



# Graphiti 0.10.0 Release Review

Planned Review Date: June 2013  
Communication Channel: [eclipse.graphiti](http://eclipse.graphiti)  
Christian Brand  
Matthias Gorning  
Tim Kaiser  
Jürgen Pasch  
Félix Velasco  
Michael Wenz

# Introduction



Graphiti is an Eclipse-based graphics framework that enables the fast and easy creation of graphical tools, which can display and edit underlying domain models using a tool-defined graphical notation. Graphiti supports the developer in getting to a first version of such a tool with very low effort by:

- Hiding platform specific technology (e.g. GEF / Draw2D on Eclipse)
- Providing rich default implementations inside the framework
- Providing a default look and feel that was designed in close co-operation with usability specialists

# Features (I)



- Make Graphiti fit for being part of the release train and eventually leaving incubation
  - Graphiti is part of the Kepler Eclipse release train since M1
  - All requirements for being part of the release train were fulfilled (see separate slide)
  - Infrastructure related topics like consumption of platform changes were taken care of
- Round-offs and ease of use
  - A lot of round-offs for framework functionality easing the development of Graphiti-based editors have gone into this release. Especially the Diagram Editor API has undergone changes. It was evolved to a Diagram Container API to enable diagrams also in views and other locations. The documentation has been improved in many areas.

# Features (II)



- New functionality (in order of their appearance)
  - Single-Click features
  - File system example plugin showing the usage of patterns
  - Shortcuts for triggering features
  - Rich text support
  - ID patterns
  - Text decorators
  - API access to export as image and print features
  - Diagrams in views
- Additionally a lot of round-offs have been implemented
- For the full list of changes and implementation details see [Graphiti New & Noteworthy page for 0.10.0](#)

# Security



Currently there is no indication that any of the known vulnerabilities affect Graphiti

# Non-Code Aspects (I)



- User documentation is available as part of the Graphiti SDK installation from the update site
- A large part of this documentation is a tutorial that introduces users step by step to the most common features of Graphiti
- The tutorial and documentation have been updated to reflect the changes done within the framework
- A new example (Filesystem diagram) based on patterns has been added
- Further examples are part of our tests and can be synced from Git

# Non-Code Aspects (II)



- Website at <http://www.eclipse.org/graphiti>
- Graphiti can be downloaded via p2 update sites and as a zipped version
- Available features
  - Graphiti Feature (the framework)
  - Examples Feature (includes examples, doc plugin with eclipse help integration and javadoc)
  - Export Feature (diagram export to SVG)
  - Tools Feature (tools that can be used to create Graphiti-based editors)
  - SDK Feature (framework, examples, additionally includes sources to enable debugging)
  - SDK+ Feature (additionally includes optional parts, currently the SVG exporter)

# Non-Code Aspects (III)



- Graphiti uses a Buckminster- and Hudson-based build and test infrastructure
- The framework plugins are provided with dedicated test plugins checking their valid behavior
- Unit and SWTBotTests are executed as a part of the Buckminster Build on the Eclipse [Hudson](#)
- Each new build is tested at least with Eclipse 4.3 (Kepler), another test environment is Eclipse 4.2/3.8 (Juno)
- Graphiti is part of the Release Train since Eclipse Indigo (M4)

- All non-API code is in «internal» packages
- APIs are consolidated and are high quality including API contract in form of JavaDoc
- API will further evolve based on community input
- Balance between hiding GEF/Draw2d from clients to reduce complexity and opening up the API to be heavily customizable

# Architectural Issues



- Graphiti's architecture is solid and basically unchanged since the basis of the framework was already productive within SAP AG for several years
- Nevertheless several reworks have been done in the 0.10.0 timeframe to improve Graphiti's architecture to better support an open communities' varied requirements
- Some parts of the framework coding needing some rework (e.g. in the rendering area) could not be addressed within this release and will be targeted for the next release

# Tool Usability



- Tutorial supports the getting started process for new users
- Good results in tool building can be achieved early
- Short turnaround cycles support incremental development
- Easy debugging enables fast problem solving (compared to frameworks based on code generation)
- Test preferences allow user to enable additional means to identify and analyze issues besides debugging

# End of Life Issues / Changed APIs



- The diagram editor API has undergone a major rework
  - The new API also supports diagrams in views and other plain UIs
  - Common parts are shared in new class `DiagramBehavior` and reused from both editors and views
  - For details see [Bug 352120](#)
- Some minor API clean-ups
  - API deprecated for 0.9.0 has been removed
  - Image provider API takes additional parameter to separate between different diagram types
- No deprecated or removed features

