Managing Installations and Provisioning of OSGi Applications

Carsten Ziegeler
cziegeler@apache.org
About

- Member of the ASF
- Current PMC Chair of Apache Sling
- Apache Sling, Felix, ACE, Portals
- (Cocoon, Avalon/Excalibur, Incubator, Commons)
- RnD Team at Adobe Research Switzerland
- Article/Book Author, Conference Speaker
- Technical Reviewer
- JSR 286 Spec Group (Portlet API 2.0)
Apache Sling
OSGi Installer
OSGi Application

- OSGi is about modules
- Typical (web)application
  - OSGi framework
  - Bundles
  - Configurations
Installation I

- One-by-one installation
- Deployment units
  - Deployment packages
  - Custom assemblies
Many different tools
  - Java API, Web Console, Shell…
  - FileInstall, **Apache Sling OSGi Installer**

What do use?
Interaction?
Creating a distributable?
Apache Sling OSGi Installer

- General service covering
  - Install
  - Update
  - Uninstall
- Of
  - Bundles
  - Configurations
  - Custom artifacts
Apache Sling OSGi Installer

- General service covering
  - Install
  - Update
  - Uninstall
- Of
  - Bundles
  - Configurations
  - Custom artifacts
Installer Providers

- Just provide artifacts or inform about removal
- Don’t need knowledge about artifacts
- Simple trackers
Apache Sling Installer File Provider

- Service for monitoring directories in the file system
  - Configurable through properties
- Informs the OSGi Installer about
  - Add
  - Update
  - Delete
- Of artifacts
- Like Apache Felix FileInstall but delegates work!
Apache Sling Installer JCR Provider

- Support for Java Content Repository (Apache Jackrabbit)
- Service for monitoring nodes in the JCR
  - Configurable through Config Admin
- Informs the OSGi Installer about
  - Add
  - Update
  - Delete
- Of artifacts
Transformers

- Transformers inspect artifacts
- Detect artifact type
  - Bundle, configuration etc.
- Might transform artifact
- Provide unique id
  - Symbolic name + version
  - Configuration PID
Installer Task Factories

- Register for artifact types
- Get transformed artifacts or removal event
- Perform their task
- OSGi Installer manages retry and failure
- A task might add another task
  - Like bundle refresh after all installs etc.
Advantages of the OSGi Installer Family

- Pluggable and highly customizable
  - New artifact types: transformer + task factory
  - New artifact sources: provider
- Batch handling
- Retry mechanism in the installer core
  - Bundle installation order doesn’t matter
- State management in the installer core
  - Defined workflow
  - Copes with 3\textsuperscript{rd} party changes
Example I – Always Install Highest Version

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2
Example I – Always Install Highest Version

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

Install directory:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2
Example I – Always Install Highest Version

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

Install directory:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2
Example I – Always Install Highest Version

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

Install directory:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

Install directory:
- Bundle A, version 1.0

Installed Bundle:
- Bundle A, version 1.0
Example II – State Management

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2
Example II – State Management

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

User installs through Web Console:
- Bundle A, version 1.4

Installed Bundle:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.4
Example II – State Management

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

User installs through Web Console:
- Bundle A, version 1.4

Installed Bundle:
- Bundle A, version 1.4

Install directory:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.4
Example II – State Management

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

User installs through Web Console:
- Bundle A, version 1.4

Install directory:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.4

Install directory:
- Bundle A, version 1.0

Installed Bundle:
- Bundle A, version 1.4

Installed Bundle:
- Bundle A, version 1.4

Installed Bundle:
- Bundle A, version 1.4
Example III – State Management + Highest Version

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2
Example III – State Management + Highest Version

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

User installs through Web Console:
- Bundle A, version 1.1

Installed Bundle:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.1
Example III – State Management + Highest Version

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

User installs through Web Console:
- Bundle A, version 1.1

Installed Bundle:
- Bundle A, version 1.1

Install directory:
- Bundle A, version 1.1

Installed Bundle:
- Bundle A, version 1.1
Example III – State Management + Highest Version

Install directory:
- Bundle A, version 1.0
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

User installs through Web Console:
- Bundle A, version 1.1

Installed Bundle:
- Bundle A, version 1.1

Install directory:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2

Install directory:
- Bundle A, version 1.2

Installed Bundle:
- Bundle A, version 1.2
Example IV – Multiple Providers

<table>
<thead>
<tr>
<th>Install directory A:</th>
<th>Install directory B:</th>
<th>Installed Bundle:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Bundle A, version 1.0</td>
<td>- Bundle A, version 1.1</td>
<td>- Bundle A, version 1.2</td>
</tr>
<tr>
<td>- Bundle A, version 1.2</td>
<td>- Bundle A, version 1.2</td>
<td></td>
</tr>
</tbody>
</table>

