Jython and Java™ Technology: Plug-and-Play

Otmar Humbel

Developer
Bison Schweiz AG
http://jython.extreme.st

Session TS-9574

Goal of This Talk

What you will gain

Learn how to seamlessly integrate Jython with your Java™ applications.

Agenda

Why Jython?

Getting Familiar With Jython

Using Jython Objects in Java Technology

Using Java Objects in Jython

Script Deployment

What's Next?

Q&A

We Do Not Need Scripting—Reflection Already Has All the Dynamics!

```
import st.extreme.jython.Clock;

try {
  Class cCls = Class.forName("st.extreme.jython.Clock");
  Class <?> fCls =
    Class.forName("st.extreme.jython.ClockFrame");
  Object frame = fCls.newInstance();
  Object clock = cCls.newInstance();
  Method clockMethod = fCls.getMethod("setClock",
    new Class[] { Clock.class });
  clockMethod.invoke(frame, new Object[] { clock });
  Method visibleMethod = fCls.getMethod("setVisible",
    new Class[] { boolean.class });
  visibleMethod.invoke(frame, new Object[] { true });
} catch (Exception e) {
  e.printStackTrace();
}
```

Jython

- Gives you a dynamic scripting environment
- Lets you access in one environment:
  - All of your Java libraries
  - Most of the Python libraries
- Is a good choice for your toolbox
  - Stable
  - Fast
  - Easy to integrate

Python

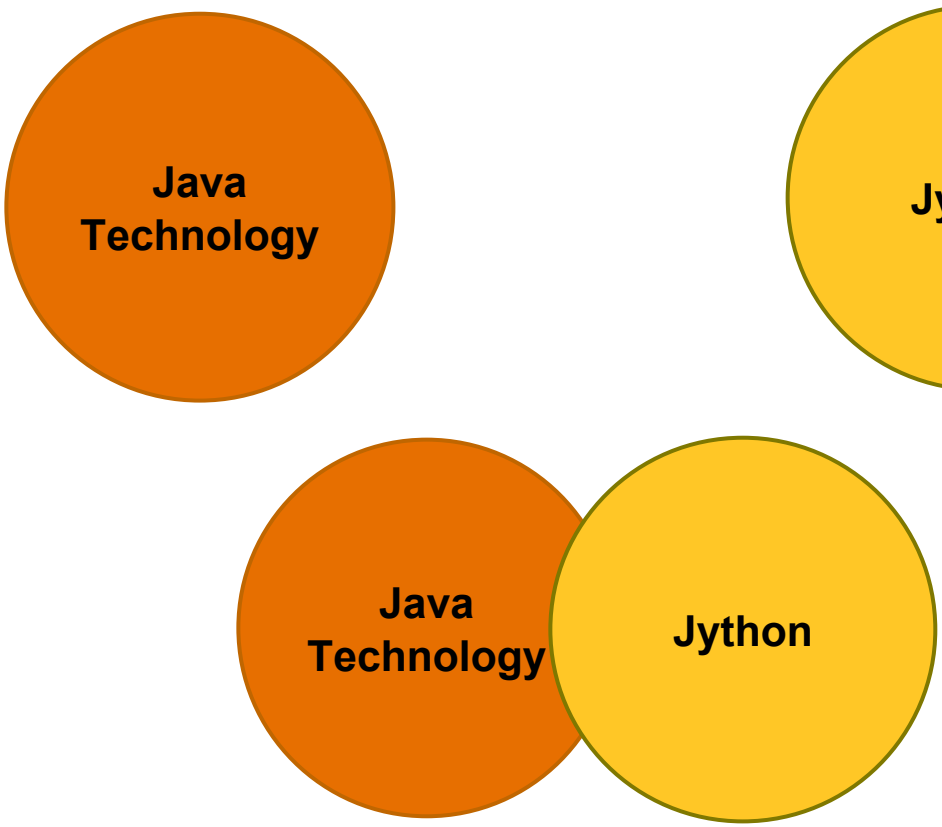
- Easy to learn
- Freely usable and distributable
- Extensive standard libraries and third-party modules for virtually every task
- Well documented
- You can do amazing things in very few lines of code

Python

A working SMTP server, printing messages to stdout

```
from smtpd import *
import asyncore

try:
  DebuggingServer(('localhost', 25), ('localhost', 25))
  asyncore.loop(timeout=2)
except KeyboardInterrupt:
  print "Crtl+C pressed. Shutting down."
```

Is This Your Picture of Java Technology/Jython Integration?



Agenda

Why Jython?

Getting Familiar With Jython

Using Jython Objects in Java Technology

Using Java Objects in Jython

Script Deployment

What's Next?

