



Quality-Driven Build Scripts for Java Applications

Duy (uu-eee) B. Vo
Graduate Student
San José State University
Department of Computer Science



Some Measures of a Quality Software Product

- Is the product thoroughly tested?
- Is the product well-documented?
- Is the coding style consistent?
- Is the product well-designed?



Use Cases

- Supports Test-Driven Development
- Provides information about the product
- Enforces organizational policies



Quality-Driven Build Script Defined

“A build script that fosters efforts to produce quality code.”

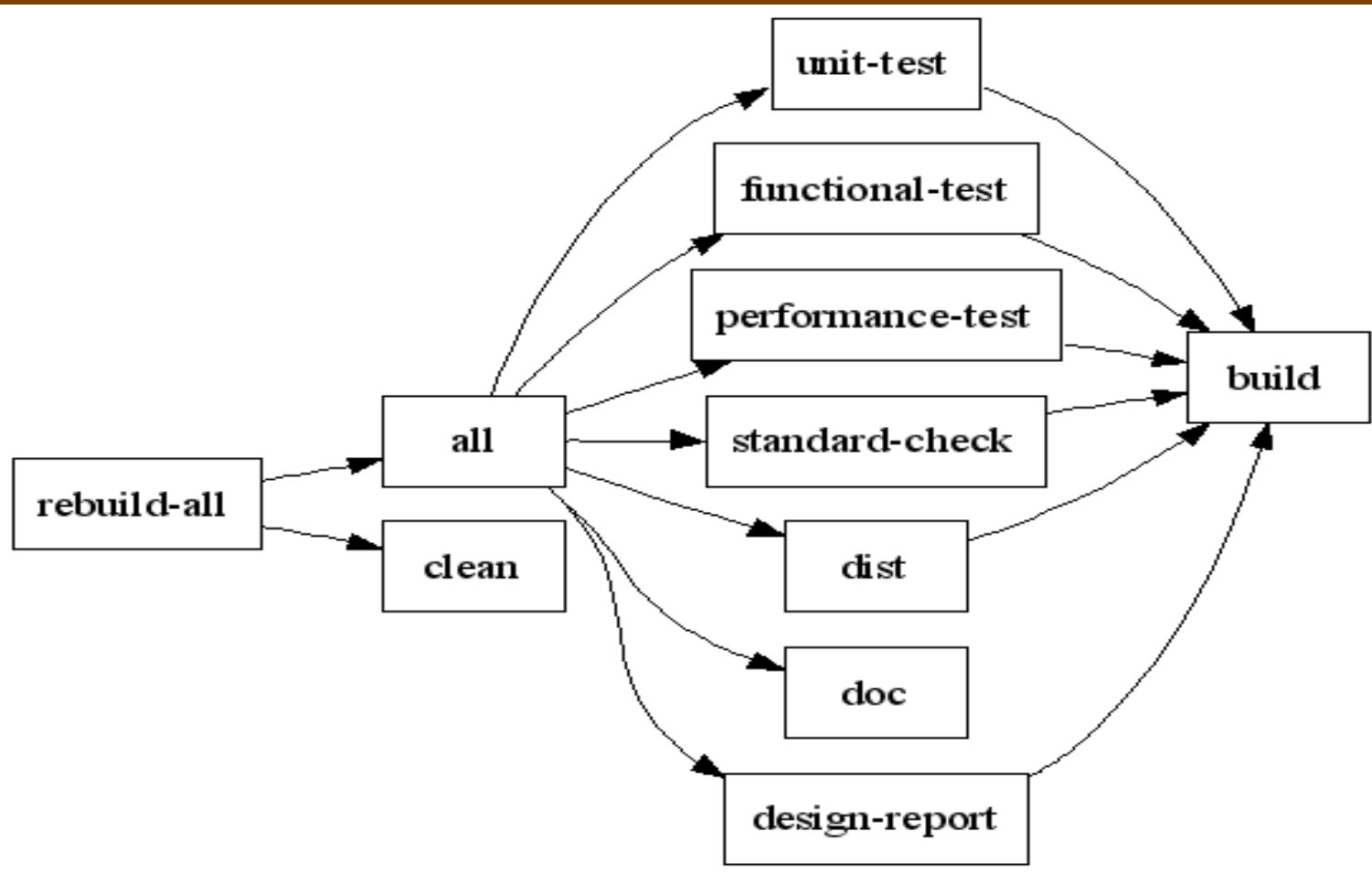


Implementation - Ant (v 1.65)

