



Scripting VI SDK with Jython

Steve JIN
June, 2008

Agenda

- What is Jython?
- What do I need?
- HelloVM.py
- Calling Java from Jython
- Executing Jython from Java

What is Jython?

- “Jython is an implementation of the high-level, dynamic, object-oriented language Python written in 100% Pure Java, and seamlessly integrated with the Java platform. It thus allows you to run Python on any Java platform.”

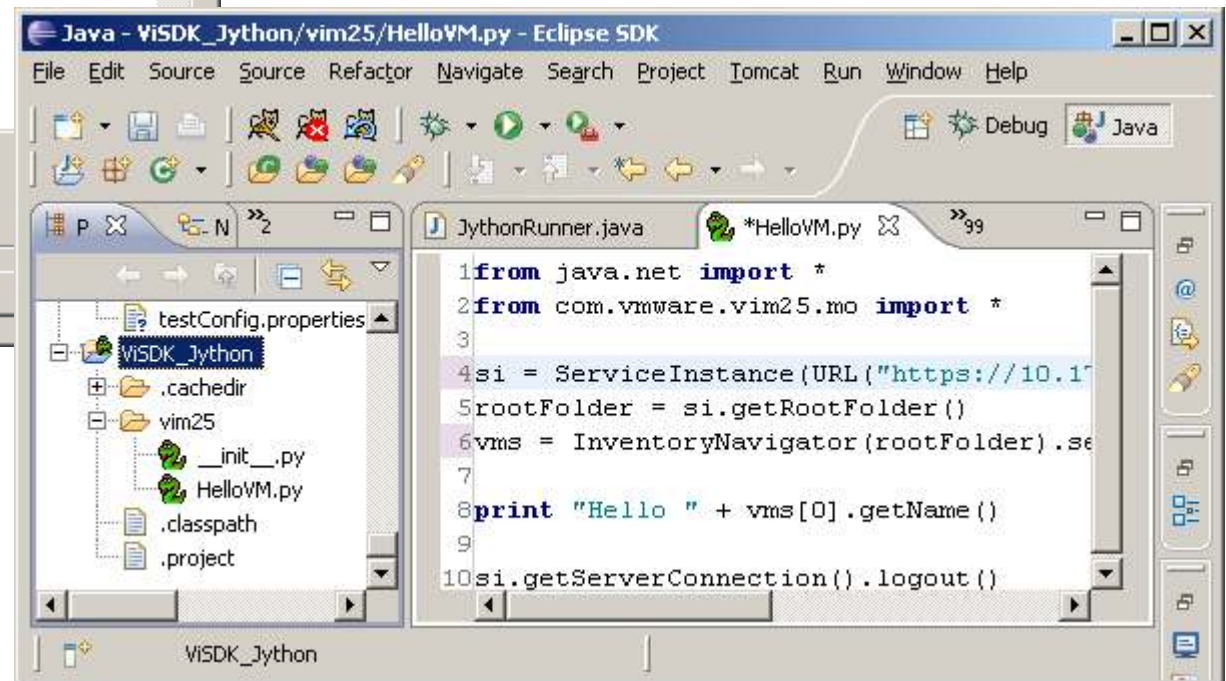
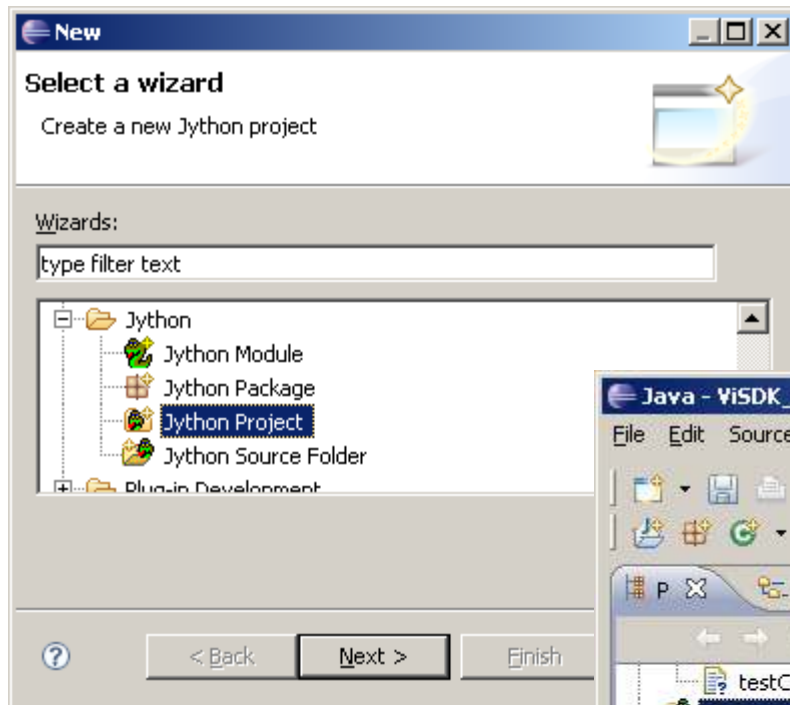
<http://www.jython.org/Project/index.html>

