WindowBuilder Graduation & 1.0.0 Release Review

http://www.eclipse.org/windowbuilder
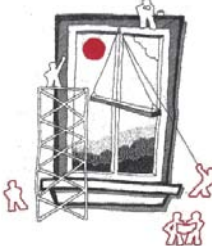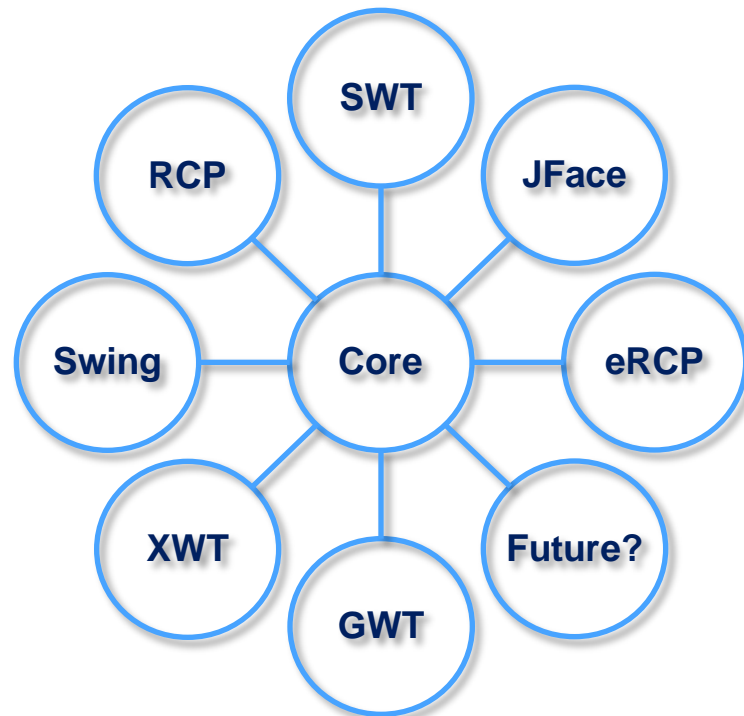
Planned Review Date: June 2011
Communication Channel: WindowBuilder Forum
Eric Clayberg (Project Lead)

History

WindowBuilder has a very long history spanning multiple technologies and companies

Smalltalk

- 1991   Original release for Smalltalk/V by Cooper & Peters
- 1993   VisualSmalltalk release by ObjectShare
- 1994   VisualAge Smalltalk release by ObjectShare
  - 1996   Briefly owned by ParcPlace-Digitalk
- 1997   VisualAge Smalltalk release by Instantiations

Eclipse/Java

- 2003   New Eclipse/Java version for SWT/RCP (SWT Designer)
- 2004   Swing support added (Swing Designer)
- 2006   Google Web Toolkit (GWT) support added (GWT Designer)
- 2009   Eclipse community award for Best Commercial Add-on
- 2010   Acquired by Google and released free to the world
- 2011   Contributed to Eclipse.org as new open-source project

Same Team

Overview

- Available now from http://www.eclipse.org/windowbuilder

- Composed of WindowBuilder Engine, SWT, eRCP, XWT & Swing Designer

- WindowBuilder Engine provides a rich API for creating UI designers

  - Java-based UI frameworks
    (e.g., Swing, SWT/RCP, eRCP, GWT)

  - XML-based UI frameworks
    (e.g., XWT, GWT UiBinder)

- Exemplary tool examples:

  - SWT Designer

  - Swing Designer

  - eRCP Designer

  - XWT Designer

- 3rd Party Tools

  - GWT Designer for
    Google Web Toolkit

SWT

RCP   JFace

Swing   Core   eRCP

XWT   Future?