- Easy to learn
- Easy to implement
- Feature-rich
- Widely-supported
- Platform-independent
- Our case study will be Cobertura



Build Target Structure





Build Target Definitions (Support)

- ***rebuild-all*** - explicitly removes all derived objects and executes all targets
- ***clean*** - explicitly removes all derived objects
- ***all*** – executes all targets without explicitly removing derived objects
- ***dist*** – prepares our product for distribution
- ***build*** (default target) – compiles our code incrementally



Build Target Definitions (Main)

- ***standard-check*** – enforces architectural, coding, and documentation guidelines
- ***dist*** – prepares our product for distribution
- ***functional-test*** – executes integrated tests and generates reports
- ***performance-test*** – executes usability tests on our product and generates reports
- ***unit-test*** – instruments production code, executes unit tests, and generates reports
- ***doc*** – generates documentation for this project
- ***design-report*** – generates metrics about design information including package and class dependencies



Use Case 1

- Implement unit tests with JUnit framework and extensions
- Functional tests should be implemented based on various requirements
- Performance tests should be implemented based on various requirements
- Reports should be generated



JUnit Details Report

```
<property name="path.src.doc" value="src/doc"/>
<property name="java.ext.dirs" value="C:\Program Files\Java\jre 1.5.0_06\lib\ext"/>
<property name="user.dir" value="D:\workspace\Cobertura"/>
<property name="line.separator" value=" "/>
<property name="java.vm.name" value="Java HotSpot(TM) Client VM"/>
<property name="env.PATHEXT" value=".COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH"/>
<property name="env.Path" value="%PATH%;C:\Program Files\Java\jdk 1.5.0_06\bin;C:\Program Files\ThinkPad\Utilities;C:\WINDOWS\system32;C:\WI
<property name="basedir" value="D:\workspace\Cobertura"/>
<property name="ant.java.version" value="1.5"/>
<property name="env.USERNAME" value="duy"/>
<property name="env.SystemDrive" value="C:"/>
<property name="file.encoding" value="Cp1252"/>
<property name="path.junit-details" value="build/doc/JUnitDetails"/>
<property name="env.USERDOMAIN" value="BETSY5"/>
<property name="path.build.test" value="build/test-classes"/>
<property name="java.specification.version" value="1.5"/>
<property name="env.PROCESSOR_LEVEL" value="6"/>
</properties>
<testcase classname="net.sourceforge.cobertura.merge.MergeMainTest" name="testNewDestinationFile" time="0.29"/>
<testcase classname="net.sourceforge.cobertura.merge.MergeMainTest" name="testExistingDestinationFile" time="0.021"/>
<testcase classname="net.sourceforge.cobertura.merge.MergeMainTest" name="testBaseDir" time="0.03"/>
- <system-out>
  Cobertura: Saved information on 1 classes. Cobertura: Saved information on 2 classes. Cobertura: Loaded information on 38 classes. Cobertura null - GNU GI
  GPL License (NO WARRANTY) - See COPYRIGHT file Cobertura: Loaded information on 1 classes. Cobertura: Loaded information on 1 classes. Cobertu
</system-out>
<system-err></system-err>
</testsuite>
```



JUnit Aggregate Report

[Home](#)

Packages

[net.sourceforge.cobertura.coverage](#)
[net.sourceforge.cobertura.merge](#)
[net.sourceforge.cobertura.reporting](#)
[net.sourceforge.cobertura.reporting](#)
[net.sourceforge.cobertura.reporting](#)

Classes

[ClassDataTest](#)
[CommandLineBuilderTest](#)
[ComplexityCalculatorTest](#)
[CopyFilesTest](#)
[CoverageDataFileHandlerTest](#)

Unit Test Results

Designed for use with [JUnit](#) and [Ant](#).

