Simplifying builds with Gradle

Saager Mhatre

6TH IndicThreads.com Conference On JAVA

2,3 DEC 2011

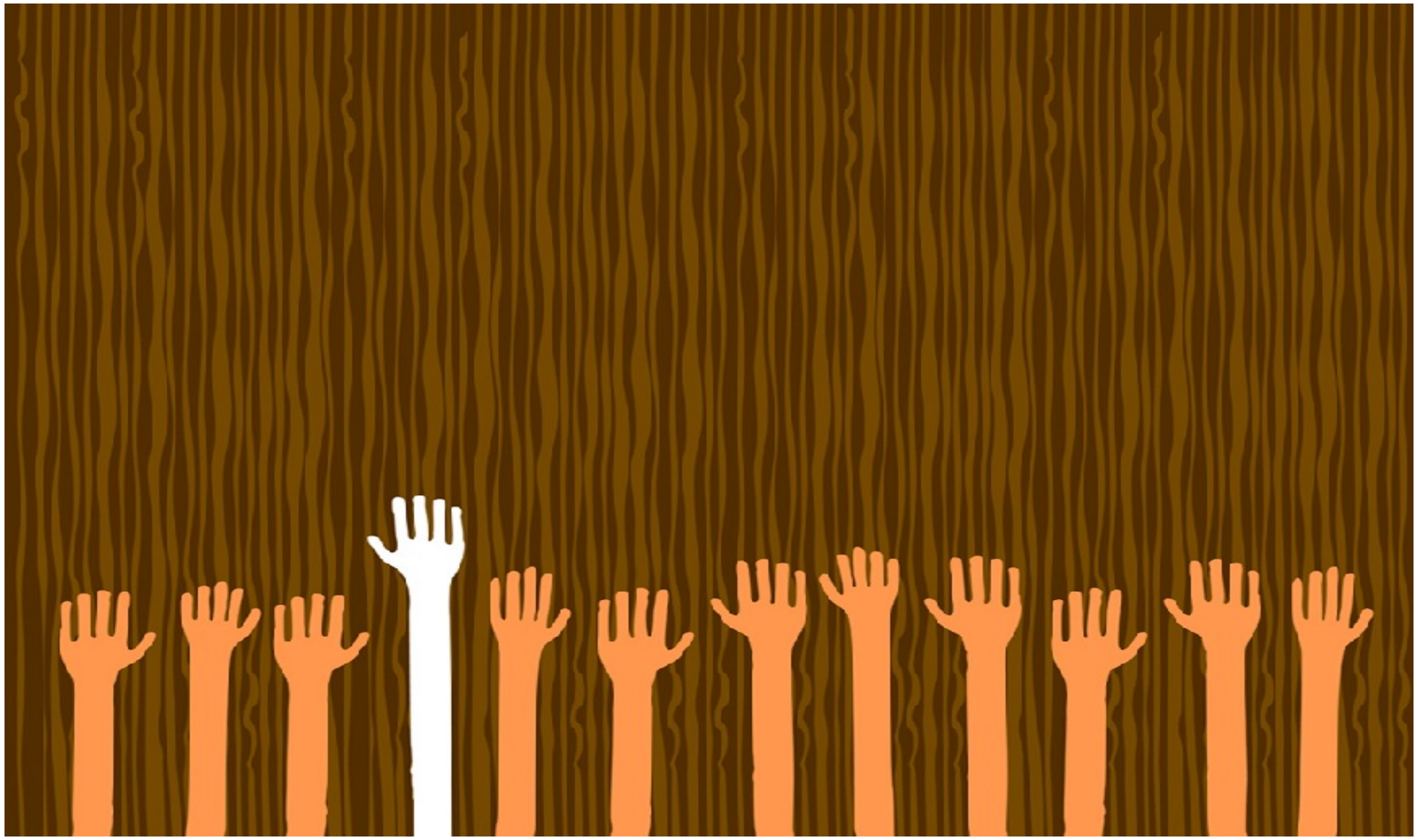
PUNE, INDIA

dexterous

errata

simplifying builds ^script with gradle

disclaimer



simplifying build scripts with gradle

build script

build

source $\longrightarrow$ artifacts

source $\rightarrow$ artifacts

source $\Rightarrow$ artifacts

code
configuration

$$\text{source} \longrightarrow \text{artifacts}$$

code

configuration

$$\text{source} \xrightarrow{\quad} \text{artifacts}$$

code
configuration

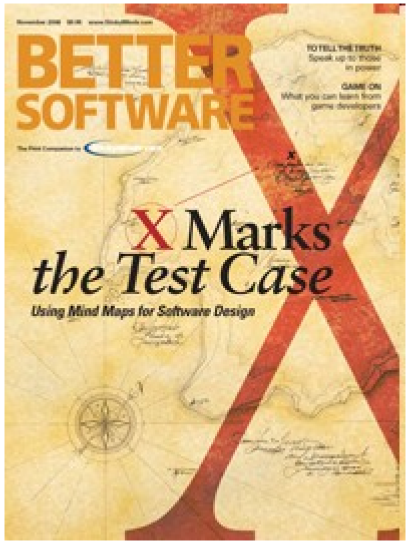
classes
generated code

script

more expressiveness
less ceremony

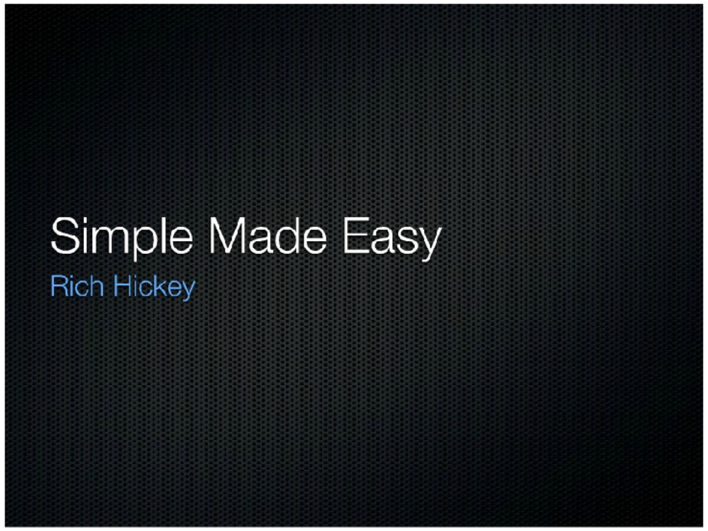
simplify

simplify... ?


