

# Add bundle information to jar manifest

<b>Motivation:</b>	Geotools shall be usable in OSGi environments, particularly in Eclipse applications like uDig
<b>Contact:</b>	<a href="#">Harald Wellmann</a>
<b>Tracker:</b>	<a href="http://jira.codehaus.org/browse/GEOT-2105">http://jira.codehaus.org/browse/GEOT-2105</a>
<b>Tagline:</b>	OSGi deployment is hampered by many jars

This page represents the **current** plan; for discussion please check the tracker link above.

- [Description](#)
- [Status](#)
  - [Tasks](#)
  - [POM Changes](#)
  - [API Changes](#)
  - [Documentation Changes](#)

## Description

Currently, uDig wraps up the GeoTools library (and all its jars) into a single bundle. This has been an okay stop-gap solution - but now more projects are making use of OSGi bundles and are being held up by the steps required to unpack our jars; and repackage them with a descriptive MANIFEST.MF file.

For a detailed discussion of different options for Geotools to become fully OSGi compliant, see [03 GeoTools and Eclipse or OSGi](#). This proposal focuses on a first step which will solve the problem at least for all Eclipse users and which will not require any source code changes in Geotools. It will only require adding a MANIFEST.MF file to every Geotools module and changing each pom.xml to use this manifest in the package phase.

In the end, every Geotools JAR gt-foo.jar will be an OSGi bundle, simply by adding some extra headers to the JAR manifest. The Factory SPI mechanism, which was assumed to be incompatible with OSGi, will continue to work both in OSGi (at least with the Eclipse Equinox implementation) and plain old classpath environments.

There are a number of organizational prerequisites:

1. Define a naming scheme for the OSGi bundle symbolic name. Proposal: org.geotools.<artifactId without "gt-" Prefix>. Thus, the bundle symbolic name of gt-referencing.jar would be org.geotools.referencing.
2. Decide on the location of the MANIFEST.MF files. According to Maven defaults, they would be placed in src/main/resources/META-INF, but unfortunately Eclipse PDE expects them to be in <project root>/META-INF. Proposal: Use <project root>/META-INF.
3. Decide on how to deal with external dependencies of Geotools, like vecmath.jar or jsr108.jar. Each of these third-party libraries also needs to be turned into a separate OSGi bundle with the required manifest headers. For some of these libraries, there are OSGi-compliant Maven artifacts, e.g. in the SpringSource Enterprise Bundle repository. However, some of these are buggy, and others are not available at all, so Geotools needs to define a procedure for providing its own OSGi compliant Maven artifacts for external dependencies. Once a naming convention and a Maven repository location for such external dependencies have been defined, each external artifact foo can be osgified by defining a new artifact foo-osgi which depends on the original foo. The maven-bundle-plugin can be used to create a foo-osgi.jar with the required manifest headers by extracting the class files from the original jar and repackaging them in the OSGi jar. (Unfortunately, this approach does not work for Geotools itself due to the large number of split packages and the use of META-INF/services. In some cases, one has to be prepared for extra manual work when external libraries have split packages or cyclic dependencies.)

## Status

This proposal is open for discussion.

Voting has not started yet:

- [Andrea Aime](#)
- [Ian Turton](#)
- [Justin Deoliveira](#)
- [Jody Garnett](#)
- [Martin Desruisseaux](#)
- [Simone Giannecchini](#)

## Tasks

*This section is used to make sure your proposal is complete (did you remember documentation?) and has enough paid or volunteer time lined up to be a success*

	no progress		done		impeded		lack mandate/funds/time		volunteer needed
--	-------------	---	------	---	---------	---	-------------------------	---	------------------

1. API changed based on BEFORE / AFTER
2. Update default implementation
3. Update wiki (both module matrix and upgrade to to 2.5 pages) |
4. Remove deprecated code from GeoTools project
5. Update the user guide
6. Update or provided sample code in demo
7. review user documentation

## POM Changes

1. Identify an OSGi compliant Maven artifact for each external dependency of Geotools.
2. Change the Geotools POMs to reference these OSGi compliant dependencies instead of the original ones.
3. Create an OSGi compliant MANIFEST.MF for each Geotools module. Each dependency (both Geotools and external) has to be listed in the Require-Bundle header with its bundle symbolic name. In addition, each bundle contributing a service to the Factory SPI has to include a header Eclipse-RegisterBuddy: org.geotools.metadata. The org.geotools.metadata manifest has to include a header Eclipse-BuddyPolicy: registered.
4. Configure the Maven archiver to use the provided manifests.

## API Changes

None.

## Documentation Changes

Pages affected by this proposal:

- [Home Developer Guide](#) should be updated; probably the release process where version numbers are changed
- [03 GeoTools and Eclipse or OSGi](#)