



Building Amazing UIs with WindowBuilder

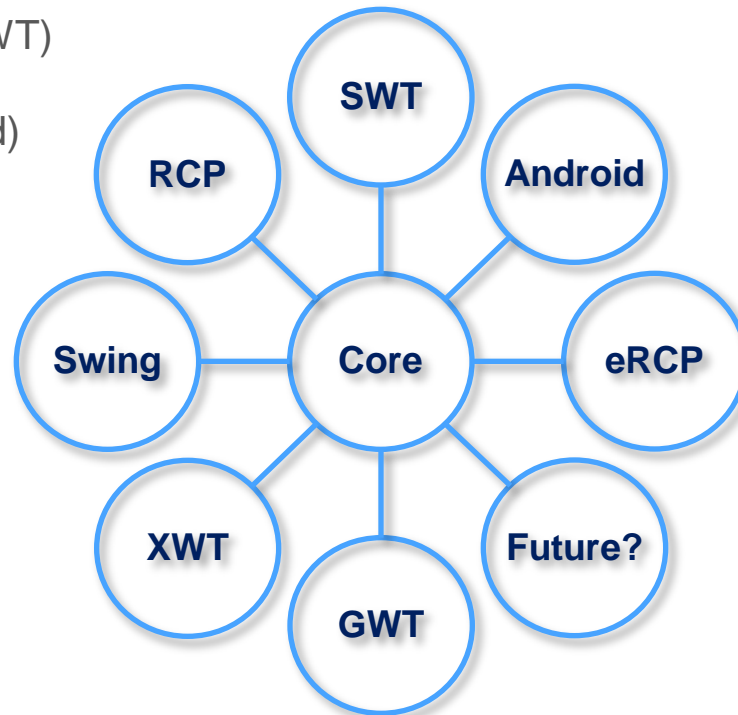
Eclipse Day at the Googleplex 2012
December 12, 2012

Eric Clayberg
Software Engineering Manager
Google, Inc.

WindowBuilder has a very long history spanning multiple technologies and companies

- 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
 - 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;
Part of Indigo & Juno release trains (Eclipse 3.7, 3.8 & 4.2)
- 
- 
- Smalltalk
- Eclipse/Java
- Same Team

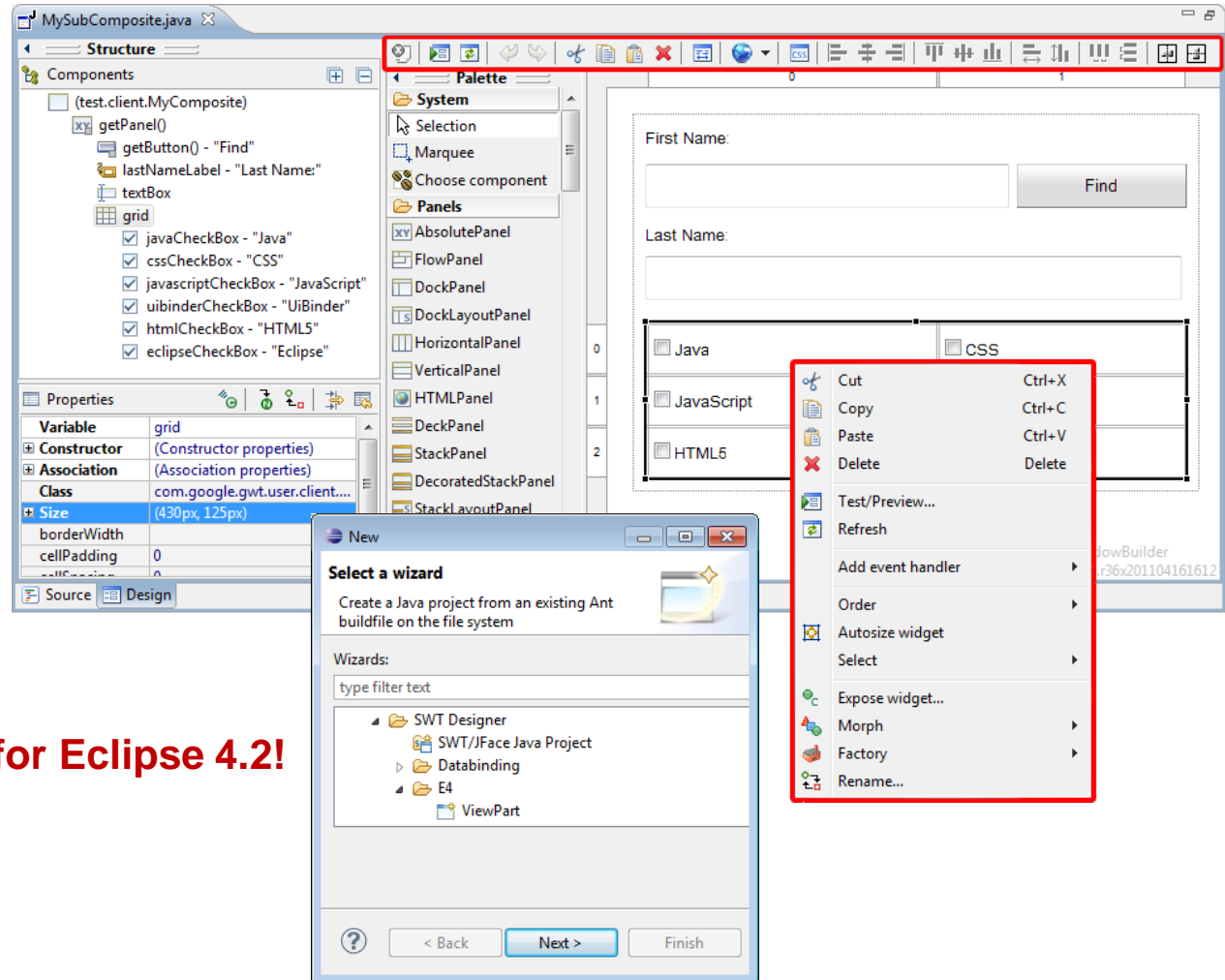
- Available now from <http://www.eclipse.org/windowbuilder>
- Composed of WindowBuilder Engine, SWT, eRCP, XWT & Swing Designer
- WindowBuilder Engine provides a rich API for creating UI designers
 - Very modular with dozens of extension points
 - Pluggable support for different languages and parsers
 - Java-based UI frameworks (e.g., Swing, SWT/RCP, eRCP, GWT)
 - XML-based UI frameworks (e.g., XWT, GWT UiBinder, Android)
- Exemplary tool examples:
 - SWT Designer
 - Swing Designer
 - eRCP Designer
 - XWT Designer
- 3rd Party Tools
 - JBuilder Swing Designer
 - GWT Designer
 - Android Designer



User Interface

WindowBuilder is composed of the following major components

- Source View
- Design View
- Component Tree
- Property Pane
- Palette
- Wizards
- Toolbars & Context Menus



New e4 ViewPart wizard for Eclipse 4.2!

