
# Eclipse Modeling Framework 2.2 Release Review

May 26, 2006

 

# Agenda

- Features
- Non-code Aspects
- API
- Tool Usability
- Architectural Issues
- End of Life
- Bugzilla
- Standards
- UI Usability
- Schedule
- Process
- Communities
- IP Issues
- Project Plan

Based on Release Review Guidelines version 032 (Jan 15, 2006)

 

# EMF 2.2 Features

- Planned Features – Completed:
  - SDO support for loading XML without schema (104717)
  - XML ease-of-use-utilities: XMLProcessor (104718)
  - EMF.Edit enhancements: reset/deletion support (105964)
  - Content adapter for reverse of 1-way references (75922)
  - Cross resource containment (105937)
  - XMI 2.1 support (76538)
  - Improve XSD generation (104893)
  - Simplify customization of build infrastructure (105923)
  - Decouple JMerger implementation from JDOM (78076)
  - Generator extensibility (75925)
  - Improve code generation error reporting and handling (104727)
  - Performance optimizations (116307)
- Planned Features – Uncompleted:
  - Definition and diagnosis of Ecore model constraints (75933)

 

# EMF 2.2 Features

- Additional Noteworthy Features:
  - Reader/Writer support in resources (115389, 115393, 117555, 141170)
  - Virtual delegation pattern (114922)
  - Standalone generator support (117022)
  - Package literals interface pattern (117353)
  - Feature ID based reflective methods (117547)
  - Minimal reflective methods pattern (118040)
  - Pure API, no API patterns (118424)
  - GenModel annotations (119287)
  - Model exporter framework (109300)
  - Improved schema annotation/roundtripping (121765, 139447, 139433, 139434)
  - Editor support for containment proxies (118688)
  - Enum support for arbitrary literals (103080)
  - Encryption support in resources (126093)
  - Change recorder based command (136356)
  - Problem indication on editors (139533)
  - Ant task to generate GenModel and code from Ecore file (141141)
  - Example installer wizard (139189)

 

# Non-code Aspects

- Update of existing documentation and localization in progress
- Documentation made available on help.eclipse.org
- Maven repository created to facilitate use by other open source communities (e.g. Apache)
- EMF articles by committers and contributors in Dr. Dobb's Journal, DevX.com, and Eclipse Review

 

# API


- Historically, clients of EMF have treated every class in EMF as API; we are left with no choice but to try our best to treat every class as API.
- We resist changes even to implementation packages and protected methods whenever possible. When deciding whether changes are necessary for new features or performance improvements, we consider each change carefully and weigh benefit against risk. If changes are made, we work with EMF clients to ensure that no breaking changes are introduced.
- We maintain backwards compatibility with old code generation in our runtime: code generated by EMF 1.0.1 still runs on EMF 2.2 runtime.

 

# API Changes

- New BasicEObject delegation pattern to implement feature ID based reflective methods eGet(), eSet(), etc.
  - Provides backwards compatibility with existing generated subclasses, but not with a mix of old and new patterns
  - Before regenerating a model you must ensure that any base model it extends has already been regenerated with EMF 2.2
- JETException no longer extends CoreException
  - Use DiagnosticException.toCoreException() to convert
- New getEFactory() methods added to EPackage.Descriptor and EPackage.Registry interfaces

 

# Tool Usability

- Although tool polish has not traditionally been EMF's focus, we were able to make usability improvements in EMF 2.2:
  - New problem indication page for EMF-based editors
  - New model exporter wizard
  - Improved error reporting in model importer wizard and code generator tools
  - Improved handling of read-only resources in EMF-based editors

# Architectural Issues


- Code generator (JET and GenModel) was refactored to remove hard dependency on Eclipse core runtime, for stand-alone support
- Introduction of model exporter framework led to refactoring of existing importer framework, with common elements now residing in a converter plug-in
- JMerge was replaced by an equivalent, facade-based implementation to facilitate replacement of JDOM

 

# End of Life

- No APIs were removed. Deprecated APIs will be kept operational until at least the next major release.
- Deprecated:
  - org.eclipse.emf.codegen.jmerge.* replaced by equivalents in org.eclipse.emf.codegen.merge.java and org.eclipse.emf.codegen.merge.properties
  - Eclipse-dependent CodeGenUtil methods replaced by CodeGenUtil.EclipseUtil equivalents
  - GenBase, GenModel and GenPackage methods for code generation replaced by Generator equivalents

 

# Bugzilla

- Between July 8, 2005 and today:
  - 701 reports were created
  - 505 were fixed
  - 176 were resolved without changing code (duplicate, invalid, etc.)
- EMF 2.2 contains:
  - 170 enhancements
  - 316 bug fixes
- Zero blockers or critical bugs remain

 

# Standards

- OMG XMI support updated to 2.1
- Continuing interchange capabilities with OMG MOF 2.0 – can read and write Ecore as EMOF
- Continuing support for SDO 1.0 (JSR-235)
  - We are also working with and supporting the Apache Tuscany effort to implement SDO 2.0

 

# UI Usability

- EMF tools use Eclipse platform APIs to provide accessible UI
  - No outstanding accessibility bugs
- All text is externalized, and has been translated into 23 languages, including BiDi languages
  - Arabic, Simplified Chinese, Traditional Chinese, Czech, Danish, Dutch, Finnish, French, German, Greek, Hebrew, Hungarian, Italian, Japanese, Korean, Norwegian, Polish, Portuguese, Brazilian Portuguese, Russian, Spanish, Swedish, Turkish
  - Translation verification testing in progress, to complete mid-June

# Schedule


- Development schedule closely followed the Eclipse Project's
  - Milestones every 6 weeks, trailing Eclipse's by one week
  - API freeze and feature complete at M6
- End game was not well planned – timing of release candidates initially uneven, as coordination was worked out with Callisto
  - Now settled in to weekly RCs, scheduled to end June 5, in advance of June 28 release

 

# Process

- EMF is developed using an open, transparent, and inclusive processes – this release follows its charter principles
- EMF makes appropriate use of Bugzilla, mailing lists and newsgroups
  - All changes are described by a Bugzilla report
  - Bugzilla used to compile release notes for every integration, milestone, and release candidate build
  - Release notes link bugs to CVS commits
- Committer changes:
  - Bertrand Portier gave up his status

 

# Communities


- Continuous effort to build community through newsgroup and Bugzilla responsiveness
  - Continually ranked #1 or 2 in liveness on Eclipse dashboards
- Tutorials and presentations by EMF committers at conferences
  - EclipseWorld, OOPSLA, EclipseCon
- Cross-project co-operation at Eclipse and with other communities (e.g. Apache Tuscany)
- Through the establishment of the EMFT and GMF technology projects and the creation of the top-level Modeling Project, a larger community is being built around modeling at Eclipse

 

# IP Issues

- About files and license files are complete and correct
- There have been no significant non-committer code contributions requiring review by the Eclipse Foundation's legal staff
- All third-party libraries have been documented in the release and reviewed by the Foundation's legal staff
- A Project Log has been created and reviewed by the Foundation's legal staff, and is available at
  http://www.eclipse.org/emf/eclipse-project-ip-log.csv

# Project Plan

- Focus for next release is likely to be Java 5.0 support
- No draft plan is available yet

 