Jython
For fun and profit

Barcamp Dublin
21 April 2007

Sean McGrath

sean.mcgrath@propylon.com
http://www.propylon.com
http://seanmcgrath.blogspot.com
Contents

• First...Python
• Then Jython...
• The Glue Language Concept
• Four Examples of Jython in Action
• Jython in a wider context: CPython, IronPython, PyPy
• The Good news and the bad news about Jython
• The Dynamic Languages Revolt : Jython in an even wider context: DynLangs, Agile Dev, XP, WebApps...
• Some Resources
First...Python...
First...Python...
The Who/What/Where/When and Why of Jython

- **What**
  - Python is written in C...
  - Rewrite in Java...Python on the JVM
    - The Java Virtual Machine is not the same thing as Java the Language
    - Can target JVM from many languages. Mix'n'match dynamic and static programming language techniques

- **Who:**
  - Jim Hugunin, CNRI, 1997, JPython
  - ...
  - Frank Wierzbicki - Red Hat - Current dev lead – now called Jython
Glue

- Big Slow moving Enterprise Apps meet
  - Changing business requirements
  - Constantly changing integration requirements
  - Insanely rapid churning of “front end” technologies...Web, Mobile, Second Life...

- ...Glue languages bolt everything together
  - Long history
  - .bat files...bash...rexxx...awk...perl...visual basic...ruby...python...lisp...lua...
Examples

- Databases like Oracle, messaging systems like MQSeries, Document Management Systems like Documentum, App Servers like WebSphere...
- Common Denominator = JVM Connectivity
- BUT
  - Java is not a pleasant Glue Language
Enter Jython

• Jython for
  – “Java” GUI Development
  – XML Munging
  – Servlets
  – Java Unit Testing
Jython for GUI Dev

• Will work with any of the GUI toolkits (since they are just libraries at the end of the day)
• AWT – equally ugly everywhere
• SWING – equally buggy everywhere
• SWT – So much for the portable GUI library
Jython/SWT

![Sample SWT Jython App window]

![Hello world dialog box]
Simple SWT Example
Sean McGrath

```
from org.eclipse.swt.events import *
from org.eclipse.swt.graphics import *
from org.eclipse.swt.layout import *
from org.eclipse.swt.widgets import *
from org.eclipse.swt.layout.GridData import *
from org.eclipse.swt import *
```
Jython/SWT

class gui:

    def messagebox (self,s):
        mb = MessageBox (self.shell,SWT.OK)
        mb.setMessage (s)
        if mb.open() == SWT.OK:
            print "OK"
def main(self):
    self.display = Display()
    self.shell = Shell(self.display,text="Sample SWT Jython App")
    # Two Column Grid Layout
    self.shell.setLayout (GridLayout(numColumns=2))

    self.menuBar = Menu (self.shell,SWT.BAR)

    self.fileMenuHeader = MenuItem (self.menuBar,SWT.CASCADE)
    self.fileMenuHeader.setText ("&File")

    self.fileMenu = Menu (self.shell,SWT.DROP_DOWN)
    self.fileMenuHeader.setMenu (self.fileMenu)

    self.fileExitItem = MenuItem (self.fileMenu,SWT.PUSH)
    self.fileExitItem.setText ("&Exit")
    self.fileExitItem.addSelectionListener (FileExitItemListener(self.shell,self.display))

    self.shell.setMenuBar (self.menuBar)
# row1
self.r1c1 = Label(self.shell, SWT.SINGLE,
    text="Label 1:",
    layoutData=GridData (HORIZONTAL_ALIGN_FILL, horizontalSpan=1))

self.r1c2 = Text(self.shell, SWT.SINGLE|SWT.BORDER,
    layoutData=GridData (HORIZONTAL_ALIGN_FILL, horizontalSpan=1))
self.r1c2.setText('"")
# row2
self.r2c1 = Label(self.shell, SWT.SINGLE,
    text="Label 2:",
    layoutData=GridData (HORIZONTAL_ALIGN_FILL, horizontalSpan=1))

self.r2c2 = Combo(self.shell, SWT.NONE,
    items=('Apple', 'Orange'),
    layoutData=GridData (HORIZONTAL_ALIGN_FILL, horizontalSpan=1))
# row3
self.r3 = Button (self.shell,SWT.PUSH,text="Click Me",
    widgetSelected = self.button_handler,
    layoutData=GridData (HORIZONTAL_ALIGN_FILL, horizontalSpan=2))
# Boilerplate Event Loop Voodoo

```
self.shell.pack()
self.shell.open()

while not self.shell.isDisposed():
    if not self.display.readAndDispatch():
        self.display.sleep()

self.display.dispose()
```