BETTER SOFTWARE
TO TELL THE TRUTH Speak up to those in power
GAME ON What you can learn from game developers
X Marks the Test Case
Using Mind Maps for Software Design

simplify… ?

http://www.stickyminds.com/s.asp?F=S11633_MAGAZINE_2



Simple Made Easy

Rich Hickey

simplify... ?

http://www.infoq.com/presentations/Simple-Made-Easy

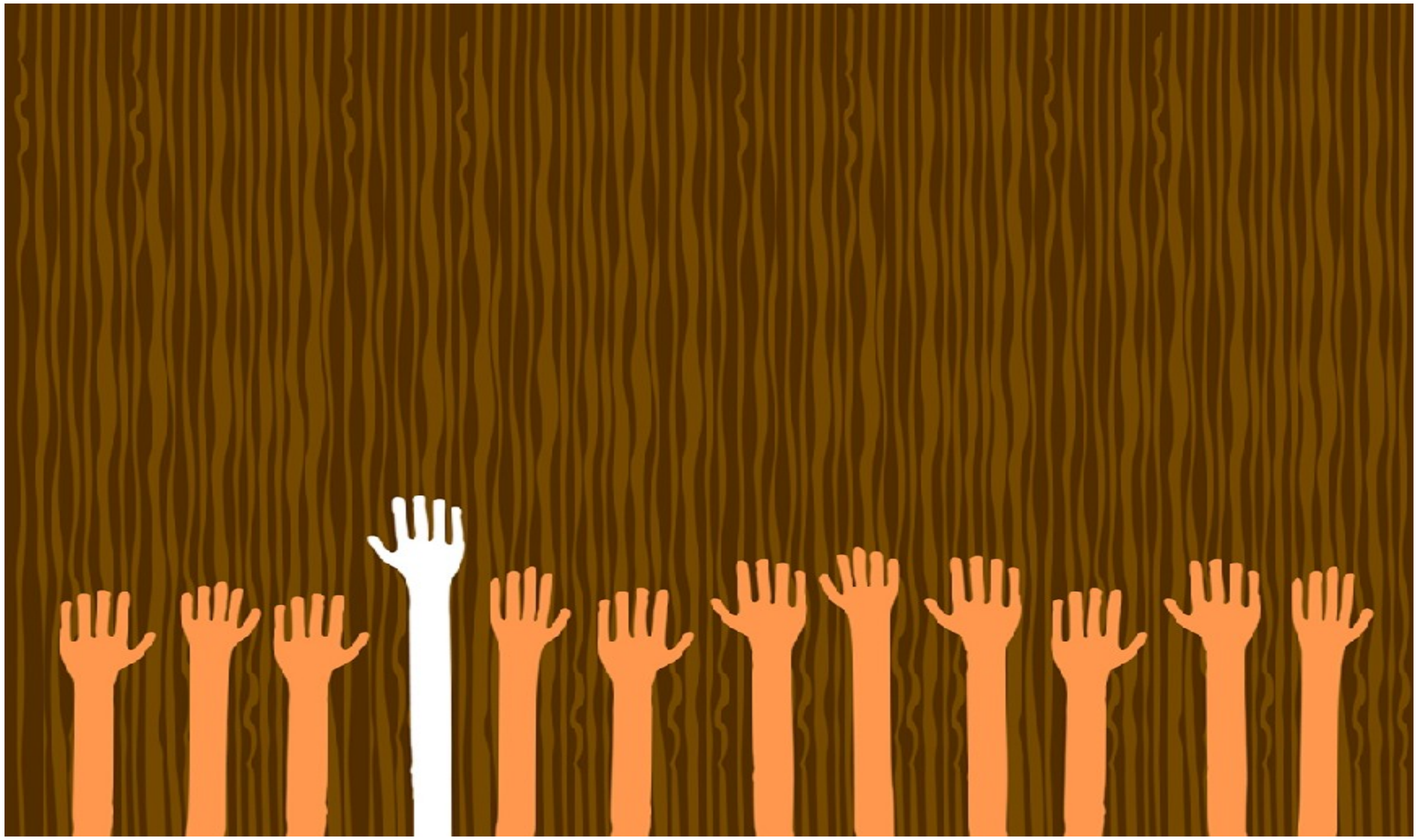
Clojure conj

(:when "October 2010"
:where "Durham, NC"
:who (+ me others))

Stuart Halloway
"Simplicity Ain't Easy"

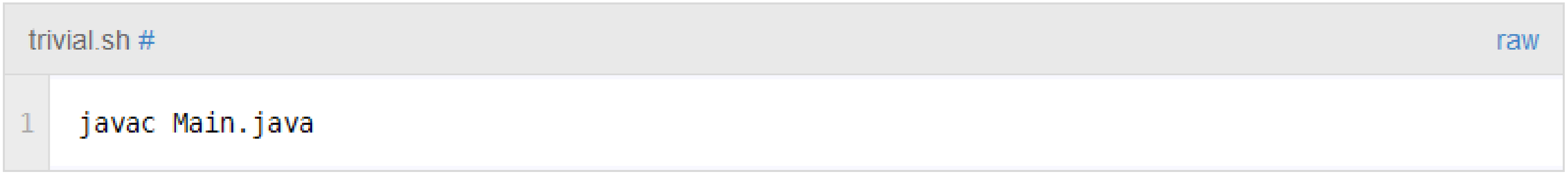
simplify... ?

