



Java Scripting: One VM, Many Languages

Mike Grogan
Sun Microsystems

LOCK
OPPORTUNITY

What will you open?



SUN TECH DAYS 2006-2007
A Worldwide Developer Conference



UNL
OP

Wha

Goals of Talk

- Explain motivation for Scripting Support in the Java Platform.
- Explain and demonstrate new Scripting Features.
- Future Directions.

Agenda

- **Background / Motivation**
- Java SE 6 Scripting Features
- Demo
- Future Directions
- Resources
- Q & A

Why Scripting?

- Java Language != Java Platform
 - > VM runs “language-neutral” bytecode
 - > Rich set of Class libraries are “language-neutral”
 - > “**Write once run anywhere**” applies to Platform
 - > Leverage programmer skills and advantages of particular languages.
- Time-tested technologies
 - > Open-source projects for various languages.
 - > Jakarta BSF.

Scripting Scenarios

- Class Files written in other language.
 - > Groovy
 - > Jython Compiler
 - > Kawa Scheme
- Java Applications execute script programs
 - > Stand-alone interpreter
 - > Macro interpreters
 - > Web Scripting
- In both cases programs use Java Objects/Libraries

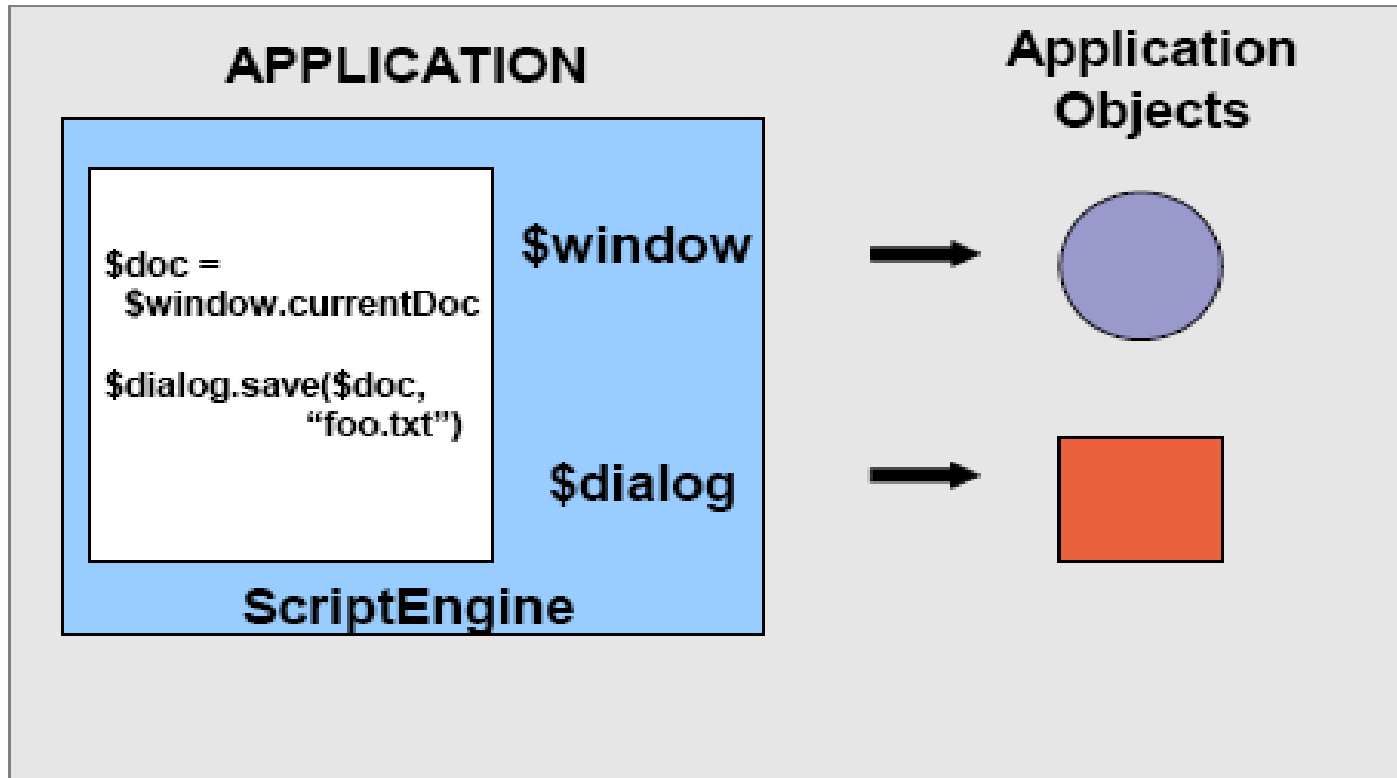
Agenda

- Motivation / Background
- **Java SE 6 Scripting Features**
- Demo
- Future Directions
- Resources
- Q & A

