

PUG DE Herbstkonferenz 2009 - Köln

# WebClient deployment made easy !

Gilles QUERRET – Riverside Software

# Topic

---

- Main topic is deployment automation for Webclient applications
- Implies having a standard build process
- Useful open source tools which cover those needs
- Mix everything, and watch the cycle from SCM to the end-user PC

# Agenda

- A standard build process, using Apache ANT and PCT
- Continuous integration, using Hudson (Sun Microsystems)
- WebClient deployment, using Padeo (Riverside Software)
- Demo !

# Standard build process

- How long does it take to get a new team member working productively ?
- How many steps after initial checkout before getting an installer ?
- Any human interaction during the build process ?
- Do you have nightly builds ?

# Standard build process

- Application compiler from a Progress session
- Makefile (syntax, multiple implementations, no extensions)
- Shell scripts (not portable)
- Roundtable
  
- You should at least be able to generate a ZIP package

# Ant

- Part of the Jakarta project, a sub-project of Apache
  - ▣ Jakarta is the sub-heading for all the Java oriented Apache projects
- Open source, BSD style license application providing build functions for Java programs
- Available since mid 2000
- <http://jakarta.apache.org/ant/>



# Ant basics

- ❑ Ant replaces the popular 'make' utility used by many C/C++ developers. 'Make' lacks a file format that is easily parsed, has variations in implementations, and lacks extensibility.
- ❑ Ant files are simple XML files, where nodes represents tasks
- ❑ Tasks may be dependant on each others



# Ant basics

- Since Ant is written in Java, it is available on any Java platform
- Cross platform : provides a way to deal with file system and process
- Custom tasks can be dropped in Ant directory.
- Custom tasks are indistinguishable from any other task in Ant.



# PCT

- Set of custom tasks for OpenEdge
- Developed by... myself
- Project started in 2003
- Apache licence (BSD like)
- Stable
- No stats on usage, but used by PSC itself



# PCT tasks

- Running .p procedures
- Compilations
- PL
- Dump/load and incremental
- Proxygen
- Additional tasks provided by vendors



# Eclipse / Architect

- ❑ Native support for ANT within Eclipse
- ❑ Disabled by default in Architect



```
168
169 <target name="proxygen" depends="init" description="Progress Proxygen">
170   <PCTCompile destDir="${build}/progress" graphicalMode="true" dlcHome="${DLC}" md5="false" minSize="false"
171     cpinternal="iso8859-15" cpstream="iso8859-15" inputChars="16384" debugListing="true">
172     <fileset dir="${src-progress}/hachette" includes="**/*.p,**/*.w" />
173     <fileset dir="${src-progress}/phx" includes="**/*.p,**/*.w" />
174     <PCTConnection dbName="ged" dbDir="${db}" singleUser="${singleUser}" />
175     <PCTConnection dbName="cust" dbDir="${db}" singleUser="${singleUser}" />
176     <propath refid="progress.compilation.propath" />
177   </PCTCompile>
178   <echo file="${build}/progress/build_${BUILD_NUMBER}.txt" message="${BUILD_ID}" />
179   <copy file="src/progress/ImageIn.xpxg.tmpl" tofile="ImageIn.xpxg">
180     <filterset>
181       <filter token="BASEDIR" value="${basedir}" />
182       <filter token="JAVAC" value="${JAVAC}" />
183     </filterset>
184   </copy>
185   <PCTProxygen workingDirectory="." srcFile="ImageIn.xpxg" dlcHome="${DLC}" />
186   <delete file="ImageIn.xpxg" />
187   <fixCrLf srcdir="${build}/proxygen" includes="**/*.java" eol="dos" />
188   <jar destfile="${libs}/ImageInProxy.jar">
189     <fileset dir="${build}/proxygen" includes="**/*.class" />
190     <fileset dir="${build}/proxygen" includes="**/*.java" />
191   </jar>
192 </target>
193
```

# Standard build

---

- Empty database + load DF
- Compilation
- Libraries or set of rcode
- Generate version files
- Create ZIP packages

# Agenda

- A standard build process, using Apache ANT and PCT
- Continuous integration, using Hudson (Sun Microsystems)
- WebClient deployment, using Padeo (Riverside Software)
- Demo !

# Continuous integration

- Maintain code repository
- Automate the build
- Make the build self-testing
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automate deployment

# Hudson

- Open source project at [java.net](http://java.net)
- Emphasis on ease of installation and use
  - ▣ “java -jar hudson.war” execution
  - ▣ Configure everything from browsers
- Extensible architecture
  - ▣ 140+ community-developed public plugins
  - ▣ By 150+ contributors
- Estimated 13,000 installations



# Hudson

- Check out the source code
  - ▣ Subversion, Perforce, Git, Mercurial, CVS, ...
- Do builds and/or tests
  - ▣ Ant, Maven, MSBuild, shell script, ...
- Record results
  - ▣ Binary, test results, code coverage, static analysis
- Notify people
  - ▣ E-mail, IM, RSS, tray apps, IDEs



# Hudson – Matrix projects

- Run the same thing on different environments, such as OSes, JDKs
- Aggregate the results
- OpenEdge matrix available soon



# Hudson – Promoted builds

- Builds may be promoted, i.e. visual notification and actions may be taken
- May be used to generate WebClient packages and deploy to an appserver



# Agenda

- A standard build process, using Apache ANT and PCT
- Continuous integration, using Hudson (Sun Microsystems)
- WebClient deployment, using Padeo (Riverside Software)
- Demo !

# Webclient limitations

- ❑ Manual steps when creating packages. Poor UI
- ❑ Manual steps when installing packages
- ❑ Administrator rights problems
- ❑ Low compression
- ❑ No server-side push
- ❑ Only one command line

# Padeo

- Use standard Windows installers
- Prowcapp generated on the fly
- One UI to « bind them all »
- WebServices enabled
- Multiple deployments servers
- Not a deployment tool

# Integration with Ant & Hudson

- Padeo API is accessible through WebServices
- A set of Ant tasks is available, which uses this API
- From Hudson, starting a Ant task calling Padeo is easy

# MSI deployment

---

- Pull : batch script at logon
- Pull : custom scripts based on location
- Push : group policies (AD)
- Push : Microsoft SMS

# Agenda

- A standard build process, using Apache ANT and PCT
- Continuous integration, using Hudson (Sun Microsystems)
- WebClient deployment, using Padeo (Riverside Software)
- Demo !



**Questions ?**

# Resources

---

- ANT : <http://ant.apache.org>
- PCT : <http://pct.sourceforge.net>
- Hudson : <http://hudson-ci.org>
- Padeo (live demonstration) :  
<http://padeo.riverside-software.fr>