



## *Riena Release Review 1.2*

Project Lead: Christian Campo  
([Christian.Campo@compeople.de](mailto:Christian.Campo@compeople.de))

Riena Mailing List: [riena-dev@eclipse.org](mailto:riena-dev@eclipse.org)

IP Log: [http://www.eclipse.org/projects/ip\\_log.php?projectid=rt.riena](http://www.eclipse.org/projects/ip_log.php?projectid=rt.riena)



# Agenda

- Features
- Non-Code Aspects
- APIs
- Architectural Issues
- Tool Usability
- End-of-Life
- Bugzilla
- Standards
- UI Usability
- Schedule
- Communities
- IP Issues
- Future plan

# Riena - Mission



The Riena platform is a foundation for building multi-tier enterprise client/server applications.

As such Riena broadens the usage of the service oriented architecture of OSGi/Equinox by providing access to local and remote services in a transparent way.

The Riena user interface components supports an enduser focused navigation metaphor which can be used as an alternative to the default workbench layout.

Riena builds new or leverage existing components that are need to implement client/server applications based on Eclipse technology.

- Riena project
  - proposed in 10/2007
  - approved 12/2007
  - 1.0.0 release Januar 2009
  - 1.1.0 release with Galileo

# Features



- Distributed Enterprise Applications
  - Remote OSGi Service based on Webservice model
- Enhanced UI Model based on RCP
  - Based on RCP but with a different look and navigation model (enduser focused)
  - separation between controller/view using Ridget API
  - Validator, Converters, Markers, UI Filters
- Riena Core
  - Injector for services and extensions
- Security
  - using equinox security
  - supports authentication and authorization (client and server)
- Object Transaction
  - follows object net modifications in a distributed environment
- Monitoring
  - transfer logs, statistics etc from client to server
- New and Noteworthy → [http://wiki.eclipse.org/Riena\\_New\\_And\\_Noteworthy](http://wiki.eclipse.org/Riena_New_And_Noteworthy)



## *New in 1.2.0*

- Focused on collecting feedback, fix bugs (97 bugs)
- Keyboard support in navigation panel
- more ridgets (List, CCombo, Link, Browser, Slider, Scale, Spinner)
- improvements MasterDetail (feedback from the community)
- AttentionMarker, priorities for Markers
- trigger relayout View from Controller for Views with dynamic content
- reworked extensionpoint id schema, added extension doc and more javadocs for APIs (where it was missing)



## *Non-Code Aspects*

- Riena Wiki pages which document a number of aspects in Riena
- Various samples to showcase the individual UI aspects of Riena that are supplied as an extra feature
- Client/Server demo application
- snippet project (and wiki page) that show how to use Ridgets
- Many JUnit TestCases (2080 on last count) to show use of the Riena API
- Javadoc

# APIs



- Many Testcases (2080) to validate that APIs are working
- Many extension and services where users can replace Riena functionality with their custom implementation.
- Two bundles (o.e.riena.ui.ridgets.swt and o.e.riena.security.common) were set to version 2.0. because we broke existing API
- The product version is still 1.2. because we think that most apps will not implement their own ridgets and have only a few areas where they hit the security API issues. So existing Riena apps should run out of the box in most cases.

# APIs



- We plan on switching to version 2.0 after 1.2. because we want to refactor a number of APIs in the following release. That will mean some broken APIs in Riena 2.0.
- Changes will be documented as early as possible in the New and Noteworthy. We will use deprecation for old APIs if possible.





## *Architectural Issues*

- We use OSGi Services wherever that is possible.
  - easily consumable by others
  - replaceable by custom services
- We use Extensions to make Riena extendable by others
- Riena can be consumed in pieces in the steps:
  - core, remote services, ridgets, with navigation
- OSGi Services of Riena are registered with a low Service Rank so that they can easily be replaced by a custom services.



## *Architectural Issues (2)*

- Dependency Injection used in all placed for services and extensions using the Injector APIs
  - making components themselves independant of the Injector API
  - make them easily testable using Mocks
  - automatically tracked coming and going services and extensions

## *Tools Usability*



- Riena currently has no tools

## *End of Life*



- We plan to stay around for longer 😊.

# Bugzilla



		Status					Total	
		NEW	ASSIGNED	REOPENED	RESOLVED	VERIFIED	CLOSED	Total
Severity	critical	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>2</u>
	major	<u>1</u>	<u>0</u>	<u>0</u>	<u>10</u>	<u>0</u>	<u>1</u>	<u>12</u>
	normal	<u>79</u>	<u>5</u>	<u>1</u>	<u>257</u>	<u>3</u>	<u>3</u>	<u>348</u>
	minor	<u>7</u>	<u>0</u>	<u>0</u>	<u>20</u>	<u>1</u>	<u>0</u>	<u>28</u>
	trivial	<u>6</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>6</u>
	enhancement	<u>45</u>	<u>1</u>	<u>0</u>	<u>68</u>	<u>0</u>	<u>1</u>	<u>115</u>
	Total	<u>138</u>	<u>6</u>	<u>1</u>	<u>356</u>	<u>4</u>	<u>6</u>	<u>511</u>

# Standards



- Attempt to base as much as possible using OSGi standards
- Remote Services are based on the concept of Webservices
- RFC 119 (standard for remote services since 9/2009) is certainly interesting for Riena. Currently no pull from the community to support it. Possible feature in a future release of Riena.
- JSR 330 (Dependency Injection) could be a good standard to be used by Riena's Injection framework which injects OSGi Services and Extensions. We havnt investigated the combination of the two.

## UI Usability



- Built on top of Eclipse RCP
- Use proven technologies from their like Views, Perspectives, Databinding
- Adds a different navigation model on top that is more driven by the requirements of endusers. (known by them from other enduser products)
- Ability to customize Look and Feel for SWT views for various use cases and customers.

# Schedule



- Release 1.2.0 milestones were started directly after the Galileo Release
- Currently on schedule
- There were 3 milestones and 2 release candidates since July 2009.



# Process: Community and Activities



- 11 committers currently, 7 contributors that are actively submitting patches.
- Growing interest in the community
- Feedback from people that are beyond the play-stage and are building real enterprise applications with Riena
- Open Source framework „Redview“ (non-Eclipse project) is based on Riena.
- RAP recently show interest and developed code to allow the use of Riena's ridgets in RAP applications.
- Riena committers are also contributing to other projects i.e. p2, core.net
- Growing number of bugs and requests are filed by non-committers.
- Open release planning and tracking processes
  - Bugzilla used to request and track all defects and enhancements
  - Bugzilla entries from committers and Riena users
- Coordination/cooperation with other Eclipse projects
  - dependency to Nebula exists (CompositeTable)

## Process: Community



- Evangelism and outreach into the community
  - Presentations on Eclipse Summit and Eclipse Con and Jax
- Riena is often consumed in pieces
  - first Remote Service, then Ridgets, then the whole UI with the alternative navigation model
- Ekke Gentz (now committer) is still very active (<http://www.ekkes-corner.org/>), blogs, Eclipse Demo Camps etc.
- Compeople is running multiple projects based on Riena.
- Riena is used by a number of people in production that we know of.

## *IP Issues*



- Compeople offered an initial code contribution which was submitted to IPZilla.
- Riena also uses some third party software that was submitted through IPZilla.
- Everything else was written by one of the committers himself or a patch was applied from a bugzilla request.
- IPLog is submitted and already approved



## *Future plan*

- Riena plans to release a 2.0. version of Riena together with Helios.
- As indicated under „API“ we plan to ease and refactor some of the API in that step to 2.0.