Summary

Tests	Failures	Errors	Success rate	Time
65	1	0	98.46%	34.742

Note: *failures* are anticipated and checked for with assertions while *errors* are unanticipated.

Packages

Name	Tests	Errors	Failures	Time(s)
net.sourceforge.cobertura.coveragedata	25	0	0	1.463
net.sourceforge.cobertura.merge	10	0	0	0.832
net.sourceforge.cobertura.reporting	8	0	0	0.781



JUnit Implementation

```
<junit fork="yes" haltonfailure="false">
  <sysproperty key="net.sourceforge.cobertura.datafile"
    file="${path.build.instrument}/data.ser"/>
  <classpath>
    <pathelement location="${path.build.instrument}"/>
    <pathelement location="${path.build.classes}"/>
    <pathelement location="${path.build.test}"/>
    <path refid="project.classpath"/>
  </classpath>
  <formatter type="xml"/>
  <batchtest todir="${path.build.test-reports}/JUnitDetails">
    <fileset dir="${path.src.test}" includes="**/*Test.java"/>
  </batchtest>
  <jvmarg value="-Djava.compiler=NONE"/>
</junit>
```



JUnit Reports Implementation

```
<junitreport
  todir="${path.build.test-reports}/JUnitReport">
  <fileset
    dir="${path.build.test-reports}/JUnitDetails"
    includes="*TEST-*.xml"/>
  <report format="frames"
    todir="${path.build.test-reports}/JUnitReport"/>
</junitreport>
```



Cobertura Report

Packages

[All](#)

[net.sourceforge.cobertura.ant](#)

[net.sourceforge.cobertura.check](#)

[net.sourceforge.cobertura.coveragedata](#)

[net.sourceforge.cobertura.instrument](#)

[net.sourceforge.cobertura.merge](#)

[net.sourceforge.cobertura.reporting.html](#)

[net.sourceforge.cobertura.reporting.html.files](#)

[net.sourceforge.cobertura.reporting.xml](#)

[net.sourceforge.cobertura.util](#)

All Packages

Classes

[CheckTask \(0%\)](#)

[ClassData \(N/A\)](#)

[ClassInstrumenter \(0%\)](#)

Coverage Report - All Packages

Package ^	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	36	55%	60%	2.54
net.sourceforge.cobertura.ant	7	0%	0%	1.851
net.sourceforge.cobertura.check	2	0%	0%	3.2
net.sourceforge.cobertura.coveragedata	10	N/A	N/A	2.196
net.sourceforge.cobertura.instrument	4	0%	0%	2.897
net.sourceforge.cobertura.merge	1	86%	100%	5.5
net.sourceforge.cobertura.reporting	2	81%	96%	2.882
net.sourceforge.cobertura.reporting.html	3	81%	90%	4.5
net.sourceforge.cobertura.reporting.html.files	1	87%	100%	4.5
net.sourceforge.cobertura.reporting.xml	1	100%	100%	1.421
net.sourceforge.cobertura.util	5	91%	100%	3.105

Report generated by [Cobertura](#) 1.7 on 12/21/05 2:27 AM.



Cobertura Implementation

```
<cobertura-report format="html"  
  datafile="${path.build.instrument}/data.ser"  
  destdir="${path.build.test-reports}/Cobertura">  
  <fileset  
    dir="${path.src.java}"  
    includes="**/*.java"/>  
</cobertura-report>
```



Use Case 2

- API Documentation
- Dependency analysis
- Architectural / Design Documentations
- UML / State Diagrams



Javadoc

[All Classes](#)

Packages

- [net.sourceforge.cobertura.a](#)
- [net.sourceforge.cobertura.c](#)
- [net.sourceforge.cobertura.c](#)
- [net.sourceforge.cobertura.ir](#)
- [net.sourceforge.cobertura.n](#)