# Bugzilla



- Bugzillas during 0.10.0 timeframe
  - Opened: 81
  - Closed: 50
  - Deferred: 1
- Outstanding (mainly enhancements):
  - P1: 0
  - P2: 1
  - P3: 22
  - P4: 61
  - P5: 6
- [All open Graphiti issues in Bugzilla](#)

# Standards / Execution Environment



- Execution Environment JavaSE 1.5, but also runs on Java 6 and Java 7
- Supports Eclipse Platforms 4.3, 4.2 and 3.8
- We will support Eclipse 3.8 and 4.2 equally, and all the functionality will be the same
- Based on Draw2D/GEF and SWT, de-facto standards for graphics in Eclipse
- Integrated with and based on EMF, de-facto standard for modeling in Eclipse
- No native components, therefore there are no further requirements to the execution environment

# Simultaneous Release Specific



- Build offset: +3
- [Project plan](#)
- [IP Log](#)
- Direct communication: Michael Wenz, Matthias Gorning
- No deviations for simultaneous release requirements
- [New & Noteworthy](#)
- Retention policy
  - Released versions will be kept forever.
  - Final milestones (announced via mail and the downloads page on the Graphiti page) for an upcoming release will be kept until the final release is available.
  - Nightly (dev) builds might be removed without prior notice. The same is valid for milestone builds that happen each day on the way towards each milestone and integration builds.
  - All source code will be kept in the Eclipse repository (Git, /gitroot/gmp/org.eclipse.gmp.graphiti)

# User Interface Usability



- Usability
  - UI designers were part of the initial development of the framework at SAP AG
- Eclipse User Interface Guidelines are adhered to
- Globalization is supported
  - All strings in separate resources
  - Graphiti is enabled for translation via the Babel project
- Graphiti diagrams can also be used via keyboard only and support displaying and editing while the underlying operation system runs in high contrast mode (or any other mode that eases handling for disabled people)
- Screen readers are currently **not supported**

# Schedule



- Graphiti is part of the Kepler release train since M1
- All milestones since then have been met
- Because of the reduction of the team capacity, Graphiti had to adapt the original scope
- However, the Graphiti team achieved the main targets of the scope for this release and unplanned topics thanks to contributions
- Additional unplanned topics
  - [Bugzilla 352120](#): Diagrams in views
- Topics needed to be dropped during the release
  - Revise implementation of GFText (framework internal code cleanup)

# Communities (I)

- [Bugzilla](#)
  - Bugs and enhancements are tracked using Bugzilla
  - User feedback and enhancement request received
  - Many contributions (bug fixes and enhancements) received
- [Newsgroup](#)
  - Is used as active communication channel (~280 threads since last release 0.9.0)
  - Many detail questions, discussions and enhancement ideas start there
- Conferences
  - Bof session at EclipseCon Europe 2012
  - Presentations at Eclipse Demo Camps both for Juno (Bonn, Walldorf and Vienna) and Kepler – talk is agreed for Walldorf



# Communities (II)

- Conferences (cont.)
  - Several talks at EclipseCon 2012/13 referred Graphiti
- Eclipse projects using Graphiti
  - [eTrice](#) diagram editor
  - [JPA Editor](#) diagram editor
  - [CDO/Dawn](#) integrates with Graphiti
  - [BPMN2 Modeler](#) diagram editor
- Other open source projects
  - [Activity BPM Platform](#) offers an Eclipse Designer built on Graphiti
  - [The KIELER research project](#) integrates their layouting algorithms with Graphiti
  - [Spray](#) supports a generative approach on top of Graphiti

# IP Log



- The Eclipse IP policies and procedures have been followed
- The Graphiti IP Log can be found at [http://www.eclipse.org/projects/ip\\_log.php?projectid=modeling.gmp.graphiti](http://www.eclipse.org/projects/ip_log.php?projectid=modeling.gmp.graphiti)
- A frozen version is available at <http://www.eclipse.org/graphiti/iplog/0.10.0.pdf>

# IP Issues



The EMO explicitly asks during the Release Review if any Member would like to assert that this release infringes their IP rights.

If so, the EMO and the project will follow the Eclipse IP Policy in discussions with that Member.

# Credits and Kudos



- Thanks to the community for the most valuable discussions and feedback in the forum and in bugzillas
- Also Kudos to all who contributed to the project, be it in form of feedback, suggestions, questions or most valuable code contributions
- Special thanks to:
  - Patrick Paulin
  - Benjamin Schmeling
  - Hernan Gonzalez

For providing a great amount of high-quality contributions