Q&A

Getting Familiar With Jython

How to get started

- Download the installation .jar from http://www.jython.org
- Execute the .jar file
  - Select ‘standalone’ as installation type
- Result: a file named **`jython.jar`**
- **`java -jar jython.jar`**
- **`jython.jar`** can be added to your classpath

DEMO

Play around with the interpreter
Play around with your own Java objects

What We Have Learned So Far

- Simply add **`jython.jar`** to your classpath
- Start the interactive interpreter as follows:
  - **`org.python.util.jython`**
- Use the interpreter to:
  - Play with your Java objects
  - Become acquainted with the Python language
- Java objects behave as expected

Agenda

Why Jython?

Getting Familiar With Jython

Using Jython Objects in Java Technology

Using Java Objects in Jython

Script Deployment

What's Next?

Q&A

Using Jython Objects in Java Technology

- Interpreted Mode
- Compiled Mode
- Optimized Mode

Interpreted Mode: The Clock Example

- A provider class returns Java 2D™ API shapes:
  - Hands
  - Ticks
- These shapes rotate around the centre point
- Implement the second hand in Jython
- Make our implementation look like a “normal” Java object to the application
- Change the behavior of the application at runtime

Original example from: http://www.chinaitpower.com/2005September/2005-09-13/208673.html

The Java Technology Superclass

```
package st.extreme.jython;

public class SecondHand implements IClockPath {

    public GeneralPath getPath() { .. }

    public Color getColor() { .. }

    public boolean isVisible() { .. }

}
```

SecondHand Usage

```
public class Clock extends JPanel {

  public void paint(Graphics g) {
    Graphics2D = (Graphics2D) g;

    IClockPath secondHand = getProvider().getSecondHand();
    if (secondHand.isVisible()) {
      g2.setPaint(secondHand.getColor());
      g2.fill(secondHand.getPath().
        createTransformedShape(getSecondTransform()));
  }

}
```

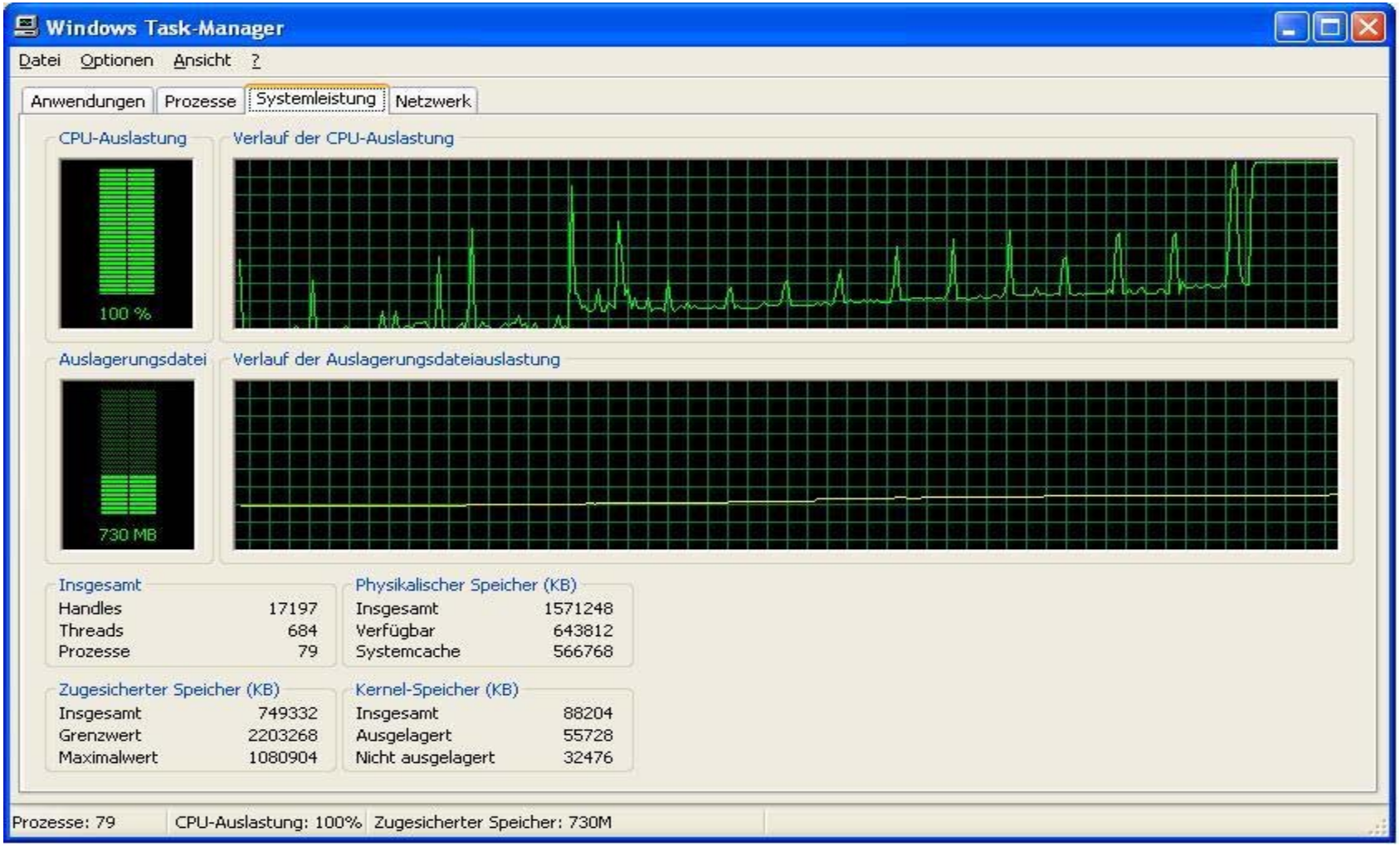
Interpreted Mode Using javax.script

```
import st.extreme.jython.util.JyClass;

public IClockPath
  createSecondHand(String secondHandScript)
{

  return JyClass.newInterpretedInstance(IClockPath.class,
    secondHandScript, "secondHand");

}
```

