

(R)evolution of Java packaging in GNU/Linux

Automating packaging

Authors:

Stanislav Ochotnický sochotnický@redhat.com

Mikołaj Izdebski mizdebsk@redhat.com

Date: *2nd February 2013*

Abstract

Packaging Java in GNU/Linux distributions is complicated by incomplete tooling. Over past 2 years, tooling and guidelines for packaging Java have changed in Fedora considerably. What used to be a 1000 line build script can soon become 100 lines of mostly metadata. We present new bleeding edge distribution-neutral tooling for packaging Maven artifacts.

Why is there a problem in the first place?

- Sort of NIH syndrome everywhere
- Each Java package a unique set of problems
 - Ant, Maven, Gradle, Ivy, 20 XML parser dependencies
- Each Linux distribution a unique set of problems
 - RPM, APT, Portage, FHS, exceptions to FHS
- Can we do better?
 - Conventions
 - Tooling
 - Sharing
 - Caring

First things first

Maven is the only widely-used Java build tool with any resemblance of conventions

RPM	Maven
Name	<artifactId/>
Version	<version/>
(Build)Requires	<dependencies/>
License	<licenses/>
%summary	<name/>
%description	<description/>
%prep	<build/>
%build	<build/>
%install	<build/>
...	...

Maven modifications in Fedora

- Custom resolver used in local mode
- Verification of models turned off in local mode
- Fix test scope dependency resolving when tests are disabled
- Approximate idea is:
 - Create a file that will map GAV to jars on filesystem
 - Maven loads this file when running in local mode
 - Return artifacts based on this mapping

Getting rid of cruft

We had this in our spec files

```
Requires(post): jpackage-utils  
Requires(postun): jpackage-utils
```

```
%post  
%update_maven_depmap
```

```
%postun  
%update_maven_depmap
```

Now we have

Fixing manual mapping for GAVs

Mapping between GAV and jar was manual

```
%add_to_maven_depmap org.apache.commons commons-io 2.5 JPP commons-io
```

Better way with the same result

```
%add_maven_depmap JPP-commons-io.pom commons-io.jar
```

Modifications of pom.xml

Old style patching

```
--- ./surefire-providers/pom.xml.sav
+++ ./surefire-providers/pom.xml
@@ -30,8 +30,10 @@
     <name>SureFire Providers</name>
     <modules>
       <module>surefire-junit</module>
+<!--
       <module>surefire-junit4</module>
       <module>surefire-testng</module>
+-->
     </modules>
     <dependencies>
       <dependency>
```

New macros

```
%pom_disable_module surefire-junit4
%pom_disable_module surefire-testng
```

File lists

Manual listing

```
%files
%defattr(-,root,root,-)
%doc LICENSE.txt NOTICE.txt RELEASE-NOTES.txt
%{_javadocdir}/*.jar
%{_mavenpomdir}/JPP-%{short_name}.pom
%{_mavendepmapfragdir}/*
```

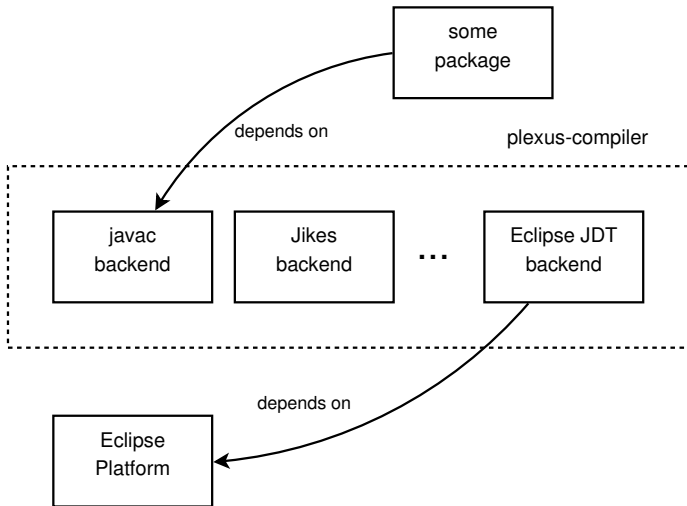
Automated way

```
%files -f .mfiles
%doc LICENSE.txt NOTICE.txt RELEASE-NOTES.txt
```


Current state

- Simple issues were solved
- Most time-consuming tasks are still manual
 - keeping dependencies up-to-date
 - installing multi-artifact packages
 - maintenance of multiple subpackages

Plexus-compiler example



A tool is needed

- Simple usage
- Powerfull
- Convention over configuration

Structure of XMvn

- Portable part
 - pure Java
 - integration with Maven
 - highly configurable
 - uses unmodified Maven
- Distribution-specific part
 - macros and shell scripts
 - integration with package manager
 - follows distribution standards
 - automatic dependency generation

Preparation for the build

Patching POM files

```
%pom_add_dep org.apache.commons:commons-io  
%pom_disable_module submod-foo
```

Launching build

```
%mvn_file : %{name}  
%mvn_build
```

During build

- Create build plan
- Read package metadata
- Call Maven to build the package
 - compile sources
 - run tests
 - generate javadocs
- Generate metadata

