

uPortal Filtered Configuration and Automated Deployment

Jennifer Bourey

Jasig 2009 Conference, March 2, 2009

© Copyright Unicon, Inc., 2009. This work is the intellectual property of Unicon, Inc. Permission is granted for this material to be shared for non-commercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of Unicon, Inc. To disseminate otherwise or to republish requires written permission from Unicon, Inc.

1. Maven filtering and overlays
2. Useful maven plugins
3. uPortal build tools
4. Configurable build case study
5. Future directions

Maven filtering and overlays

Maven Properties

- Specify properties from
 - pom file: `<property.name>value</property.name>`
 - command line: `mvn goal -Dproperty.name=value`
 - profile
- In uPortal 3, used to define database dependency

Maven Profiles

- Use profiles to
 - Set property values
 - Add additional repositories
- Define profiles in
 - pom.xml
 - *HOME_DIR*/.m2/settings.xml
- Select using `mvn goal -Pname`

Maven Filtering

- Filter resources during the build process
 - Only applies to processed resources
 - Other files ([web.xml](#)) can potentially be filtered by plugins
- Replace tokens of format `${token}`
- Filter values may be specified by a properties file

Maven Filtering

```
<build>
<filters>
<filter>
    ../../filters/filter.properties
</filter>
</filters>
<resources>
<resource>
<directory>src/main/resources</directory>
<filtering>>true</filtering>
</resource>
</resources>
. . .

<build>
```

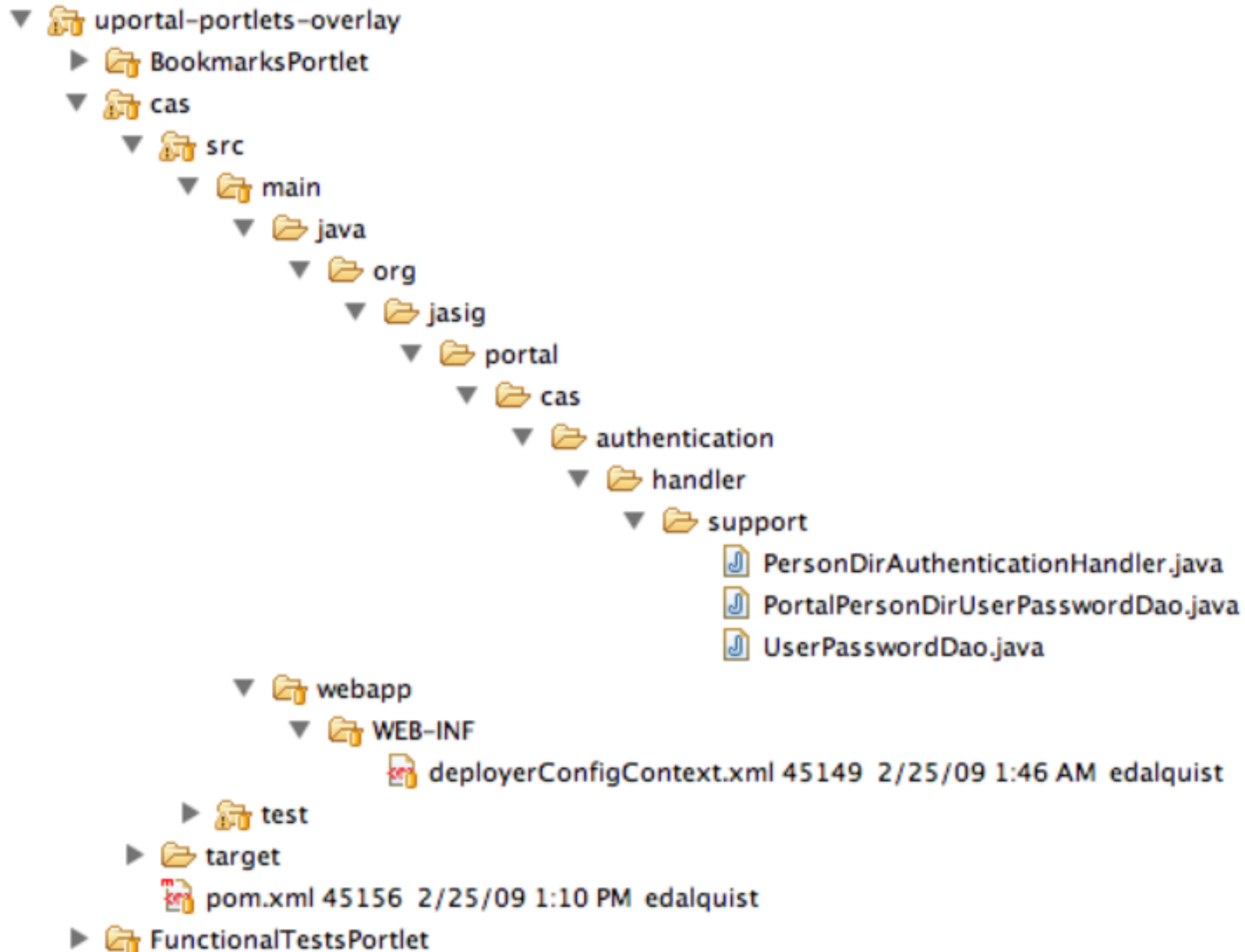
Maven Overlays

- Layer additional files on top of imported artifact
- Resources replace existing files or add to the file set
- Can be used to change the contents of a file, or to add new files and classes to a project