http://blip.tv/clojure/stuart-halloway-simplicity-ain-t-easy-4842694

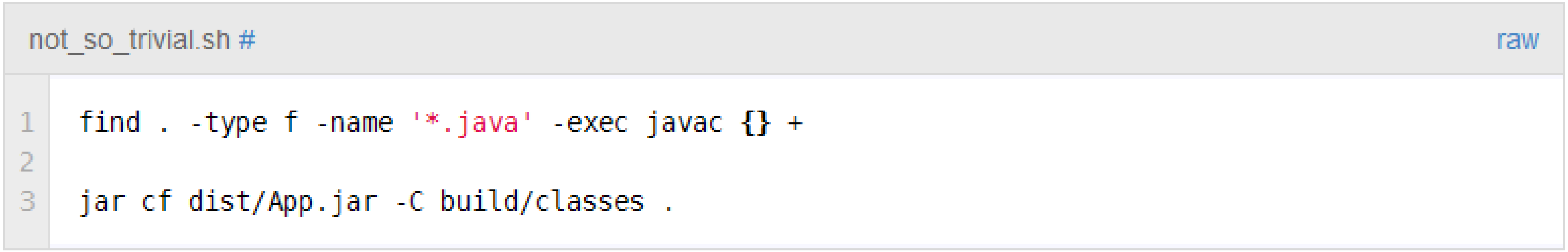


enough talk

let's build something



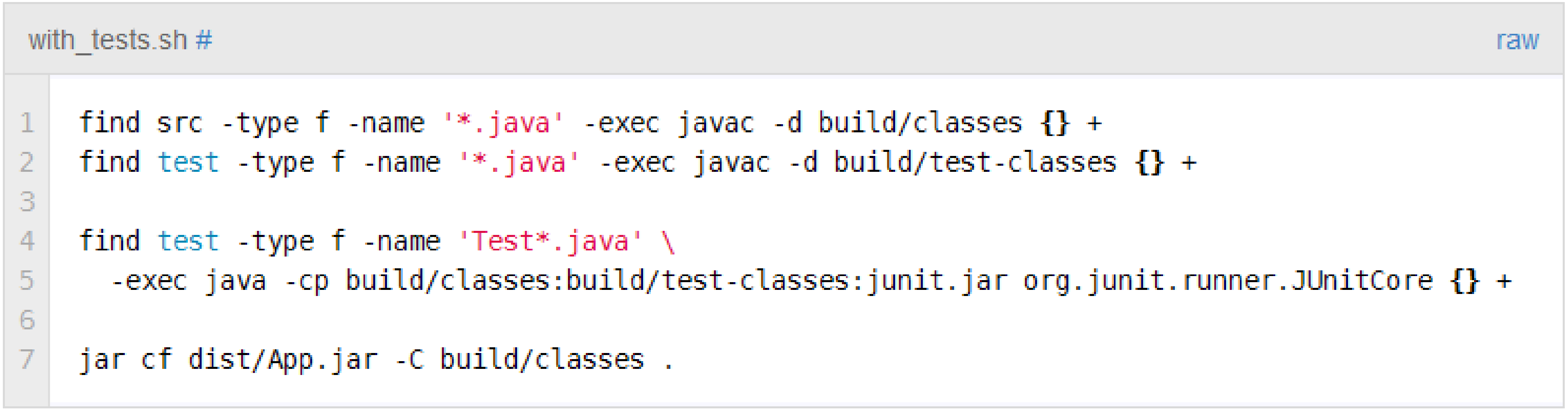
```
javac Main.java
```



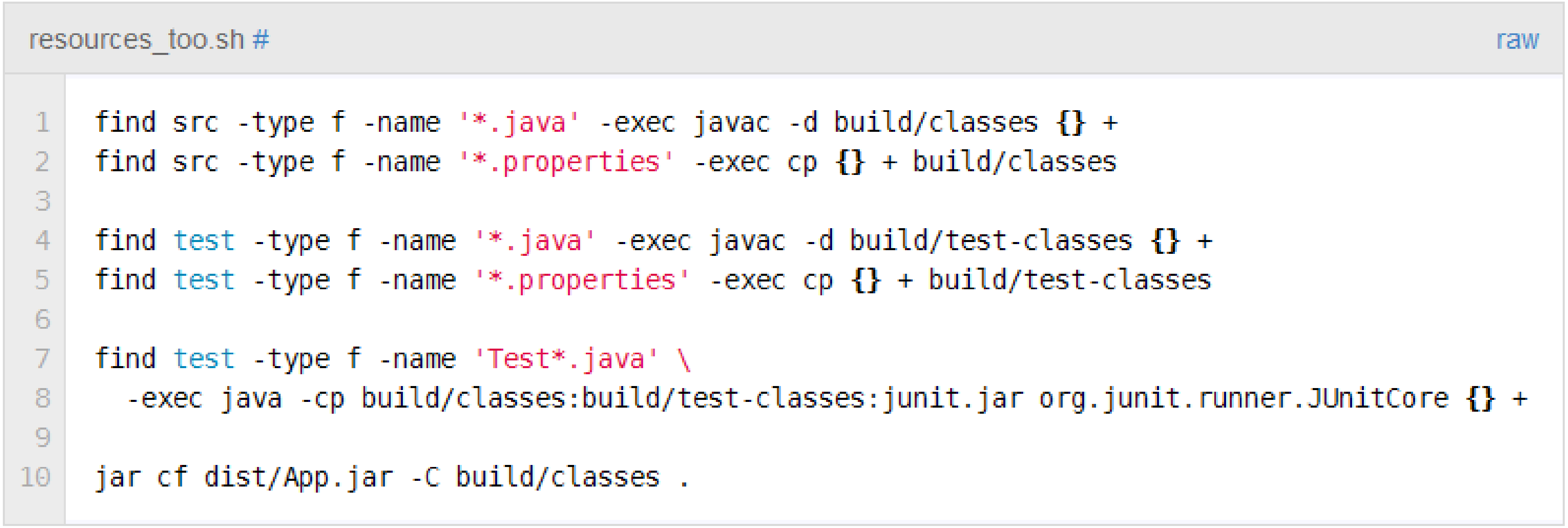
not_so_trivial.sh #

raw

```
find . -type f -name '*.java' -exec javac {} +

jar cf dist/App.jar -C build/classes .
```


with_tests.sh # raw

```
find src -type f -name '*.java' -exec javac -d build/classes {} +
find test -type f -name '*.java' -exec javac -d build/test-classes {} +

find test -type f -name 'Test*.java' \
  -exec java -cp build/classes:build/test-classes:junit.jar org.junit.runner.JUnitCore {} +

jar cf dist/App.jar -C build/classes .
```