- It looks like Python, and integrates well with Java.

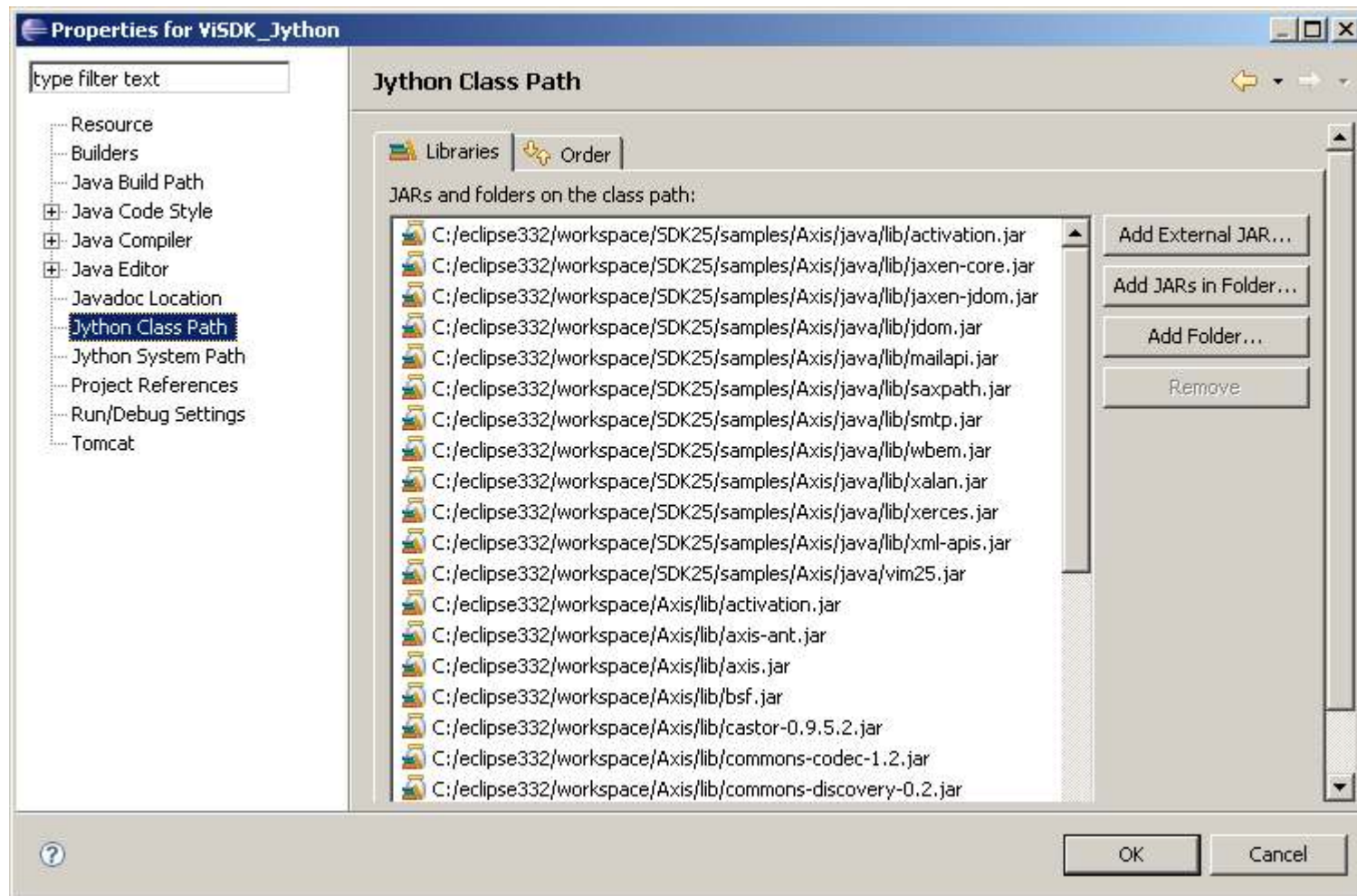
What do I need?

