The Stairway to Maven

The JAKARTA Project Build
And Comprehension Tool

DI Siegfried GÖSCHL
siegfried.goeschl@it20one.at
The Problem

How do you build and maintain complex software projects written in Java?!

- The Chinese Approach ...  
  - You can always find a student or two for the build

- Make or better not to Make – that’s the question ...  
  - Ever tried to run a UNIX makefile on Windows?!

- All ANTs are equal and some are even more equal ...  
  - Tired of copy/pasting ANT scripts?!
  - Need something like ANT with more bells and whistles?!
Maven was initially developed for building Turbine
Maven matured into an open source software engineering platform
The core functionality is automated project building, distribution and project website creation
A project is described with a XML Project Object Model (POM)

- The POM defines how to build a project and defines the external dependencies
- The project build result are created in the local repository
- Dependent JARs are downloaded from a remote repository
- The Maven functionality is implemented in terms of plugins
- The plugins are written in Jelly
What is in the Maven Box?!

CLOVER  SIMIAN  JavaNCSS  DOCBOOK
Word2HTML  Canoo WebTest  RPM

Custom Plugins

Source Cross Ref  JavaDoc Creation  Project Distribution  Source Metrics
Regression Test  Site Creation  Coding Style Check  CVS Reports

Core Plugins

MAVEN

project.xml  maven.xml  project.properties  build.properties
Project Object Model  Custom Maven Scripts  Project Configuration  Build Configuration
Project Object Model (POM)

- Verbose project description
- Company information
- List of developer and their roles
- Mailing list and CVS server access
- Source code and unit test code location
- Resources needed for a build
- Definition of JAR dependencies
- Assembling of artifacts, e.g. JAR
The Core Plugins

Build Plugins

- The distribution plugin allows building a binary or a source distribution from the command line
- An **ANT** build file can be generated and distributed to allow **ANT** users to build the distribution
- Other plugins allows
  - Creating a JAR file
  - Creating a WAR file
  - Creating an EAR file
  - Deploying the distribution automatically

No more thrills building your project!
The Core Plugins

Site Plugin

- Site creation based on Project Object Model
- Site generation uses XDOC
  - Maven generated XDOC
  - Manually written documentation
- JSTL transforms XDOC into HTML
- Site layout is defined through `xdoc/navigation.xml`
- Site appearance is customizable through properties defined in `project.properties`
The Core Plugins

Site Plugin

**PartyCodeIdTypes**

A PartyCodeIdType classifies an identifier for looking a party

<table>
<thead>
<tr>
<th>PartyCodeIdType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>internal</td>
<td>The internally used UUID for a party</td>
</tr>
<tr>
<td>external</td>
<td>A unique external id, usually the customer id of the target system</td>
</tr>
<tr>
<td>duns</td>
<td>A DUNS code of the party</td>
</tr>
<tr>
<td>iln</td>
<td>An ILN code of the party</td>
</tr>
<tr>
<td>goverment_id</td>
<td>A unique goverment id such as &quot;Firmenbuchnummer&quot;</td>
</tr>
</tbody>
</table>

**Language Codes**

The [ISO 639-1 Codes](https://en.wikipedia.org/wiki/ISO_639-1) are used
**The Core Plugins**

### Site Plugin

#### Project Reports
- PMD Report

#### Project Documentation
- Front Page
- Project Info
- Project Reports
  - Change Log
  - Developer Activity
- File Activity
- Unit Tests
- Metric Results
- Checkstyle Report
- JavaDocs
- Source XReference
- Development Process