Scripting Framework

- Pluggable Framework for third-party Script Engines.
 - > Resembles BSF ActiveX Scripting
 - > “Java application runs script programs” scenario.
- **javax.script** package
- Part of Java SE 6
- Available for Java 5.0

Scripting Components Diagram



Example code

```
import javax.script.*;

public class Main {
    public static void main(String[] args) throws ScriptException {
        // create a script engine manager
        ScriptEngineManager factory = new ScriptEngineManager();

        // create JavaScript engine
        ScriptEngine engine = factory.getEngineByName("JavaScript");

        //add a script variable whose value is a Java Object
        engine.put("greeting", new Exception("Hello World!"));

        // evaluate JavaScript code from String
        engine.eval("print(greeting.toString());");
    }
}
```

Scripting API

- ScriptEngine
 - > Components that execute scripts.
- ScriptEngineManager
 - > Used by host application to locate and instantiate ScriptEngines.
- ScriptContext/Bindings
 - > Provide view of host application to ScriptEngine
 - > Script variables<====>Application objects.

ScriptEngine

- ScriptEngine interface - **required**
 - Execute scripts - “eval” methods
 - Map Java objects to script variables (“put”/”get” methods)
- Invocable interface – **optional**
 - Invoke script functions/methods
 - Implement Java interface using script functions/methods
- Compilable interface – **optional**
 - Compile Script to intermediate form
 - Execute multiple times without recompilation

ScriptEngineManager

- Concrete class
- ScriptEngine discovery/instantiation
 - Services mechanism
 - Thread context class loader
 - By name, extension, MIME type
- Global scope variables visible to all ScriptEngines

Javascript Engine

- Based on Mozilla Rhino 1.6v2
- Features omitted for security/footprint reasons.
 - > Optimizer (script-to-bytecode compiler – only interpreter support)
 - > E4X (XML language support) – depends on xmlbeans.jar
 - > Rhino command line tools (shell, debugger etc.)
- Security Tweaks

Scripting Tools / Samples

- Tools
 - > <JDK>/bin directory
 - > jrunscript
 - > Interactive command-line interpreter.
 - > jhat
 - > Processes heap analysis tool output
 - > jconsole scripting plugin
- Samples
 - > Script notepad
 - > Swing application mostly implemented in Javascript
 - > Fancy Javascript programming.

Agenda

- Motivation / Background
- Mustang Scripting Features
- **Demo**
- Future Directions
- Resources
- Q & A

Programmable Calculator

- From “Scripting for the Java Platform” by John O'Connor
<http://java.sun.com/developer/technicalArticles/J2SE/Desktop/scripting/>
- 100% Java Swing Application
- Customizable using end-users' scripts
- Uses Mustang Javascript engine
- Enhanced to use any JSR 223 Engine

Agenda

- Motivation / Background
- Mustang Scripting Features
- Demo
- **Future Directions**
- Resources
- Q & A

Language JSRs

- invokedynamic Bytecode – JSR 292
 - > <http://www.jcp.org/en/jsr/detail?id=292>
- Groovy – JSR 241
 - > <http://groovy.codehaus.org/>
- BeanShell – JSR 272
 - > <http://www.beanshell.org>

Server-side Scripting - Phobos

- <http://phobos.dev.java.net>
- Borrows from Ruby on Rails
 - > Speed of development
 - > Well-organized application structure
- Access to Enterprise Java
- Javascript libraries
- Support for other Technologies
 - > AJAX
 - > RSS / Atom

Agenda

- Motivation / Background
- Mustang Scripting Features
- Demo
- Future Directions
- **Resources**
- Q & A

Resources - scripting.dev.java.net

- BSD License
- Scripting Engines
 - > jruby, groovy, beanshell, jacl, jaskell, java, jawk, jelly, jexl, jruby, javascript, jython, ognl, pnuts, scheme, sleep, xpath, xslt
- Applications
 - > NetBeans 5.0 Scripting module
- Also see coyote.dev.java.net
 - > NetBeans 5.0 Scripting IDE
 - > Jython, groovy support

Resources - references

- JSR-223
 - > <http://jcp.org/en/jsr/detail?id=223>
- A. Sundararajan's Blog
 - > <http://blogs.sun.com/sundararajan>
- Roberto Chinnici's Blog (serverside scripting)
 - > <http://weblogs.java.net/blog/robc/>
- JavaScript Developer Connection
 - > <http://java.sun.com/javascript>

Agenda

- Motivation / Background
- Mustang Scripting Features
- Demo
- Future Directions
- Resources
- Q & A



Java Scripting: One VM, Many Languages

Mike Grogan
Sun Microsystems
mike.grogan@sun.com

LOCK
OPPORTUNITY

What will you open?



SUN TECH DAYS 2006-2007
A Worldwide Developer Conference



UNL
OP

Wha