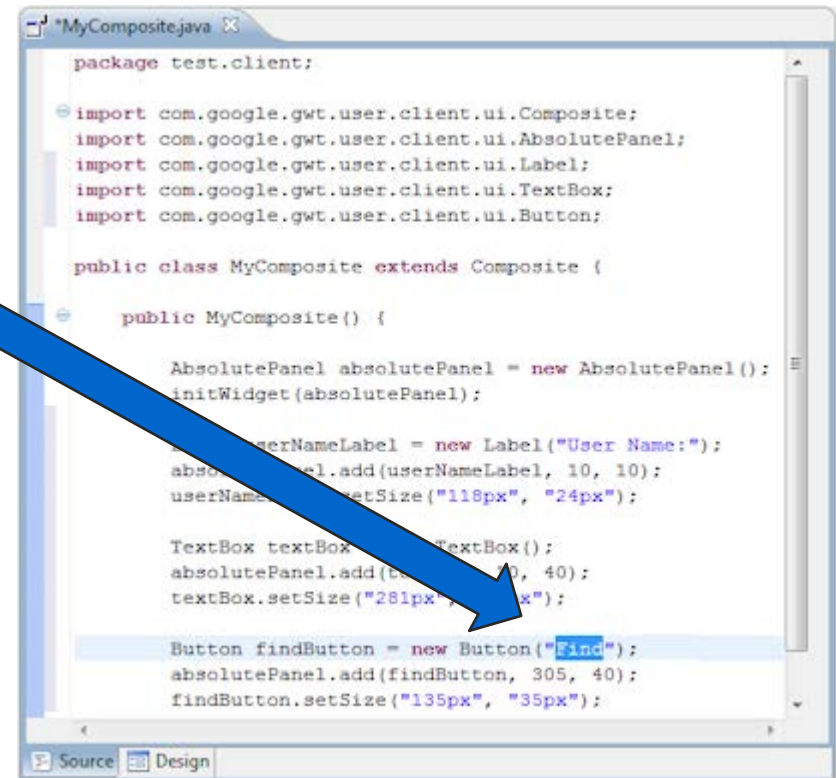
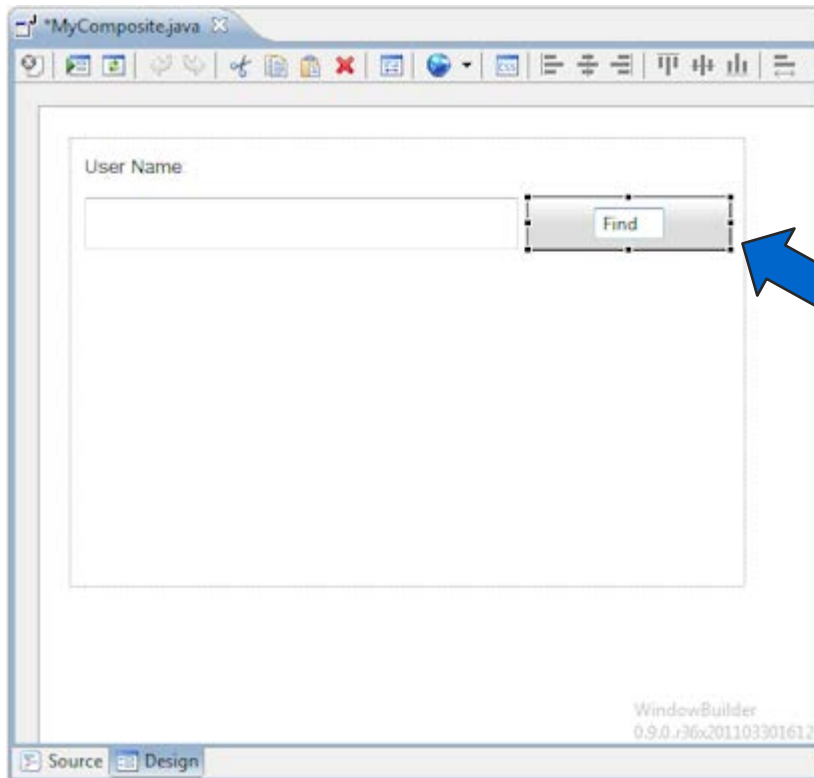
WindowBuilder supports many state-of-the-art features

- WYSIWYG & Bi-directional Code Generation
- Powerful & Flexible Code Parser
- Read & Write Any Format or Style
- Internationalization
- Visual Inheritance
- UI Factories
- Morphing
- Widgets & Layout Managers
- Graphical Menu Editing
- Data Binding



WYSIWYG & Bi-directional Code Generation

- WYSIWYG editing in design view
- Bi-directional Code Generation
- Micro edits result in smallest possible code change



Powerful & Flexible Code Parser



- Can parse its own code and code written by hand
- No protected code blocks
- Understands data flow
- Ignores & preserves non-UI code
- Refactoring friendly and resilient to hand-made changes
- One-to-one relationship between UI and Java/XML code
- No intermediate metadata file to get lost or out of sync

```
MyComposite.java X
package test.client;

import com.google.gwt.user.client.ui.Composite;
import com.google.gwt.user.client.ui.AbsolutePanel;
import com.google.gwt.user.client.ui.Label;
import com.google.gwt.user.client.ui.TextBox;
import com.google.gwt.user.client.ui.Button;

public class MyComposite extends Composite {

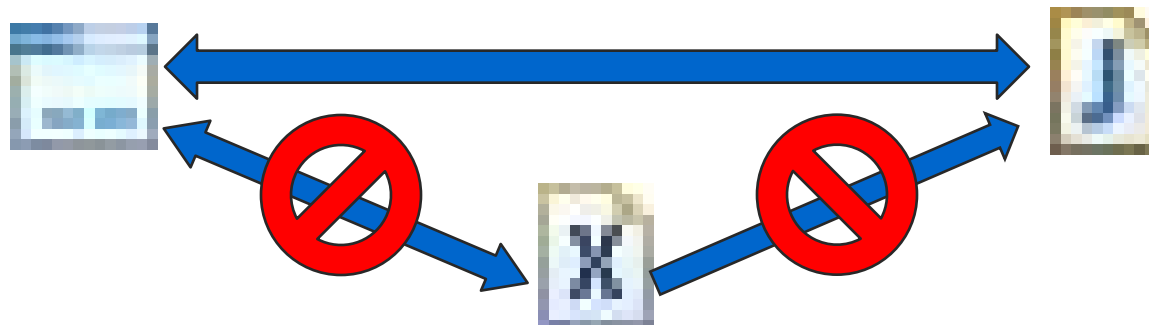
    public MyComposite() {

        AbsolutePanel absolutePanel = new AbsolutePanel();
        initWidget(absolutePanel);

        Label userNameLabel = new Label("User Name:");
        absolutePanel.add(userNameLabel, 10, 10);
        userNameLabel.setSize("118px", "24px");

        TextBox textBox = new TextBox();
        absolutePanel.add(textBox, 10, 40);
        textBox.setSize("281px", "27px");

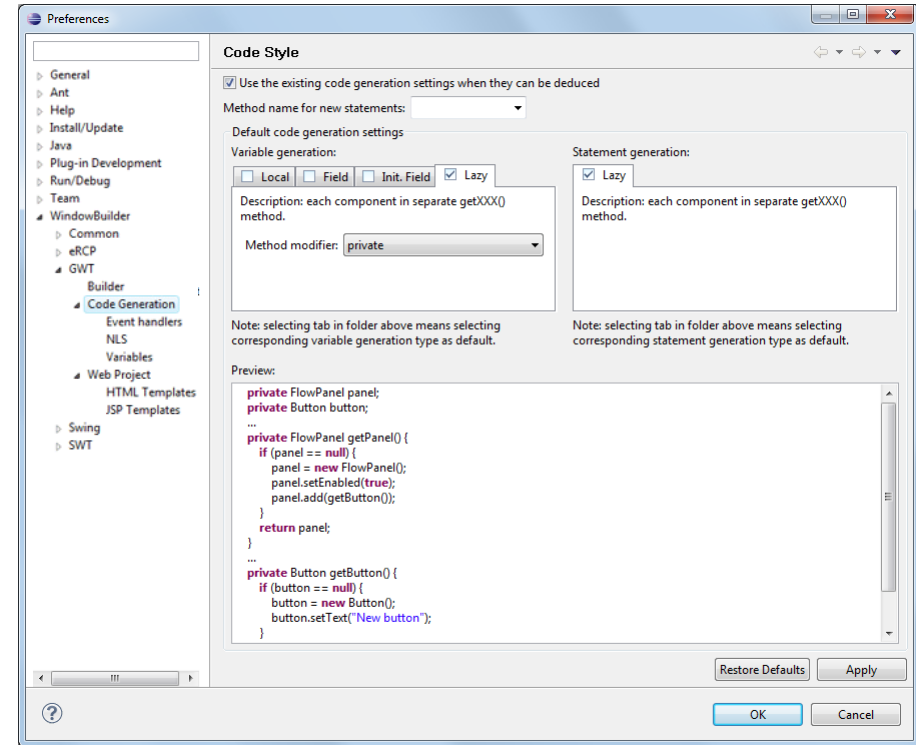
        Button findButton = new Button("Find");
        absolutePanel.add(findButton, 305, 40);
        findButton.setSize("135px", "35px");
    }
}
```