### Overview

<table>
<thead>
<tr>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Log</td>
<td>This document provides a history of the most recent changes made to the project.</td>
</tr>
<tr>
<td>Developer Activity</td>
<td>This document provides a report of activity in terms of CVS commits and breaks it out by developer.</td>
</tr>
<tr>
<td>File Activity</td>
<td>This document provides a report of activity in terms of CVS commits and breaks it out by file.</td>
</tr>
<tr>
<td>Unit Tests</td>
<td>This document provides the results of the unit tests that are part of this project. Successes and failures are noted.</td>
</tr>
<tr>
<td>Metric Results</td>
<td>This document provides information on various source code metrics that have been computed. These metrics can provide useful information regarding the abstractness and total number of classes.</td>
</tr>
<tr>
<td>Checkstyle Report</td>
<td>This document provides the results of the Checkstyle report. This report provides an indication of how well this project complies with its coding conventions.</td>
</tr>
</tbody>
</table>
The Core Plugins

CheckStyle Plugin

- Various coding styles can be defined
  - Sun coding convention
  - Turbine coding convention
  - Roll your own coding convention (if you really have to)

- Rules are customizable through XML file
  - Maximum line length of 100 instead of 80 characters
  - Patterns for constants and variables
  - Setting a different tab width

Just in case you have a coding convention ...
The Core Plugins

CheckStyle Plugin

Method is missing a Javadoc comment.

org/apache/maven/app/PluginManager.java

<table>
<thead>
<tr>
<th>Error</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unused import - java.net.URL</td>
<td>64</td>
</tr>
<tr>
<td>Unused import - java.net.URLClassLoader</td>
<td>65</td>
</tr>
<tr>
<td>Unused import - java.utilEnumeration</td>
<td>67</td>
</tr>
<tr>
<td>Unused import - java.util.HashMap</td>
<td>68</td>
</tr>
<tr>
<td>'static' modifier out of order with the JLS suggestions.</td>
<td>108</td>
</tr>
<tr>
<td>'static' modifier out of order with the JLS suggestions.</td>
<td>113</td>
</tr>
<tr>
<td>'static' modifier out of order with the JLS suggestions.</td>
<td>118</td>
</tr>
<tr>
<td>'static' modifier out of order with the JLS suggestions.</td>
<td>123</td>
</tr>
<tr>
<td>'static' modifier out of order with the JLS suggestions.</td>
<td>128</td>
</tr>
<tr>
<td>variable 'log' must match pattern '^[A-Z][A-Z0-9]+$'.</td>
<td>128</td>
</tr>
<tr>
<td>cast needs to be followed by whitespace.</td>
<td>173</td>
</tr>
<tr>
<td>Expected an @return tag.</td>
<td>217</td>
</tr>
<tr>
<td>Expected @param tag for 'pluginName'.</td>
<td>217</td>
</tr>
<tr>
<td>';' needs to be followed by whitespace.</td>
<td>219</td>
</tr>
</tbody>
</table>
The Core Plugins

Activity Plugin

- Plugin accesses CVS directly to extract development activity information
- Extracts information about
  - Changes and commits
  - Number of commits per developer
  - Number of changes for a file
- Ongoing work to support VSS and ClearCase

No more digging through CVS changelog
## Activity Plugin

### Change Log

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Files/Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-09-04</td>
<td>Siegfried Goeschl</td>
<td>junitpp.jpr.local - v1.3 maven.xml - v1.4</td>
</tr>
<tr>
<td>09:37:12</td>
<td></td>
<td>project.properties - v1.3 project.xml - v1.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xdocs/commandline.xml - v1.1 xdocs/index.xml - v1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xdocs/stylesheets/project.xml - v1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cleaned up site and added documentation</td>
</tr>
<tr>
<td>2002-09-03</td>
<td>Siegfried Goeschl</td>
<td>maven.xml - v1.3</td>
</tr>
<tr>
<td>20:18:20</td>
<td></td>
<td>Commented out the PMD target</td>
</tr>
<tr>
<td>2002-09-03</td>
<td>Siegfried Goeschl</td>
<td>JUNIT.library - v1.2</td>
</tr>
<tr>
<td>20:11:08</td>
<td></td>
<td>junitpp.jpr - v1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>junitpp.jpr.local - v1.3</td>
</tr>
</tbody>
</table>
The Core Plugins