# Event Handlers by the boat load

```
def button_handler(self, evt):
    self.messageBox("Hello world")
```
# More boilerplate

class FileExitItemListener (SelectionListener):

    def __init__(self,shell,display):
        self.shell = shell
        self.display = display

    def widgetSelected(self,evt):
        self.shell.close()
        self.display.dispose()

    def widgetDefaultSelected(self,evt):
        self.shell.close()
        self.display.dispose()

if __name__ == '__main__':
    app = gui()
    app.main()
Jython for XML Munging

- Xerces/Xalan
- SAX API Example
Jython for XML Munging

XML Processing Example using Xerces SAX

```
from org.xml import sax as java_sax
from java.io import FileReader

class XMLHandler (java_sax.helpers.DefaultHandler):
    def startDocument (self):
        print "START DOCUMENT"

    def startElement(self, uri, name, qname, attrs):
        print "START ELEMENT %s" % name

fr = FileReader("test.xml")
input_source = java_sax.InputSource(fr)
xml_handler = XMLHandler()
xr = java_sax.helpers.XMLReaderFactory.createXMLReader()
xr.setContentHandler(xml_handler)
xr.parse(input_source)
fr.close()
```
Jython for XML Munging

```
test.xml
<test>
<p>Hello world</p>
</test>
```

```
$jython xml.py
```

```
START DOCUMENT
START ELEMENT
test
START ELEMENT p
```
Jython for Servlets

```python
from java.io import *
from javax.servlet.http import HttpServlet
from JythonServletUtils import *
import calendar, time

class JythonServlet (HttpServlet):
    def doGet(self, request, response):
        self.doPost(request, response)

    def doPost(self, request, response):
        toClient = response.getWriter()
        response.setContentType("text/html")
        toClient.println("<html><head><title>Jython Servlet Test</title>"
        )
        toClient.println("<body><h1>Calendar</h1><pre>%s</pre></body></html>" %
        calendar.calendar(time.localtime()[0]))
```
Jython for testing

• interactive shell a great boon for application testing
• junit/unitttest
• pushtotest/grinder etc.
• Some big Java shops use Jython extensively for testings but don't talk about it in public
The Wider Python World

• CPython also begat...
  – IronPython
  – PyPy
• OLPC, OpenOffice, Chandler, Plone, TurboGears, Django
Jython – the bad news

- Implementation progress keeping in synch with CPython is slow. No “Big Co” very involved
- Sometimes the join is not completely seamless – java arrays, wrapper classes can get in the way e.g. logging
- Jython lags – and probably alway will lag - CPython
Jython – the good news

- Incredible productivity compared to pure Java Programming Language app dev
- Even if you only use Jython for test, use it! You will be amazed at the difference it makes to productivity
The Dynamic Languages Revolt

- “scripting languages” grrrrr....
- Static languages
  - Prevent certain types of failure by preventing certain types of success
- “A program is correct if it gets the job done before your boss fires you” -- Larry Wall
- “Ship the prototype” -- MPRTDV
- Dynamic Languages/Agile Dev/XP
- “Behind DNS, nobody hears you redeploying your gradually improving app, every day”
- The Ruby on Rails Effect
Resources

- http://www.jython.org
- Frank's blog: http://fwierzbicki.blogspot.com/
- Jython Essentials; Pedroni, Rappin (ISBN: 0596002475)
- Barry Feigenbaum:
  - Excellent two part Jython (and Python) tutorial:
    http://www.jython.org/j-jython1-ltr.pdf
    http://www.jython.org/j-jython2-ltr.pdf
- Jython Servlet Tutorial (part 1):
Thank You

• (Question,Answer?)*