



EMF 2.1 Release Review

Ed Merks, Elena Litani, Marcelo Paternostro

June 30, 2005



Agenda

- Features
- Non-Code Aspects
- APIs
- Architectural Issues
- End-of-Life
- Bugzilla
- Standards
- Schedule
- Process
- Community
- IP Issues
- Project Plan

Features

- 2.1 Plan available at http://dev.eclipse.org/viewcvs/indextools.cgi/~checkout~/emf-home/docs/dev-plans/emf_project_plan_2.1.html
- Major features
 - **XML Support**
 - Load any XML. Provided a resource implementation that allows to load any XML document without a need to specify a model for this document (uses xsi:schemaLocation, noNamespaceSchema location attributes)
 - DOM and XPath. Provide mechanism to load EMF resources using DOM as an input source, as well as serializing resources to DOM trees. Using serialization to DOM user can use XPath to query EMF instance models.

Features (Continued)

- Major features (continued)
 - **EMF/SDO runtime performance.**
 - Improved performance of serialization and deserialization of XML/XMI. Improved performance for copy. Reduced footprint of EMF models.
 - **Change Model.**
 - Support “pause” when recording changes
 - Provide a mechanism for recording and applying changes for FeatureMaps to help to provide complete redo/undo capabilities for models that use FeatureMaps.



Features (Continued)

- Major features (continued)
 - **Improving generator capabilities**
 - EMF now provides an extension-point so that model importers can be added to the framework. The existing importers (Rose, XML Schema, Java annotated interfaces and Ecore model) were refactored into this design, being contributed by different plugins. We also created an API that can be used by anyone that is providing an importer.
 - The user can now specify a directory where the code generator will look for customized templates.
 - The JET now supports conditional inclusion.
 - The users can now easily customize the packages in which the classes will be generated, allowing them to, for example, generate their implementation classes in a “internal” package.

Non-Code Aspects

- XML Schema Infoset Model (XSD) has moved to become part of EMF project.
- New translations (for 2.1 changes) are not available in 2.1 release. Similar to the platform translations, these should be available for Eclipse 3.1.1 timeframe.
- Documentation available at <http://eclipse.org/emf/docs.php>
 - [FAQ](#) (updated), Improved Javadoc
- Generated standalone tests for EMF instance models
 - Includes basic model tests
 - Includes basic standalone tests for loading the EMF resources from XML streams.
- Ant tasks
 - In 2.1 we provide Ant tasks to automate most of the code generation actions available in EMF: invoke JET and JMerger, and code generation from Rose and XSD.

Non-Code Aspects

- Alignment with Eclipse standards
 - EMF plugins
 - Uses the OSGi Bundle Manifest file
 - Deployed as jars
 - Available through Update sites
 - The model artifacts are not placed in the source directory
 - Doc plugins shipped with the search indexes
 - Generated plugins
 - Uses the OSGi Bundle Manifest file
 - Prepared to be deployed as jars (build.properties)

APIs

- Historically, clients of EMF have treated every class in EMF as API. Therefore, we are left with no choice but to try our best to treat every class as API.
- Since changes are required (e.g. when we need to implement new features, improve performance) and it is not always possible avoid changing *impl packages and/or protected methods. In such situation each change is considered carefully and we weight benefit against risk. In the case we decide to go ahead and implement the changes, we work with EMF clients to ensure that no breaking changes are introduced. For example, during last release we worked closely with VE/GEM project to ensure that changes introduced in do not break them.
- It is worth mentioning that the code generated by EMF 1.0.1 still runs on EMF 2.1 runtime.



Architectural Issues

- Improved architectural quality
 - Re-factored code generator code including providing new importer plug-ins
 - Improving EMF runtime performance
 - Change recording model was modified to solve the problem on how to record data on FeatureMaps

End-of-Life

- EMF.Codegen
 - Introduced new common utility `util.CodeGenUtil` and deprecated corresponding methods in other classes (see deprecated methods in `org.eclipse.emf.codegen.CodeGen`)
 - `GenTypedElement` has now common methods for `GenOperation`. Methods on `GenOperation` are deprecated
- EMF.Ecore
 - `ResourceImpl` deprecates `addModificationTrackingAdapters`, `removeModificationTrackingAdapters` instead `attachedHelper` is used.
 - Deprecated `EList` related methods in `EcoreUtil`. Use methods `ECollections` (common plugin)
 - `XMLResource` deprecated methods that allow manipulation of IDs from `Map`. Instead directly use `setId/getId` methods

Bugzilla

- 503 bugs/feature requests opened for 2.1 release (not including Theme bugs)
- 481 bugs/feature requests resolved or closed
- 22 bugs/feature requests deferred

Standards

- SDO 1.0 implementation is based on JSR-235



Schedule

- Our commitment was to provide a stable EMF driver one week after Eclipse Milestone driver is available. We have met this goal.

Process

- This release has been developed using open, transparent, and inclusive processes.
- This release has followed its charter principles.
- The EMF project makes appropriate use of Bugzilla, mailing lists (emf-dev@eclipse.org) and newsgroups (eclipse.tools.emf).
 - Almost all changes are described by a bugzilla.
 - Every time a build is published, we also release a list of all the bugzillas implemented for that build (release notes)
 - “What’s new CVS” – has links to all the new commits to CVS, with a bugzilla number attached to those changes
- Committer changes:
 - Kenn Hussey was added.
 - All the non EMF XSD committers gave up their status



Community

- Continue to encourage interaction in Bugzilla and continue answering all questions on the newsgroup.
- Worked with several external contributors to initiate EMF technology project
 - EMFT proposal
 - CDO
- Increased coordination with UML2 tools project.
- EMF is the 3th most popular download at Eclipse.org.
- EMF is heavily used by other Eclipse projects (WTP, UML2, TPTP, VE, etc)

IP Issues

- The EMF project verifies that:
 - the about files and use licenses are in place
 - all contributions (code, documentation, images, etc) have been committed by individuals who are Members of the Foundation
 - there were no external contributions during this release



Project Plan

- Draft 2.2 plan is not available yet.
- Candidates for 2.2 release are:
 - J2SE 5 support
 - Continue improving performance of EMF runtime
 - Provide Ant tasks to generate code from Java Annotated interfaces and Ecore



Thank you!

Q&A

June 30, 2005