Test Plugin

- Runs **JUnit** test case
- The plugin creates a **XDOC** report
- The **XDOC** report is then transformed to HTML
- The **JUnit** test report is then integrated within the generated project site

I hope you have some tests?!
The Core Plugins

Test Plugin

<table>
<thead>
<tr>
<th>Name</th>
<th>Tests</th>
<th>Errors</th>
<th>Failures</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllTests</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>32,000</td>
</tr>
</tbody>
</table>

Test Cases

- [summary]  [package list]  [test cases]

AllTests

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Type</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>testGetKeyWithBaseNameKey</td>
<td>Success</td>
<td></td>
<td>0,000</td>
</tr>
<tr>
<td>testGetKeyWithBaseKey</td>
<td>Success</td>
<td></td>
<td>0,000</td>
</tr>
<tr>
<td>testGetKeyWithClassNameKey</td>
<td>Success</td>
<td></td>
<td>0,000</td>
</tr>
<tr>
<td>testGetKeyWithKey</td>
<td>Success</td>
<td></td>
<td>0,000</td>
</tr>
<tr>
<td>testGetInt</td>
<td>Success</td>
<td></td>
<td>0,000</td>
</tr>
<tr>
<td>testGetUndefinedKey</td>
<td>Success</td>
<td></td>
<td>0,000</td>
</tr>
</tbody>
</table>

Stairway To Maven
The Core Plugins

JDepend Plugin

- Is using **JDepend** from ClarkWare
- Analyzes class files
- Reports package dependencies
- Finds cyclic dependencies
- Generates a HTML report to be included into project website
### JDepend Plugin

The JUNITPP Project - JDepend Source Code Analysis - Microsoft Internet Explorer

Address: http://junitpp.sourceforge.net/depend-report.html

#### Packages

- **summary**
- **packages**
- **cycles**
- **explanations**

#### junit.extensions

<table>
<thead>
<tr>
<th>Afferent Couplings</th>
<th>Efferent Couplings</th>
<th>Abstractness</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>17%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abstract Classes</th>
<th>Concrete Classes</th>
<th>Used by Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigurableTest</td>
<td>ActiveTest</td>
<td>test.junit.extensions</td>
</tr>
<tr>
<td></td>
<td>ConfigurableTestCase</td>
<td>test.junit.extensions.util</td>
</tr>
<tr>
<td></td>
<td>ConfigurableTestSetup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PPTestResult</td>
<td>test.junit.extensions</td>
</tr>
<tr>
<td></td>
<td>PPTestRunner</td>
<td>test.junit.extensions.util</td>
</tr>
</tbody>
</table>
The Core Plugins

Java Cross Reference Plugin

- Transforms source files into HTML files
- Provides syntax coloring and hypertext linking
- Allows linking between reports and actual source code as done with
  - CheckStyle plugin
  - PMD plugin
Java Cross Reference Plugin

```java
35  public class ConfigurableTestCase extends TestCase implements Configur
36  
37  /** Cached configuration. */
38  protected Configuration conf;
39  
40  /**
41  * Creates instance of configurable test case.
42  *
43  * @param name Name of this test case.
44  */
45  public ConfigurableTestCase(String name) {
46      super(name);
47      ConfigurationFactory.init(this.getClass());
48  }
49  
50  /**
51  * Retrieves configuration of this test case.
52  */
```
The Core Plugins

JavaDoc Plugin

Package junit.extensions

Interface Summary

| ConfigurableTest | Test that provides access to configuration parameters. |

Class Summary

| ActiveTest       | A Decorator that runs a test in a separate thread. |
| ConfigurableTestCase | Base class for test cases which are configured using a configuration file instead of hardcoding test data in the fixture. |
| ConfigurableTestSetup | A Decorator to set up and tear down additional fixture state using configuration. |
| PPTestResult     | A PPTestResult collects the results of executing a test case and implements verbose output and delaying. |
### PMD Plugin