DEMO

Interpreted Mode

What Went Wrong? (1/2)



What Went Wrong? (2/2)

- Every second, 60 ticks are created
- Each tick compiles and loads a class
- Improvements
  - Use compiled scripts
  - Reuse the 60 ticks

Compiled Mode Using javax.script

```
public Tick createTick(String tickScript,
                       int drawingMinute)
{
  Bindings bindings = new SimpleBindings();
  bindings.put("drawingMinute",
               new Integer(drawingMinute));

  // compile only once
  CompiledScript compiledTickScript =
    getCompiledTickScript(tickScript, bindings);

  return JyClass.newCompiledInstance(Tick.class,
    compiledTickScript, bindings, "tick");
}
```

Reuse the 60 Ticks

```
private Tick[] _tickCache = new Tick[60];

public Tick getTick(int drawingMinute,
  int realtimeSecond)
{
  Tick tick = _tickCache[drawingMinute];
  if (tick == null) {
    tick = /* create it as shown ... */
    _tickCache[drawingMinute] = tick;
  }
  tick.setRealtimeSecond(realtimeSecond);
  return tick;
}
```

Java Code Developers Have a Dream:

What if we could just write a class definition…

```
from st.extreme.jython import Tick

class OptimizedTick(Tick):
  def __init__(self, drawingMin):
    Tick.__init__(self, drawingMin)

  def isVisible(self):
    return 1
```

- Compile this into a Java class…
- Instantiate the same Java class over and over?

Optimized Mode

- Applies the “pattern”:
  - `JyClass jyClass = JyClass.forScript(…)`
  - `jyClass.newInstance(…)`

- 1:1 Relation between script and class in Java Virtual Machine™ (JVM™)

The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform.

Optimized Mode for Tick Creation

```
private JyClass _tickJyClass;

private JyClass getTickJyClass(String tickScript) {
  if (_tickJyClass == null) {
    _tickJyClass =
      JyClass.forScript(tickScript, Tick.class);
  }
  return _tickJyClass;
}

public Tick createOptimizedTick(String tickScript,
  int drawingMinute)
{
  JyClass tickJyClass = getTickJyClass(tickScript);
  return tickJyClass.newInstance(Tick.class,
     new Integer(drawingMinute));
}
```

DEMO

1. Compiled Mode
2. Optimized Mode

What We Have Learned So Far

- Jython objects look like Java objects
- Take care when creating objects (reuse if possible)
- Optimize use of embedded scripts
  - Use compiled scripts
  - Use `JyClass.forScript()/newInstance()`
- Handle exceptions
  - At compile time
  - At run time

Agenda

Why Jython?

Getting Familiar With Jython

Using Jython Objects in Java Technology

Using Java Objects in Jython

Script Deployment

What's Next?

Q&A

The Alarm Class

```
public class Alarm {
  private JyDecimal _hour;
  private JyDecimal _minute;

  // bean property participant
  public void setHour(Object hour) {
    if (hour instanceof JyDecimal) {
      _hour = (JyDecimal) hour;
    } else {
      _hour.setValue(hour);
    }
  }
  // bean property participant
  public Object getHour() {
    return _hour;
  }
}
```

DEMO

A simple alarm clock

JyDecimal Explained (Subtraction)

```
public JyDecimal sub(Object subtrahend) {
  return new
    JyDecimal(_value.subtract(makeNumeric(subtrahend)));
}

public Object __sub__(Object subtrahend) {
  return sub(subtrahend);
}

public Object __rsub__(Object minuend) {
  return new
    JyDecimal(makeNumeric(minuend).subtract(_value));
}

public JyDecimal __isub__(Object subtrahend) {
  setValue(sub(subtrahend));
  return this;
}
```

What We Have Learned So Far

- Bean properties for setter/getter pair methods
- Hook methods like `__sub__(Object obj)` (enable) operators
- Java objects behave like Python ones
- Familiar syntax for script authors

Agenda

Why Jython?

