m2e 1.3 Release Review

Planned Review Date: 2013-02-20

Communication Channel: m2e-wtp@eclipse.org

Igor Fedorenko (ifedorenko@sonatype.com)

Jason van Zyl (jason@sonatype.com)

Fred Bricon (fbricon@gmail.com)

Introduction

- m2e provides both a framework for building Maven centric Eclipse tools and a set of tools and user interface elements directly consumable by Eclipse Users.
- As a framework, m2e provides facilities to import and configure Maven projects in Eclipse workspace as well as APIs that allow m2e extensions to access Maven project metadata and participate in Maven project configuration and workspace build.

Introduction cont'ed

- For the end user, m2e provides
  - wizards to import existing and to create new Maven projects
  - rich form-based and text-based pom.xml file edito
  - launch configuration types to launch Maven build directly from Eclipse IDE
  - access Maven repository index to browse repository contents and in various code-assist
- JDT integration, both directly useful to endusers and extensible

Project history

- m2e started as “m2eclipse” at codehaus in 2005
- Codebase moved to Sonatype public SVN repository in 2008 and to a public Github repository in 2010
- Proposed as Eclipse project in 2008 but actual move was delayed due to m2e dependency on prereleased version of Maven 3.0 (released on 2010-10-08)
- m2e 1.0 was released in June, 2011, as part of Eclipse Indigo simultaneous release
- m2e has been ranked #1 or #2 most downloaded plugin on Eclipse Marketplace for the past 12 months

New features

- The following features have been added or significantly reworked in m2e 1.3 release
  - New APIs to allow 3rd party adopters to better control the project conversion to Maven experience
  - New API to better control the classpath of launch configurations, when using dependencies with classifiers
  - Improvements on the archetype selection page
  - A new discovery catalog for m2e 1.3 compatible 3rd party extensions

Non-Code Aspects

- Localization, Internationalization, Accessibility
  - All m2e strings are externalized, but only English strings are provided.
  - m2e development team did not do any localization and/or accessibility testing due to resource constraints.
- Active user and development mailing lists as well as #m2eclipse IRC channel at codehaus.
- User and Developer Documentation and Wiki are outdated and largely not applicable to m2e 1.0+

API changes

- The following behaviors changed in m2e 1.3
  - The behavior for <execute/> mapping of maven plugins has been changed to runOnIncremental=false by default, to prevent random build loops
  - The JDT conversion participant is now restricted to the “jar” packaging by default. 3Rd party adopters need to explicitely request the maven-compiler-plugin to be configured, via an extension point

New APIs

- Added new <conversionParticipantConfiguration> extension point to explicitely request ProjectConversionParticipant invocation per packaging type
- Introduced a new ResolverConfiguration#lifecycleMappingId attribute, if specified, the corresponding ILifecycleMapping will be used regardless of any other lifecycle mapping metadata
- Introduced new flavour of AbstractProjectConfigurator#addNature that accepts updateFlags parameter that allows finer control over new nature creation.
- Added new AbstractJavaProjectConfigurator#getOutputLocation method to provide better extensibility

New APIs

- Added new <conversionParticipantConfiguration> extension point to explicitely request ProjectConversionParticipant invocation per packaging type
- Added a new ConversionEnabler, providing the ability to
  - preselect the packaging type during project conversion
  - exclude project conversion participants for certain projects
- Added new attributes to order project conversion participants execution
- Added new API for Runtime classpath resolution of dependencies referenced with classifiers

New APIs cont'ed

- Introduced a new ResolverConfiguration#lifecycleMappingId attribute, if specified, the corresponding ILifecycleMapping will be used regardless of any other lifecycle mapping metadata
- Introduced a new flavor of AbstractProjectConfigurator#addNature that accepts updateFlags parameter that allows finer control over new nature creation.
- Added new AbstractJavaProjectConfigurator#getOutputLocation method to provide better extensibility

Workspace compatibility

- There were no workspace metadata format changes between version 1.2 and 1.3. m2e 1.3 is expected to work with workspaces created with 1.2.

Architectural Issues

- m2e continues to suffer from lack of proper support for nested workspace projects
- Lack of support for proper classpath separation between test and main/production source paths in JDT is another long-standing problems that affects m2e.

Tool Usability

- Generic tools for working with Maven projects inside Eclipse workspace
  - Automatic discovery and installation of required m2e extensions based on pom.xml
  - Feature-rich pom.xml editor, error markers and quick-fixes for common problems and best practices
  - Maven launch configuration types
  - Automatic project dependency resolution
- Support for “plain” Java projects out-of-the box via JDT integration
  - JDT project configuration based on pom.xml
  - Automatic compile classpath managemen
  - Maven specific classpath in RunAs JavaApplication and JUnitTest

Tool Usability cont'ed

- Conversion of existing Eclipse project to Maven
  - Wizard for generating a new pom.xml
  - Extensible mechanism for generation of Maven model in pom.xml

Bugzilla

- 18 bugzilla records were closed as fixed in m2e 1.3

Standards

- m2e 1.3 is fully compatible with pom.xml format used by Maven 3.0.x, 2.1.x and 2.0.x
- m2e 1.3 is fully compatible with Maven “default” repository layout and Maven repository Index format
- m2e 1.3 provides limited support for Maven Archetype descriptor format

UI Usability

- m2e team did not evaluate m2e conformance to the User_Interface_Guidelines
- The code is expected to support multiple languages but this has not been tested by m2e team

Project Health

- Moderately active codebase
  - No git commit statistics were collected for m2e 1.2
- Separate git repository at Github for unit and integration tests due to significant effort involved in moving test code and its dependencies to Eclipse
- m2e 1.2 did not have publicly accessible CI build work is being done to setup CI system at http://ci.tesla.io:8080/job/m2e/

Schedule

- m2e build and release schedule is aligned with Eclipse yearly simultaneous release

Communities

- Active user community
  - No community activity statistics were collected for m2e 1.2
- Active developer and adopter communities
- All project decisions are discussed on m2e-dev mailing list
- Active irc://irc.codehaus.org/m2eclipse IRC channel
- m2e team maintains a number of opensource extensions at Github that are both
- useful to endusers and as working examples for adopters
- Close collaboration with Apache Maven; collaboration with m2e/wtp, JBossTools and other opensource projects

Committer Diversity

- Developers from Sonatype and Red Hat work on m2e core

IP Log

- m2e 1.3 IPLog has been submitted for Eclipse IP team approval.

IP Issues

Notes

Credits and Kudos

- Thanks to Marcos Muñoz and Roberto Sanchez Herrera for actively contributing patches to m2e core