Read & Write Any Format or Style

Window > Preferences > WindowBuilder > GWT | Swing | SWT > Code Generation

- Local variables vs. Fields
- Flat vs. Block
- Initialized fields
- Lazy declaration



```
...  
FlowPanel panel = new FlowPanel();  
panel.setEnabled(true);  
  
Button button = new Button();  
button.setText("Add customer...");  
panel.add(button);  
...
```

```
private FlowPanel panel;  
private Button button;  
...  
private FlowPanel getPanel() {  
    if (panel == null) {  
        panel = new FlowPanel();  
        panel.setEnabled(true);  
        panel.add(getButton());  
    }  
    return panel;  
}  
...  
private Button getButton() {  
    if (button == null) {  
        button = new Button();  
        button.setText("New button");  
    }  
    return button;  
}
```

```
{  
    FlowPanel panel = new FlowPanel();  
    panel.setEnabled(true);  
    {  
        Button button = new Button();  
        button.setText("Add customer...");  
        panel.add(button);  
    }  
    ...  
}
```

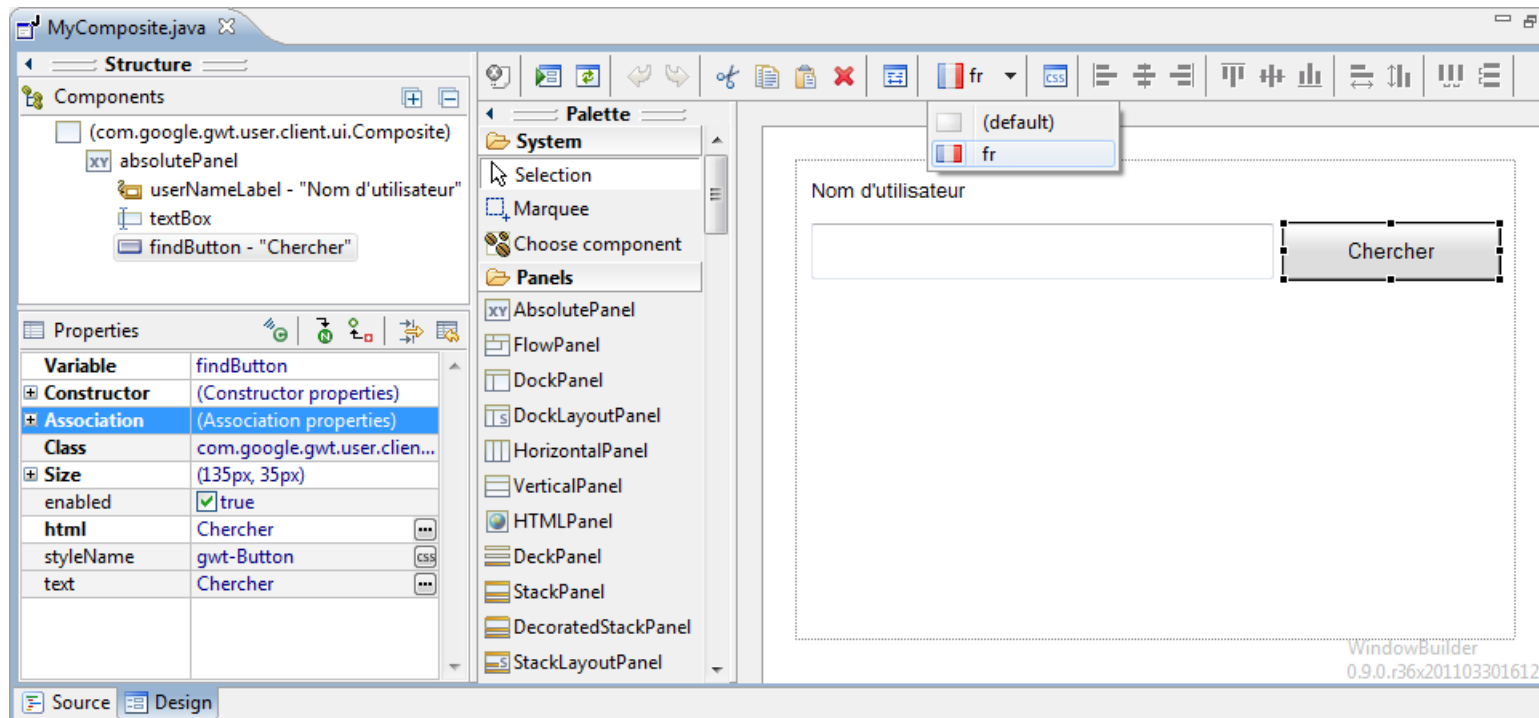
```
private final FlowPanel panel = new FlowPanel();  
private final Button button = new Button();  
...  
panel.setEnabled(true);  
  
panel.add(button);  
button.setText("Add customer...");  
...
```

```
private FlowPanel panel;  
private Button button;  
...  
panel = new FlowPanel();  
panel.setEnabled(true);  
  
button = new Button();  
button.setText("Add customer...");  
panel.add(button);  
...
```


Internationalization



Offers easy-to-use Internationalization and Localization tools



```
AppConstants.properties
#GWT variable: CONSTANTS
findButton_html=Search
userNameLabel_text=User Name:
```

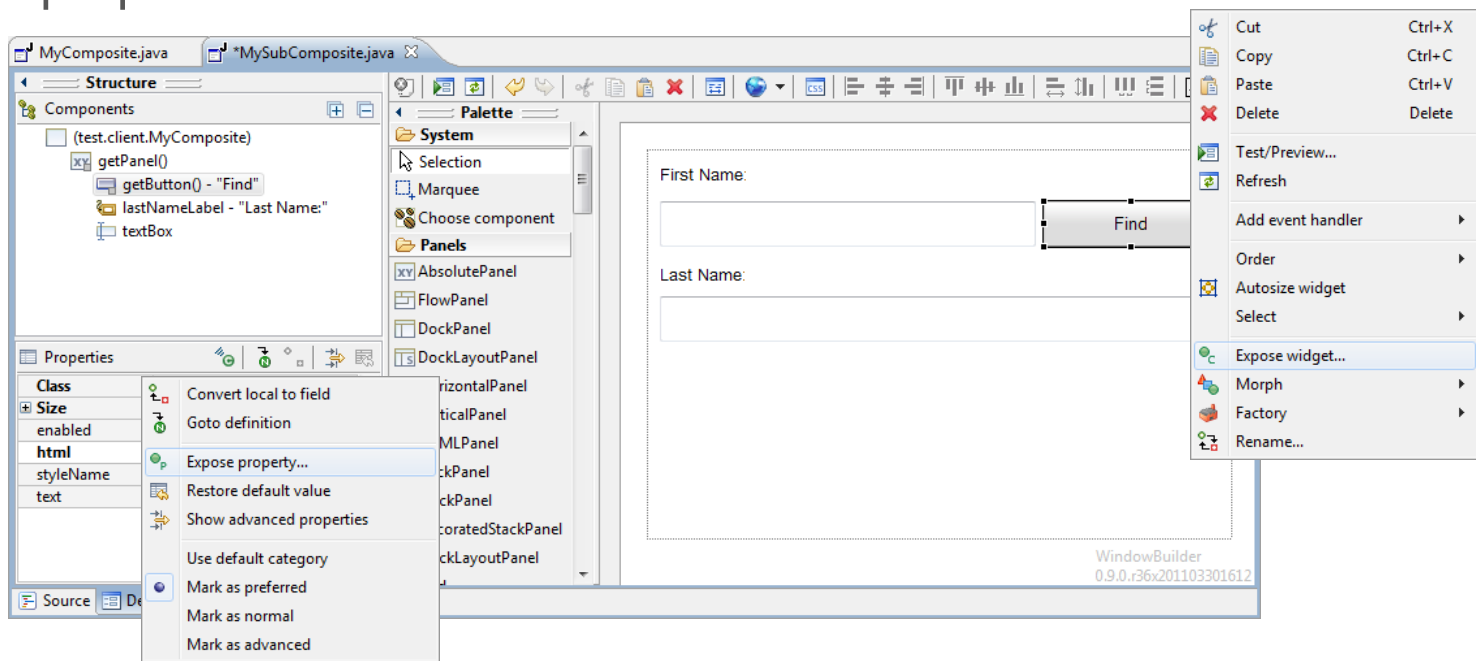
```
AppConstants_fr.properties
#GWT variable: CONSTANTS
findButton_html=Chercher
userNameLabel_text=Nom d'utilisateur:
```

