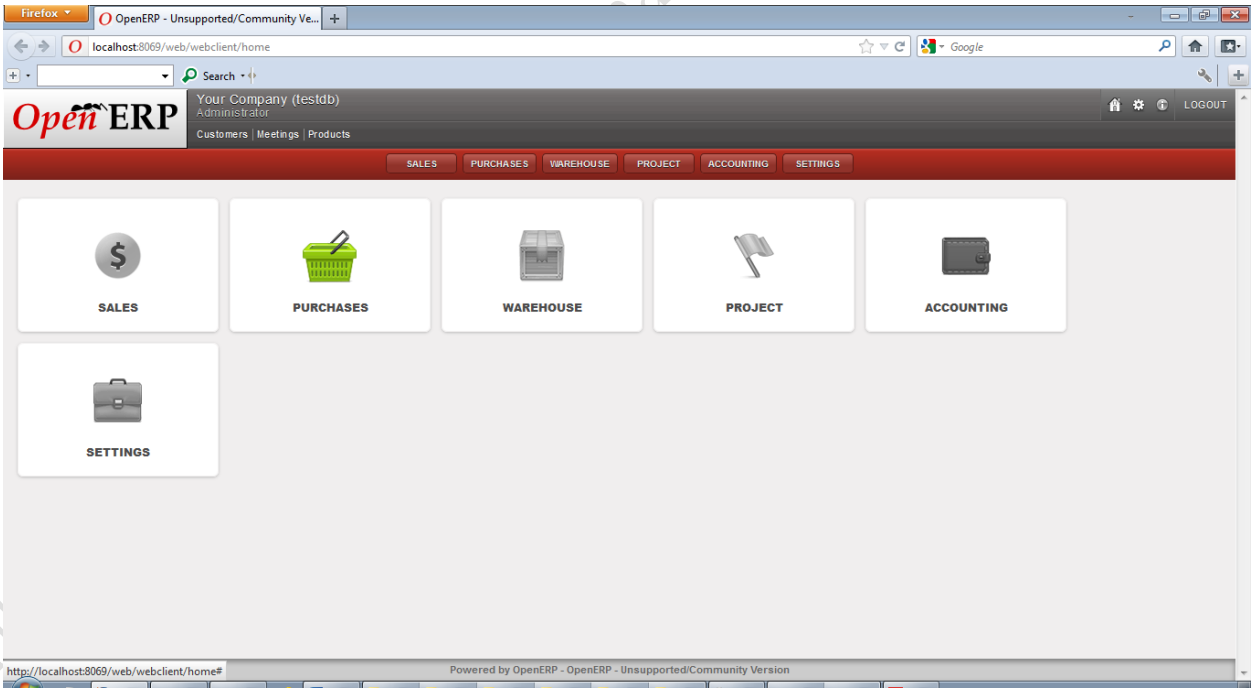

FIGURE 41 – CHECKING WEB CLIENT RUNNING VIA ECLIPSE THROUGH WEB BROWSER ON PORT 8069