Sikuli Workbook(2012)

Overview

- Install JRE6 i586(**Win32**)
  - http://www.oracle.com/technetwork/java/javase/downloads/jre-6u32-downloads-1594646.html
  - http://stackoverflow.com/questions/5272216/is-it-possible-to-install-both-32bit-and-64bit-java-on-windows-7
  - **Installation Path for EnvVars:** C:\Program Files (x86)\Java\jre6
- Get Sikuli IDE
  - http://www.sikuli.org/download.shtml#win
  - use zip and put into “\lib”
  - D:\lib\Sikuli-X-1.0rc3 (r905)-win32\Sikuli-IDE
  - Run with “Sikuli-IDE.bat”
- Get PyDev for Eclipse
- Get Jython Download:
  - http://sourceforge.net/projects/jython/files/jython/2.5.2/jython_installer-2.5.2.jar/download
  - http://wiki.python.org/jython/InstallationInstructions
  -
-

Install JRE6(i586)

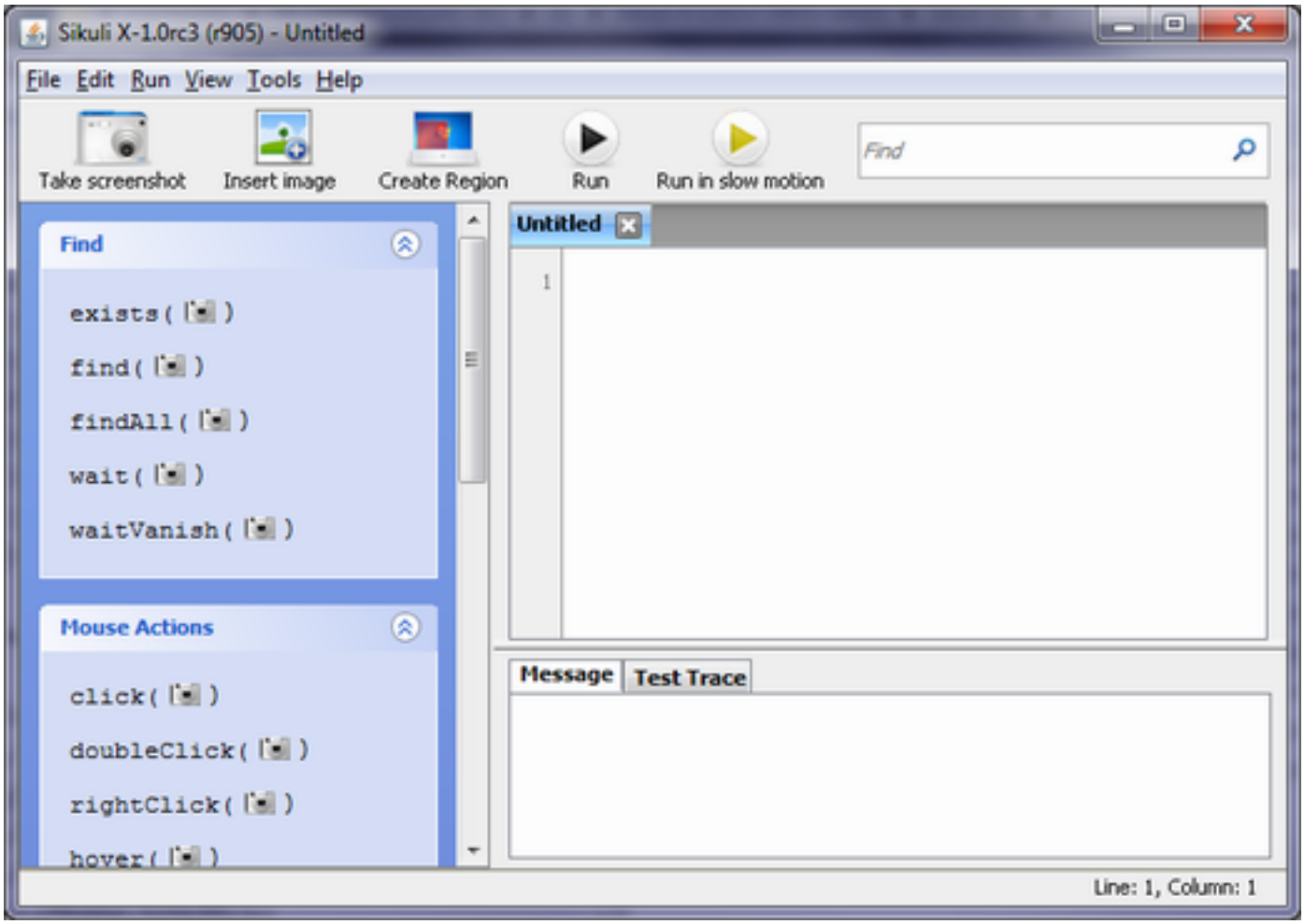
Note for 64bit Windows: Please download a **32bit Java RE** to run Sikuli.

- http://www.oracle.com/technetwork/java/javase/downloads/jre-6u32-downloads-1594646.html
- http://stackoverflow.com/questions/5272216/is-it-possible-to-install-both-32bit-and-64bit-java-on-windows-7
  - You can install multiple Java runtimes under Windows (including Windows 7) as long as each is in their own directory.
  - For example, if you are running Win 7 64-bit, or Win Server 2008 R2, you may install 32-bit JRE in "C:\Program Files (x86)\Java\jre6" and 64-bit JRE in "C:\Program Files\Java\jre6", and perhaps IBM Java 6 in "C:\Program Files (x86)\IBM\Java60\jre".
  - The Java Control Panel app theoretically has the ability to manage multiple runtimes: Java tab >> View... button
  - There are tabs for User and System settings. You can add additional runtimes with Add or Find, but once you have finished adding runtimes and hit OK, you have to hit Apply in the main Java tab frame, which is not as obvious as it could be - otherwise your changes will be lost.

- If you have multiple versions installed, only the main version will auto-update. I have not found a solution to this apart from the weak workaround of manually updating whenever I see an auto-update, so I'd love to know if anyone has a fix for that.
- Most Java IDEs allow you to select any Java runtime on your machine to build against, but if not using an IDE, you can easily manage this using environment variables in a cmd window. Your PATH and the JAVA_HOME variable determine which runtime is used by tools run from the shell. Set the JAVA_HOME to the jre directory you want and put the bin directory into your path (and remove references to other runtimes) - with IBM you may need to add multiple bin directories. This is pretty much all the set up that the default system Java does. You can also set CLASSPATH, ANT_HOME, MAVEN_HOME, etc. to unique values to match your runtime.

Sikuli Installation

- Get ZIP package
- Copy to \lib
- Run with “Sikuli-IDE-w.bat”



Downloads

Windows

We provide a self-extracting installer and a portable version for Windows.

Prerequisite

Make sure you have Java Runtime Environment (JRE) 6 installed. If you see an error when launching Sikuli like "java.language.UnsatisfiedLinkError: ..... Win32Util.dll .....", please read the solution here.

- Setup Sikuli Envorinment on 64bit OS
  - https://answers.launchpad.net/sikuli/+question/140519
- Library Errors Encountered
  - https://answers.launchpad.net/sikuli/+question/140519

Note for Sikuli 0.10 users: Please uninstall Sikuli 0.10 before installing Sikuli X.
Latest version: X 1.0 rc3 (r930)

- Sikuli-X-1.0rc3 (r905)-win32.exe (Self-extracting installer. For 32bit & 64bit Windows.)
- Sikuli-X-1.0rc3 (r905)-win32.zip (Portable version for 32bit Windows. For 64bit Windows, please run the Sikuli-IDE-w.bat instead of Sikuli-IDE.exe.)
- Sikuli X r930 (Portable version for 32bit Windows. For 64bit Windows, please run the Sikuli-IDE-w.bat instead of Sikuli-IDE.exe.)

Linux

Linux version is a portable zip, no installation required. Just run sikuli-ide.sh in Sikuli-IDE/ to start it.
Make sure you have Java Runtime Environment (JRE) 6, wmctrl, and OpenCV 2.1 (libcv2.1, libcvaux2.1, libhighgui2.1) installed.

- Sikuli-X-1.0rc3 (r905)-linux-i686.zip (For 32bit Linux)
- Sikuli X r931 (For 32bit Linux)
- Sikuli-X-1.0rc3 (r905)-linux-x86_64.zip (For 64bit Linux)

Command Line Tool

Sikuli IDE can be used under command line to run a Sikuli script or a Sikuli test case. It is a little bit different to invoke the Sikuli IDE under command line on each platform.
Please read the command line usage in our documentation.

Source Code

The source code is hosted on github. You can check out the development tree using Git or download the zipped source at github.
If you would like to send patches, please read How to get involved.

Jython Installation

Contents

Jython 2.5.2 and 2.2.x

Basic Install

Jython 2.5.2 and 2.2.1 are distributed as executable jar file installers. After downloading it, either double click the jython_installer-2.5.2.jar or run java with the -jar option
java -jar jython_installer-2.5.2.jar
This will start the regular GUI installer on most systems, or a console installer on headless systems. To force the installer to work in headless mode invoke the installer as:
java -jar jython_installer-2.5.2.jar --console
The installer will then walk through a similar set of steps in graphical or console mode: showing the license, selecting an install directory and JVM and actually copying Jython to the filesystem. After this completes, Jython is installed in the directory you selected. Executing a script in the install directory, jython on Unix-like systems or jython.bat on Windows, will start up the Jython console, which can be used to dynamically explore Jython and the Java runtime, or to run Jython scripts.