- Jython
 - <http://www.jython.org/Project/download.html>
 - FAQ: <http://www.jython.org/Project/userfaq.html>
- Jython IDE (JyDT plugin for Eclipse)
 - Download the corresponding Eclipse plug-in, unzip it to a local directory and point your Eclipse “Install/Update” wizard to the directory.
 - <http://www.redrobinsoftware.net/jydt/>
 - Optionally, you can point to the online update site directly.
 - <http://www.redrobinsoftware.net/jydt/installation/installation.html>
 - Create and configure a project
 - <http://www.redrobinsoftware.net/jydt/projects/creatingproject.html>
- VI SDK 2.5
- VI Java API

Create a Jython project



Configure a Jython project

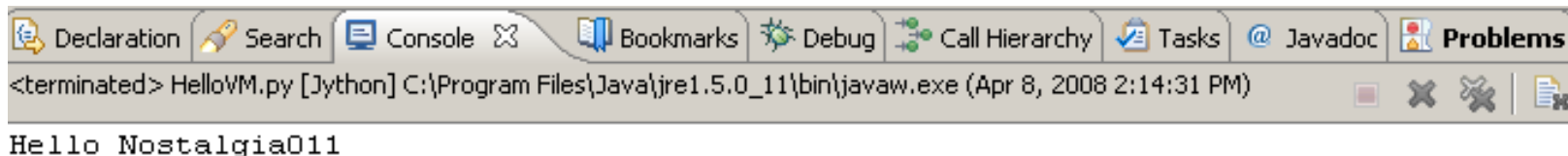


HelloVM.py

```
1 from java.net import *
2 from com.vmware.vim25.mo import *
3
4 si = ServiceInstance(URL("https://10.17.218.174/sdk"), "root", "password", True)
5 rootFolder = si.getRootFolder()
6 vms = InventoryNavigator(rootFolder).searchManagedEntities("VirtualMachine")
7
8 print "Hello " + vms[0].getName()
9
10 si.getServerConnection().logout()
```

T

- Output:



The screenshot shows an IDE console window with a toolbar at the top containing icons for Declaration, Search, Console, Bookmarks, Debug, Call Hierarchy, Tasks, Javadoc, and Problems. The console text reads: <terminated> HelloVM.py [Jython] C:\Program Files\Java\jre1.5.0_11\bin\javaw.exe (Apr 8, 2008 2:14:31 PM) Hello Nostalgia011

Calling Java from Jython

- First include the Java libraries in the Jython class path
- Import the Java class in Jython code:

```
import java.net.URL as URL
```

```
import com.vmware.vim25.mo.ServiceInstance as ServiceInstance
```

Or **from** com.vmware.vim25.mo **import** *

- Use the Java object in your code:
 - `si = ServiceInstance(URL("https://10.17.218.174/sdk"), "root", "vmware", True)`
- More info @ <http://www.jython.org/Project/userguide.html>

Executing Jython from Java

- Include jython.jar in the Java classpath
- Use the Jython Interpreter as follows:

```
package com.vmware.vim25.mo.samples;

import org.python.util.*;

public class JythonRunner
{
    public static void main(String[] args)
    {
        PythonInterpreter pi = new PythonInterpreter();
        pi.setOut(System.out);
        pi.exec("C:/eclipse332/workspace/ViSDK_Jython/vim25/HelloVM.py");
    }
}
```

- Line by line execution is also possible with `PythonInterpreter.eval()` method.
- More advanced techniques:
 - <http://www.jython.org/Project/userfaq.html#extending-jython>

Conclusion

- Jython is a good scripting language for managing Virtual infrastructure. It has the advantages of both worlds - Python and Java.
- Open source VI Java API provides an easy to use object model for the scripting the VI SDK. More info can be found at its project home. (<http://vijava.sf.net>)