resources_too.sh # raw

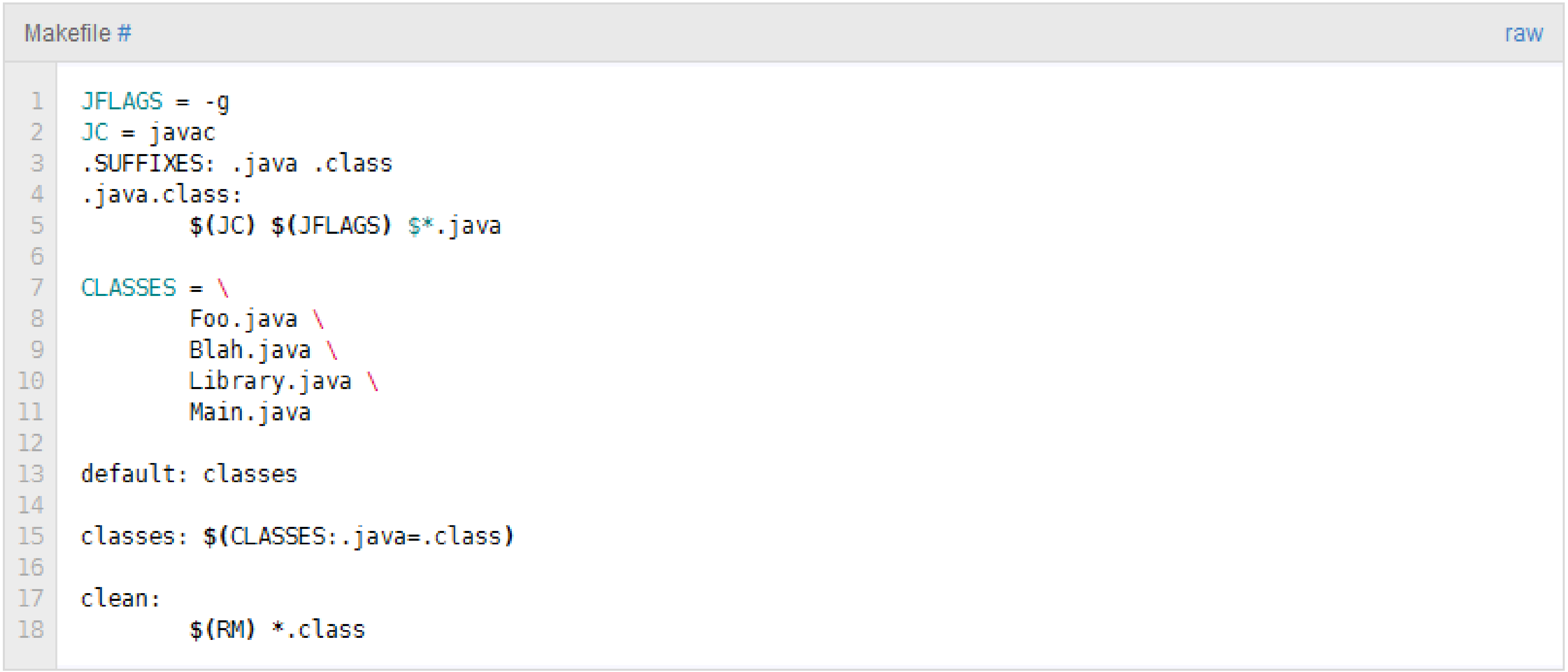
```
find src -type f -name '*.java' -exec javac -d build/classes {} +
find src -type f -name '*.properties' -exec cp {} + build/classes

find test -type f -name '*.java' -exec javac -d build/test-classes {} +
find test -type f -name '*.properties' -exec cp {} + build/test-classes

find test -type f -name 'Test*.java' \
  -exec java -cp build/classes:build/test-classes:junit.jar org.junit.runner.JUnitCore {} +

jar cf dist/App.jar -C build/classes .
```

