

WindowBuilder



Guillermo Esquivel González

Nicolás Hernández González

Programación de Aplicaciones Interactivas



Index

What will we talk about?

- What is WindowBuilder
- How to install WindowBuilder
- Distribution of the window
- Short introduction for a quick start
- Features
- Wizards
- Examples



What is WindowBuilder?

- Java GUI designer
- Makes easy creating complicated windows
- Visual Designer and Java Code
- Drag-and-Drop
- Add event handlers



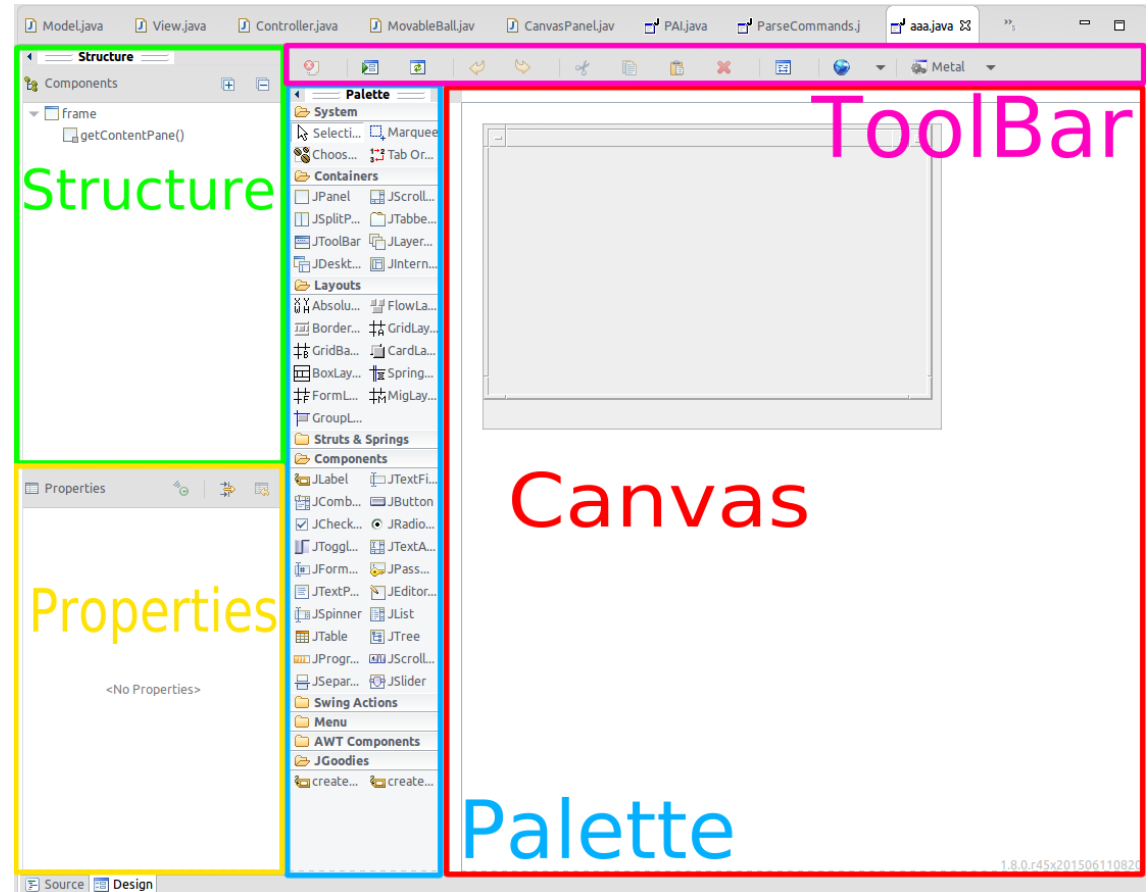
Installation

- In Eclipse, click **Help > Install New Software...**
- Get the proper link from <http://www.eclipse.org/windowbuilder/download.php>
- **In the Available Software** dialog, enter the product's update site location and select all the items to install. Click **Next** to continue.
- Click **Next** to confirm installation.
- Accept the license agreement. Click **Finish**.
- Restart Eclipse.