All Classes

- [CheckTask](#)
- [ClassData](#)
- [CommandLineBuilder](#)
- [CommonMatchingTask](#)
- [ComplexityCalculator](#)
- [CopyFiles](#)
- [CoverageData](#)
- [CoverageDataContainer](#)
- [CoverageDataFileHandler](#)
- [CoverageRate](#)
- [CustomAttribute](#)
- [FileFinder](#)
- [HasBeenInstrumented](#)
- [Header](#)
- [HTMLReport](#)
- [Ignore](#)
- [InstrumentTask](#)
- [IOUtil](#)
- [JavaToHtml](#)
- [JavaToHtml.State](#)
- [LineData](#)
- [Main](#)

Overview [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#) [FRAMES](#) [NO FRAMES](#)

Packages

net.sourceforge.cobertura.ant	The ant tasks for Cobertura and assorted helper classes.
net.sourceforge.cobertura.check	
net.sourceforge.cobertura.coveragedata	A hierarchy of classes used for storing metadata about each of your classes.
net.sourceforge.cobertura.instrument	
net.sourceforge.cobertura.merge	
net.sourceforge.cobertura.reporting	
net.sourceforge.cobertura.reporting.html	
net.sourceforge.cobertura.reporting.html.files	
net.sourceforge.cobertura.reporting.xml	
net.sourceforge.cobertura.util	

Overview [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#) [FRAMES](#) [NO FRAMES](#)



Javadoc Implementation

```
<javadoc version="true"
  destdir="${path.build.doc}/APIDocs" author="true"
  classpathref="project.classpath"
  sourcepath="${path.src.java}"
  packagenames="net.sourceforge.cobertura.*">
  <link offline="true" href="http://ant.apache.org/manual/api/"
    packagelistLoc="${path.src.doc}/ant-javadoc"/>
  <link href="http://java.sun.com/j2se/1.4.2/docs/api"/>
</javadoc>
```



Some Package-Level Dependency Metrics – A Tangential Discussion

- Defined by Robert Martin in “OO Design Quality Metrics”
- Metrics on packages
- Count of abstract classes and interfaces (A_c)
- Count of concrete classes (C_c)
- Afferent coupling (C_a) metric quantifies the package’s responsibility
- Efferent coupling (C_e) metric quantifies the package’s dependence
- Instability defined as $I = C_e / (C_a + C_e)$
- Abstractness defined as $A = A_c / (A_c + C_c)$
- Cycle counts metric determines cyclic dependencies
- The Euclidean distance from the ideal line defined as $A(I) = 1 - I$.
Ideally, a package with many responsibilities should be stable (high A_c) whereas a package with no responsibility can be unstable (High C_c)



Some Class-Level Dependency Metrics – A Tangential Discussion

- McCabe's Cyclomatic Complexity
 - Complexity information that determines the code paths of all methods within a class. The higher the complexity count, the more code paths the class has
- Class-level couplings and dependencies
- Class layer information

JDepend HTML Report

JDepend Analysis

Designed for use with [JDepend](#) and [Ant](#).

Summary

[\[summary\]](#) [\[packages\]](#) [\[cycles\]](#) [\[explanations\]](#)

Package	Total Classes	Abstract Classes	Concrete Classes	Afferent Couplings	Efferent Couplings	Abstractness	Instability	Distance	
net.sourceforge.cobertura.ant	7	1	6	6	0	1	0.14	1	0.14
net.sourceforge.cobertura.check	2	0	2	2	0	2	0	1	0
net.sourceforge.cobertura.coveragedata	10	4	6	6	6	1	0.4	0.14	0.46
net.sourceforge.cobertura.instrument	4	0	4	4	0	2	0	1	0
net.sourceforge.cobertura.merge	1	0	1	1	0	2	0	1	0
net.sourceforge.cobertura.reporting	3	0	3	3	2	4	0	0.67	0.33
net.sourceforge.cobertura.reporting.html	4	1	3	3	1	4	0.25	0.8	0.05
net.sourceforge.cobertura.reporting.html.files	1	1	0	0	1	0	1	0	0
net.sourceforge.cobertura.reporting.xml	1	0	1	1	1	3	0	0.75	0.25
net.sourceforge.cobertura.util	5	3	2	2	8	0	0.6	0	0.4