conditional execution

static analysis

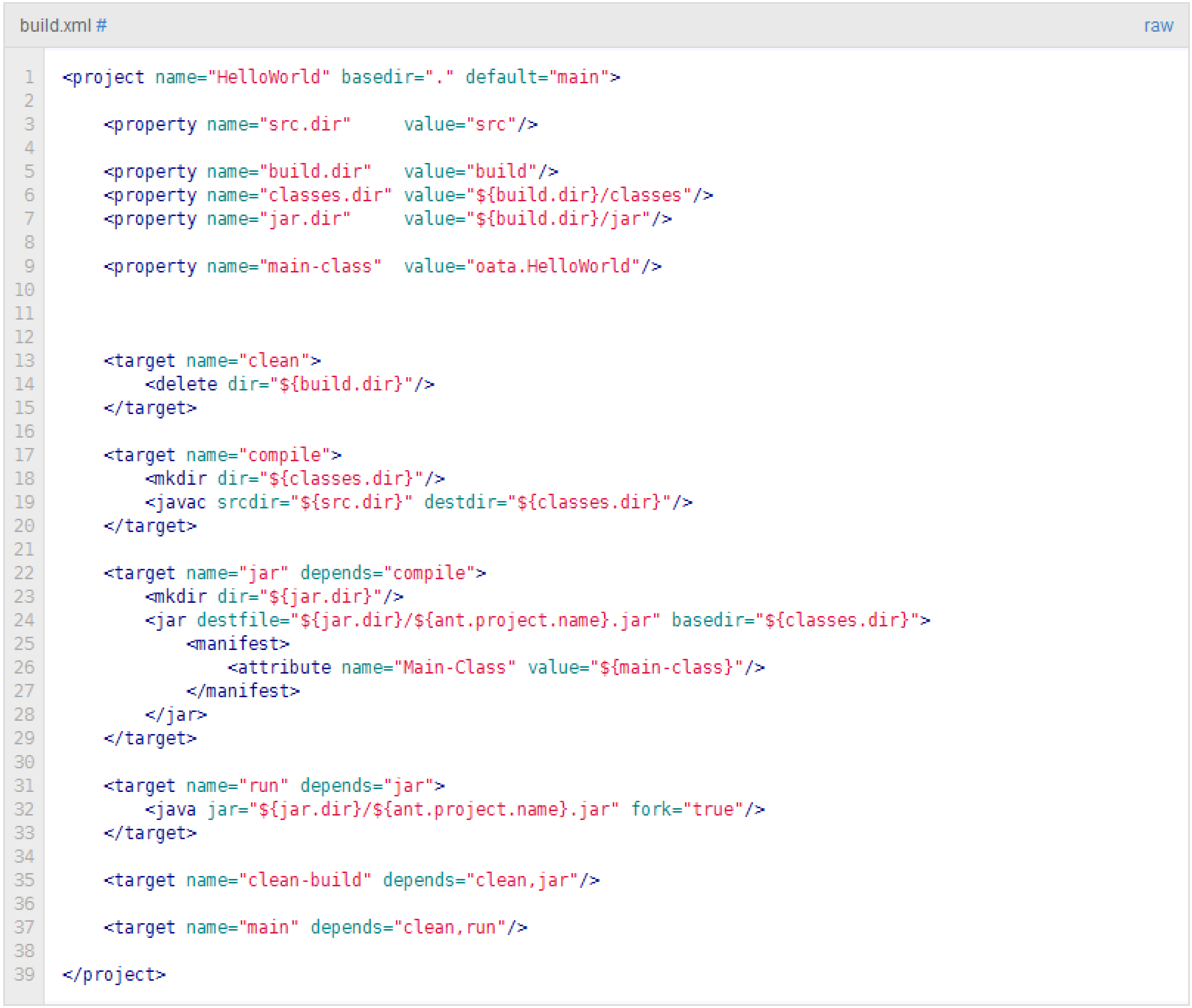
execution profiling

insanity ensues


Makefile # raw

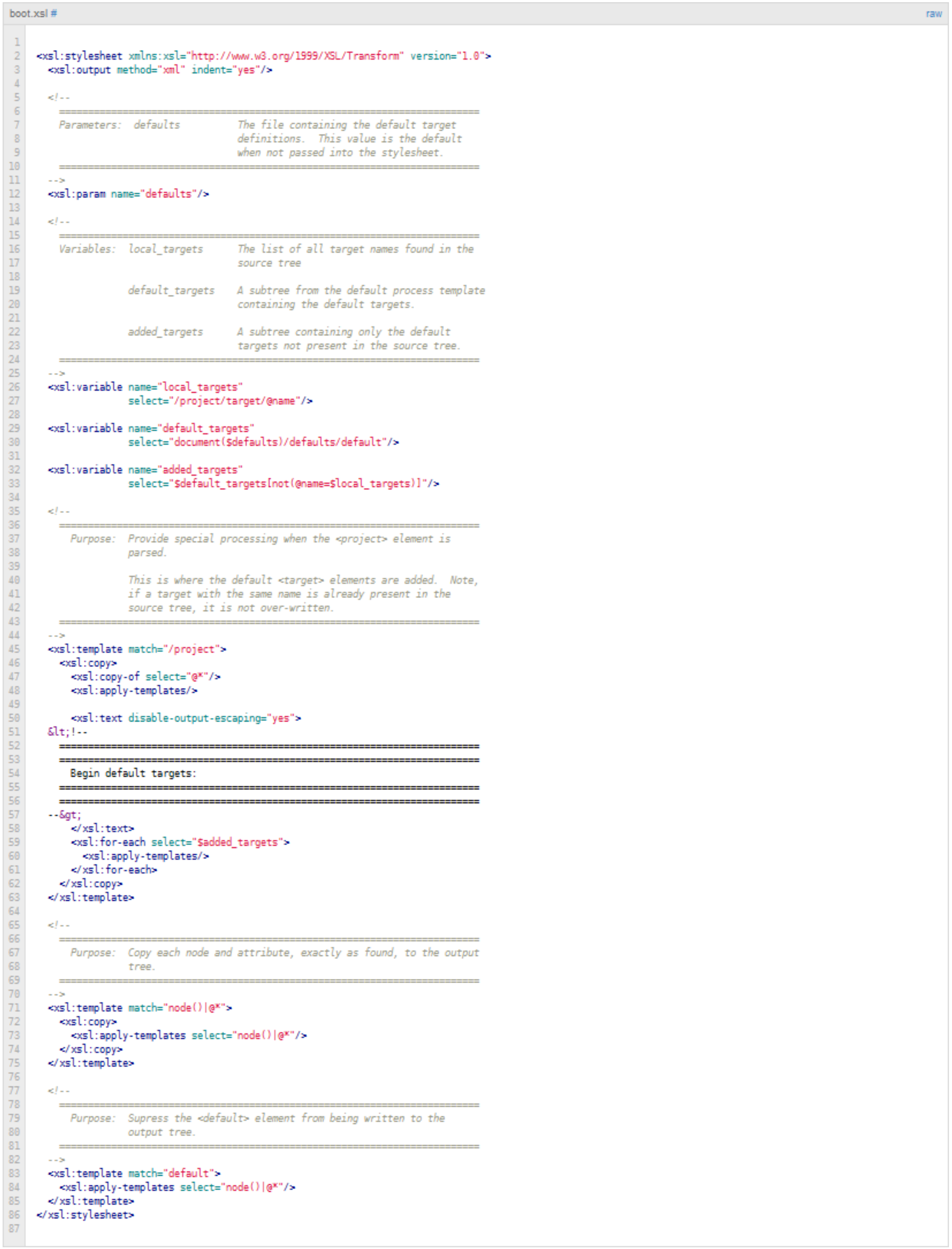
```makefile
JFLAGS = -g
JC = javac
.SUFFIXES: .java .class
.java.class:
	$(JC) $(JFLAGS) $*.java

CLASSES = \
	Foo.java \
	Blah.java \
	Library.java \
	Main.java

default: classes

classes: $(CLASSES:.java=.class)

clean:
	$(RM) *.class
```