GWT

Quotes

"WindowBuilder delivers the kind of GUI building productivity that we used to have before we converted to Java. WindowBuilder not only dramatically improves productivity for design and maintenance, but it also enables us to significantly improve the look-and-feel of our GUIs without costing days of coding. Until discovering WindowBuilder, I had forgotten just how much fun and easy it can be building Java GUIs."
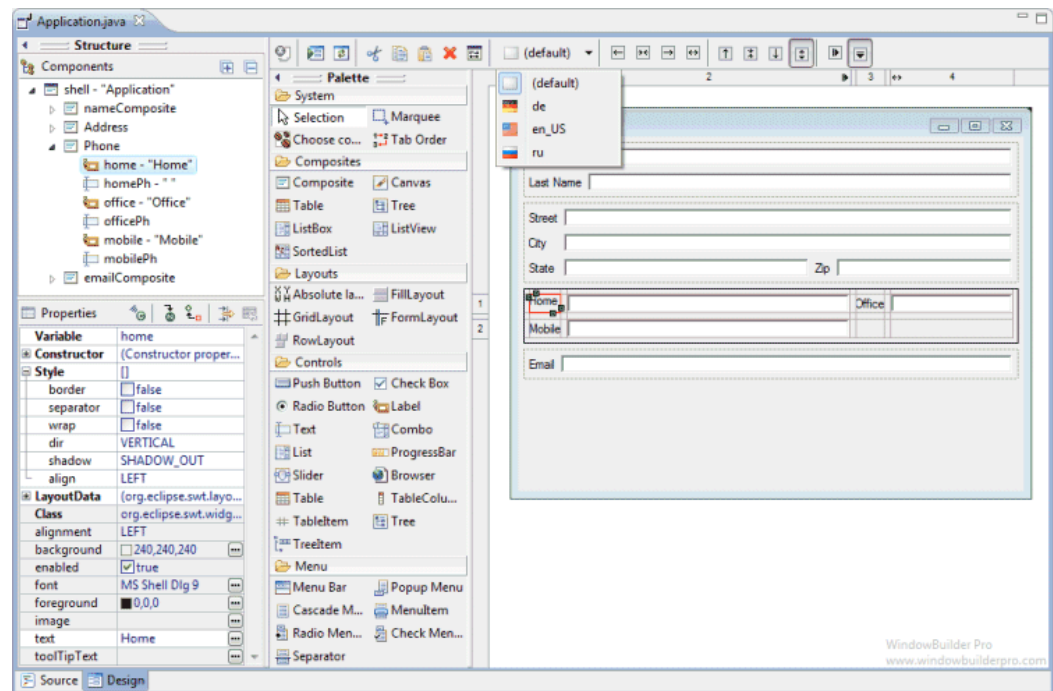
Sally Rich,
Senior Software Engineer at RSS Solutions Inc

"In 25 years of software development I have used a plethora of development tools. I can honestly say that WindowBuilder is head and shoulders above anything I have used for serious development. The features I particularly like include the bi-directional edit process, the native look and feel of cross platform GUIs and the manner in which component management is greatly simplified. It all adds up to allowing the developer to get on with the process of creating an application rather than worrying about the technology beneath it."

John Bond,
Developer

4

Features

WindowBuilder supports many state-of-the-art features

- WYSIWYG & Bi-directional Code Generation

- Powerful & Flexible Code Parser

- Single Representation

- Read & Write Any Style

- Micro Edits

- Internationalization

- Visual Inheritance

- UI Factories

- Morphing

- Graphical Menu Editing

- Widgets & Layout Managers

Non-Code Aspects

- User Documentation
  - 250+ pages of user documentation available as Eclipse Help or on-line
  - Customization API doc available for adding new components to the palette

- Tutorials:
  - Adding New Components
  - SWT/RCP/Swing Databinding

- All strings externalized to resource files and localization ready (English resource files only)

- Presentations
  - WindowBuilder 101 at EclipseCon 2011
  - WindowBuilder at Eclipse DemoCamps in Atlanta & Boston [upcoming]
  - GWT+UI Designer at Google I/O 2011

API

- The APIs in this release are Eclipse Quality

- As part of the contribution process, all code was initially marked internal and limited classes and methods were made API on a very conservative basis

- Additional API will be made public in the future and will be a major focus of the Juno development cycle

Architectural Issues

- The WindowBuilder Engine was designed to be very modular and offers a great many extension points