Visual Inheritance

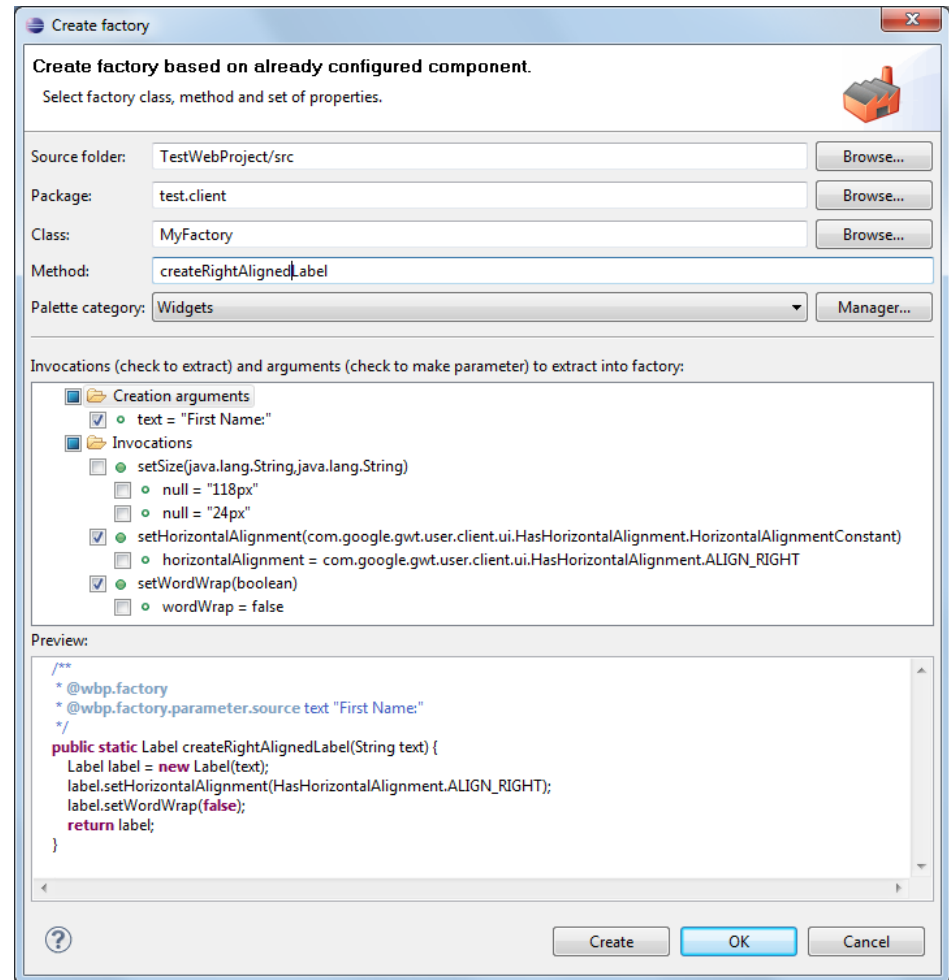
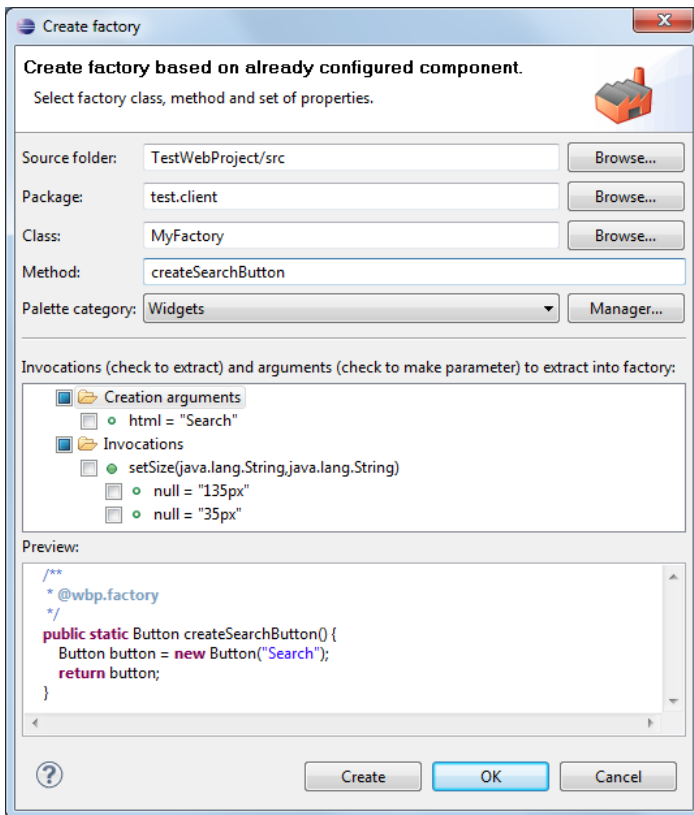
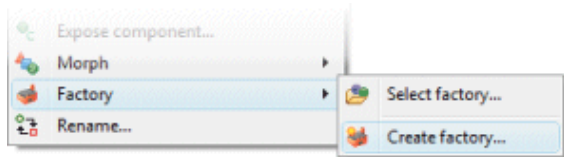


Provides visual inheritance so that code features can be easily inherited from a parent – child hierarchy

- Easily expose fields and properties
- Add components & event handlers to inherited fields
- Change public properties of inherited components
- Change properties of inherited fields