Standalone mode

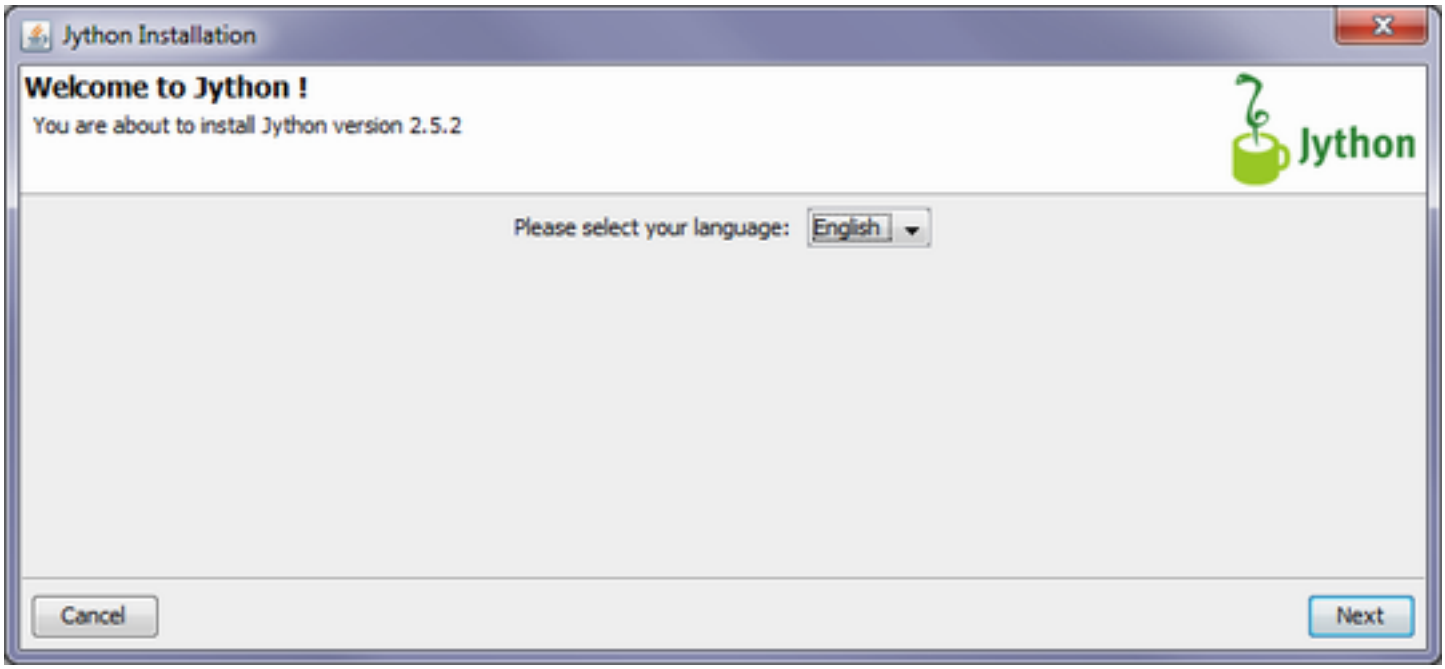
The standalone option does no caching and so avoids the startup overhead (most likely at the cost of some speed in calling Java classes, but I have not profiled it)
You can try it out by running the installer:
$ java -jar jython_installer-2.5.2.jar
then when you come to the "Installation type" page, select "Standalone".
The installation will generate a jython.jar with the Python standard library (/Lib) files included, which can be run as:
$ java -jar jython.jar
Of course you can run scripts just by calling them as you might expect:
$ java -jar jython.jar script.py

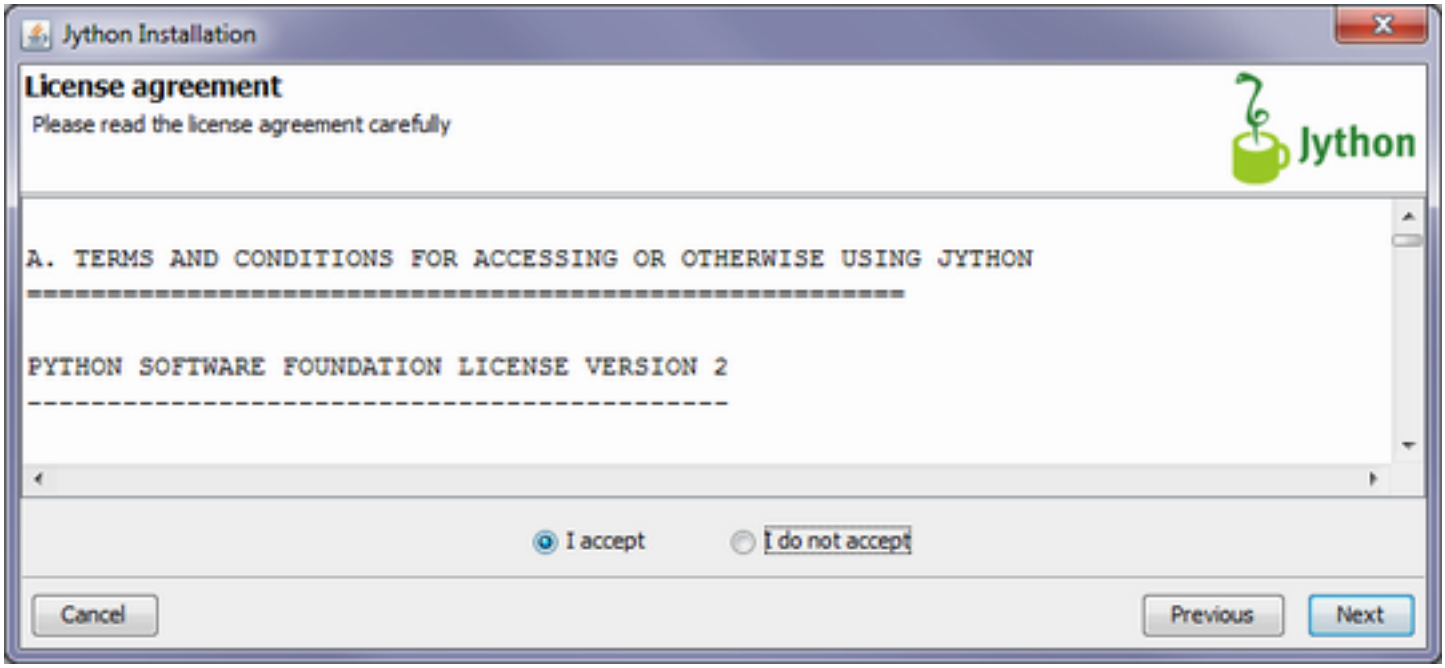
Or, add this file to the classpath of your application.