Maven Filtered Overlay

```
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-war-plugin</artifactId>
<version>${maven-war-plugin.version}</version>
<configuration>
<webResources>
<resource>
<directory>src/main/webapp</directory>
<includes>
<include>WEB-INF/portlet.xml</include>
</includes>
<filtering>>true</filtering>
</resource>
</webResources>
</configuration>
</plugin>
```

uPortal 3 Example



Maven plugins

Maven Pluto Plugin

- Pluto-fy [web.xml](#) and repackage
- Used in uPortal 3+ for deploying included portlets
- Performs same task as `deployPortletApp`

```
<plugin>  
<groupId>org.apache.pluto</groupId>  
<artifactId>maven-pluto-plugin</artifactId>  
</plugin>
```

Maven War Plugin

- Produce a .war artifact
- Used in uPortal 3 to generate uPortal.war and cas.war artifacts
- Also used by portlet projects
- Allows filtering of additional files

Maven War Plugin

```
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-war-plugin</artifactId>
<configuration>
<warName>BookmarksPortlet</warName>
<webXml>
    ${basedir}/src/main/webapp/WEB-INF/web.xml
</webXml>
</configuration>
</plugin>
```

YUI Compressor Plugin

- Use the Yahoo UI compressor to minify javascript and CSS
- Validate with jslint
- Combine files
- uPortal 3 usage: minify uPortal theme resources

YUI Compressor Plugin

```
<plugin>
<groupId>net.sf.alchim</groupId>
<artifactId>yuicompressor-maven-plugin</artifactId>
<version>0.7.1</version>
<executions>
<execution>
<goals><goal>compress</goal></goals>
</execution>
</executions>
<configuration>
<excludes>
<exclude>**/*.min.js</exclude>
</excludes>
<linebreakpos>10000</linebreakpos>
<suffix>.min</suffix>
</configuration>
</plugin>
```


JSP Pre-compilation Plugin

- Pre-compile JSPs on maven build
- Requires adding placeholder to [web.xml](#)
- May want to tie to profile in order to disable during development

Hibernate Plugin

- Initialize hibernate-configured tables via maven goal
- Can also output SQL scripts
- Potential solution for portlet “initportal”

uPortal Build

uPortal Ant Commands

- **initdb**
 - run all database initialization tasks and import entities
- **deploy-war**
 - update the uPortal war
- **deploy-ear**
 - update all deployed components, including uPortal war, portlets, and shared libraries
- **deployPortletApp -DportletApp=/path/file.war**
 - pluto-fy and deploy a portlet

uPortal Import/Export

- Add or update uPortal entities via the command line
- Export uPortal database entities to XML format
- Default dataset resides in uportal-impl

```
ant crn-import -Ddir=uportal-  
impl/src/main/resources/entities/layout -  
Dpattern=admin.layout
```

Deploying Portlets

- Using uportal-portlets-overlay
 - add new module to uportal-portlets-overlay
 - require new artifact from uportal-ear/pom.xml
- Advantages
 - Automatic deployment of correct portlet artifact
 - Information about portlet versions preserved in subversion

Case Study

Goals

- Install portal and portlets on development and production server without lots of configuration
- Enable developers to check out and build in their local environment
 - should get all necessary portlets and data
 - build regardless of preferred database, servername, etc.
- Create nightly development build

Strategy

- Configurable build using maven filtered overlays
- One filter for each environment
 - server hostname
 - database configuration
 - other environment-specific parameters
- Filter applied to portal as well as all portlets

Steps

- Define new “environment.name” property
- Configure uPortal and portlets to use filter file based on environment property value
- Edit build.xml to pass property to maven goals
- Potentially edit build.xml to choose import directory based on property name

Build Command

```
ant clean initportal  
-Denvironment.name=local -Djdbc.artifactId=postgresql -  
Djdbc.groupId=postgresql -Djdbc.version=8.3-603.jdbc3
```

Advantages

- Possible to implement without extensive changes
- Choose build environment with a few parameters
 - production deployment requires less configuration
 - easier to set up automated nightly build
- Allows developers to easily check out and build environment

Disadvantages

- Same artifact name for each environment
 - Have to remember to include correct parameters
- Have to specify database dependency on the command line
- Need additional configurability to handle spring configuration differences
- Approach requires passwords in subversion

The Future

Future Goals

- Per-environment artifacts
- Separate artifact production from ear deployment

Questions?



Jennifer Bourey

Jbourey@unicon.net

www.unicon.net