Getting Groovy *with Java*

Trenton Computer Festival 2008
IT Professional Conference
April 25, 2008

Michael P. Redlich

mike@redlich.net

My Background (1)

- **Degree**
  - B.S. in Computer Science
  - Rutgers University (go Scarlet Knights!)
- **"Petrochemical Research Organization"**
  - Senior Research Technician (1988-1998, 2004-present)
  - Systems Analyst (1998-2002)
- **Ai-Logix, Inc.**
  - Technical Support Engineer (2003-2004)
- **Amateur Computer Group of New Jersey (ACGNJ)**
  - Java Users Group Leader (2001-present)
  - President (2007-present)
  - Secretary (2006)

My Background (2)

- **Publications**
  - Java Boutique (http://www.javaboutique.com/)
    - ❖ Co-authored with Barry Burd
    - ❖ Design Patterns
  - http://publications.redlich.net/
- **Presentations**
  - Trenton Computer Festival (TCF) since 1998
  - TCF IT Professional Seminars since 2006
  - Emerging Technologies for the Enterprise 2008
  - Princeton Java Users Group
  - Capital District Java Developers Network
  - New York Software Industry Association (NYSIA)

Objectives

- **What is Groovy?**
- **What is a Domain Specific Language?**
- **Differences Between Groovy and Java**
- **Getting Started with Groovy**
- **GroovyBeans**
- **GStrings**
- **Closures**
- **Groovy Builders**
- **Source Code Examples (yea!)**

What is Groovy?

- **"An agile dynamic language for the Java platform"**
  - ❑ **Codehaus Opensource Software**
- **"Builds upon the strength of Java but has additional power features inspired by languages like Python, Ruby, and Smalltalk"**
  - ❑ **Codehaus Opensource Software**
- **A scripting language that runs on top of the Java Virtual Machine (JVM)**
  - ❑ **Sanctioned by Sun Microsystems**
  - ❑ **JSR 241**
- **A Domain Specific Language (DSL)**

What is a Domain Specific Language (DSL)?

- "A limited form of computer language designed for a specific class of problems"
  - Neal Ford, ThoughtWorks
- The trend for development is:
  - Better
  - Lighter
  - Faster
- Implicit context

```
Venti half-caf, non-fat, extra hot, no foam, no
whip latte
```

Ordering Coffee in Java

```
Coffee latte = new Latte(Size.VENTI);
latte.setStrength(5);
latte.setFatContent(FatContent.NON_FAT);
latte.setTemperature(Temperature.EXTRA_HOT);
latte.setFoam(Foam.NONE);
latte.setWhip(Whip.NONE);
```

Ordering Coffee in a DSL

```
Coffee latte = Latte.order()
     .size(VENTI);
     .strength(5);
     .fatContent(NON_FAT);
     .temperature(EXTRA_HOT);
     .foam(NONE);
     .whip(NONE);
```

Differences Between Groovy and Java (1)

- **Semicolons are optional**
  - ❑ You can use them if you wish
  - ❑ You must use them to put several statements on one line
- **The `return` keyword is optional**
- **Methods and classes are public by default**
- **`protected` in Groovy has the same meaning as `protected` in Java**
  - ❑ You can have friends in the same package and derived classes can also see protected members

Differences Between Groovy and Java (2)

- **Inner classes are not supported at this time**
  - ❑ **Use closures instead**
- **There is no difference between checked and unchecked exceptions**
- **You will not get compile errors for using undefined members or passing arguments of the wrong type**

- **Download Groovy**
  - http://groovy.codehaus.org/
- **Setup Groovy environment**
  - `set GROOVY_HOME=C:\groovy-1.5.5`
  - `set PATH=%PATH%;%GROOVY_HOME%\bin`
  - Copy your favorite database driver to `%GROOVY_HOME%\lib`
- **Use the Groovy shell:**
  - `C:\> groovsh`
- **Use the Groovy console:**
  - `C:\> groovyConsole`

Getting Started With Groovy (2)

Use the Groovy compiler

- `C:\groovy-apps\sql> groovyc ReadDb.groovy`

GroovyBeans

- A JavaBean defined in Groovy
- Plain Old Groovy Object (POGO)
- Automatic generation of getter/setter methods

```
class Book {
    String title
    }
```

GStrings

- GStrings can be denoted with either single or double quotes
  - double quote: string literal
  - Single quote: placeholder for future dereference

```
me = 'Tarzan'
you = 'Jane'


println "Me $me, you $you"
// prints Me Tarzan, you Jane
```

Closures

- **A section of code wrapped up as an object**
  - Accept arguments
  - Return a value
- **Resembles a Java anonymous inner class**

```
def adder = {
     a,b -> print a + b
     }


adder(3,9) // prints '12' to the console
```

Builders

- **Provide a convenient way to build hierarchal data models**
- **Available builders include:**
  - **MarkupBuilder**
  - **NodeBuilder**
  - **SwingBuilder**

Groovy Resources (1)

- **Groovy Web Site**
  - http://groovy.codehaus.com/
- **Scott Davis Blog**
  - http://aboutgroovy.com/

Further Reading (1)

- **Groovy Recipes: Greasing the Wheels of Java**
  - Scott Davis
  - ISBN
  - http://pragprog.com/titles/sdgrvr/groovy-recipes/
- **Groovy in Action**
  - Dierk Koenig
  - ISBN 1-932394-84-2
  - http://www.manning.com/koenig/
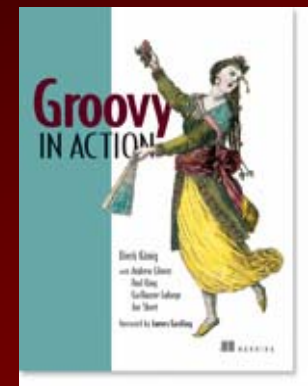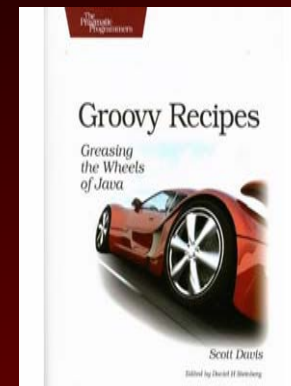
Local Java User Groups

- **ACGNJ Java Users Group**
  - facilitated by Mike Redlich
  - http://www.javasig.org/
- **Princeton Java Users Group**
  - facilitated by Yakov Fain
  - http://www.myflex.org/princetonjug/
- **NYJavaSIG**
  - facilitated by Frank Greco
  - http://www.javasig.com/
- **Capital District Java Developers Network**
  - facilitated by Anthony DeBonis
  - http://www.cdjdn.com/