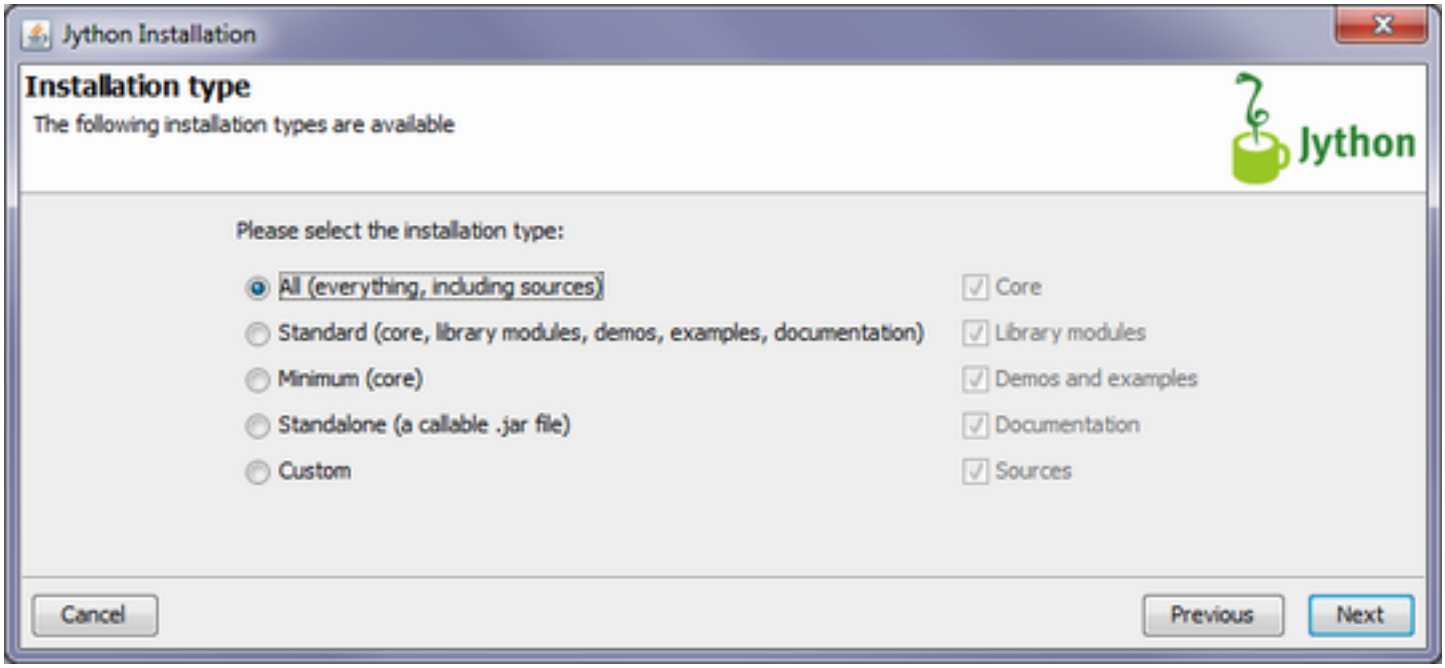
Installation options

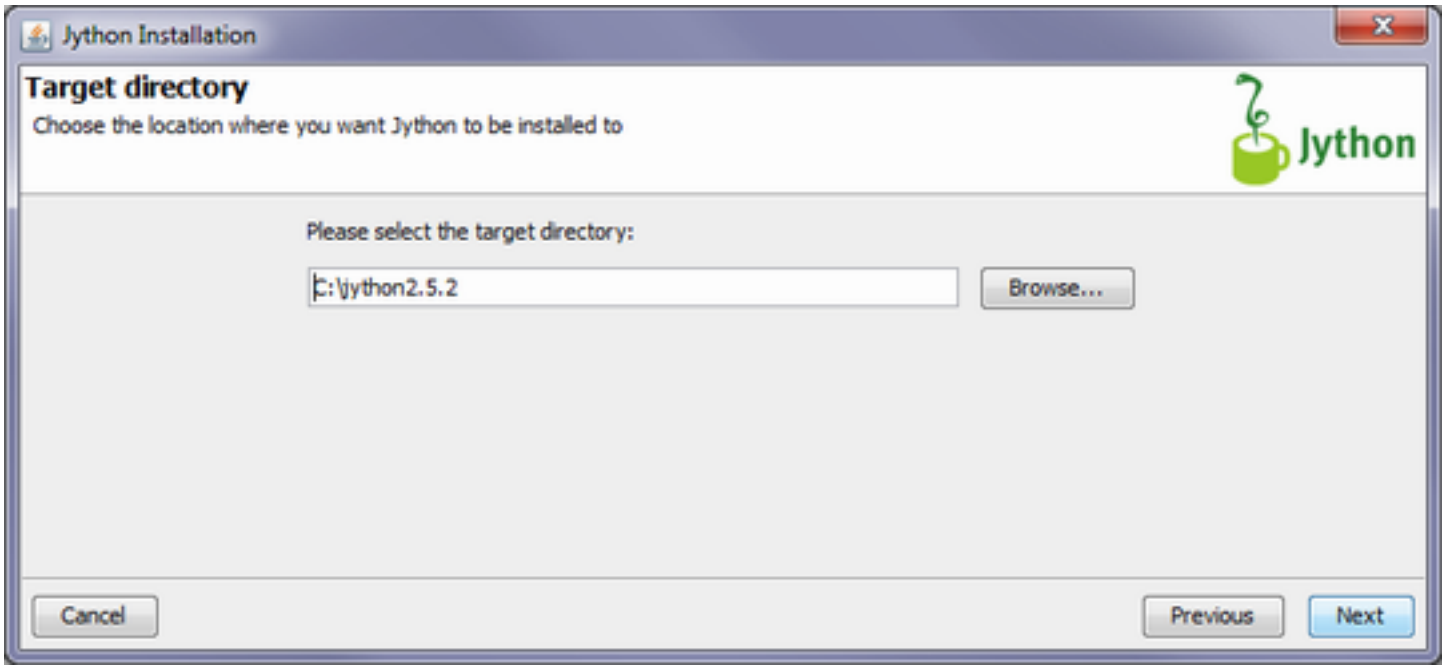
You can get a list of installer options (to install Jython unattended, for example) by running:

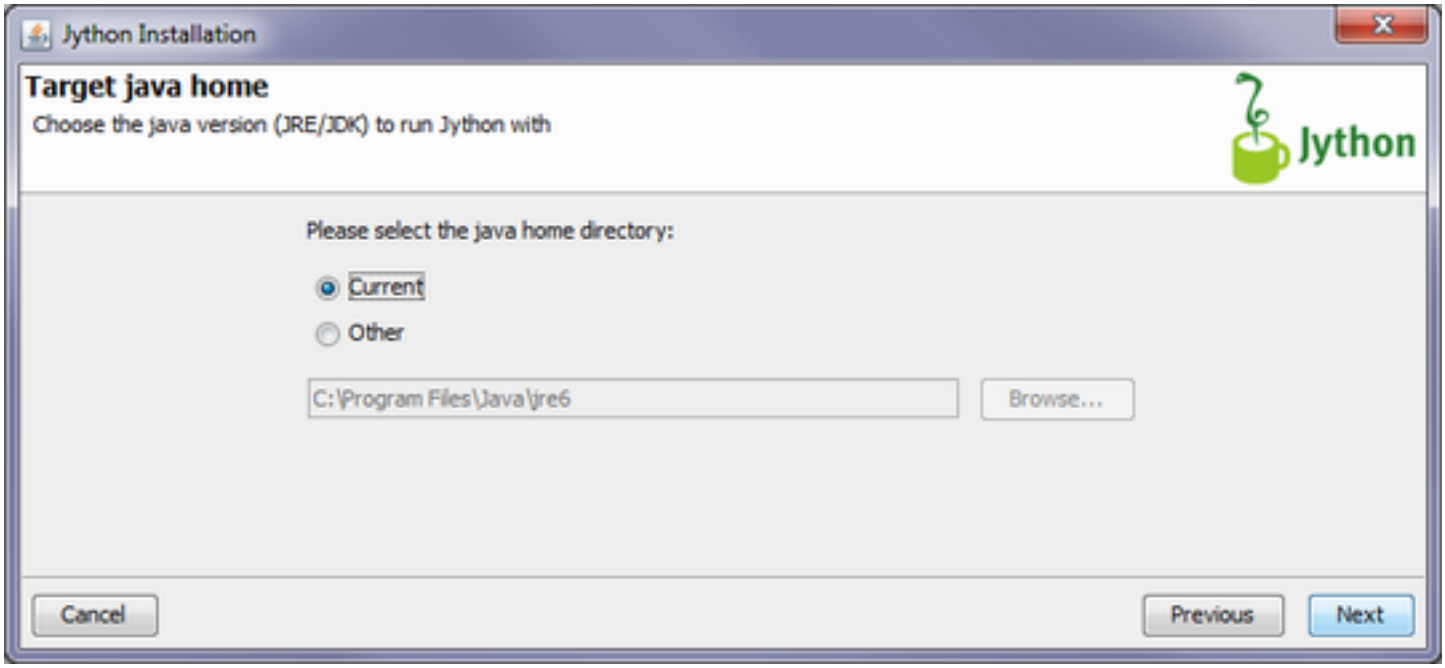
```
$ java -jar jython_installer-2.5.2.jar --help
```

