Agenda

• dm Server v1.0.x recap
• dm Server v2.0
  • provisioning: plans, repositories
  • web application support
  • regions
  • medic
  • other enhancements
• Supporting tools: bundlor, STS
dm Server v1.0

- Open source, GPL licensed. Commercial license available
- OSGi-based kernel + subsystems + application programming model\(^1\)
  - built on Equinox
- Provisioning from local repository
- Enhanced serviceability & problem diagnostics
- Existing enterprise library compatibility
- Supports deployment of:
  - any OSGi bundle
  - war files (embedded tomcat)
  - web-module bundles
  - par files

\(^1\) Spring Dynamic Modules programming model is Apache Licensed
dm Server 1.0 – par files

- set of bundles
  - unit of packaging
  - unit of deployment
  - runtime scope
    - type visibility
    - service visibility
    - synthetic context
dm Server 1.0 – web support
dm Server 2.0
Plan files

- generalize par file concept
  - (optionally) atomic lifecycle (deploy, start, stop, ...)
  - (optionally) scoped
- more than just bundles

```xml
<plan name="greenpages.bundles" version="1.0.0" abstract="true">
  <artifact type="bundle" name="greenpages.db" version="1.0.1"/>
  <artifact type="bundle" name="greenpages.app" version="1.0.0"/>
  <artifact type="bundle" name="greenpages.jpa" version="1.0.3"/>
  <artifact type="bundle" name="greenpages.web" version="1.0.0"/>
</plan>

<plan name="greenpages.dev" version="1.0.0" scoped="true" atomic="true">
  <artifact type="properties" name="greenpages.jdbc.dev" version="1.0.0"/>
  <artifact type="plan" name="greenpages.bundles" version="1.0.0"/>
</plan>

<plan name="greenpages.prod" version="1.0.0" scoped="true" atomic="true">
  <artifact type="properties" name="greenpages.jdbc.prod" version="1.0.0"/>
  <artifact type="plan" name="greenpages.bundles" version="1.0.0"/>
</plan>
```
Plans

- Plans are just artefacts in a repository
- as is everything they refer to
- properties files provisioned from a repository are deployed as configuration admin dictionaries
- configure dm Server with an initial plan
Provisioning: bundle dependencies

• Bundle dependencies:

• Attempted installation
  -> fail to resolve!
Provisioning

- dm Server detects missing dependencies and automatically provisions bundles from repository to satisfy

Repository of bundles that *might* be installed
- Disk footprint only

Minimal runtime footprint maintained
dm Server 2.0

- Provisioning from chained repositories
  - may be local
  - or remote

```json
{
...

  "remote-repo" : {
    "type" : "remote",
    "uri" : "http://my-host:8080/com.springsource.server.repository/my-hosted-repo",
    "indexRefreshInterval" : 30,
  },
  "repositoryChain" : [
    "bundles-subsystems",
    "bundles-ext",
    "bundles-usr",
    " libraries-ext",
    "libraries-usr",
    "remote-repo"
  ]
}
```
Web application support

- web modules deprecated
- replaced by RFC-66 Reference Implementation
- + standard tomcat configuration files
- standard dm Server deployment + URL protocol support
  - install webbundle: `file:formtags.war` [?Web-ContextPath=...]
    - adds necessary manifest entries
      - symbolic name, version, imports etc.
    - sets up Bundle-Classpath to include WEB-INF/classes and all jars in WEB-INF/lib
- all dm Server war options supported (shared services, shared library)
On the criteria for decomposing systems into modules...

horizontal decomposition
On the criteria for decomposing systems into modules...

- We want **vertical decomposition** by business function as primary decomposition mechanism.

![Diagram showing web bundle, infrastructure bundle, and functions A, B, C. There is a note saying 'can't partition web layer!' ]
Web slices
Version isolation

This works in OSGi

This doesn't
Regions

- Regions isolate user code from dm server kernel code
- Kernel and users can use different versions of Spring
Version isolation

- Application A
  - Spring 2.5.6
  - Hibernate 3.3.2

- Application B
  - Spring 2.5.6
  - Hibernate 3.2.6

*Spring bundle(s) can't simultaneously bind to both Hibernate 3.3.2 and Hibernate 3.2.6*
Cloning

- Cloning enables
  - maximum flexibility, minimal footprint
  - also solves “shared static” issues
medic

• dm Server serviceability suite
  • tracing
  • logging
  • dumping
• available for application use
  • logback configuration
  • control format, destination for any bundle or group of bundles
    • independent configuration
Admin console improvements

![SpringSource dm Server](image)

**Admin Console**

Result of the last operation: 'Applications Listed'.

**Deployed Artefacts**

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Origin</th>
<th>Date</th>
<th>Undeploy</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.springsource.server.admin.plan</td>
<td>2</td>
<td>Hot Deployed</td>
<td>03-Jul-2009 14:46:15 BST</td>
<td>(N/A)</td>
</tr>
</tbody>
</table>

- **Associated Modules:**
  - com.springsource.server.admin.web
  - com.springsource.server.admin.content
  - com.springsource.server.admin-plan-synthetic-context
  - (type: Web Bundle) /admin
  - (type: Bundle) No personality identifier
  - (type: Bundle) No personality identifier
Admin console improvements

**Dump Inspector**

- Dumps available for inspection:
  - 20090703144707-0 - 03-Jul-2009 at 14:47:08

- Dump entries available for inspection:
  - osgi_20090703144707-0.zip
  - summary_20090703144707-0.txt
  - system_20090703144707-0.json
  - thread_20090703144707-0.txt
  - trickle_20090703144707-0.log

**Dump Entry Viewer**

- Date: 03-Jul-2009
- Time: 14:47:07 BST
- dm Server: 2.0.0.D-20090701121909
- Cause: MANUAL
Blueprint Service

• Support for OSGi Blueprint Service (4.2 specification)
  • aka “RFC 124”
• Standardization of Spring Dynamic Modules + core of Spring container as an OSGi Compendium Service
• Track the Spring Dynamic Modules 2.0 release
tools
Bundlor

- Consistent management of manifest across ant, maven, command-line, and your IDE
Bundlor benefits

- Automatic management of imports based on scanning project / jar
  - + Spring config, Hibernate, JPA, properties, web.xml, ...
- Import-Template supports specification of version information and optional imports without redundancy
  - Import-Template:
    
      org.springframework.*;version="[2.5.4.A,3.0.0)"

- Automatic management of versioned exports based on bundle version
  - can provide explicit version info with Export-Template
  - can restrict with Excluded-Exports
- Property substitution
- Version-range policies
Bundlor benefits

- Automated manifest generation inside Eclipse
- removes common developer pain point
SpringSource Tool Suite

- OSGi development support
  - Bundle Classpath Container (inc. test.mf support)
  - Automated manifest creation using Bundlor
- dm Server support in WTP
  - drag-and-drop deployment
  - repository management
  - console access
- bundle, package, service visualizations
- bundle inspection
- cluster deployment on EC2, VMware LabManager
SpringSource Tool Suite
Summary

• dm Server v2.0
  - provisioning: plans, repositories
  - web application support
  - full version isolation
  - medic
  - blueprint service, other enhancements

• Supporting tools: bundlor, STS

• Keep up at blog.springosource.com