

Case Study: Using Jenkins to Build WebSphere Portal Applications for the Enterprise



Sam Alexander

Senior Managing Consultant

IBM Software Services for Collaboration

June 18, 2014

#jenkinsconf

Topics

- Typical Enterprise Portal Projects
- Portal artifacts
- How does Jenkins Help?
 - Integration with SCM
 - Integration with Build tools
 - Deployment
- Future Plans



Typical Enterprise Portal Projects



- Big projects
- Strict delivery deadlines
- Many roles
 - Developers
 - Infrastructure Specialists
 - Testers
 - Business Analysts
 - Project Managers
- Many environments
 - Local
 - Integration
 - Testing
 - Staging
 - Production
- Many artifacts

Topics

- Typical Enterprise Portal Projects
- **Portal artifacts**
- How does Jenkins Help?
 - Integration with SCM
 - Integration with Build tools
 - Deployment
- Future Plans



Portal Artifacts

- Standard artifacts
 - Portlet WARs
 - Standalone WARs
 - Portal Theme EAR
 - Portal Theme Artifacts
 - Java libraries
 - PAA (Portal Application Archive)
- Other artifacts
 - Feature Sets (reusable code assets)
 - IBM Forms components



Topics

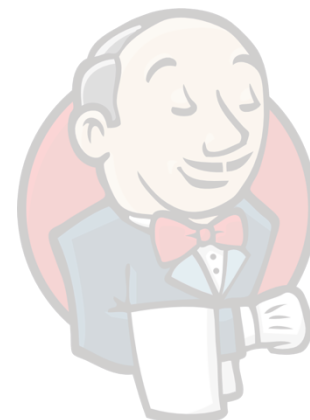
- Typical Enterprise Portal Projects
- Portal artifacts
- **How does Jenkins Help?**
 - Integration with SCM
 - Integration with Build Tools
 - Deployment
- Future Plans



How does Jenkins Help?

- Rapidly create a build pipeline
 - Easy installation
 - Easy configuration
 - Built in functionality
 - Add Plugins
- Quick and easy integration with
 - SCM
 - Build tools





SCM Integration

- OOTB, Simple integration with Subversion
- Point at branch

Source Code Management

- ☐ CVS
☐ CVS Projectset
☐ None
☒ Subversion

Modules

Repository URL

Local module directory (optional)

Repository depth option

Ignore externals option

☐

[Add more locations...](#)

Check-out Strategy

Use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to remain when a new build starts.

Repository browser

Developing and Building Portal Solutions

- Agile development methodology
 - Several iterations
 - Daily SNAPSHOT builds
 - Scheduled Release builds
- Build discrete parts of the application
 - Components grouped logically into projects
 - Limits risk to only one component
- Maven well suited for this
 - Declarative build scripts
 - Concept of modules

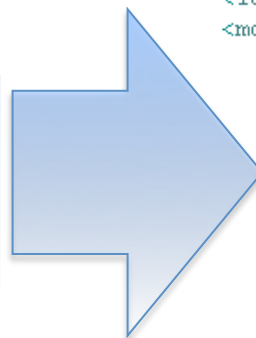




Jenkins Integration with Maven

- Jenkins “understands” Maven pom files and project structure
- Very little configuration
- Overall portal-parent Maven project defines modules to build
- Configure Jenkins build configuration to portal-parent pom

Build	
Root POM	portal-parent/pom.xml
Goals and options	-X clean deploy -P build-profile



```
<profiles>
  <profile>
    <id>local-dev-profile</id>
    <modules>
      <module>./portal-parent</module>
      <module>./portal-parent-frontend</module>
      <module>./portal-parent-backend</module>
      <module>./portal-parent-portal</module>
      <module>./portal-parent-portal-frontend</module>
      <module>./portal-parent-portal-backend</module>
      <module>./portal-parent-portal-portal</module>
      <module>./portal-parent-portal-portal-frontend</module>
      <module>./portal-parent-portal-portal-backend</module>
      <module>./portal-parent-portal-portal-portal</module>
      <module>./portal-parent-portal-portal-portal-frontend</module>
      <module>./portal-parent-portal-portal-portal-backend</module>
      <module>./portal-parent-portal-portal-portal-portal</module>
    </modules>
  </profile>
</profiles>
```

Jenkins Integration with Maven (continued)



- Flexibility in building
 - Build with parameters, specifying build number
 - Easily configure release builds with maven-release-plugin

Build Environment

<input checked="" type="checkbox"/> Maven release build	
Release goals and options	<input type="text" value="-Dresume=false -X clean release:prepare release:perform -P build-profile"/>
DryRun goals and options	<input type="text" value="-Dresume=false -DdryRun=true release:prepare -P build-profile"/>
Number of successful release builds to keep	<input type="text" value="0"/>
Default versioning mode	<input type="text" value="None"/>
Preselect custom SCM comment prefix	<input type="checkbox"/>
Preselect append Jenkins username	<input type="checkbox"/>
Preselect 'specify SCM login/password'	<input checked="" type="checkbox"/>

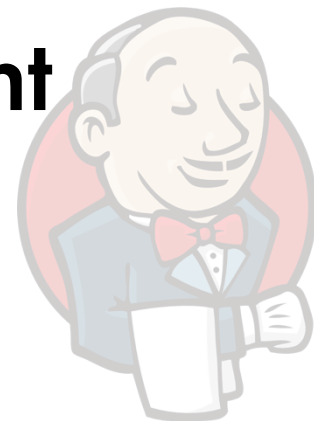
[Advanced...](#)

Jenkins and Maven for Deployment

- Potentially many steps to deploy an Enterprise Portal
 - Many components
 - Portlets, JAR files, Theme components, etc
 - Many configurations
- Portal Application Archive (PAA) simplifies deployment of large or small applications
- PAA file format defines how to install the application & directory structure for installable artifacts

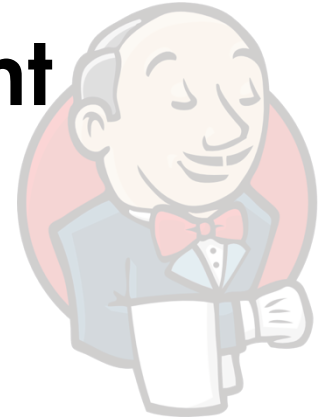


Jenkins and Maven for Deployment (continued)



- Dedicated Maven project to build PAA
 - *maven-resources-plugin* copies built artifacts to build area
 - *maven-assembly-plugin* to arrange artifacts in PAA directory structure & build zip file
- Last module in portal-parent POM file builds PAA project

Jenkins and Maven for Deployment (continued)



- Future:
 - Configure Jenkins to run scripts to automatically deploy PAA file to an environment
 - Configure Jenkins build scripts to run Selenium tests

Thank You To Our Sponsors

Platinum



Gold



Silver