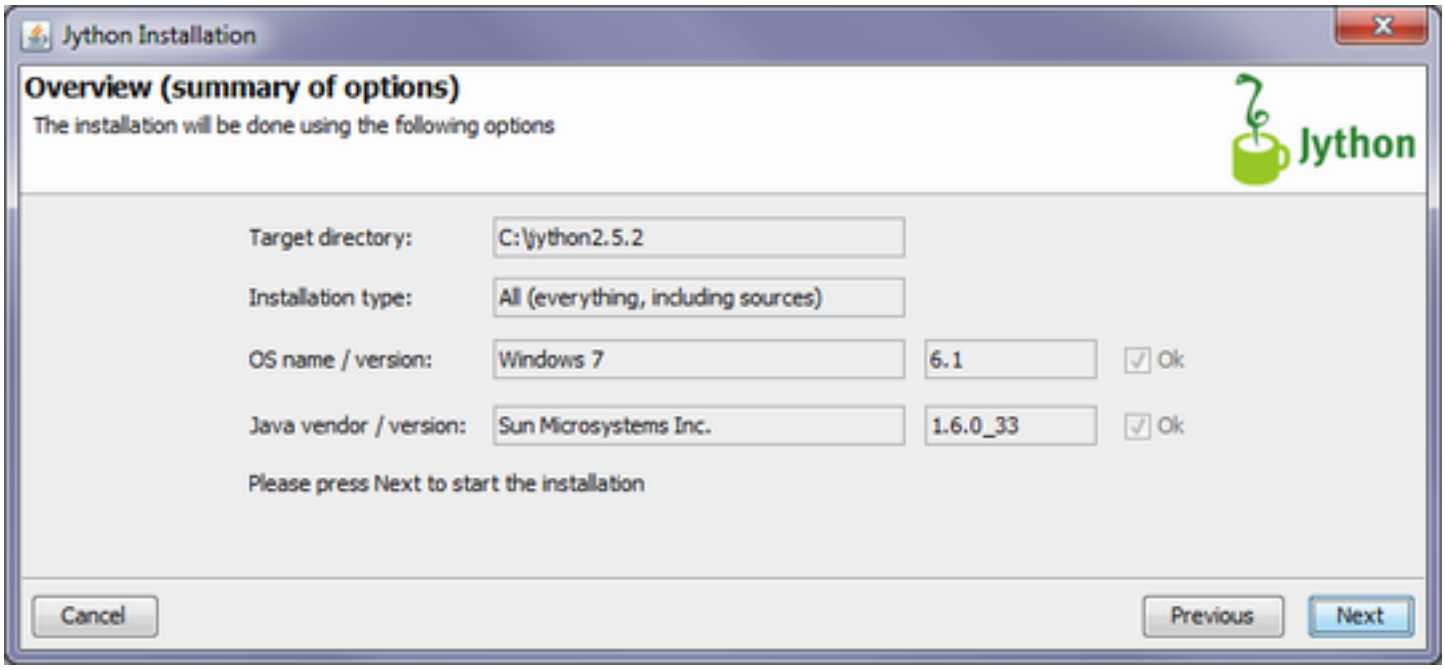


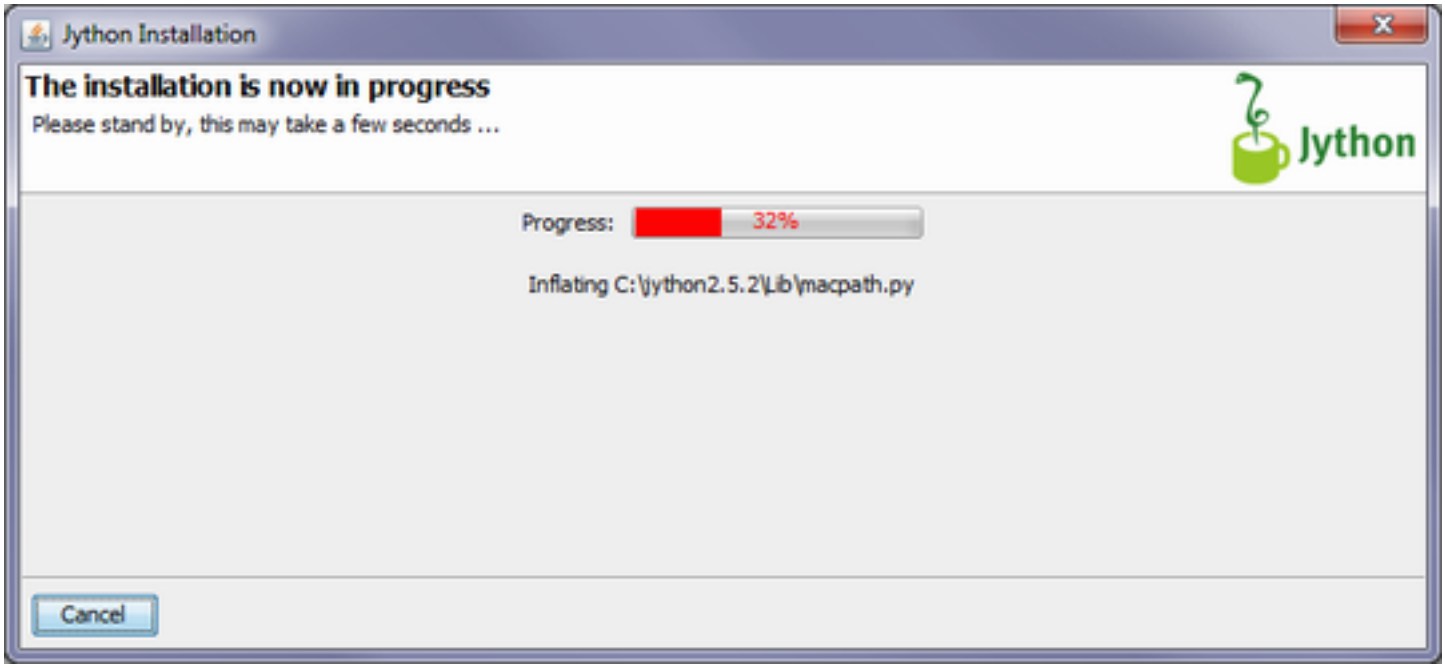


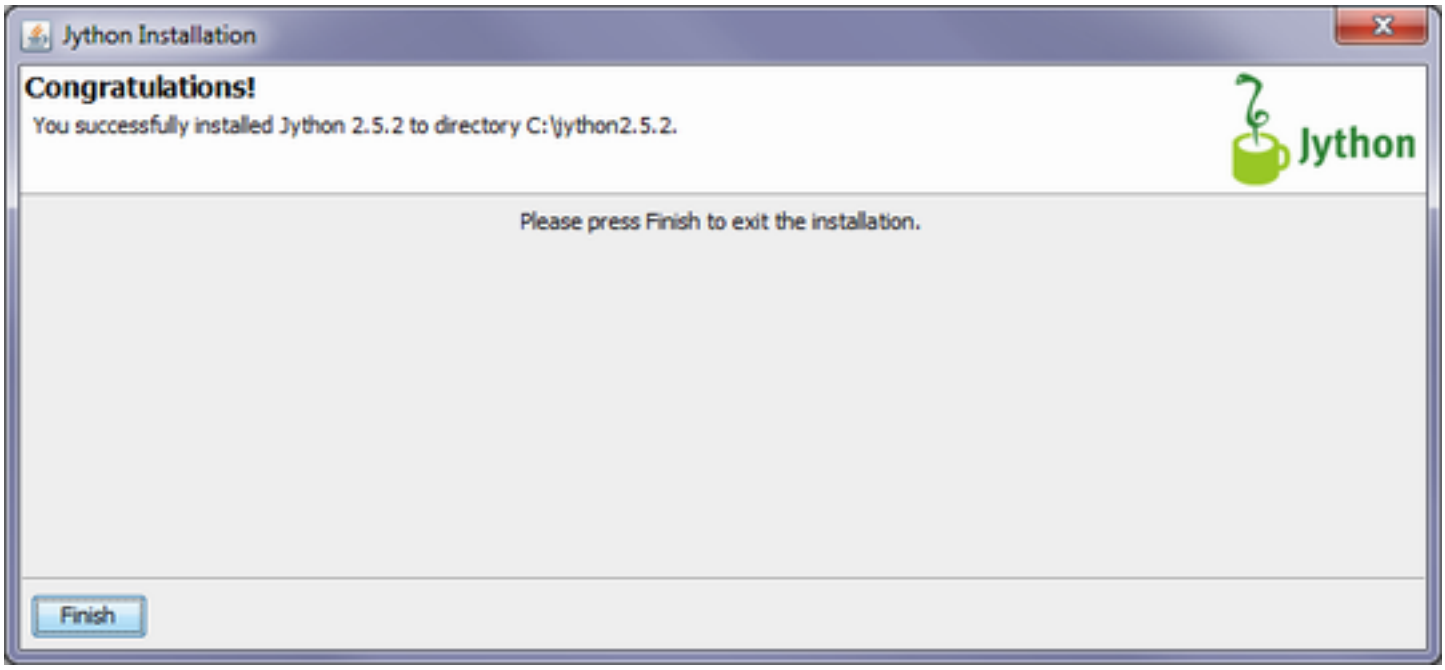

Jython Installation
Installation type
The following installation types are available
Jython
Please select the installation type:
All (everything, including sources)
Standard (core, library modules, demos, examples, documentation)
Minimum (core)
Standalone (a callable .jar file)
Custom
Core
Library modules
Demos and examples
Documentation
Sources
Cancel
Previous
Next


Jython Installation
Target directory
Choose the location where you want Jython to be installed to
Jython
Please select the target directory:
C:\jython2.5.2
Browse...
Cancel
Previous
Next


Jython Installation
Target java home
Choose the java version (JRE/JDK) to run Jython with
Jython
Please select the java home directory:
Current
Other
C:\Program Files\Java\jre6
Browse...
Cancel
Previous
Next


Jython Installation
Overview (summary of options)
The installation will be done using the following options
Jython
Target directory:
C:\Jython2.5.2
Installation type:
All (everything, including sources)
OS name / version:
Windows 7
6.1
Ok
Java vendor / version:
Sun Microsystems Inc.
1.6.0_33
Ok
Please press Next to start the installation
Cancel
Previous
Next





Testing Installation

```
C:\jython2.5.2>jython.bat
*sys-package-mgr*: processing new jar, 'C:\jython2.5.2\jython.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files (x86)
\QuickTime\QTSystem\QTJava.zip'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre6\lib\resources.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre6\lib\rt.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre6\lib\jsse.jar'
```

```
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre6\lib\jce.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre6\lib\charsets.jar'
*sys-package-mgr*: processing new jar, 'C:\Program Files\Java\jre6\lib\ext\dnsns.jar'
*sys-package-mgr*: processing new jar, 'C:\Program
Files\Java\jre6\lib\ext\localedata.jar'
*sys-package-mgr*: processing new jar, 'C:\Program
Files\Java\jre6\lib\ext\sunjce_provider.jar'
Jython 2.5.2 (Release_2_5_2:7206, Mar 2 2011, 23:12:06)
[Java HotSpot(TM) 64-Bit Server VM (Sun Microsystems Inc.)] on java1.6.0_33
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- import os <enter>
- import sys <enter>
- for e in sys.path: print e <enter> <enter>
- type exit() <enter> (to leave interactive Jython)

```
>>> import os
>>> import sys
>>> for e in sys.path: print e
...