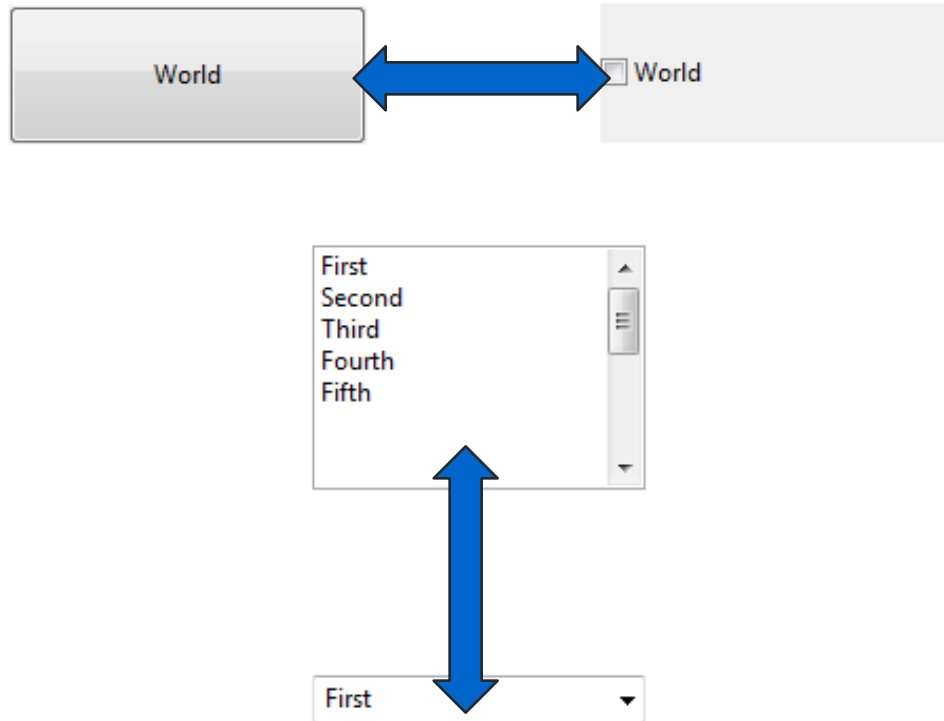


Support for UI Factories and reusable customized GUI elements



Morphing

Provides a Morphing tool to easily change one widget type into another

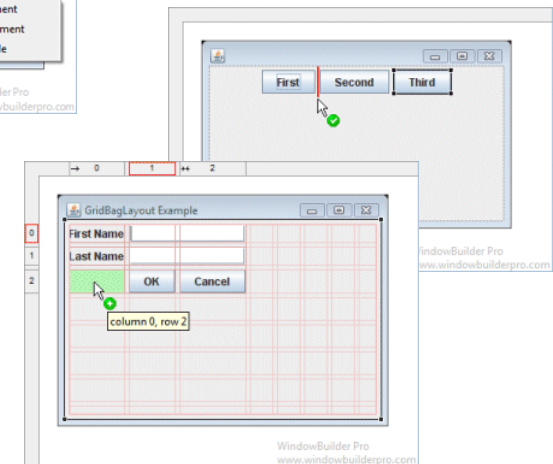
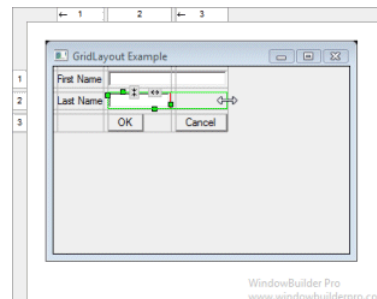
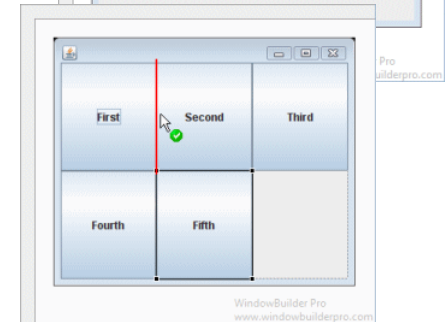
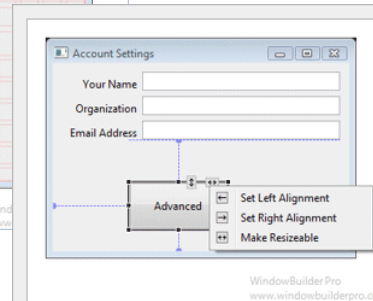
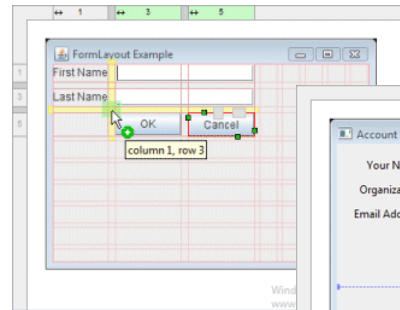
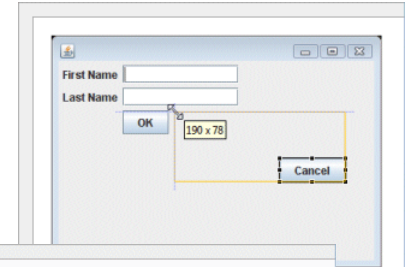
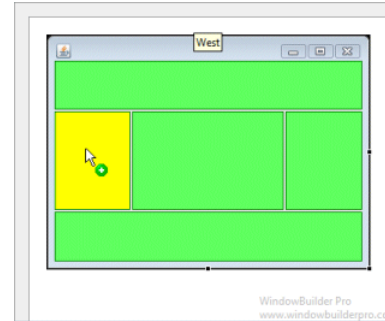


Widgets & Layout Managers



Fully supports all standard widgets and layout managers as well as select third-party widgets and layout managers

- Composites
 - Composite
 - Canvas
 - Table
 - Tree
 - ListBox
 - ListView
 - SortedList
- Layouts
 - Absolute layout
 - FillLayout
 - GridLayout
 - FormLayout
 - RowLayout
 - StackLayout
 - GroupLayout
 - FlowLayout
 - BoxLayout
 - BorderLayout
- Controls
 - Push Button
 - Check Box
 - Radio Button
 - Label
 - Text
 - Combo
 - List
 - ProgressBar
 - Slider
 - Browser
 - Table
 - TableColumn
 - TableItem
 - Tree
 - Treeltem

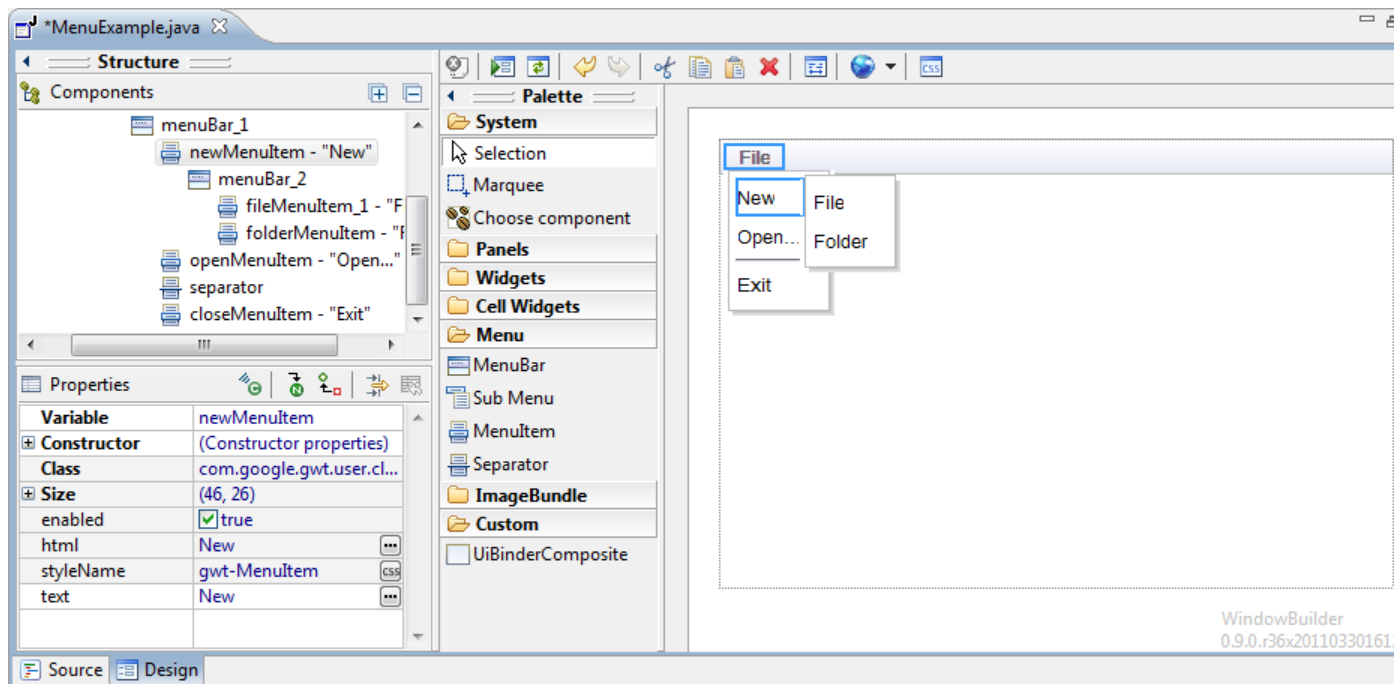
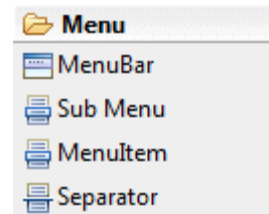


Graphical Menu Editing



Supports WYSIWYG Graphical Menu Editing

- Graphical edit menubars and menuitems
- Use drag/drop to rearrange menus
- Direct edit menu labels