Time
### Example IV – Multiple Providers

<table>
<thead>
<tr>
<th>Install directory A:</th>
<th>Install directory B:</th>
<th>Installed Bundle:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Bundle A, version 1.0</td>
<td>- Bundle A, version 1.1</td>
<td>- Bundle A, version 1.2</td>
</tr>
<tr>
<td>- Bundle A, version 1.2</td>
<td>- Bundle A, version 1.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Installed Bundle:</strong></td>
</tr>
<tr>
<td><strong>Installed Bundle:</strong></td>
<td></td>
<td>- Bundle A, version 1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Install directory A:</td>
<td>Install directory B:</td>
<td>Installed Bundle:</td>
</tr>
<tr>
<td>- Bundle A, version 1.0</td>
<td>- Bundle A, version 1.1</td>
<td>- Bundle A, version 1.2</td>
</tr>
<tr>
<td></td>
<td>- Bundle A, version 1.2</td>
<td></td>
</tr>
</tbody>
</table>

**Time**
### Example IV – Multiple Providers

<table>
<thead>
<tr>
<th>Time</th>
<th>Install directory A:</th>
<th>Install directory B:</th>
<th>Installed Bundle:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Bundle A, version 1.0</td>
<td>- Bundle A, version 1.1</td>
<td>- Bundle A, version 1.2</td>
</tr>
<tr>
<td></td>
<td>- Bundle A, version 1.2</td>
<td>- Bundle A, version 1.2</td>
<td>- Bundle A, version 1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bundle A, version 1.1</td>
<td>- Bundle A, version 1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bundle A, version 1.1</td>
<td>- Bundle A, version 1.1</td>
</tr>
</tbody>
</table>
Versioned artifacts (e.g. bundles)

- Versions are unique
  - (Maven) snapshot handling supported
- Always install highest version
- Handle 3rd party interaction
State Management II

- Unversioned artifacts (e.g. configurations)
  - Last action wins
  - Handle 3rd party interaction
    - Comparing artifacts
    - Write back mechanism
Example IV – State Management Unversioned Artifact

Install directory:
- Config PID A, state 1

Installed Configuration:
- Config PID A, state 1
Example IV – State Management Unversioned Artifact

Install directory:
- Config PID A, state 1

Installed Configuration:
- Config PID A, state 1

User updates configuration through Web Console:
- Config PID A, state 2

Installed Configuration:
- Config PID A, state 2
Example IV – State Management Unversioned Artifact

Install directory:
- Config PID A, state 1

Installed Configuration:
- Config PID A, state 1

User updates configuration through Web Console:
- Config PID A, state 2

Installed Configuration:
- Config PID A, state 2

Install directory:

Installed Configuration:
- Config PID A, state 2
Example IV – State Management Unversioned Artifact

Install directory:
- Config PID A, state 1

User updates configuration through Web Console:
- Config PID A, state 2

Install directory:
- Config PID A, state 2

Install directory:
- Config PID A, state 3

Installed Configuration:
- Config PID A, state 1

Installed Configuration:
- Config PID A, state 2

Installed Configuration:
- Config PID A, state 3
Installer checks for changes from other sources:
  - On next action for an artifact

Supports write back of 3rd party changes:
  - Pluggable, usually done by task factory
  - Built-in support for configurations
  - Provider might support write back of changes
    - File and JCR provider do!
Apache Sling OSGi Installer

- Central mechanism for handling
  - Install, Update, Delete
- Highly customizable
- Download and install:
  - org.apache.sling.installer.api
  - org.apache.sling.installer.core
  - org.apache.sling.installer.provider.file
  - org.apache.sling.installer.factory.configuration
Apache Sling Launchpad
Creating a distribution is easy
  • Standalone jar file and/or web application

Maven Launchpad Plugin
  • Define bundle list
  • Add other artifacts
    • Configurations

Leverages OSGi installer
  • Through launchpad provider
<?xml version="1.0"?>
<bundles>
  <startLevel level="boot">
    <bundle>
      <groupId>org.apache.sling</groupId>
      <artifactId>org.apache.sling.commons.log</artifactId>
      <version>2.1.2</version>
    </bundle>
    ...
  </startLevel>
  <startLevel level="9">
    <bundle>
      <groupId>org.apache.felix</groupId>
      <artifactId>org.apache.felix.eventadmin</artifactId>
      <version>1.3.2</version>
    </bundle>
    ...
  </startLevel>
  <startLevel level="0">
    <bundle>
      <groupId>org.some.company</groupId>
      <artifactId>a.strange.artifact</artifactId>
      <version>1.2.0</version>
      <type>zip</type>
    </bundle>
  </startLevel>
</bundles>
Apache Sling Launchpad

- Bootstrap code
- Includes Apache Felix
  - Handles framework updates
- Launch an OSGi framework
  - Standalone jar with http service
  - As a web application with http bridge
- Deploy packaged bundles
- Configurable
  - sling.properties
  - (system properties / servlet parameters)
OSGi Standalone App or OSGi Webapp with Launchpad

**JAR File**
- Launchpad Main Bootstrap Code
- OSGi Framework
- OSGi HTTP Service
- sling.properties

**Deployable Artifacts**

![Diagram of JAR File Deployable Artifacts]

**WAR File**
- Launchpad Servlet Bootstrap Code
- OSGi Framework
- HTTP Bridge
- sling.properties

**Deployable Artifacts**

![Diagram of WAR File Deployable Artifacts]
Apache Sling Goodies

- Try out the Sling OSGi installer & friends
- Think about the Sling Launchpad
- Share your experience
  - [users|dev]@sling.apache.org
Give Feedback on the Sessions

1. Sign In: www.eclipsecon.org

2. Select Session Evaluate

3. Vote +1  0  -1