C:\jython2.5.2\Lib
__classpath__
__pyclasspath__/
C:\jython2.5.2\Lib\site-packages
>>> exit()
```

Eclipse PyDev Installation/Config

Now we **install the PyDev plugin** from inside Eclipse: Menu **Help** -> **Eclipse Marketplace**. Either search it or find it on the Popular tab's lower part. Simply click the Install button and do what you are asked to do ;-)
The next step is to tell Eclipse PyDev, where it can find the Jython interpreter. Navigate to the *Eclipse Preferences* pane and open the category *PyDev* and inside the subcategory *Interpreter-Jython*. First try to *Auto Config* by clicking the appropriate button. If this does not work, click the button *New*, name the entry and navigate to the folder, where Jython was installed and select *jython.jar*. After clicking ok, a window *Selection needed* might come up: click *Select all* and *ok* to finalize this step.
Other options with PyDev are available, but not relevant for our actual matter (see documentation).

Configure for using Sikuli script features at runtime
Again we have a difference to Netbeans: The PyDev plugin does not allow library references to folders inside of jar-files in the respective configuration dialog (it does not insert them to the Python path), though Jython itself accepts them, when specified on the Python path. So if you want to run and debug your script in Eclipse, you have to extract the folder **Lib/sikuli** from sikuli-script.jar (*find one HowTo here*).
As with Netbeans, the library configuration is done in the project itself. So we open a new project:

- Menu **File** -> **New** -> **Project** -> **PyDev** -> (open sublist) -> **PyDev Project** -> click **Next** button.

On the configuration pane name your project, select Jython as Project type, grammar version 2.5 (higher Python language versions are not supported by Jython 2.5.x) and click the **Finish** button. Your project is created. Add at least one *you-name-it.py* file to the source folder and put *from sikuli.Sikuli import ** as the first line.
In the last step, we tell PyDev, where to find the Sikuli libraries.
Goto Menu **Project** -> **Properties** -> select category **PyDev - PYTHONPATH** and go to the tab **External Libraries**.
We need a reference to *path-to/sikuli-script.jar* and another one to the extracted folder *Lib* containing the folder *sikuli*.

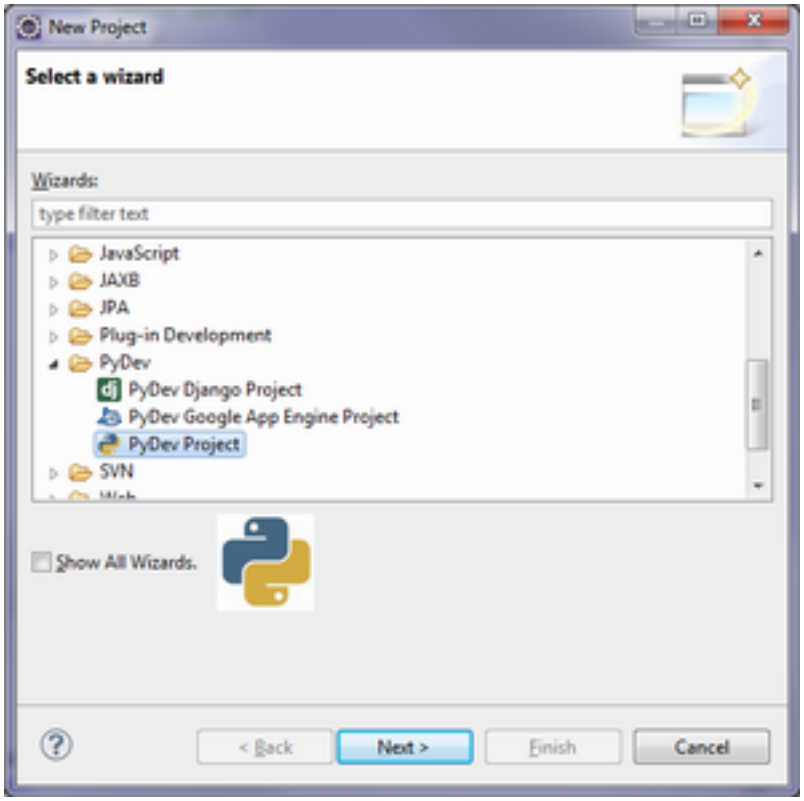