Getting Familiar With Jython

Using Jython Objects in Java Technology

Using Java Objects in Jython

Script Deployment

What's Next?

Q&A

Python Modules

- A Python module is a directory
  - Containing an `__init__.py` file (usually empty)
  - And any number of `*.py` files
- Example
  - `mymodules/__init__.py`
  - `mymodules/foo/__init__.py`
  - `mymodules/foo/bar.py`

See also: http://www.python.org

Easy Deployment

- You can pack all your modules in the standalone **`jython.jar`**
- Place the **`/mymodules`** directory in **`/Lib`**
  - **`jython.jar/Lib/mymodules/__init__.py`**
  - **`jython.jar/Lib/mymodules/foo/__init__.py`**
  - **`jython.jar/Lib/mymodules/foo/bar.py`**
- Deploy the **`jython.jar`** with your application
- You can then import as follows:
  - **`from mymodules.foo import bar`**

Tips

- Use different names for Java code packages and Python modules
- Physically separate directories containing Python modules from those containing Java code packages
- When importing Java class files, always use the following pattern:
  - from some.cool.java.package import AFancyClass
- Avoid inner classes named py or PyInner in your Java class files

Agenda

Why Jython?

Getting Familiar With Jython

Using Jython Objects in Java Technology

Using Java Objects in Jython

Script Deployment

What's Next?

Q&A

Progress on Jython

- Active mailing lists
- Increasing number of committers
- Build bots
- Automated tests
- Lots of other activities, for example:
  - Jim Baker/Michael Taylor working on an ANTLR parser for Python 2.5
  - “leouser” working on **`jythonx`**

Current Development Status

(As of March 2007)

- 2.2b1 released
- 2.2b2 in progress
- 2.2 (final) targeted for spring 2007
- Many 2.3 features already implemented
- Far less time expected for 2.2 → 2.3, compared to 2.1 → 2.2

Jython Roadmap

- Next release after 2.2
  - Primarily clean-up
    - Will catch up with latest CPython version (2.3, 2.4, or even 2.5 ?)
    - Improve Java code integration
- Jython 2.x (release after that)
  - Performance improvements
  - Enable CPython frameworks (Django?)
- Jython 3.0 (“Jython 3000”)
  - Twin of CPython 3.0
  - Backwards incompatible

See also: http://www.jython.org/Project/roadmap.html

Getting Involved

[Sure we need an active community!]

- Start with the wiki page at:
  - http://wiki.python.org/jython/HowToGetInvolved
- Pick an unimplemented module
- Solve bugs and provide patches:
  - http://sourceforge.net/projects/jython/

Summary

- Jython is easy…
  - To install
  - To use
  - To deploy
- Jython integrates seamlessly with your Java applications
- Jython dynamically extends your Java applications
- You get three for two ☺

For More Information

- Your backup
  - http://jython.extreme.st
- Python-related links
  - http://www.python.org
  - http://www.djangoproject.com

For More Information (Cont.)

- Jython-related links
  - http://www.jython.org
  - http://wiki.python.org/jython
  - https://scripting.dev.java.net
  - http://pydev.sourceforge.net/index.html

For More Information (Cont.)

Books

- **Jython Essentials**; by Samuele Pedroni and Noel Rappin
- **Jython for Java Programmers**; by Robert W. Bill
- **Python Programming With the Java Class Libraries, A Tutorial for Building Web and Enterprise Applications with Jython**; by Richard Hightower
- **Python in a Nutshell**; by Alex Martelli
- **Learning Python, Second Edition**; by Mark Lutz
- **Beginning Python, From Novice to Professional**; by Magnus Lie Hetland

Q&A

Otmar Humbel

http://jython.extreme.st

Thanks to …

Charles Oliver Notter, Sun
for making it possible to speak here

A. Sundararajan, Sun
for implementing the javax.script engine

The Jython developers on the mailing lists
for their support and review

Ruben Bakker, uncomplex.net
for showing me JPython years ago,
for the clock idea and the SMTP server

Renate Willimann, Gustaf Hansen, and Lars Steiger, bison-group.com
for helping me with the presentation

Jython and Java™ Technology: Plug-and-Play

Otmar Humbel

Developer
Bison Schweiz AG
http://jython.extreme.st

Session TS-9574