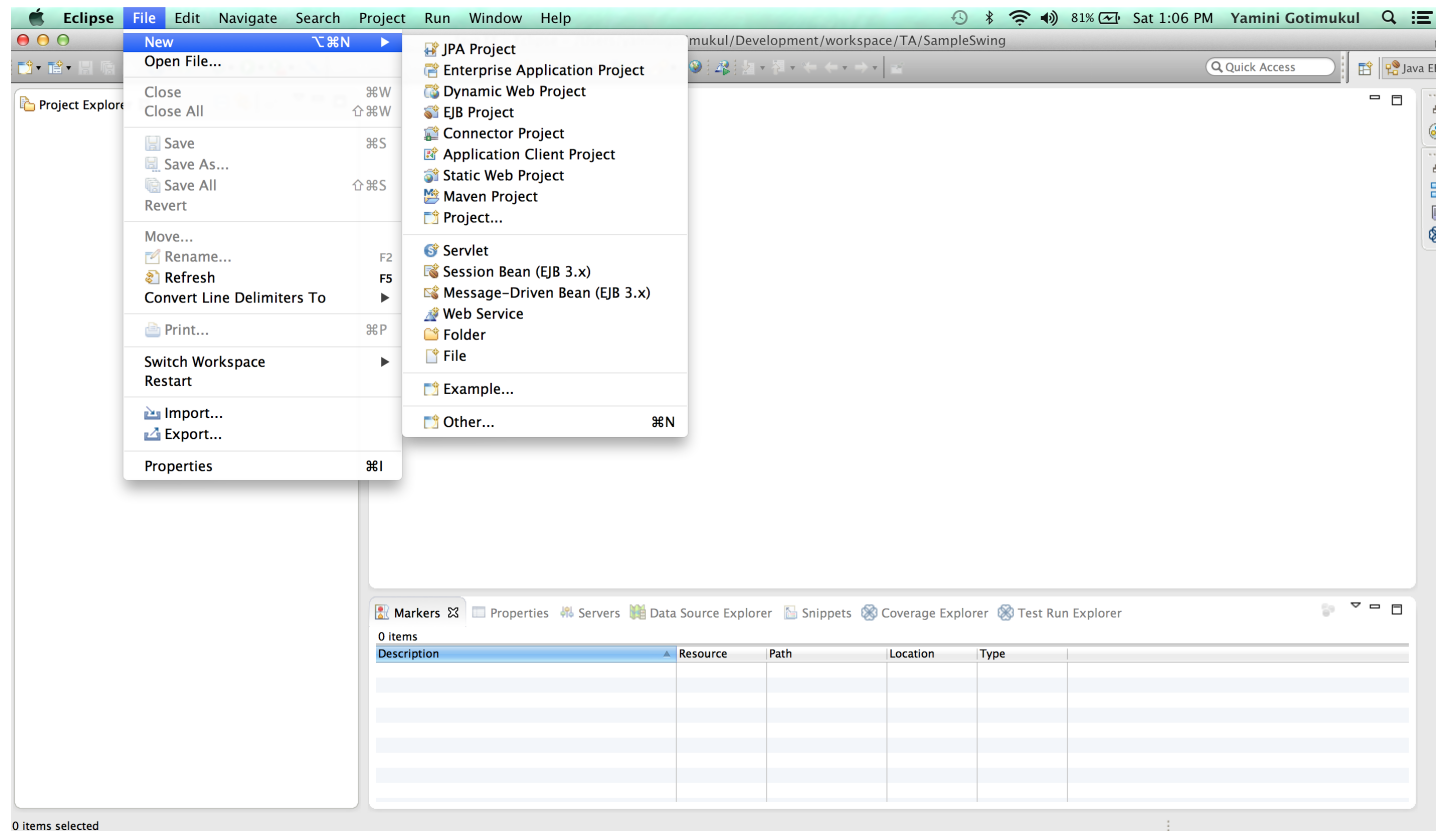
Distribution of the window

- Design View
- Source View
- Structure View
 - Component Tree
 - Property Pane
- Palette
- Toolbar
- Context Menu