  - Pluggable support for different languages and parsers

  - 5 public exemplary tools built using this architecture so far supporting SWT/RCP, eRCP, XWT, Swing and GWT

  - Multiple private/OEM tools built using this architecture
    - UI design tool for unnamed cell phone manufacturer
    - UI designer embedded in Embarcadero's JBuilder
    - UI designer for popular phone OS at prototype phase

- Core branding elements are optional for easy customization by Eclipse members wishing to build custom tools

- Core engine offers an auto-discovery capability for loading new UI toolkit editors using p2 (makes it easy to load support for things like eRCP, XWT, Swing, etc. with just a few clicks)

Tool Usability

- WindowBuilder based tools widely used by >200K developers world wide since 2003
  - SWT Designer released in 2003
  - Swing Designer released in 2004
  - GWT Designer released in 2006

- GWT Designer embedded in Google Plug-in for Eclipse (GPE)

- Swing Designer embedded in Embarcadero's JBuilder

- Eclipse Magazin (Germany) Reader's Choice Award in 2006 for "Best Eclipse-Based Commercial Software Solution"

- Eclipse Community Award in 2009 for "Best Commercial Eclipse Development Tool"

End-of-Life

- The following features did not make it through the Eclipse IP process:
    - Support for SWT GroupLayout (based on Swing-based LGPL code)
    - Support for various LGPL Swing Look & Feels

- These features were moved to a new Eclipse Labs project called WindowBuilder-Extras and will be maintained by the current committer team

- In addition, the LGPL JSR 295 Swing Data Binding jar is no longer included with WindowBuilder and must be installed separately by the user

Bugzilla / FogBugz

- **Eclipse Bugzilla**
  - 17 Bugs – all closed
  - 1 open enhancement request

Status			
Severity	**NEW**	**CLOSED**	**Total**
major	.	4	4
normal	.	13	13
enhancement	1	.	1
Total	1	17	18

- **Google FogBugz during last six months since mid-December**
  - 604 Bugs & Inquiries – all closed
  - 36 Enhancement Requests – all closed
  - Users are being encouraged to report all bugs to Eclipse Bugzilla going forward
  - Google obligated to provide commercial support to WindowBuilder users until August 2011

- **Historic WindowBuilder FogBugz data (Instantiations & Google)**
  - 10173 Bugs & Inquiries – all closed
  - 347 Enhancement Requests – all closed

- **WindowBuilder includes an automated bug reporting tool to make it easy for users to report problems/exceptions**
  - Historically pointed at Instantiations/Google FogBugz
  - Now pointed at Eclipse Bugzilla

- **Committer team is committed to responding to all bug reports, inquiries and enhancement requests very quickly and has an excellent track record in this regard throughout the eight-year history of the project**

Standards

- WindowBuilder requires Java 5 or later

- Supports SWT, RCP, eRCP, XWT and Swing

- Supports Swing JSR 295 and 296

UI Usability

- WindowBuilder conforms to the Eclipse user interface standards

- WindowBuilder was certified as "Ready for Rational" (which requires conformance to the Eclipse user interface standards)

- WindowBuilder provides the following accessibility features:
  - Keyboard equivalents for most actions. For example, widgets may be moved in any free form layout using the arrow keys. Likewise, the palette and property pane can be traversed using the arrow keys.
  - Visual focus indicators are provided for selected widgets in the component tree and design view as well as for the currently selected property.
  - Color preferences are provided to control feedback for most layout managers.
  - Color is used as an enhancement to various feedbacks in the design view.
  - System high contrast settings are respected
  - System fonts, colors and sizes are inherited
  - Blinking text or objects are avoided
  - Documentation is provided in an accessible format
  - No audio alerts are used

Schedule

- Google announces WindowBuilder contribution on 12/15/2010

- First Eclipse-branded build available on 1/16/2011 from code.google.com

- Initial code contribution to Eclipse.org on 3/9/2011

- First Eclipse.org build available on 4/7/2011

- New builds available weekly

- 1.0.0 release scheduled for 6/22/2011

Communities

- Bugzilla is in active use by users and contributors

- New WindowBuilder forum very active
  - 43 threads and 141 posts
  - Lots of questions about how to extend the tool