Packages

[\[summary\]](#) [\[packages\]](#) [\[cycles\]](#) [\[explanations\]](#)

net.sourceforge.cobertura.ant

[Afferent Couplings](#): 0

[Efferent Couplings](#): 1

[Abstractness](#): 0.14

[Instability](#): 1

[Distance](#): 0.14

Abstract Classes	Concrete Classes	Used by Packages	Uses Packages
net.sourceforge.cobertura.ant.CommonMatchingTask	net.sourceforge.cobertura.ant.CheckTask net.sourceforge.cobertura.ant.Ignore net.sourceforge.cobertura.ant.InstrumentTask net.sourceforge.cobertura.ant.MergeTask net.sourceforge.cobertura.ant.Regex net.sourceforge.cobertura.ant.ReportTask	None	net.sourceforge.cobertura.util

net.sourceforge.cobertura.check

[Afferent Couplings](#): 0

[Efferent Couplings](#): 2

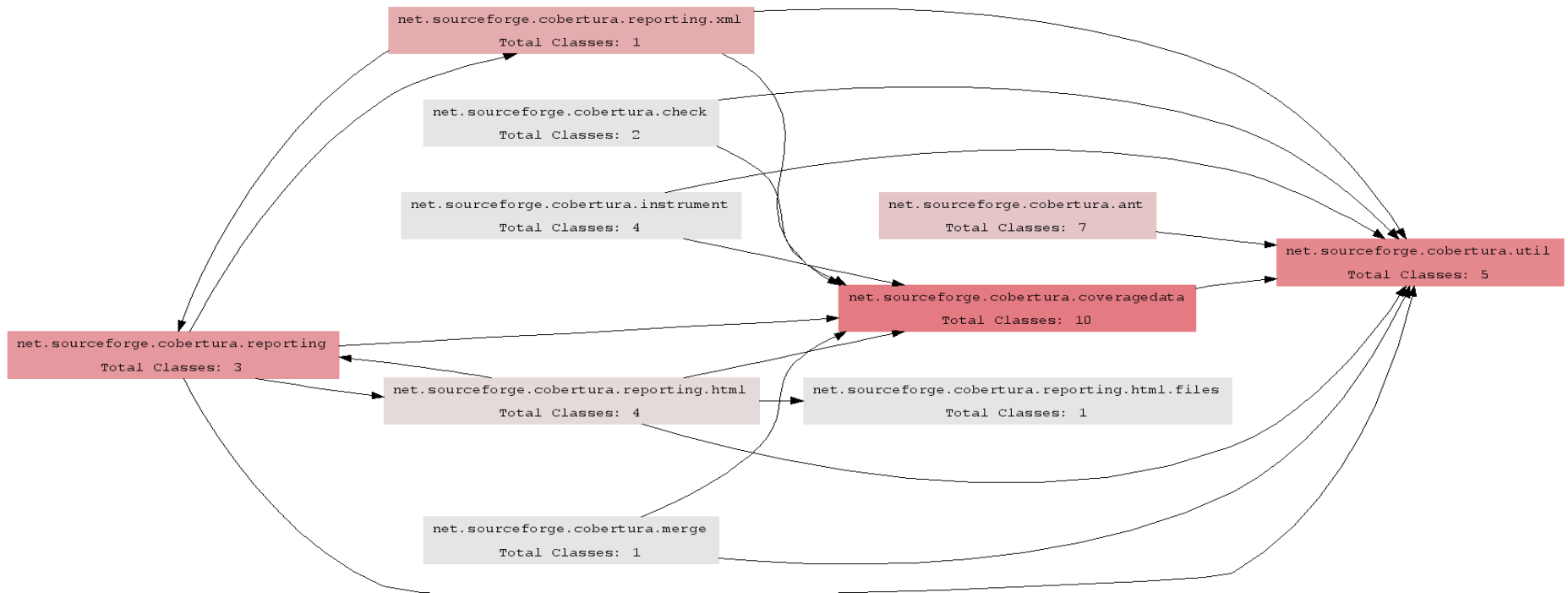
[Abstractness](#): 0

[Instability](#): 1

[Distance](#): 0

Abstract Classes	Concrete Classes	Used by Packages	Uses Packages
None	net.sourceforge.cobertura.check.CoverageRate	None	net.sourceforge.cobertura.coveragedata

JDepend PNG Report








JDepend Implementation





















```
<jdepend
  outputfile="${path.jdepend}/jdepend-report.xml"
  format="xml" haltonerror="yes">
  <exclude name="java.*"/>
  <exclude name="javancss*" />
  <exclude name="org.*" />
  <classpath>
    <pathelement location="${path.build.classes}" />
  </classpath>
</jdepend>
```



Classycle Report

Classes and Packages

Click on  or  to go to the cycle to which the class/package belongs.
Click on  behind a number and a window will pop up showing more details.

Class	Size	Used by	Uses internal	Uses external	Layer
 net.sourceforge.cobertura.ant	7	1 	2 	7 	1
 net.sourceforge.cobertura.ant.CheckTask	4917	0 	1 	21 	1
 net.sourceforge.cobertura.ant.CommonMatchingTask	5127	3 	2 	21 	1
 net.sourceforge.cobertura.ant.Ignore	556	1 	0 	2 	0
 net.sourceforge.cobertura.ant.InstrumentTask	2573	0 	3 	8 	2



Classycle Implementation

```