#### Avoid empty catch blocks

**Violation**

<table>
<thead>
<tr>
<th>Description</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid unused local variables such as 'result'</td>
<td>103</td>
</tr>
<tr>
<td>Avoid using 'if...else' statements without curly braces</td>
<td>137</td>
</tr>
<tr>
<td>Avoid using 'if...else' statements without curly braces</td>
<td>139</td>
</tr>
<tr>
<td>Avoid using 'if...else' statements without curly braces</td>
<td>141</td>
</tr>
<tr>
<td>Avoid using 'if...else' statements without curly braces</td>
<td>143</td>
</tr>
</tbody>
</table>

**junit\extensions\PPTestRunner.java**

<table>
<thead>
<tr>
<th>Description</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid unused local variables such as 'result'</td>
<td>112</td>
</tr>
</tbody>
</table>

**test\junit\extensions\ConfigurableTestCaseTest.java**

<table>
<thead>
<tr>
<th>Description</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid unused local variables such as 'result'</td>
<td>112</td>
</tr>
</tbody>
</table>
PMD Plugin

- **PMD** is a static source code analyzer
- **PMD** is rule driven
  - Basic rules
  - Import rules
  - Unused code rules
  - Naming rules
  - Design rules
- The plugin creates an HTML report with linking to the Source Cross Reference
Maven currently comes with more than 70 plugins

For additional functionality
  – Check out SourceForge and Google
  – Write your own plugin and contribute it ...

Some plugins can’t be shipped with Maven due to incompatible licences
  – **CLOVER** Code Coverage Analyzer
  – **SIMIAN** Code Similarity Analyzer
  – **Canoo WebTest** Functional WebTesting Tools
Custom Plugins

Clover Test Coverage Plugin

- Commercial software from *The Cortex*
- Instruments the *JUnit* test case code
- Generates a HTML test coverage report
  - Conditional Coverage
  - Statement Coverage
  - Method Coverage
- Integrated into generated project site through Maven

Another good reason for unit tests!!
Custom Plugins

Clover Test Coverage Plugin

http://www.thecortex.net/clover/eg/checkstyle/report/index.html - Microsoft Internet Explorer

Checkstyle 2.1 unit tests
Clover coverage report

Overview

All Classes
AuditEvent (89.9%)
CheckSMileTask (0%)
CheckSMileTask.For
CheckSMileTask.For
Checker (84.1%)
Configuration (51%)
DefaultLogger (47.69)
JarArchTag (83.9%)
LeftCurlyOption (42.6)
LineTest (91.7%)
Main (9%)
MethodSignature (91)
MyCommonAST (10C)
MyModifierSet (53%)
MyToken (100%)
MyVariable (100%)

Clover coverage report - Checkstyle 2.1 unit tests
Coverage Timestap: Fri Aug 30 2002 15:20:24 EST

Overview  Package  File

Overview

Conditionals  Statements  Methods TOTAL

Project

69.3%  63.3%  65.6%  65.2%

Packages

com.puppycrawl.tools.checkstyle

69.3%  63.3%  66.6%  65.2%

Report generated by Clover 1.0
Fri Aug 30 2002 15:23:15 EST.
Continuous Integration License

Stairway To Maven
**Clover Test Coverage Plugin**

```java
/**
 * Loads the contents of a file in a String array.
 * @return the lines in the file
 * @param aFileName the name of the file to load
 * @throws IOException error occurred
 */

private String[] getLines(String aFileName) throws IOException {
    final LineNumberReader lnr = new LineNumberReader(new FileReader(aFileName));
    final ArrayList lines = new ArrayList();
    while (true) {
        final String l = lnr.readLine();
        if (l == null) {
            break;
        }
        lines.add(l);
    }
    return (String[]) lines.toArray(new String[0]);
```
Custom Plugins

Simian Plugin

- Commercial tool from Redhill Consulting
- Code Similiarity Analyzer
- Generates a HTML report
- Integrated into generated project site through Maven
Simian Plugin

The Apache Jakarta Project
http://jakarta.apache.org/

Maven

Last published: 14 August 2003 16:17 +1000  |  Doc for 1.0-rc1-SNAPSHOT

Summary

Copyright (c) 2003 RedHill Consulting, Pty. Ltd. All rights reserved.

