Java Code Coverage

Marc R. Hoffmann
The Java Specialists Symposium, Crete
2011-09-01
Code Coverage

„Legacy Code is simply code without tests.“

Michael Feathers:
Working Effectively with Legacy Code
Usage Scenario

Unit Tests

INDIRECT METRIC

System Test
Automated UI Test
Manual Execution

White Box Testing

Black Box Testing
Coverage Units

- Control Flow Coverage
  - Classes
  - Methods
  - Lines
  - Statements
  - Branches
  - Paths

Coverage Ratio = \frac{Covered Units}{Total Units}
Statement Coverage

```java
public int clip(int lower, int upper, int x) {
    if (x < lower) { ✓
        x = lower; ✓
    }
    if (x > upper) { ✓
        x = upper; ✓
    }
    return x; ✓
}
```

Test Set for Full Statement Coverage:

- clip(1, 9, 0)
- clip(1, 9, 10)
Branch Coverage

```java
public int clip(int lower, int upper, int x) {
    if (x < lower) {
        x = lower;
    }
    if (x > upper) {
        x = upper;
    }
    return x;
}
```

Test Set for Full Branch Coverage:
- `clip(1, 9, 0)`
- `clip(1, 9, 10)`
Path Coverage

```java
public int clip(int lower, int upper, int x) {
    if (x < lower) {
        x = lower;
    }
    if (x > upper) {
        x = upper;
    }
    return x;
}
```

Test Set for Full Path Coverage:

- clip(1, 9, 0)
- clip(1, 9, 10)
- clip(1, 9, 5)
- clip(9, 1, 5)
EclEmma – Code Coverage for Eclipse
Current Status of EMMA

- Great Tool! 😊
- Last Release 😞
  - 2.0.5312, 2005-06-13
- Project Activity 😞
  - Latest Commit 2006-02-23
- No Java 7 Support 😞

<table>
<thead>
<tr>
<th>Element</th>
<th>Instruction Coverage</th>
<th>Missed Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.vladium.emma</td>
<td>0%</td>
<td>5.412 / 5.412</td>
</tr>
<tr>
<td>com.vladium.util</td>
<td>0%</td>
<td>2.462 / 2.462</td>
</tr>
<tr>
<td>com.vladium.jdc</td>
<td>0%</td>
<td>1.531 / 1.531</td>
</tr>
<tr>
<td>com.vladium.logging</td>
<td>0%</td>
<td>162 / 162</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>0%</strong></td>
<td><strong>9.567 / 9.567</strong></td>
</tr>
</tbody>
</table>
Action Required!

- Making EMMA alive?
- Using a different Library?
- Starting a new Project?
Requirements for a Code Coverage Library

- Be a Library!
  - Open for Different Usage Scenarios
  - Designed for Integration
- Regression Tests
- Runtime Framework Independent
- Scalable for Large Projects
- Fast Enough for Agile Teams
Java Code Coverage

The „JaCoCo“ Project

- **Java Code Coverage**
- Started Mid of 2009
- EPL
- Hosted within EclEmma (SourceForge)
  - www.eclemma.org/jacoco
- Integrated with
  - Ant
  - Maven
  - Sonar
  - GUI Dancer (Commercial)
JaCoCo Demo

```
T:\demo\eclipse>eclipse -vmargs -javaagent:jacocoagent.jar
T:\demo\eclipse>cd ..\report
T:\demo\report>ant
Buildfile: T:\demo\report\build.xml
report:
BUILD SUCCESSFUL
total time: 20 seconds
T:\demo\report>
```

```
org.eclipse.osgi.util

<table>
<thead>
<tr>
<th>Element</th>
<th>Instruction Coverage</th>
<th>Missed Classes</th>
<th>Missed Methods</th>
<th>Missed Blocks</th>
<th>Missed Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ManifestElement</strong></td>
<td>48%</td>
<td>0 / 1</td>
<td>9 / 21</td>
<td>73 / 153</td>
<td>83 / 189</td>
</tr>
<tr>
<td><strong>TextProcessor</strong></td>
<td>18%</td>
<td>0 / 1</td>
<td>3 / 8</td>
<td>59 / 71</td>
<td>49 / 61</td>
</tr>
<tr>
<td><strong>NLS</strong></td>
<td>53%</td>
<td>0 / 1</td>
<td>2 / 11</td>
<td>47 / 125</td>
<td>55 / 145</td>
</tr>
<tr>
<td><strong>NLS_MessagesProperties</strong></td>
<td>50%</td>
<td>0 / 1</td>
<td>0 / 2</td>
<td>6 / 15</td>
<td>9 / 23</td>
</tr>
<tr>
<td><strong>NLS privilegedAction()</strong></td>
<td>0%</td>
<td>1 / 1</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td>4 / 4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>48%</td>
<td>1 / 5</td>
<td>18 / 44</td>
<td>106 / 366</td>
<td>139 / 421</td>
</tr>
</tbody>
</table>

Created with JaCoCo 0.3.1.20100311234421
Implementation Details
Recording Code Coverage

- `javaagent:D:\java_tools\jacoco\lib\jacocoagent.jar`
Coverage Analysis

Actual *.exec  Coverage

Target *.class  Analysis

Good work, but I think we might need just a little more detail right here.
Java Agent

- `java.lang.instrument`
  - In-Memory
  - No class file preprocessing

```java
byte[] transform(ClassLoader loader,
                 String className,
                 Class<?> classBeingRedefined,
                 ProtectionDomain protectionDomain,
                 byte[] classfileBuffer)
throws IllegalClassFormatException
```
Java Code Coverage

Keep the Good Ideas of EMMA

- Byte Code Instrumentation
  - JRE and Platform Independent
- Basic Block Coverage
  - Good Tradeoff between Details and Overhead
- Using `boolean[]` Arrays for Probe Storage
  - Concurrency Possible
  - Sharing the Instance
Probes

ALOAD probearray
ICONST probeid
ICONST_1
BASTORE
Class Identity

- Issues
  - Multiple Versions of the Same Class in one VM
  - Modified Classes over Time

- Use CRC64 Hash
  - Fits into Java `long`
Avoid Coverage Runtime Dependency

- Avoid Class Loading Issues
- Use JRE APIs only!

```java
Object access = ... // Retrieve instance

Object[] args = new Object[3];
args[0] = Long.valueOf(0x89f47a04b2881d38); // class id
args[1] = "com/example/MyClass"; // class name
args[2] = Integer.valueOf(24); // probe count

access.equals(args);

boolean[] probes = (boolean[]) args[0];
```
How to Share an Object Instance?

- **The Challenge:**
  
  Share a given object instance by using JDK APIs only.

- **Current Solutions:**
  
  - Object as System Property
  - Install Custom Handler with Java Logging
  - Instrumented JRE Class
Runtime Isolation

Eating one’s own dog food:
Run JaCoCo on JaCoCo?

- Java Agent becomes part of the application classpath 🙃
- Rename classes in jacocoagent.jar during build 😊
Runtime Isolation

```
jacocoagent.jar
+-- org.jacoco.agent.rt_ouey7p
    +-- org.jacoco.agent.rt_ouey7p.asm
    +-- org.jacoco.agent.rt_ouey7p.asm.commons
    +-- org.jacoco.agent.rt_ouey7p.asm.tree
    +-- org.jacoco.agent.rt_ouey7p.core.data
    +-- org.jacoco.agent.rt_ouey7p.core.instr
    +-- org.jacoco.agent.rt_ouey7p.core.runtime
+-- META-INF
    +-- about.html
```