After the build

Installation

```
%mvn_install
```

Enumerating files

```
%files -f .mfiles  
%files javadoc -f .mfiles-javadoc
```

Example spec file (part 1)

```
Name:                maven-shared-incremental
Version:             1.0
Release:             1%{?dist}
Summary:             Maven Incremental Build support utilities
License:             ASL 2.0
URL:                 http://maven.apache.org/shared/maven-shared-incremental/
Source0:             http://repo1.maven.org/maven2/org/apache/maven/[...]
BuildArch:           noarch

BuildRequires:      maven-local
BuildRequires:      plexus-component-annotations
BuildRequires:      plexus-component-api

%description
Various utility classes and plexus components for supporting
incremental build functionality in maven plugins.

%package javadoc
Summary:             API documentation for %{name}

%description javadoc
This package provides %{summary}.
```


Example spec file (part 2)

```
%prep
%setup -q

%build
%mvn_build

%install
%mvn_install

%files -f .mfiles
%doc LICENSE NOTICE
%dir %[_javadoc]/%{name}

%files javadoc -f .mfiles-javadoc
%doc LICENSE NOTICE

%changelog
* Wed Jan 23 2013 Mikolaj Izdebski <mizdebsk@redhat.com> - 1.0-1
- Initial packaging
```

Advantages

- Simpler, more readable packages
- Easier and faster packaging and updates
- Better quality packages
- Reduced metadata redundancy
- No modifications to Maven
- Changes in guidelines are easier to introduce

Easier Maven maintenance

Maven diff

```
0001-Add-plugin-api-deps.patch | 28 --
0001-Customize-compiler-plugin.patch | 104 -----
0002-Use-custom-resolver.patch | 224 -----
0003-Use-utf-8-source-encoding.patch | 24 --
...-scope-skipping-with-maven.test.skip.patch | 160 -----
...ompiler-plugin-default-to-source-1.5.patch | 33 --
JavadirWorkspaceReader.java | 198 -----
MavenJPackageDepmap.java | 313 -----
maven-empty-dep.jar | Bin 341 -> 0 bytes
maven-empty-dep.pom | 9 -
maven-script-local | 47 ---
maven-script-rpmbuild | 93 -----
maven.spec | 269 +++-----
repo-metadata.tar.xz | Bin 3028 -> 0 bytes
14 files changed, 37 insertions(+), 1465 deletions(-)
```

Build description of maven-surefire in F-12

```
%if %{with_maven}
export MAVEN_REPO_LOCAL=$(pwd)/.m2/repository
mkdir -p $MAVEN_REPO_LOCAL
cat %{SOURCE4}
mvn-jpp -e -Dmaven.repo.local=$MAVEN_REPO_LOCAL \
    -Dmaven2.jpp.depmap.file=%{SOURCE4} \
    -Dmaven.test.skip=true install
for dir in maven-surefire-plugin maven-surefire-report-plugin \
    surefire-api surefire-booter surefire-providers/surefire-junit;
do
    (cd $dir
    mvn-jpp -Dmaven.repo.local=$MAVEN_REPO_LOCAL \
        -Dmaven2.jpp.depmap.file=%{SOURCE4} \
        javadoc:javadoc
    )
done
%else
mkdir -p lib
build-jar-repository -s -p lib classworlds junit plexus/utils
ant -Dmaven.mode.offline=true
cp -p target/*.jar ../lib/$project.jar
%endif
```

Build description of maven-surefire in F-15

```
# tests turned off because they need jmock
mvn-rpmbuild -e \
    -Dmaven.local.depmap.file=%{SOURCE1} \
    -Dmaven.test.skip=true \
    install javadoc:aggregate
```

Build description of maven-surefire in F-19

```
%mvn_build -f
```

Simplified package

maven-surefire diff between F-12 and F-18

.cvsignore		3	-
.gitignore		14	+
Makefile		21	--
maven-surefire-2.3-junit4-pom.patch		11	-
maven-surefire-booter-build.xml		64	-----
maven-surefire-build.xml		90	-----
maven-surefire-buildonlyjunit3.patch		13	-
maven-surefire-buildskiptestng.patch		12	-
maven-surefire-jpp-depmap.xml		23	--
maven-surefire-plexus12.patch		20	--
maven-surefire.spec		399	+++++-----
sources		3	+-

12 files changed, 117 insertions(+), 556 deletions(-)

Disadvantages

- Harder to debug
- Incompatibility with older systems
- Bleeding edge

Summary

- Improved packaging
- Full solution
- Backwards-compatible
- Smooth transition

Future

- Automated package generation
- Debugging tools
- Graphical tooling
- Support for more types of artifacts
- Integration with Eclipse
- Adoption by different distributions?

Links

- Code repository
 - <http://git.fedorahosted.org/git/xmvn.git>
- Fedora 19 feature
 - <http://fedoraproject.org/wiki/Features/XMvn>
- Cookbook
 - <http://mizdebsk.fedorapeople.org/xmvn/cookbook/>

The background is a dark blue, textured surface. It features several wavy, light blue lines that sweep across the frame. In the lower right quadrant, there is a cluster of small, light-colored birds in flight, scattered across the blue background.

The end.

Thanks for listening.