- reference to *path-to/sikuli-script.jar*
    - Windows: click button **Add zip/jar/egg** and select *sikuli-script.jar* from the Sikuli installation.
    - Mac: As with NetBeans, the file dialog does not allow to step inside Sikuli-IDE.app. So we use the trick, to define a *String Substitution Variable*: on the respective tab click **Add variable**, name it e.g. *sikuli-script* and enter as value: /Applications/Sikuli-IDE.app/Contents/Resources/Java/sikuli-script.jar. Go back to the tab **External Libraries** and click **Add based on variable**. In the entry field enter: ${sikuli-script} and click **OK**.
- reference to the extracted folder *Lib* containing the folder *sikuli*
    - click **Add source folder** and select the folder *Lib* in the place you had it extracted.
    - this is not needed, if you have moved the extracted folder *sikuli* to a folder, that is already on the Python path (e.g. jython-intallation/Lib/site-packages).

Now you are able to run your first script. Remember, that in every script including the main script, that you are editing now, as the first line you need *from sikuli.Sikuli import **, to have access to the Sikuli features at runtime.
Everytime later on you might come back to the project's preferences with **Project** -> **Properties**.
Code Completion works from the start without any further configuration and even steps into the Java classes where appropriate.

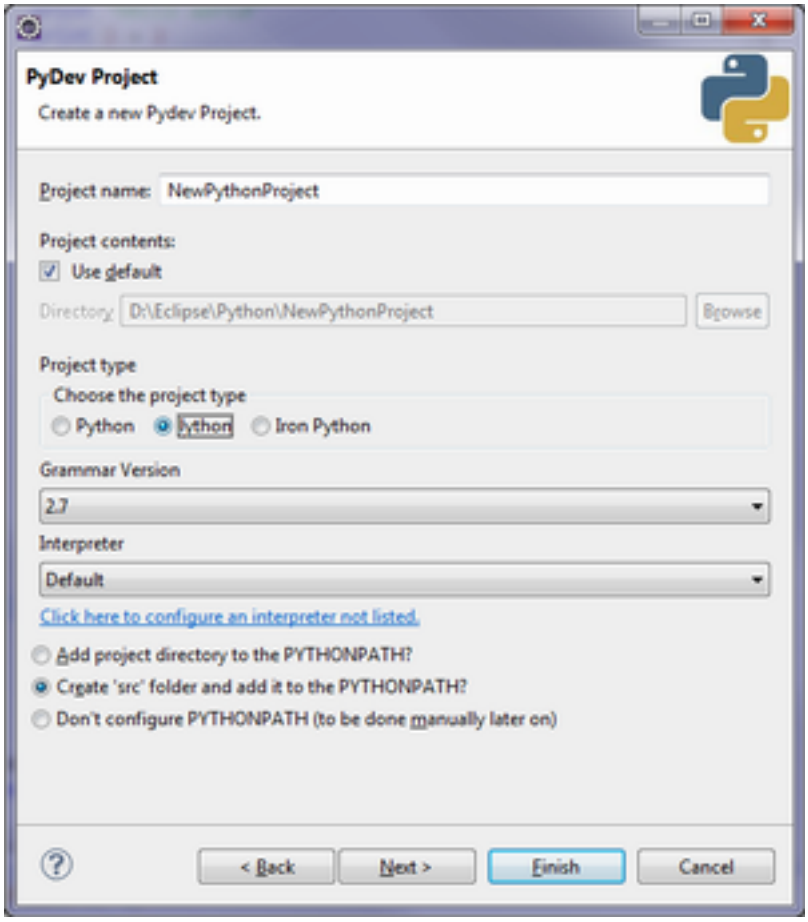
Configuring Eclipse

Open Python Workspace
New > PyDev Project
Jython Interpreter: D:\lib\jython2.5.2\jython.jar

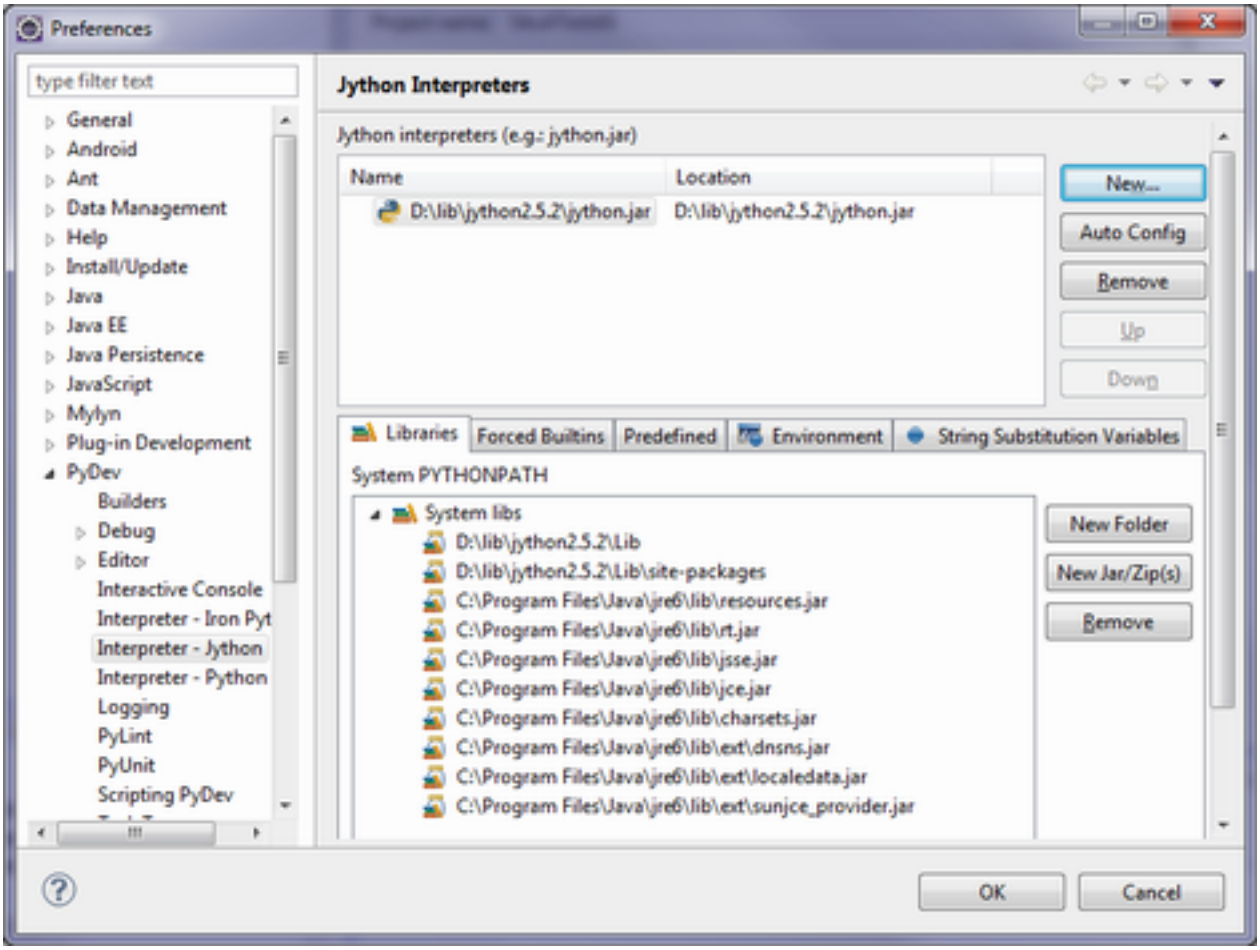
- **New > PyDev Project**



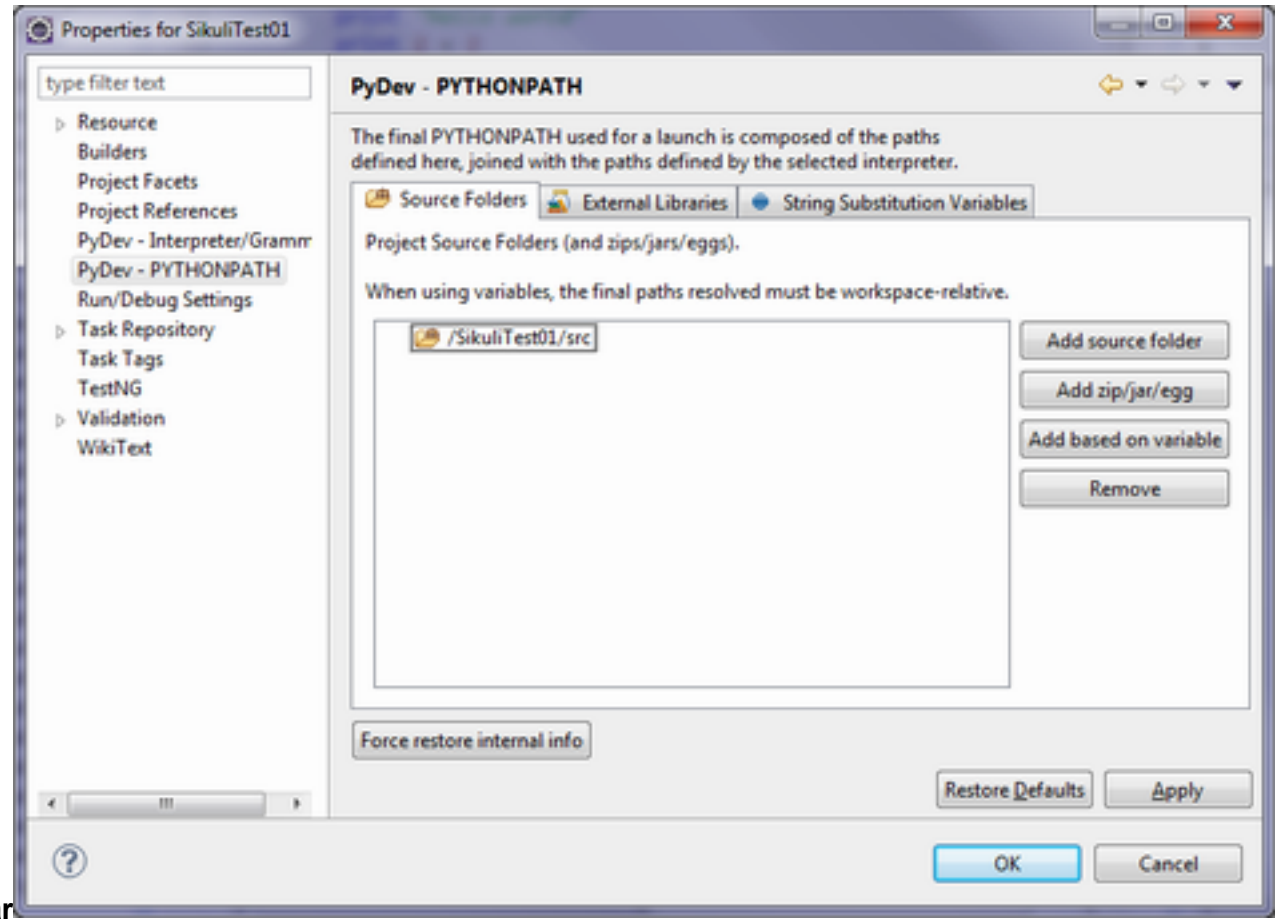
- **PyDev Project Settings(Jython)**



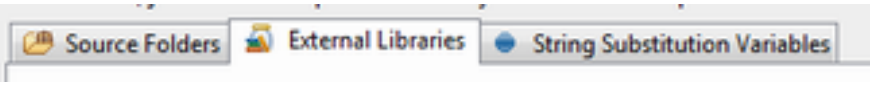
- **Jython Interpreter(jython.jar)**



- **Add sikuli-script.jar**

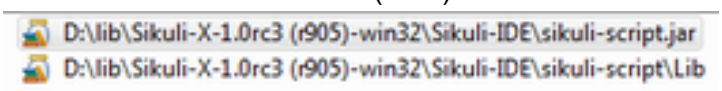


- **Go to External Libraries Tab (Add zip/jar/egg)**
- **Add sikuli-script.jar**
- **Extract '\Lib' from sikuli-script.jar, then add as 'source folder**



- D:\lib\Sikuli-X-1.0rc3 (r905)-win32\Sikuli-IDE\sikuli-script.jar

- D:\lib\Sikuli-X-1.0rc3 (r905)-win32\Sikuli-IDE\sikuli-script\Lib



- PyDev Projects for Sikuli, need to use the following import:
  - `from sikuli.Sikuli import *`

Run Script from Batch file until Execution resolved in Eclipse:

https://answers.launchpad.net/sikuli/+question/203300