http://www.cs.swarthmore.edu/~newhall/unixhelp/javamakefiles.html


build.xml # raw

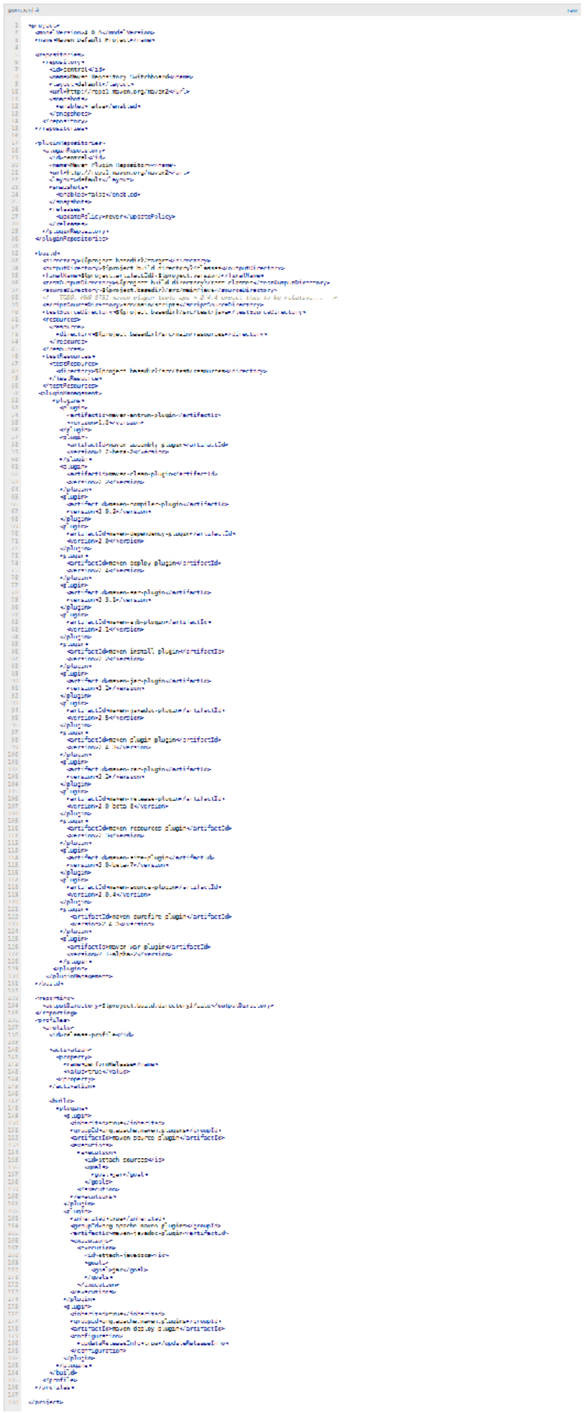
```
<project name="HelloWorld" basedir="." default="main">

    <property name="src.dir"     value="src"/>

    <property name="build.dir"   value="build"/>
    <property name="classes.dir" value="${build.dir}/classes"/>
    <property name="jar.dir"     value="${build.dir}/jar"/>

    <property name="main-class"  value="oata.HelloWorld"/>



    <target name="clean">
        <delete dir="${build.dir}"/>
    </target>

    <target name="compile">
        <mkdir dir="${classes.dir}"/>
        <javac srcdir="${src.dir}" destdir="${classes.dir}"/>
    </target>

    <target name="jar" depends="compile">
        <mkdir dir="${jar.dir}"/>
        <jar destfile="${jar.dir}/${ant.project.name}.jar" basedir="${classes.dir}">
            <manifest>
                <attribute name="Main-Class" value="${main-class}"/>
            </manifest>
        </jar>
    </target>

    <target name="run" depends="jar">
        <java jar="${jar.dir}/${ant.project.name}.jar" fork="true"/>
    </target>

    <target name="clean-build" depends="clean,jar"/>

    <target name="main" depends="clean,run"/>

</project>
```

http://ant.apache.org/manual/tutorial-HelloWorldWithAnt.html#enhance


boot.xsl #   raw

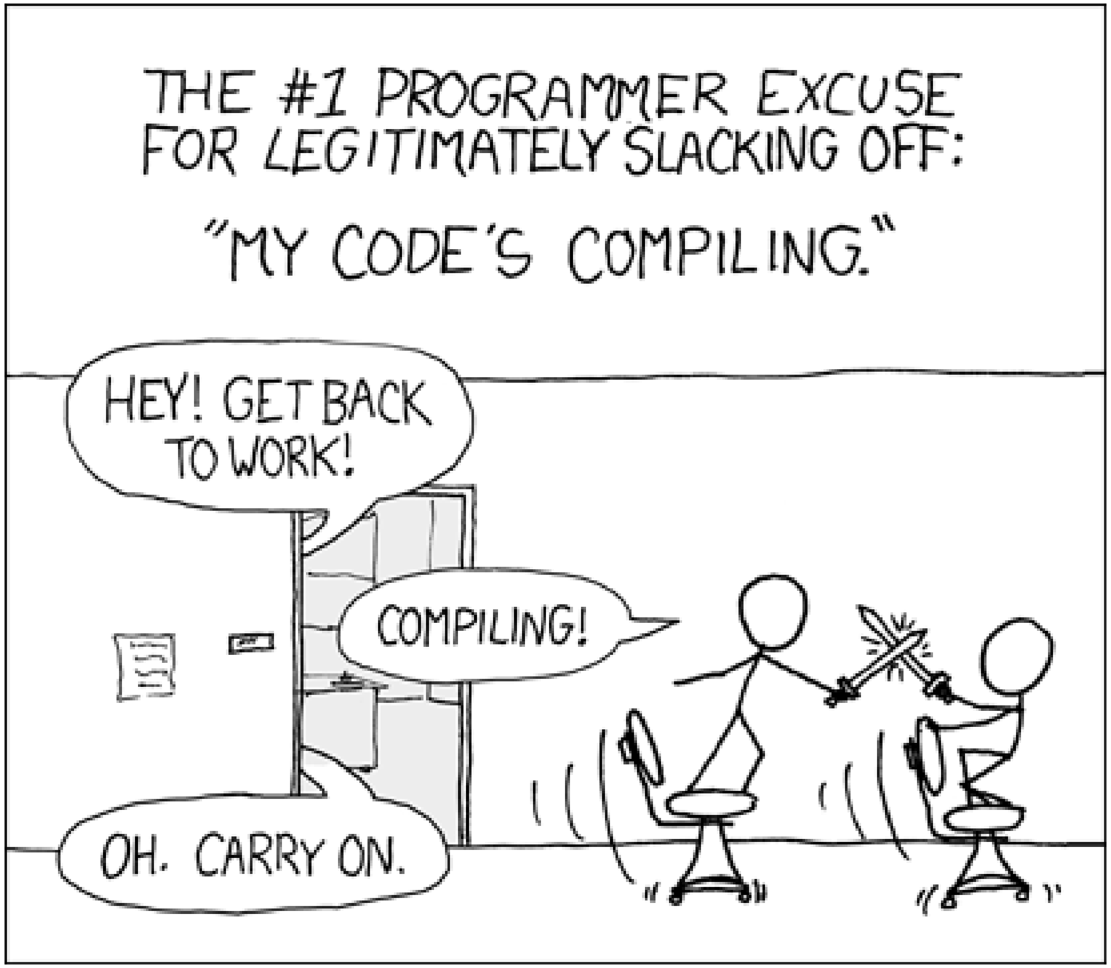
```xml

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes"/>

  <!--
    ========================================================================
    Parameters:  defaults          The file containing the default target
                                   definitions.  This value is the default
                                   when not passed into the stylesheet.
    ========================================================================
  -->
  <xsl:param name="defaults"/>

  <!--
    ========================================================================
    Variables:  local_targets      The list of all target names found in the
                                   source tree

                default_targets    A subtree from the default process template
                                   containing the default targets.

                added_targets      A subtree containing only the default
                                   targets not present in the source tree.
    ========================================================================
  -->
  <xsl:variable name="local_targets"
                select="/project/target/@name"/>

  <xsl:variable name="default_targets"
                select="document($defaults)/defaults/default"/>

  <xsl:variable name="added_targets"
                select="$default_targets[not(@name=$local_targets)]"/>

  <!--
    ========================================================================
      Purpose:  Provide special processing when the <project> element is
                parsed.

                This is where the default <target> elements are added.  Note,
                if a target with the same name is already present in the
                source tree, it is not over-written.
    ========================================================================
  -->
  <xsl:template match="/project">
    <xsl:copy>
      <xsl:copy-of select="@*"/>
      <xsl:apply-templates/>

      <xsl:text disable-output-escaping="yes">
  &lt;!--
    ========================================================================
    ========================================================================
      Begin default targets:
    ========================================================================
    ========================================================================
  --&gt;
      </xsl:text>
      <xsl:for-each select="$added_targets">
        <xsl:apply-templates/>
      </xsl:for-each>
    </xsl:copy>
  </xsl:template>

  <!--
    ========================================================================
      Purpose:  Copy each node and attribute, exactly as found, to the output
                tree.
    ========================================================================
  -->
  <xsl:template match="node()|@*">
    <xsl:copy>
      <xsl:apply-templates select="node()|@*"/>
    </xsl:copy>
  </xsl:template>

  <!--
    ========================================================================
      Purpose:  Supress the <default> element from being written to the
                output tree.
    ========================================================================
  -->
  <xsl:template match="default">
    <xsl:apply-templates select="node()|@*"/>
  </xsl:template>
</xsl:stylesheet>