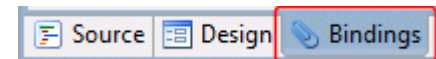


WindowBuilder
0.9.0.r36x201103301612

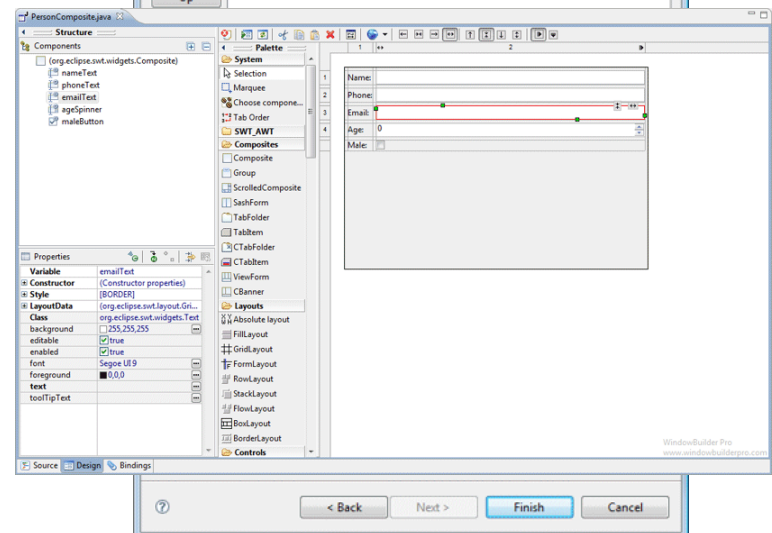
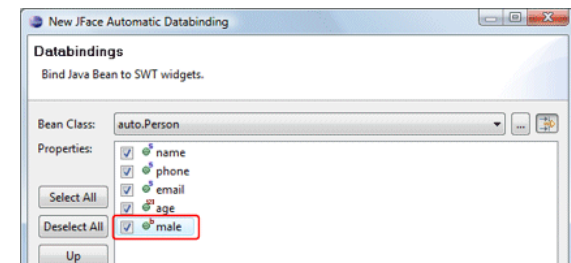
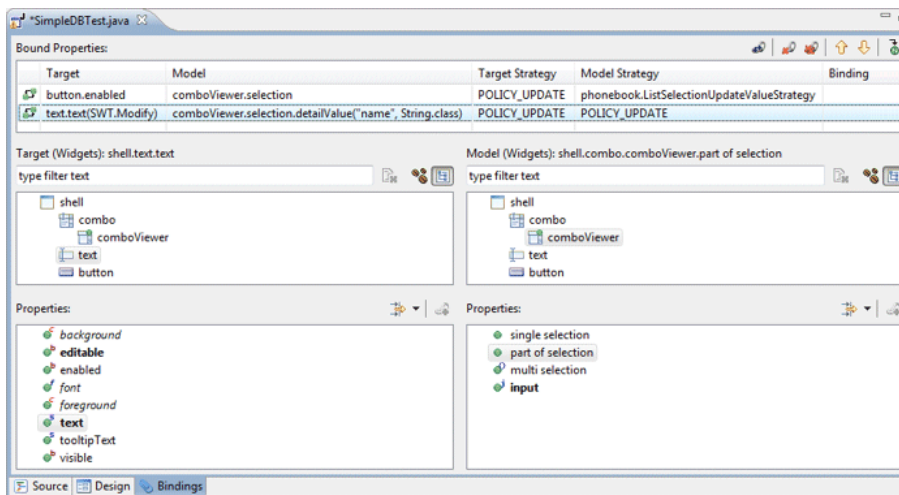
Data Binding



Supports binding UI elements and model elements



- No need to write and register listeners
- Common infrastructure for validation and conversion
- Connect data sources to widgets such as text fields, combos, tables and trees, for viewing and editing
- Manual and automatic tools provided



Gallery – SWT Designer



<http://eclipse.org/windowbuilder/>

The screenshot displays the Eclipse SWT Designer interface for a Java Swing window titled "Message". The interface is divided into several panes:

- Structure:** Shows a tree view of the component hierarchy. The selected component is a "link" widget with the text "This RCP Application".
- Properties:** A table showing the properties of the selected "link" widget.
- Palette:** A collection of UI controls and layouts available for design.
- Design View:** A visual representation of the message dialog with a white background and a grey border. The dialog contains a header with fields for "Subject", "From", and "Date", and a main text area.

Variable	link
Constructor	(Constructor properties)
LayoutData	(org.eclipse.swt.layout.Grid...
Class	org.eclipse.swt.widgets.Link
bindings	[]
Extension	(Extension properties)
name	Message
icon	icons/sample2.gif
category	
background	240,240,240
enabled	<input checked="" type="checkbox"/> true
font	Segoe UI 9
foreground	0,0,0
text	<a>nicole@mail.org
toolTipText	
touchEnabled	<input type="checkbox"/> false

The message dialog in the design view contains the following text:

Subject: This is a message about the cool Eclipse RCP!
From: nicole@mail.org
Date: 10:34 am

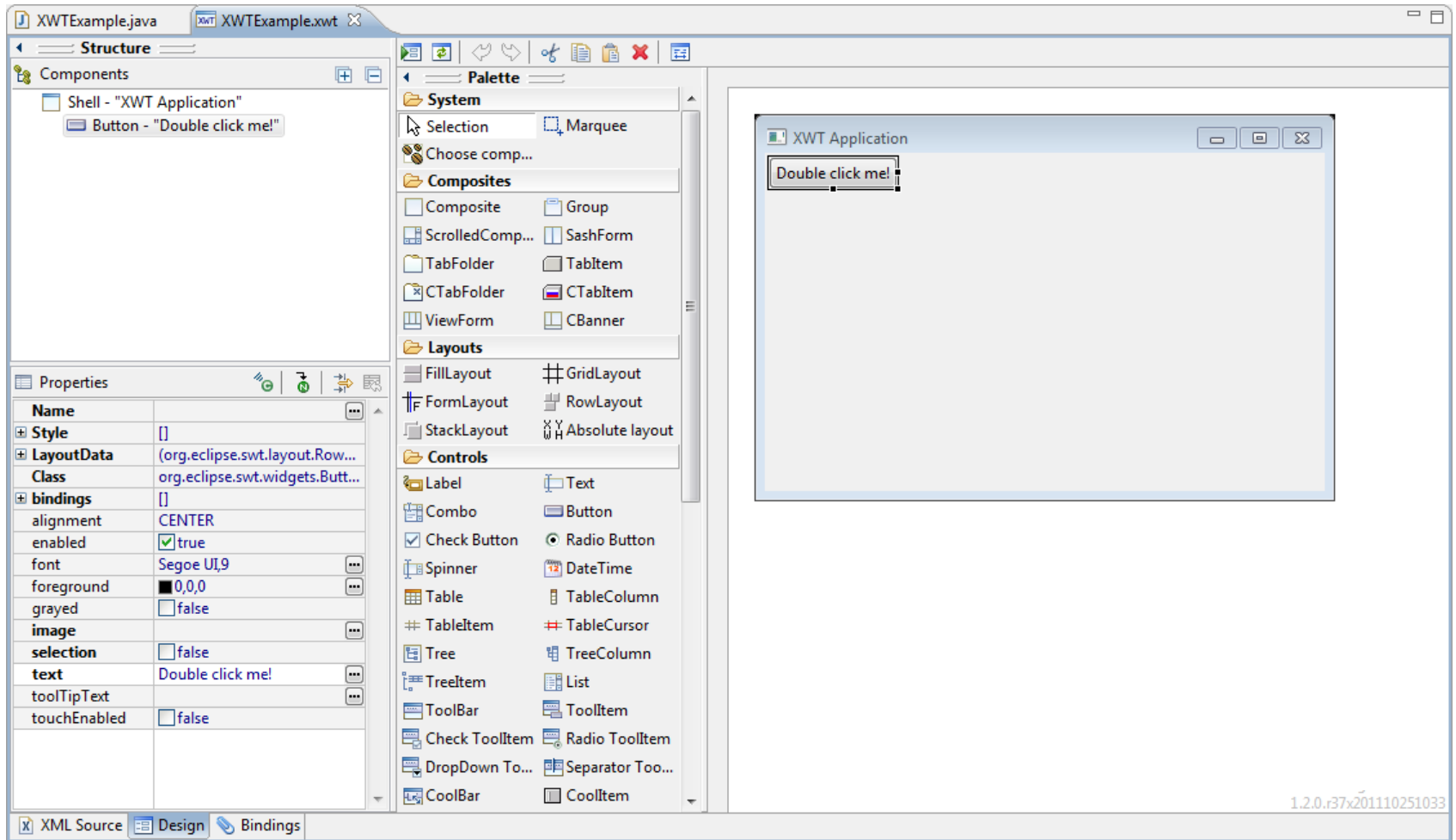
This RCP Application was generated from the PDE Plug-in Project wizard. This sample shows how to:

- add a top-level menu and toolbar with actions
- add keybindings to actions
- create views that can't be closed and multiple instances of the same view
- perspectives with placeholders for new views
- use the default about dialog
- create a product definition

