Java Builds And Code Management

South Florida Java User's Group

Bob Goldfarb

Salient Software Solutions, Inc.

bob.goldfarb@salientsoftware.net

http://www.salientsoftware.net

Overview

- No one really wants to do this job.
- Key is to automate and simplify (ex: building GNU software).
- We'll look at real-world experiences and tools.

Which Projects Need Project and Code Management?

- All
- Large projects
  - More than one engineer.
  - More than a few files in one directory.

Worst Practices

- It worked OK on my machine.
- Work like a maniac now, merge & integrate later.
- If it compiles, it's OK.
- javac *.java (by hand/in script).
- No package names.
- I.T. must be backing this stuff up.
- Only need archived code for shipping product and current development.

Benefits of Formal Tools and Procedures

- Makes developer's jobs easier and more productive.
- Reliable results.
- Ensure portability.
- Protect assets (source code).
- Retrieve code to debug customer problems.

Laying Out Source Tree

- Package directories
- Functional directories (MVC, client/server/common)
- Sub-product

Source Layout Example

```
build.xml
run
upshot.jpx
Makefile
filelist.txt
src/net
src/net/salientsoftware
src/net/salientsoftware/mgr
src/net/salientsoftware/mgr/common
src/net/salientsoftware/mgr/common/tools
src/net/salientsoftware/mgr/common/xml
src/net/salientsoftware/mgr/controller
src/net/salientsoftware/mgr/model
src/net/salientsoftware/mgr/test
src/net/salientsoftware/mgr/view
```

Source Code Control (SCM)

Orderly storage and retrieval of all software related to a project or product.

Source Code Control (SCM) Functional Requirements

- Easy to use.
- Recreate all previous builds and distributions.
- Multiplatform.

Source Code Control (SCM) Technical Requirements

- Central Repository
- Multi-developer
- Branching
- Labeling
- History
- Locking
- Developer views
- Merge capability

Source Code Control (SCM) Tools

- Lots of files (.bak, .old) – bad, not as easy as it appears. [free]
- SCCS/RCS – suitable for single user, or pre-integration. [free]
- CVS – open-source, teamwork, basic GUI available. [free]
- Perforce [$600/user], SourceSafe [$500], PVCS [$570], ClearCase [$3000/seat], Sun Forte Enterprise [$2000] – high-end teamwork, rich GUI.

Build Tools

Software that performs tasks to convert source code into end-product.

Build Tools Requirements

- Easy-to-use.
- Multiplatform.
- Full-cycle (can run part or whole):
  - Checkout
  - Organize
  - Compile
  - Package
  - Smoke-test

Build Tools
Real World Tools

- By hand – javac *.java; bad.
- Script – not portable or flexible; can have user interaction.
- Make – time tested and true; may not be portable; complicated for large tasks.
- Ant – portable; easy-to-understand.

Build Tools
Examples

- Make
- Simple Ant
- Complex Ant

Make Example 1a

```
ROOTDIR=.
DOCDIR=$(ROOTDIR)/doc
CLASSDIR=$(ROOTDIR)/classes
SRCDIR=$(ROOTDIR)/src
FILELIST=filelist.txt

# Packages used for building javadoc
PKGPRE=net.salientsoftware.mgr
PACKAGES=$(PKGPRE).common.tools $(PKGPRE).common.xml $(PKGPRE).test

ALL: build

prepare:
    @if [ ! -d $(DOCDIR) ]; then \
          mkdir $(DOCDIR); \
    fi
    @if [ ! -d $(CLASSDIR) ]; then \
          mkdir $(CLASSDIR); \
    fi
```

Make Example 1b

```
clean:
    rm -rf $(DOCDIR) $(CLASSDIR)

build: prepare compile

mkfilelist:
    find $(SRCDIR) -name "*.java" > $(FILELIST)

compile: mkfilelist
    javac -d $(CLASSDIR) @$(FILELIST)

doc: prepare
    javadoc -sourcepath $(SRCDIR) -d $(DOCDIR) -author -use -version \
          -windowtitle "Upshot" -doctitle "Upshot Management Software" \
          -bottom "Copyright &#169; 2001 Salient Software Solutions Inc. All
    Rights Reserved." \
          $(PACKAGES)
```

Ant Example 1a

```
<project name="upshot" default="build" basedir=".">

  <target name="prepare">
    <mkdir dir="classes"/>
    <mkdir dir="doc"/>
  </target>

  <target name="clean">
    <delete dir="classes"/>
    <delete dir="doc"/>
  </target>

  <target name="build" depends="prepare, compile"/>

  <target name="compile">
    <javac srcdir="src"
          destdir="./classes"
          includes="net/salientsoftware/upshot/**" />
             <!--
           debug="on" optimize="off" deprecation="off"/>
             -->
  </target>
```

Ant Example 1b

```
<target name="doc">
    <javadoc packagenames="net.salientsoftware.mgr.*"
            sourcepath="src"
            destdir="doc"
            author="true"
            version="true"
            use="true"
            windowtitle="Upshot"
            doctitle="Upshot Software"
            bottom="Copyright &#169; 2001 Salient Software Solutions Inc. All
    Rights Reserved." />
  </target>

</project>
```

Ant Example 2a

```
<project name="db" default="help" basedir=".">

  <property name="app.name"       value="db"/>
  <property name="tomcat.home"    value="/usr/local/tomcat"/>
  <property name="deploy.home"    value="${tomcat.home}/webapps/${app.name}"/>
  <property name="class.home"   value="${deploy.home}/WEB-INF/classes"/>
  <property name="dist.home"      value="${deploy.home}"/>
  <property name="dist.src"       value="${app.name}.jar"/>
  <property name="dist.war"       value="${app.name}.war"/>
  <property name="javadoc.home"   value="${deploy.home}/javadoc"/>


  <target name="prepare">
    <mkdir  dir="${deploy.home}"/>
    <mkdir  dir="${deploy.home}/WEB-INF"/>
    <copy  file="web.xml" todir="${deploy.home}/WEB-INF"/>
    <!--
    <copy  file="index.html" todir="${deploy.home}"/>
    -->

    <mkdir  dir="${deploy.home}/WEB-INF/classes"/>
  </target>
```

Ant Example 2b

```
<target name="help">
  <echo>
  Try using the "build" target.
  </echo>
</target>


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

<target name="build" depends="prepare, compile, install"/>

<target name="install">
  <copy todir="${class.home}">
    <fileset dir="classes"/>
  </copy>
</target>
```

Ant Example 2c

```
  <target name="compile" depends="prepare">

    <javac srcdir="src" destdir="./classes"
           classpath="./classes" />
             <!--
           debug="on" optimize="off" deprecation="off"/>
             -->

    <copy   todir="${deploy.home}/WEB-INF/classes">
      <fileset dir="src" includes="**/*.properties"/>
    </copy>

  </target>

  <target name="all" depends="build, dist" />

  <target name="dist" depends="prepare,compile">
       <jar jarfile="${dist.home}/${dist.war}"
         basedir="${deploy.home}" includes="*"/>
  </target>
</project>
```

Resources

http://jakarta.apache.org/ant
http://www.cvshome.org
http://www.cvshome.org/dev/
http://www.gnu.org
http://www.perforce.com
http://msdn.microsoft.com/ssafe/
http://www.rational.com/products/clearcase
http://www.merant.com/pvcs
http://www.sun.com/forte/teamware/index.html
news:comp.software.config-mgmt
FAQ: http://www.daveeaton.com/scm

- General Questions
- Configuration Management Tools
- Problem Management Tools

Java Builds And Code Management

- Q & A
- Thankyou