```

http://www.ibm.com/developerworks/xml/library/x-antxsl/

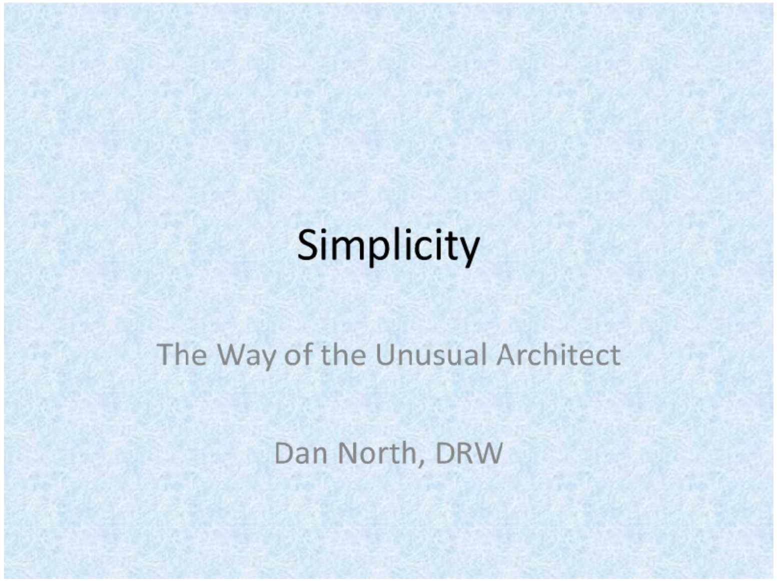

pom.xml

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <name>Maven Default Project</name>

  <repositories>
    <repository>
      <id>central</id>
      <name>Maven Repository Switchboard</name>
      <layout>default</layout>
      <url>http://repo1.maven.org/maven2</url>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
    </repository>
  </repositories>

  <pluginRepositories>
    <pluginRepository>
      <id>central</id>
      <name>Maven Plugin Repository</name>
      <url>http://repo1.maven.org/maven2</url>
      <layout>default</layout>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
      <releases>
        <updatePolicy>never</updatePolicy>
      </releases>
    </pluginRepository>
  </pluginRepositories>

  <build>
    <directory>${project.basedir}/target</directory>
    <outputDirectory>${project.build.directory}/classes</outputDirectory>
    <finalName>${project.artifactId}-${project.version}</finalName>
    <testOutputDirectory>${project.build.directory}/test-classes</testOutputDirectory>
    <sourceDirectory>${project.basedir}/src/main/java</sourceDirectory>
    <!-- TODO: MNG-3731 maven-plugin-tools-api < 2.4.4 expect this to be relative... -->
    <scriptSourceDirectory>src/main/scripts</scriptSourceDirectory>
    <testSourceDirectory>${project.basedir}/src/test/java</testSourceDirectory>
    <resources>
      <resource>
        <directory>${project.basedir}/src/main/resources</directory>
      </resource>
    </resources>
    <testResources>
      <testResource>
        <directory>${project.basedir}/src/test/resources</directory>
      </testResource>
    </testResources>
   <pluginManagement>
       <plugins>
         <plugin>
           <artifactId>maven-antrun-plugin</artifactId>
           <version>1.3</version>
         </plugin>
         <plugin>
           <artifactId>maven-assembly-plugin</artifactId>
           <version>2.2-beta-2</version>
         </plugin>
         <plugin>
           <artifactId>maven-clean-plugin</artifactId>
           <version>2.2</version>
         </plugin>
         <plugin>
           <artifactId>maven-compiler-plugin</artifactId>
           <version>2.0.2</version>
         </plugin>
         <plugin>
           <artifactId>maven-dependency-plugin</artifactId>
           <version>2.0</version>
         </plugin>
         <plugin>
           <artifactId>maven-deploy-plugin</artifactId>
           <version>2.4</version>
         </plugin>
         <plugin>
           <artifactId>maven-ear-plugin</artifactId>
           <version>2.3.1</version>
         </plugin>
         <plugin>
           <artifactId>maven-ejb-plugin</artifactId>
           <version>2.1</version>
         </plugin>
         <plugin>
           <artifactId>maven-install-plugin</artifactId>
           <version>2.2</version>
         </plugin>
         <plugin>
           <artifactId>maven-jar-plugin</artifactId>
           <version>2.2</version>
         </plugin>
         <plugin>
           <artifactId>maven-javadoc-plugin</artifactId>
           <version>2.5</version>
         </plugin>
         <plugin>
           <artifactId>maven-plugin-plugin</artifactId>
           <version>2.4.3</version>
         </plugin>
         <plugin>
           <artifactId>maven-rar-plugin</artifactId>
           <version>2.2</version>
         </plugin>
         <plugin>
           <artifactId>maven-release-plugin</artifactId>
           <version>2.0-beta-8</version>
         </plugin>
         <plugin>
           <artifactId>maven-resources-plugin</artifactId>
           <version>2.3</version>
         </plugin>
         <plugin>
           <artifactId>maven-site-plugin</artifactId>
           <version>2.0-beta-7</version>
         </plugin>
         <plugin>
           <artifactId>maven-source-plugin</artifactId>
           <version>2.0.4</version>
         </plugin>
         <plugin>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>2.4.3</version>
         </plugin>
         <plugin>
           <artifactId>maven-war-plugin</artifactId>
           <version>2.1-alpha-2</version>
         </plugin>
       </plugins>
     </pluginManagement>
  </build>