<taskdef name="classycleReport" classpath="${path.lib.dev}/classycle.jar"
  classname="classycle.ant.ReportTask"/>
<classycleReport reportfile="${path.classycle}/classycle.xml"
  title="${project.name}">
  <fileset dir="${path.build.classes}" includes="**/*.class"/>
</classycleReport>
<copy todir="${path.classycle}/images">
  <fileset dir="${path.src.doc}/classycle-images" ncludes="*.png"/>
</copy>
<xslt in="${path.classycle}/classycle.xml"
  out="${path.classycle}/classycle.htm"
  style="${path.src.doc}/reportXMLtoHTML.xsl"/>
```



Use Case 3

- Enforcing Documentation Standards
- Enforcing Coding Standards
- Enforcing Architectural Rules



Doc Check Doclet

Category 1: Class Error

Class CoverageRate

- Class missing comment.

```
/**
 *
 *
 * @author unattributed
 * @version %I% %G%
 */
```

Category 2: Member Error

CoverageRate(double, double)

- Constructor missing comment.

```
/**
 * Constructs a CoverageRate object.
 *
 * @param lineCoverageRate a double
 * @param branchCoverageRate a double
 */
```

getLineCoverageRate()

- Method missing comment, but reasonably self-evident.

```
/**
 * Returns the line coverage rate value.
 */
```

getBranchCoverageRate()

- Method missing comment, but reasonably self-evident.

```
/**
 * Returns the branch coverage rate value.
```



Documentation Check Implementation

```
<javadoc destdir="{path.doccheck}" Locale="en US"  
version="true" author="true" classpathref="project.classpath">  
  <doclet  
    name="com.sun.tools.doclets.doccheck.DocCheck"  
    path="{path.lib}/doccheck.jar">  
      <param name="-title" value="Cobertura Case Study Doc Check"/>  
    </doclet>  
    <fileset dir="{path.src.java}"/>  
  </javadoc>
```



Checkstyle Report

Files

Summary

[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)
[D:\workspace_pcm\Cobertura\...](#)

CheckStyle Audit

Designed for use with [CheckStyle](#) and [Ant](#).