- Old Instantiations WindowBuilder forum also very active
  - 69 threads and 309 posts since 12/15/2010
  - 1415 threads and 5594 posts since July 2003
  - Users encouraged to post new questions to Eclipse WindowBuilder forum
  - Scheduled to be closed on 5/27/2011

- Presentations at EclipseCon and Google I/O

- Committers demoing at Eclipse DemoCamps in June (Boston & Atlanta)

IP Issues

- The Eclipse IP policies and procedures have been followed

- IP Log available at:
  http://www.eclipse.org/projects/ip_log.php?projectid=tools.windowbuilder

- Current Eclipse SUA in place for all features

Project Plan

- The project plan is available here:
  http://www.eclipse.org/projects/project-plan.php?projectid=tools.windowbuilder

- The basic goal for this first release is simple: make the existing WindowBuilder tool available to the entire Eclipse ecosystem
  - Move existing code base to Eclipse.org namespace
  - Establish core set of public APIs
  - Externalize all strings to resource files for localization
  - Further modularize the code base to minimize its required footprint
  - Update build process to sign and compact jars
  - Fix any and all reported bugs

Graduation – Code Base

- WindowBuilder is a very stable, mature tool with >200K users

- Eclipse Magazin (Germany) Reader's Choice Award in 2006 for "Best Eclipse-Based Commercial Software Solution"

- Eclipse Community Award in 2009 for "Best Commercial Eclipse Development Tool"

- The WindowBuilder Engine was designed to be very modular and offers a great many extension points
  - Pluggable support for different languages and parsers
  - 5 public exemplary tools built using this architecture so far supporting SWT/RCP, eRCP, XWT, Swing and GWT
  - Multiple private/OEM tools built using this architecture
    - UI design tool for unnamed cell phone manufacturer
    - UI designer embedded in Embarcadero's Jbuilder
    - UI designer for popular phone OS at prototype phase

Graduation - Communities

- Very active user community at Instantiations/Google in the process of migrating to Eclipse.org

- Old and new WindowBuilder forums very active (old forum shutting down soon and all traffic actively being directed to Eclipse.org).

- Committer team represents four different companies (heavily weighted to Google at the moment)

- WindowBuilder framework in use by:
  - Google – GWT Designer
  - Embarcadero – UI Builder in JBuilder
  - Unnamed Cell Phone Manufacturer – UI design tool for proprietary UI toolkit

Graduation - Openness

- Bugzilla is in active use by users and contributors

- Old and new WindowBuilder forums very active (old forum shutting down soon and all traffic being directed to Eclipse.org)
  - Many discussions about how to extend the tool
  - Many discussions about bugs and possible enhancements

- Committer team is committed to being open and transparent but acknowledges that we are still learning the Eclipse way and have much more to do in this regard and much to learn from our mentors

Graduation - Logistics

- The project is committed to following the Eclipse Development Process and will be improving on this going forward

- The project is adhering to the Eclipse IP Policy

- Committers are participating in the larger Eclipse community:
  - WindowBuilder 101 at EclipseCon 2011
  - WindowBuilder at Eclipse DemoCamps in Atlanta & Boston [upcoming]
  - GWT+UI Designer at Google I/O 2011

- The project is a credit to Eclipse and of great benefit to the Eclipse community

Graduation - Conclusion

- WindowBuilder wishes to exit incubation to release 1.0.0

- WindowBuilder wishes to be included in the following Indigo packages:
  - Eclipse IDE for Java Developers – see Bugzilla Bug 343869
  - Eclipse for RCP and RAP Developers – See Bugzilla Bug 343871

- We are requesting an exception to the normal Eclipse time-scale based on the following:
  - WindowBuilder is a very mature project
  - The team has worked on open source projects in the past
  - We will continue to work closely with our mentors going forward toward Juno
  - We are confident that we will be able to better follow the normal milestone flow for Juno