  <reporting>
    <outputDirectory>${project.build.directory}/site</outputDirectory>
  </reporting>
  <profiles>
    <profile>
      <id>release-profile</id>

      <activation>
        <property>
          <name>performRelease</name>
          <value>true</value>
        </property>
      </activation>

      <build>
        <plugins>
          <plugin>
            <inherited>true</inherited>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-source-plugin</artifactId>
            <executions>
              <execution>
                <id>attach-sources</id>
                <goals>
                  <goal>jar</goal>
                </goals>
              </execution>
            </executions>
          </plugin>
          <plugin>
            <inherited>true</inherited>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-javadoc-plugin</artifactId>
            <executions>
              <execution>
                <id>attach-javadocs</id>
                <goals>
                  <goal>jar</goal>
                </goals>
              </execution>
            </executions>
          </plugin>
          <plugin>
            <inherited>true</inherited>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-deploy-plugin</artifactId>
            <configuration>
              <updateReleaseInfo>true</updateReleaseInfo>
            </configuration>
          </plugin>
        </plugins>
      </build>
    </profile>
  </profiles>

</project>
```

http://maven.apache.org/guides/introduction/introduction-to-the-pom.html#Super_POM



http://xkcd.com/303/



Simplicity

The Way of the Unusual Architect

Dan North, DRW

Finally, sometimes, simplicity grows out of adversity

http://www.infoq.com/presentations/Simplicity-Architect

gradle

talk is cheap,
let's see some code

https://github.com/dexterous/crave2gradle

for more gradle goodness

http://mrhaki.blogspot.com/search/label/Gradle%3AGoodness

a community
that keeps on giving

https://github.com/search?type=Repositories&language=&q=gradle+plugin

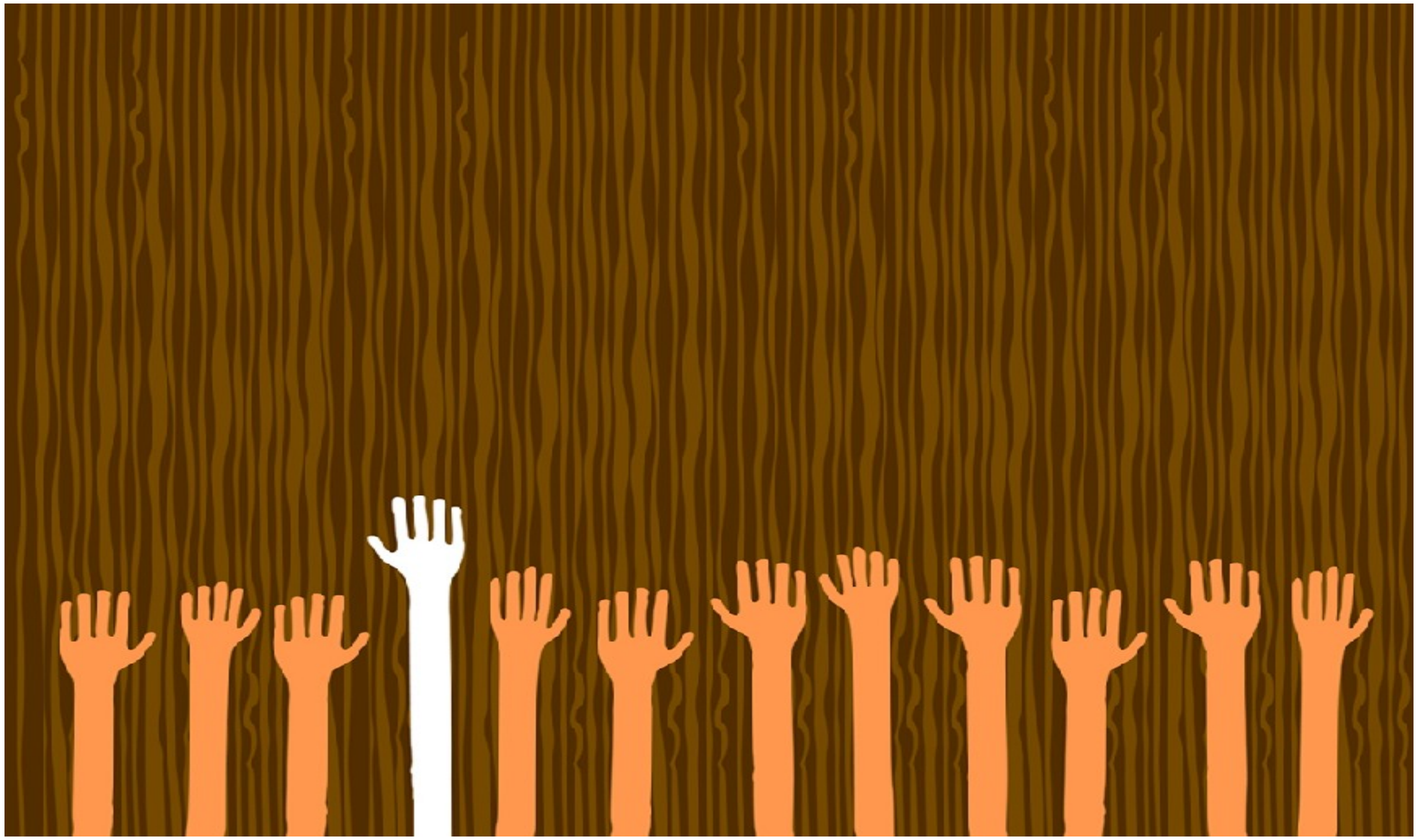
“maven has me
in its evil clutches!”

https://github.com/jbaruch/maven2gradle

gradleware

http://gradleware.com

http://spkr8.com/t/8377



http://j11.indicthreads.com/feedback