```
set PATH=%PATH%;D:\lib\Sikuli32\Sikuli-IDE\libs;C:\Windows\SysWOW64\java.exe
set SIKULI_HOME=D:\lib\Sikuli32\Sikuli-IDE\
C:\Windows\SysWOW64\java.exe -cp "%SIKULI_HOME%sikuli-script.jar"
org.python.util.jython D:\Eclipse\Sikuli\SikuliTest01\src\SikuliVolumeControlTest01.py
```

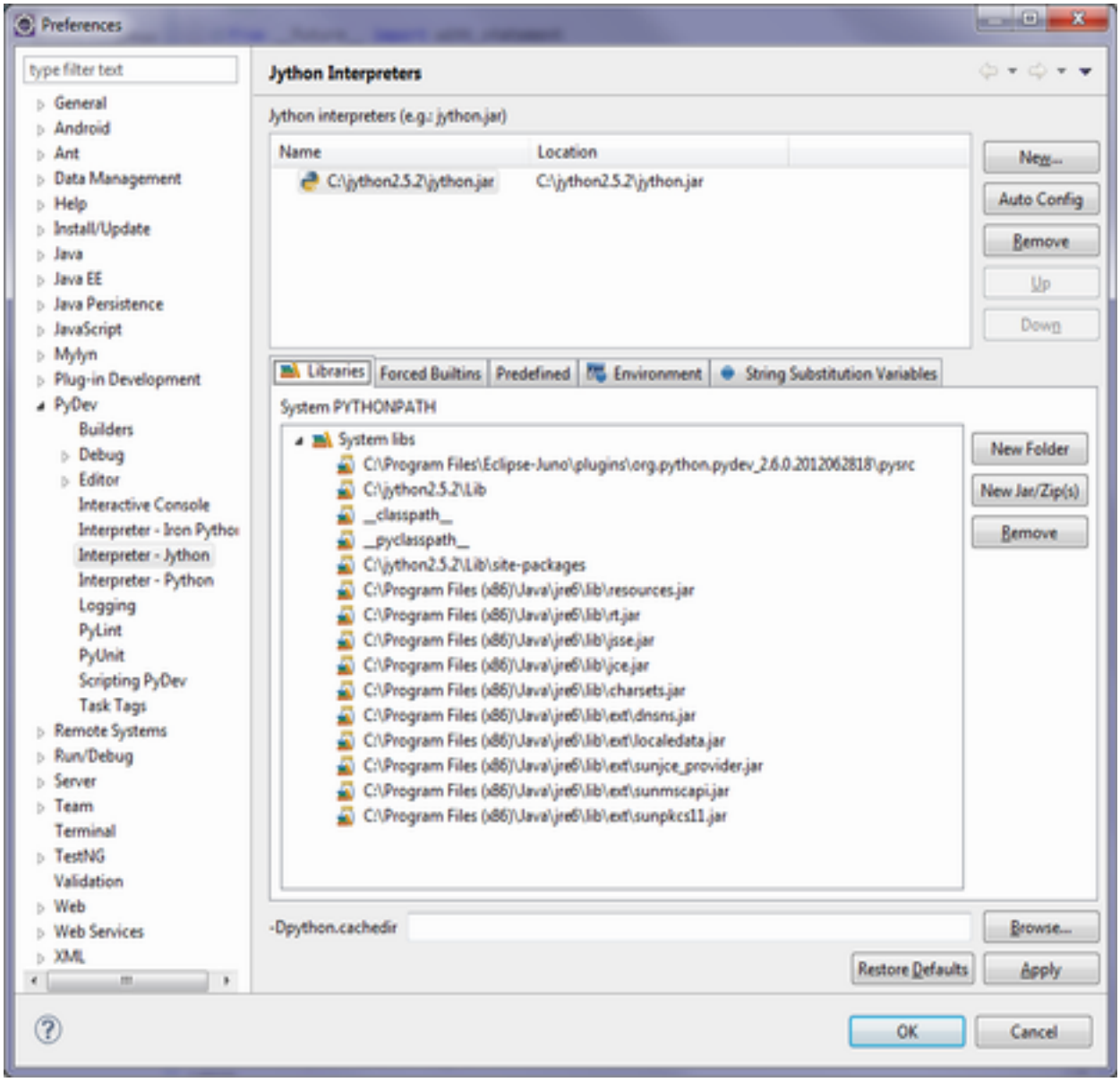
Eclipse Config for Jython/Sikuli

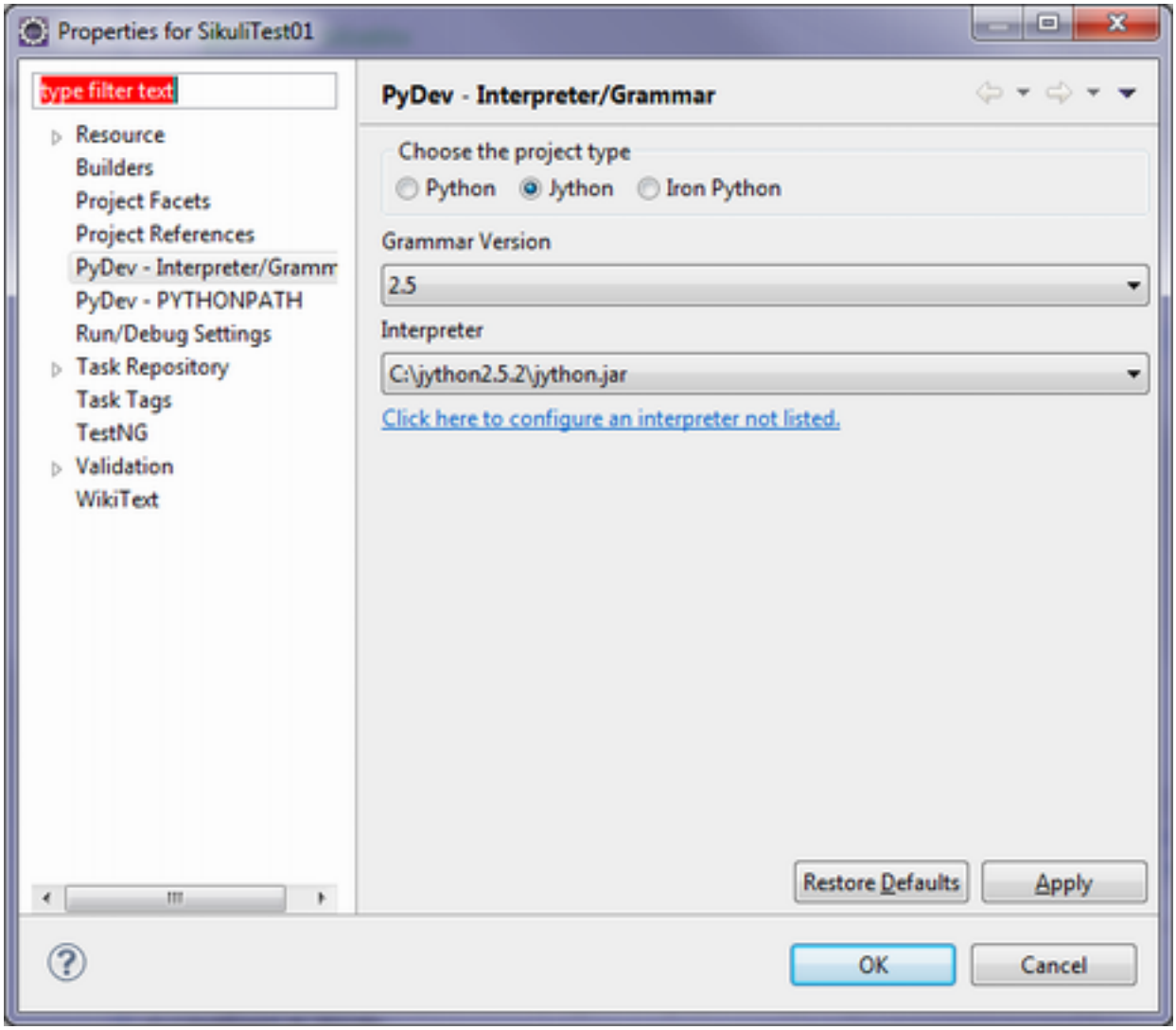
Eclipse > Window > Preferences > PyDev > Interpreter - Jython

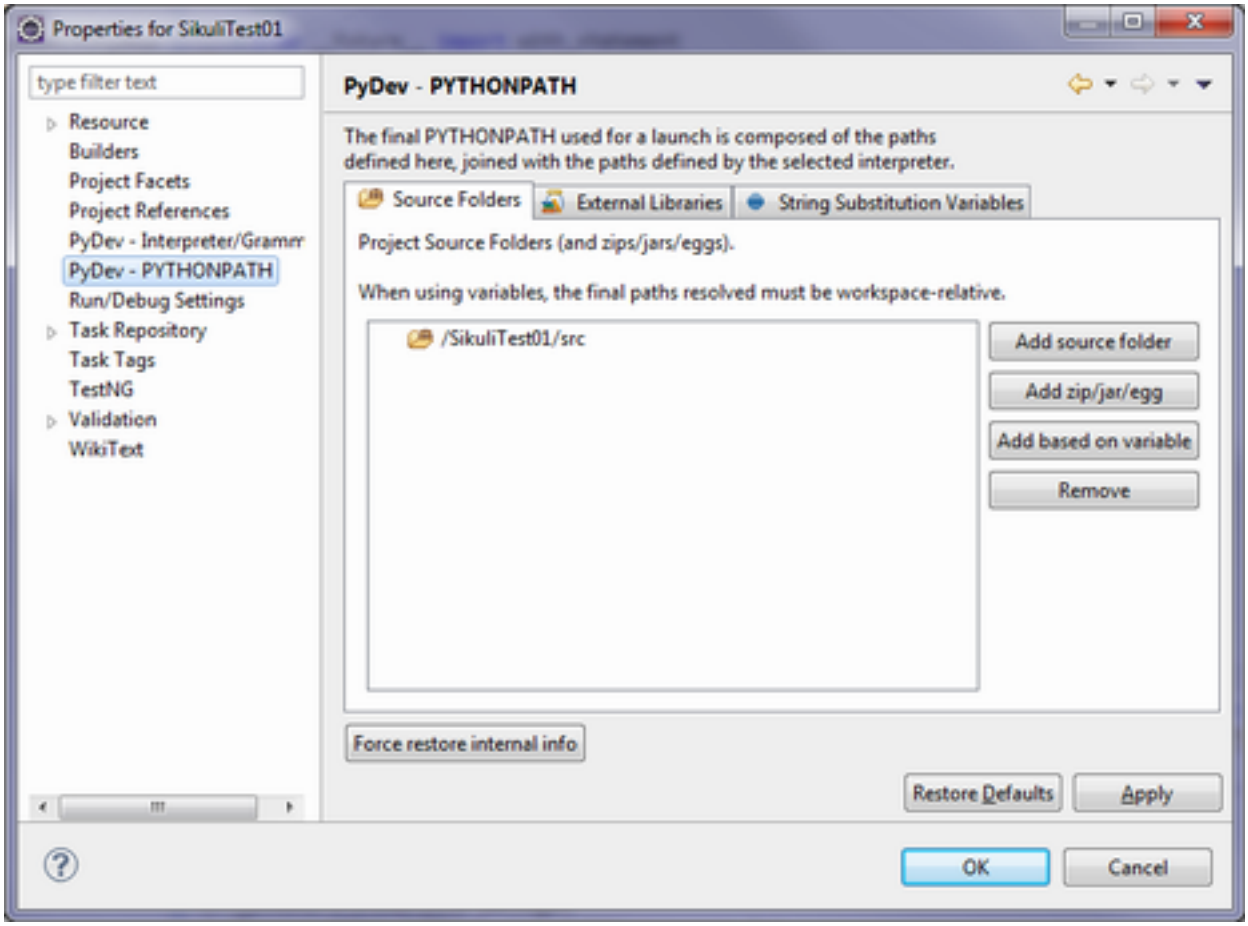
- **Interpreter - Jython** = Jython 2.5.2
- **PATH=**C:\Program Files (x86)\Java\jre6\bin;D:\Sikuli\Lib\Sikuli32\Sikuli-IDE\libs;C:\Windows\SysWOW64\java.exe;
    - Without PATH, you will get: **java.lang.UnsatisfiedLinkError:**

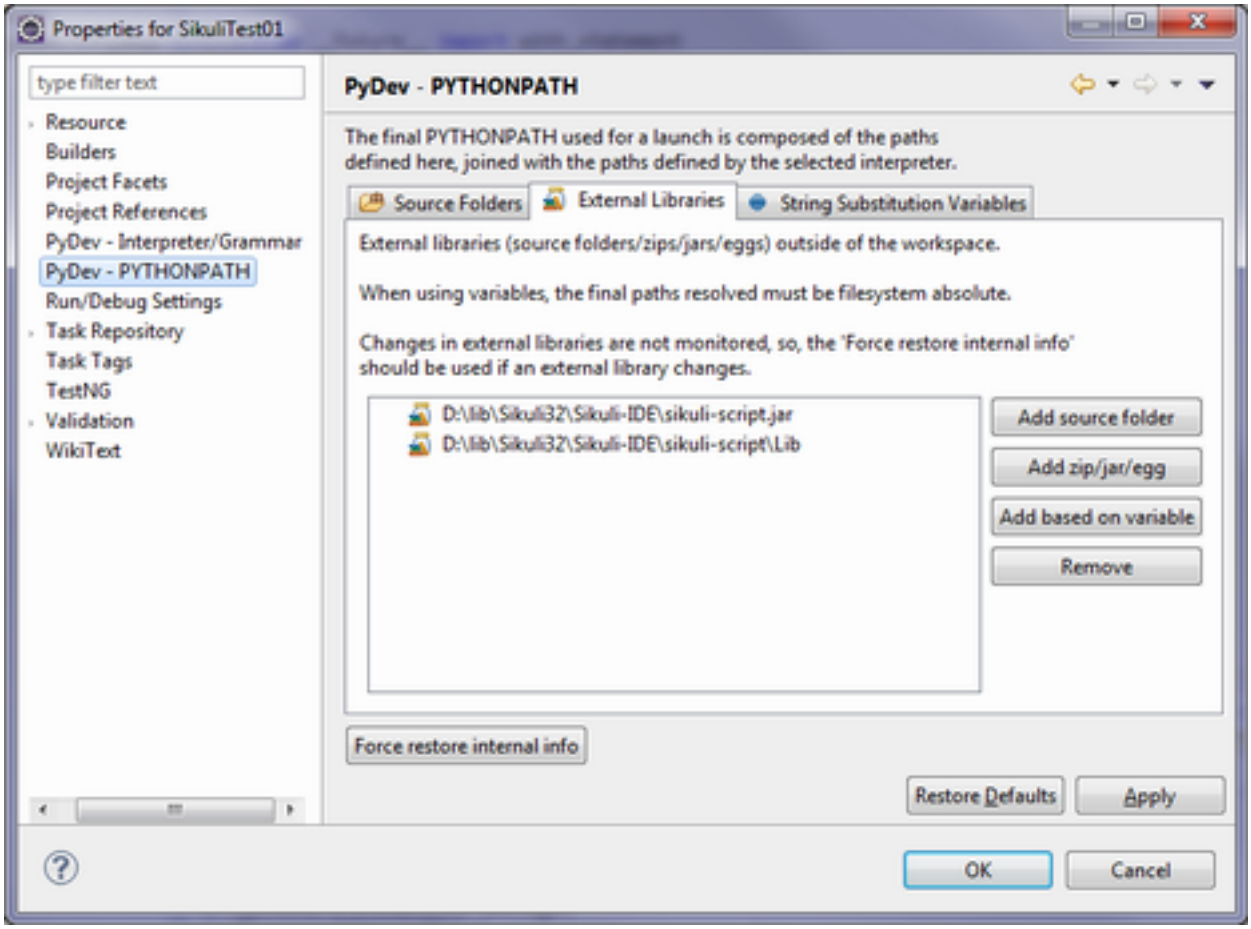
Project Preferences

- **PyDev - PYTHONPATH**
    - **Source Folders** = Project
    - **External Libraries** =
        - sikuli-script.jar
        - sikuli-script\Lib (Pull from siklui-script.jar)


Preferences
type filter text
General
Android
Ant
Data Management
Help
Install/Update
Java
Java EE
Java Persistence
JavaScript
Mylyn
Plug-in Development
PyDev
Builders
Debug
Editor
Interactive Console
Interpreter - Iron Pytho
Interpreter - Jython
Interpreter - Python
Logging
PyLint
PyUnit
Scripting PyDev
Task Tags
Remote Systems
Run/Debug
Server
Team
Terminal
TestNG
Validation
Web
Web Services
XML
Jython Interpreters
Jython interpreters (e.g.: jython.jar)
Name
Location
C:\jython2.5.2\jython.jar
C:\jython2.5.2\jython.jar
New...
Auto Config
Remove
Up
Down
Libraries
Forced Builtins
Predefined
Environment
String Substitution Variables
System PYTHONPATH
System libs
C:\Program Files\Eclipse-Juno\plugins\org.python.pydev_2.6.0.2012062818\pysrc
C:\jython2.5.2\Lib
__classpath__
__pyclasspath__
C:\jython2.5.2\Lib\site-packages
C:\Program Files (x86)\Java\jre6\lib\resources.jar
C:\Program Files (x86)\Java\jre6\lib\rt.jar
C:\Program Files (x86)\Java\jre6\lib\jsse.jar
C:\Program Files (x86)\Java\jre6\lib\jce.jar
C:\Program Files (x86)\Java\jre6\lib\charsets.jar
C:\Program Files (x86)\Java\jre6\lib\ext\dnsns.jar
C:\Program Files (x86)\Java\jre6\lib\ext\localedata.jar
C:\Program Files (x86)\Java\jre6\lib\ext\sunjce_provider.jar
C:\Program Files (x86)\Java\jre6\lib\ext\sunmscapi.jar