1.2.0.r37x201110251033

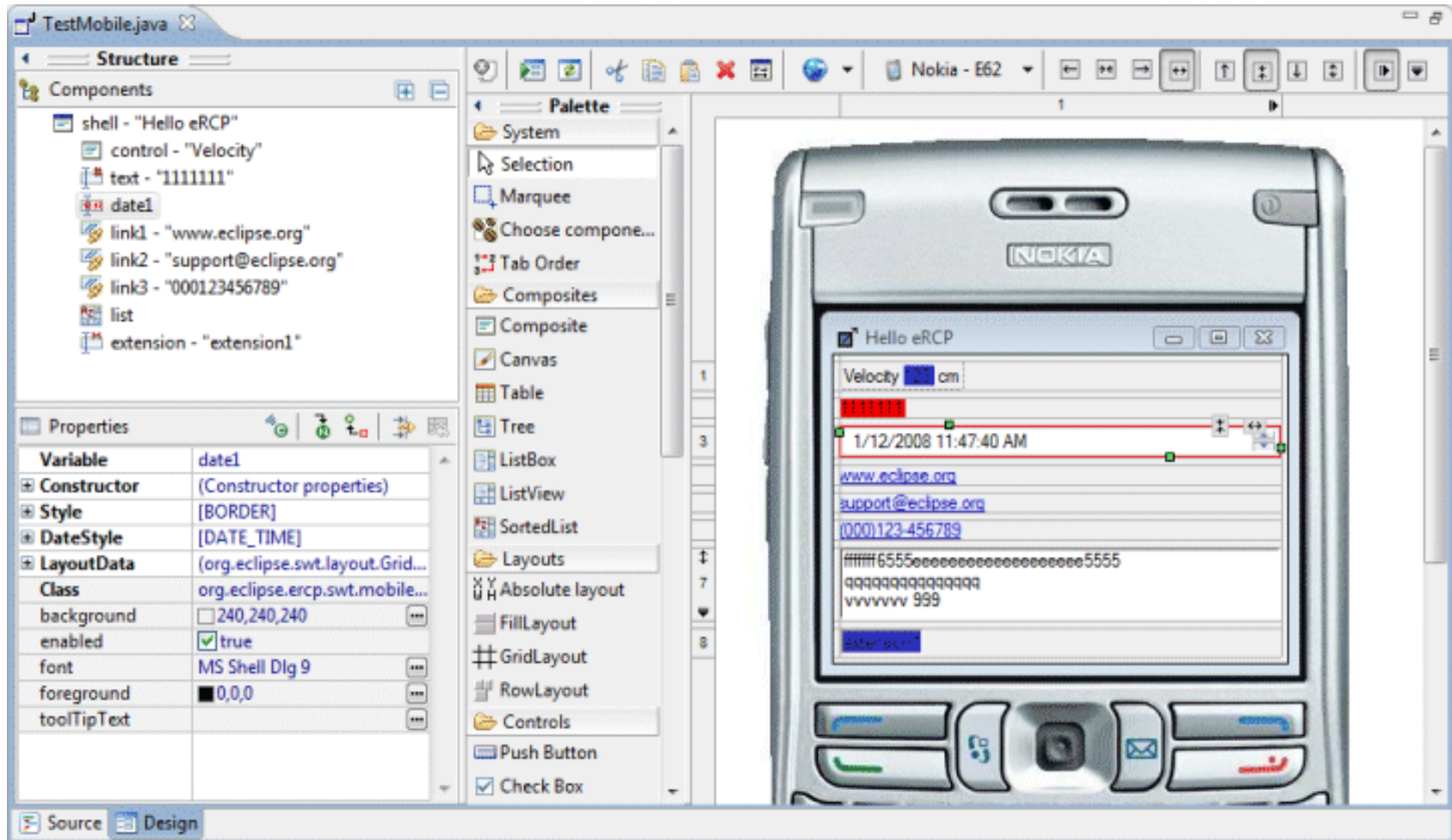
Gallery – XWT Designer

<http://eclipse.org/windowbuilder/>



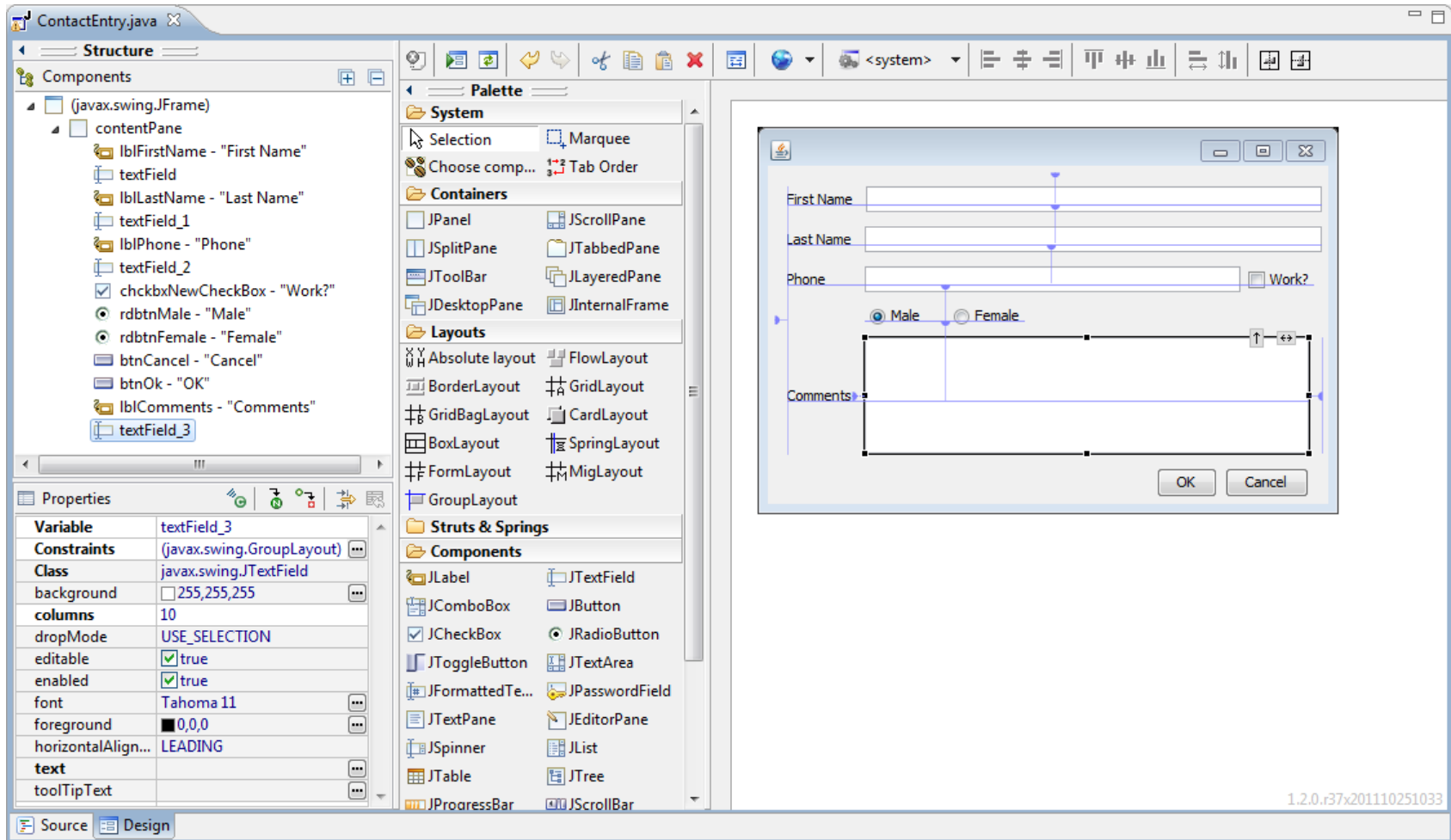
Gallery – eRCP Designer

<http://eclipse.org/windowbuilder/>



Gallery – Swing Designer

<http://eclipse.org/windowbuilder/>

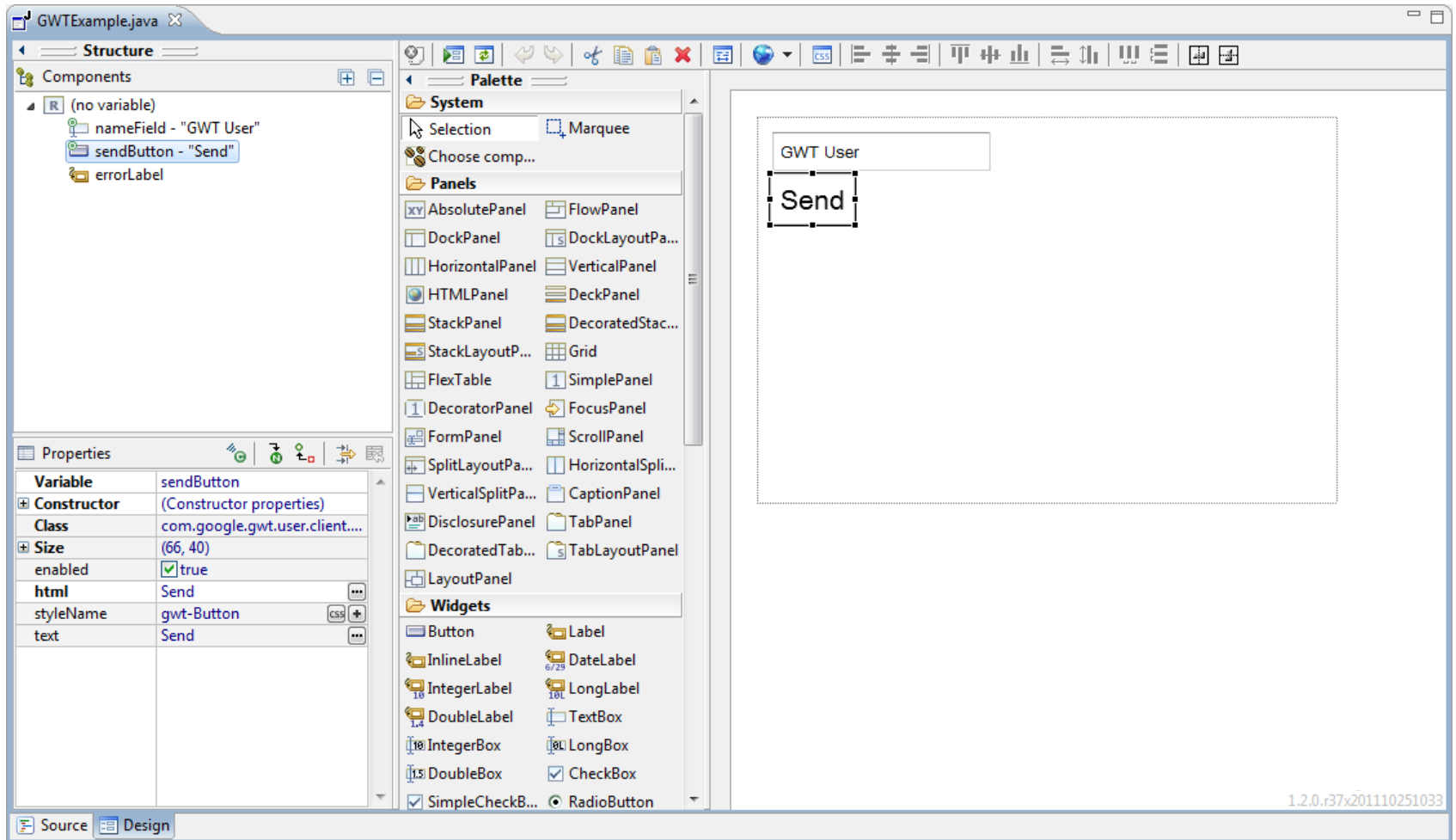


1.2.0.r37x201110251033

Gallery – GWT Designer



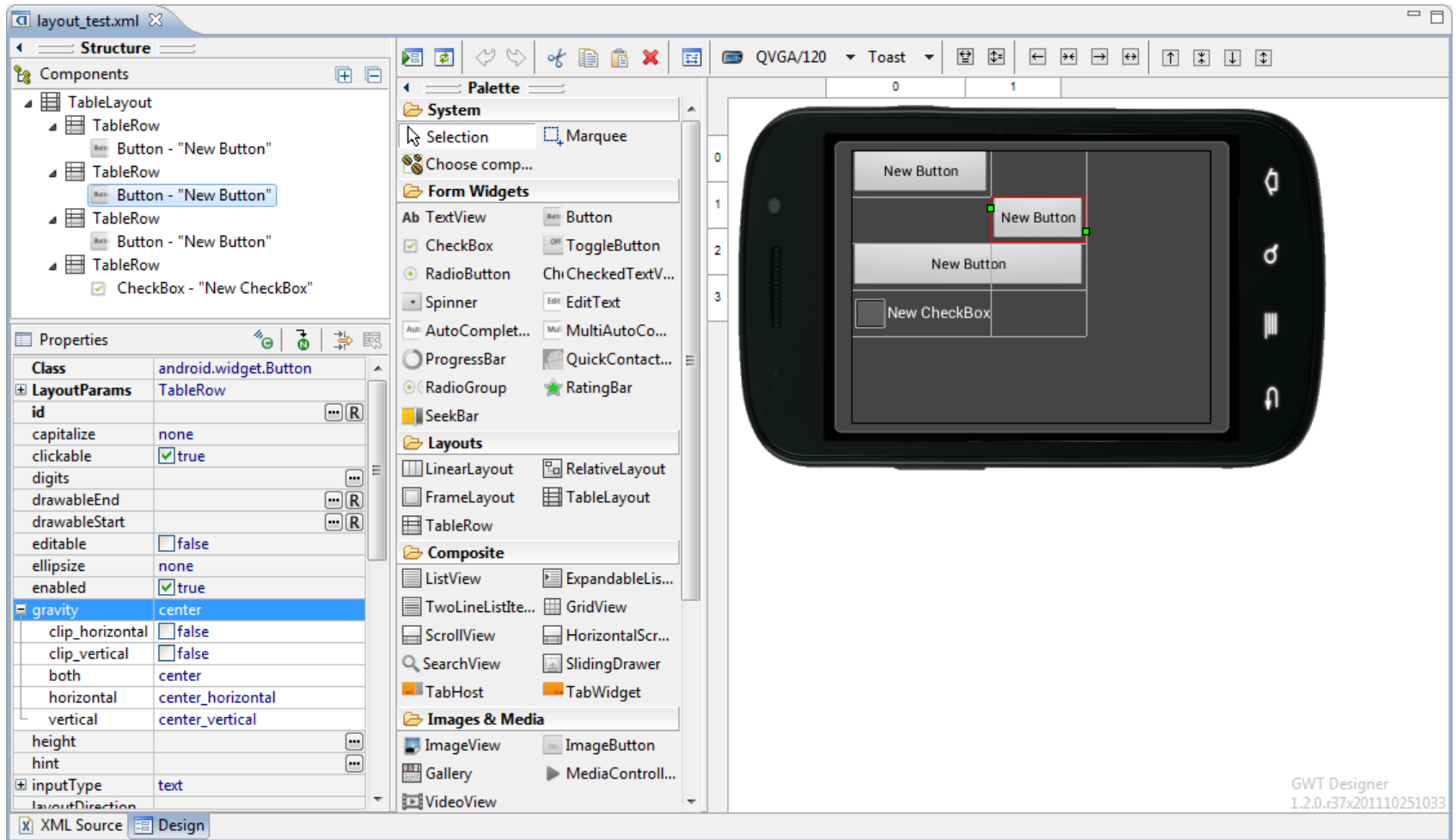
<http://code.google.com/p/gwt-designer/>



Gallery – Android Designer



<http://code.google.com/a/eclipselabs.org/p/android-designer>



Get More Info



Where to get it:

<http://eclipse.org/windowbuilder/download.php>

<https://developers.google.com/java-dev-tools/download>

<https://developers.google.com/web-toolkit/tools/download-gwt-designer>

<http://code.google.com/a/eclipseorg/p/android-designer/downloads>

Documentation:

<https://developers.google.com/java-dev-tools/wbpro/>

Issue tracker, source:

<https://bugs.eclipse.org/bugs/> → Tools > WindowBuilder

<http://dev.eclipse.org/svnroot/tools/org.eclipse.windowbuilder>

<https://svn.codespot.com/a/eclipseorg/windowbuilder-extras/trunk>

Forum:

http://eclipse.org/forums/index.php?t=thread&frm_id=214