<table>
<thead>
<tr>
<th>Similarity threshold (lines)</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of duplicate lines</td>
<td>257</td>
</tr>
<tr>
<td>Total number of duplicate blocks</td>
<td>53</td>
</tr>
<tr>
<td>Total number of files with duplicates</td>
<td>10</td>
</tr>
<tr>
<td>Total number of processed lines</td>
<td>6099</td>
</tr>
<tr>
<td>Total number of processed files</td>
<td>126</td>
</tr>
<tr>
<td>Scan time</td>
<td>701ms</td>
</tr>
</tbody>
</table>

Duplications

Duplication: 11 lines
java.org.apache.maven.jelly.tags.maven.DependencyResolverTest (159 - 160)
java.org.apache.maven.jelly.tags.maven.DependencyResolverTest (177 - 187)
Canoo WebTest Plugin

- Open Source Project from Canoo Engineering AG
- Automates functional testing of webapps
- Test scripts are written in XML
  - Can be done by test engineers
- Integration in den Daily Build with ANT or Maven
- Maven integration comes with full reporting

No more excuses for skipping tests ...
## Maven Canoo WebTest Plugin - Canoo WebTest Results

### Canoo WebTest Summary Report

The following document contains the results of Canoo WebTest.

### Summary

<table>
<thead>
<tr>
<th>Test</th>
<th>Passed</th>
<th>Failed</th>
<th>Failed Rate</th>
<th>Duration (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0.25</td>
<td>19174</td>
</tr>
</tbody>
</table>

### Test Specs

<table>
<thead>
<tr>
<th>Test</th>
<th>Success</th>
<th>Start Time</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>TestSummary1-SimpleTest1.xml</td>
<td>yes</td>
<td>Fri Jul 18 11:45:45 CEST 2003</td>
<td>3297</td>
</tr>
<tr>
<td>TestSummary2-SimpleTest2.xml</td>
<td>yes</td>
<td>Fri Jul 18 11:45:46 CEST 2003</td>
<td>4735</td>
</tr>
<tr>
<td>TestSummary3-MailingListTest.xml</td>
<td>yes</td>
<td>Fri Jul 18 11:45:53 CEST 2003</td>
<td>7657</td>
</tr>
<tr>
<td>TestSummary4-BocusTest.xml</td>
<td>no</td>
<td>Fri Jul 18 11:46:01 CEST 2003</td>
<td>3485</td>
</tr>
</tbody>
</table>
Canoo WebTest Plugin

Maven Canoo Webtest Plugin - Canoo WebTest Results - Microsoft Internet Explorer

<table>
<thead>
<tr>
<th>Summary</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShowHtmlParserOutput</td>
<td>Yes</td>
</tr>
<tr>
<td>SaveResponse</td>
<td>Yes</td>
</tr>
<tr>
<td>Protocol</td>
<td>Http</td>
</tr>
<tr>
<td>Host</td>
<td>jakarta.apache.org</td>
</tr>
</tbody>
</table>

**Executed Test Steps**

<table>
<thead>
<tr>
<th>Nr</th>
<th>Name</th>
<th>Parameter</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>invoke</strong></td>
<td>url = index.html</td>
<td>1484</td>
</tr>
<tr>
<td></td>
<td>&quot;Go to the JAKARTA home page&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>verifyTitle</strong></td>
<td>type = title; regex = true; text = The Jakarta Site.*</td>
<td>453</td>
</tr>
<tr>
<td>3</td>
<td><strong>clicklink</strong></td>
<td>label = Turbine</td>
<td>266</td>
</tr>
<tr>
<td></td>
<td>&quot;Switch to the TURBINE home page&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>clicklink</strong></td>
<td>label = Maven</td>
<td>1094</td>
</tr>
<tr>
<td></td>
<td>&quot;Switch to the Maven home page&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Write Documentation ...