C:\Program Files (x86)\Java\jre6\lib\ext\sunpkcs11.jar
New Folder
New Jar/Zip(s)
Remove
-Dpython.cachedir
Browse...
Restore Defaults
Apply
OK
Cancel


Properties for SikuliTest01
type filter text
Resource
Builders
Project Facets
Project References
PyDev - Interpreter/Gramm
PyDev - PYTHONPATH
Run/Debug Settings
Task Repository
Task Tags
TestNG
Validation
WikiText
PyDev - Interpreter/Grammar
Choose the project type
Python
Jython
Iron Python
Grammar Version
2.5
Interpreter
C:\jython2.5.2\jython.jar
Click here to configure an interpreter not listed.
Restore Defaults
Apply
OK
Cancel


Properties for SikuliTest01
type filter text
Resource
Builders
Project Facets
Project References
PyDev - Interpreter/Gramm
PyDev - PYTHONPATH
Run/Debug Settings
Task Repository
Task Tags
TestNG
Validation
WikiText
PyDev - PYTHONPATH
The final PYTHONPATH used for a launch is composed of the paths defined here, joined with the paths defined by the selected interpreter.
Source Folders
External Libraries
String Substitution Variables
Project Source Folders (and zips/jars/eggs).
When using variables, the final paths resolved must be workspace-relative.
/SikuliTest01/src
Add source folder
Add zip/jar/egg
Add based on variable
Remove
Force restore internal info
Restore Defaults
Apply
OK
Cancel


Properties for SikuliTest01
type filter text
Resource
Builders
Project Facets
Project References
PyDev - Interpreter/Grammar
PyDev - PYTHONPATH
Run/Debug Settings
Task Repository
Task Tags
TestNG
Validation
WikiText
PyDev - PYTHONPATH
The final PYTHONPATH used for a launch is composed of the paths defined here, joined with the paths defined by the selected interpreter.
Source Folders
External Libraries
String Substitution Variables
External libraries (source folders/zips/jars/eggs) outside of the workspace.
When using variables, the final paths resolved must be filesystem absolute.
Changes in external libraries are not monitored, so, the 'Force restore internal info' should be used if an external library changes.
D:\lib\Sikuli32\Sikuli-IDE\sikuli-script.jar
D:\lib\Sikuli32\Sikuli-IDE\sikuli-script\Lib
Add source folder
Add zip/jar/egg
Add based on variable
Remove
Force restore internal info
Restore Defaults
Apply
OK
Cancel