File **D:\workspace_pcm\CoberturaCaseStudy\src\javasrc\net\sourceforge\cobertura\ant\CheckTask.java**

Error Description

Error Description	Line
File does not end with a newline.	0
'{' should be on the previous line.	80
Line contains a tab character.	82
Missing a Javadoc comment.	82
Line contains a tab character.	84
Missing a Javadoc comment.	84
Variable 'regexes' must be private and have accessor methods.	84
Line contains a tab character.	86
Missing a Javadoc comment.	86
Line contains a tab character.	88
Missing a Javadoc comment.	88
Line contains a tab character.	90
Missing a Javadoc comment.	90
Line contains a tab character.	92
Missing a Javadoc comment.	92
Line contains a tab character.	94



Coding Guideline Check Implementation

```
<taskdef resource="checkstyletask.properties"  
  classpath="${path.lib}/checkstyle-all-4.1.jar"/>  
<checkstyle config="${path.src.doc}/sun checks.xml"  
  failureProperty="checkstyle.failure" failOnViolation="false">  
  <fileset dir="${path.src.java}" includes="**/*.java"/>  
  <formatter type="xml" toFile="${path.checkstyle}/checkstyle.xml"/>  
</checkstyle>  
<xslt in="${path.checkstyle}/checkstyle.xml"  
  out="${path.checkstyle}/checkstyle report.htm"  
  style="${path.src.doc}/checkstyle-frames.xsl"/>
```



Macker

```
<?xml version="1.0"?>
<macker>
  <ruleset name="example">
    <access-rule>
      <deny>
        <from class="**Print*" />
        <to class="java.**" />
      </deny>
    </access-rule>
  </ruleset>
</macker>
```

Checking ruleset: example ...

Illegal reference
from PrintHelloWorld
to java.io.PrintStream

Illegal reference
from PrintHelloWorld
to java.lang.Object

Illegal reference
from PrintHelloWorld
to java.lang.System



Summary

- Build scripts should facilitate development of quality code
 - TDD
 - Information about the product
 - Enforce guidelines
- Rolling out your own
 - Explore Ant's core and optional tasks
 - When you need a tool, look for an open-source utilities
 - Integrate important metrics and tasks into the build process

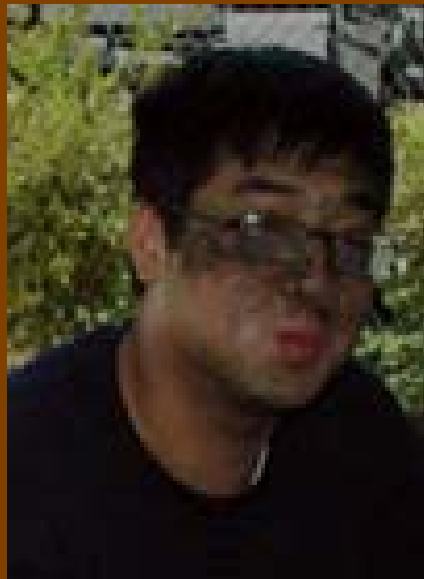


Tools

- **Ant** version 1.6.5 <<http://ant.apache.org>>
- **Checkstyle** version 4.1 <<http://checkstyle.sourceforge.net>>
- **Classycle** version 1.2 <<http://classycle.sourceforge.net>>
- **Cobertura** version 1.8 <<http://cobertura.sourceforge.net>>
- **Doc Check Doclet** version 1.2 Beta 2
<<http://java.sun.com/j2se/javadoc/doccheck>>
- **Java Development Kit** version 5 update 6
<<http://java.sun.com/j2se/1.5.0/download.js>>
- **JDepend** version 2.9 <<http://clarkware.com/software>>
- **JUnit** version 4.1 <<http://www.junit.org/index.htm>>
- **Graphviz** version 2.6.1 <<http://www.graphviz.org>>
- **Vizant** version 0.1.1 <<http://vizant.sourceforge.net>>



Contact Information



- Duy B. Vo
- dvo@email.sjsu.edu
- 408.839.1116



00000011

00011111