- Provide **XDOC** for developer documentation
  - Simple XML-based documentation
  - Can be transformed into HTML and PDF
- Simplified **DOCBOOK** for technical documents
  - Transformation to PDF using DOCBOOK plugin
- Stick to your existing **WORD** documents
  - The Word2HTML plugin converts WORD documents to HTML automatically
Multi-Project Support

- POM Inheritance Mechanism to simplify *mavenizing* multiple projects
  - Define common properties in the master POM
  - Add or override project specific properties

- Multiproject Plugin
  - Plugin triggers build of all dependent projects
  - Auto-detection of the required build order
Breaking the Ant Addiction

- Maven can do in 0 lines what ANT takes 50 lines for a simple project
- You can reuse ANT tasks in Maven
  - Calling ANT tasks directly from JELLY
  - Start a JVM running an ANT script
- You can start MAVEN from an ANT script
  - Retrofitting MAVEN into an existing ANT build
- You can always roll your own plugin
  - Enhydra XMLC plugin
  - Canoo Webtest plugin
The Maven Problems

- Incompatible directory structure?!
  - Maven can be tweaked to use a different directory layout
  - Could break plugins ...

- More than one source directories per project?!
  - Split the project into subprojects
  - Copy the source code in a `<preGoal>`
  - Add the additional classpath in a `<preGoal>`

- More than one artifact per project?!
  - Use multiple targets to create your artifacts
  - Register `<preGoals>` or `<postGoals>` to do the work
We don’t use version numbers for JARs?!  
  – Why is it good enough for third-party libraries?!  
  – You can always circumvent a missing version number using a constant, e.g. „LATEST“

Can we resurrect our existing ANT scripts?!  
  – Some ANT scripts are made obsolete by Maven  
  – Maven provides a ANT taglib calling the ANT tasks directly

Will Maven be legacy?!  
  – Maven was recently promoted to a top-level APACHE project such as ANT, Apache Webserver or Cocoon

Is Maven good enough for serious projects?!  
  – Check out the „Powered By Maven“
**Conclusion**

- **Maven** is an Open Source project build and comprehension tool developed by the [JAKARTA](https://jira.atlassian.com) community
- **Maven** is an integration platform for software engineering tools using a plug-in mechanism
- **Maven** provides a remote and local JAR repository to simplify upgrading of JARs
- **Maven** could be the answer to a lot of your software engineering problems
Any questions !?
<table>
<thead>
<tr>
<th>Related Links</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Maven</td>
<td><a href="http://jakarta.apache.org/turbine/maven/">http://jakarta.apache.org/turbine/maven/</a></td>
</tr>
<tr>
<td>JavaNCSS</td>
<td><a href="http://www.kclee.com/clemens/java/javancss/">http://www.kclee.com/clemens/java/javancss/</a></td>
</tr>
<tr>
<td>PMD</td>
<td><a href="http://pmd.sourceforge.net/">http://pmd.sourceforge.net/</a></td>
</tr>
<tr>
<td>Clover</td>
<td><a href="http://www.thecortex.net/clover/">http://www.thecortex.net/clover/</a></td>
</tr>
<tr>
<td>JDepend</td>
<td><a href="http://www.clarkware.com/software/JDepend.html">http://www.clarkware.com/software/JDepend.html</a></td>
</tr>
<tr>
<td>Jelly</td>
<td><a href="http://jakarta.apache.org/commons/sandbox/jelly/">http://jakarta.apache.org/commons/sandbox/jelly/</a></td>
</tr>
<tr>
<td>Maven Plug-ins</td>
<td><a href="http://maven-plugins.sourceforge.net">http://maven-plugins.sourceforge.net</a></td>
</tr>
<tr>
<td>ANT</td>
<td><a href="http://jakarta.apache.org/ant/index.html">http://jakarta.apache.org/ant/index.html</a></td>
</tr>
</tbody>
</table>