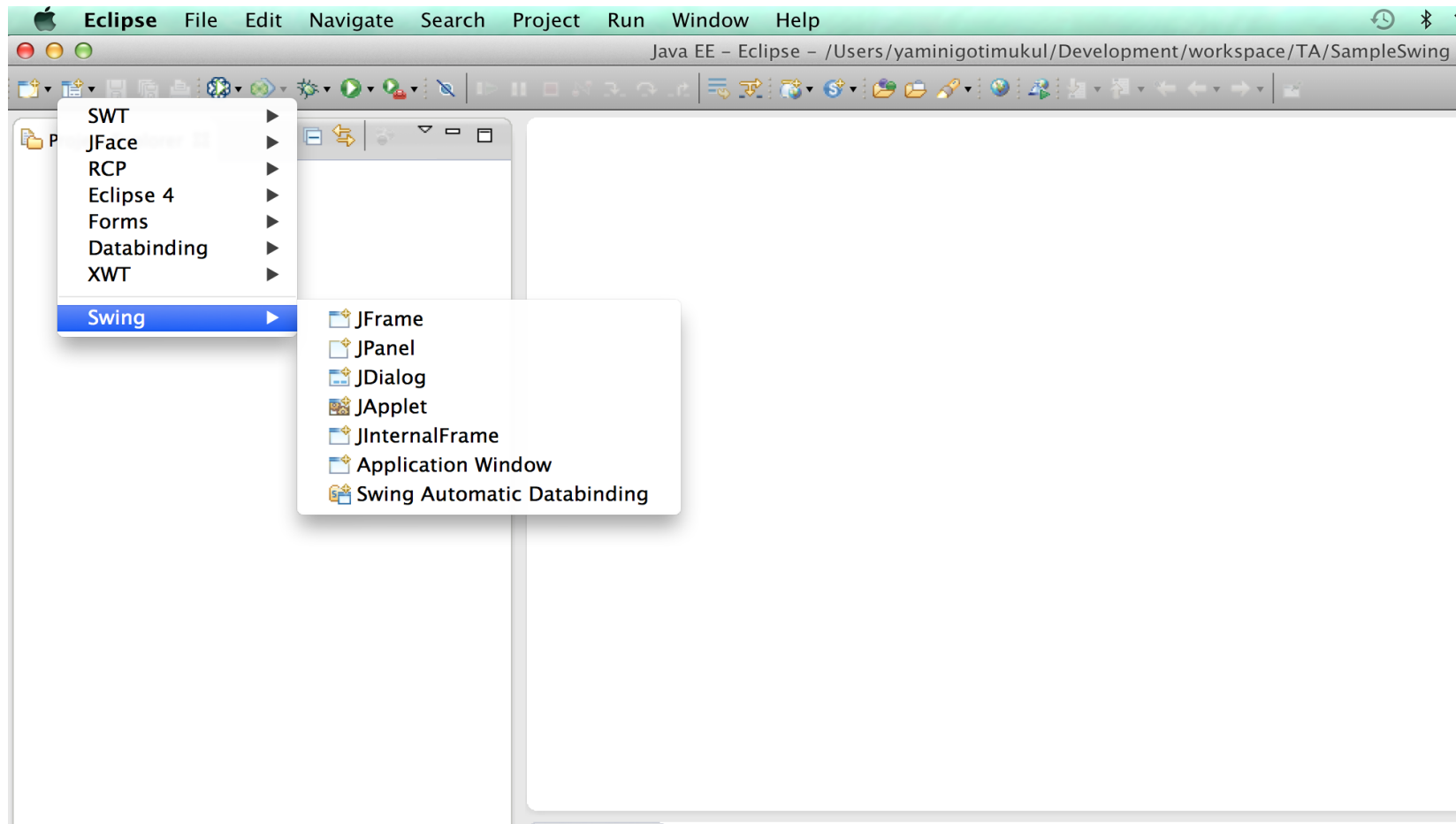
Short introduction for a quick start

Create a new project using the appropriate project wizard



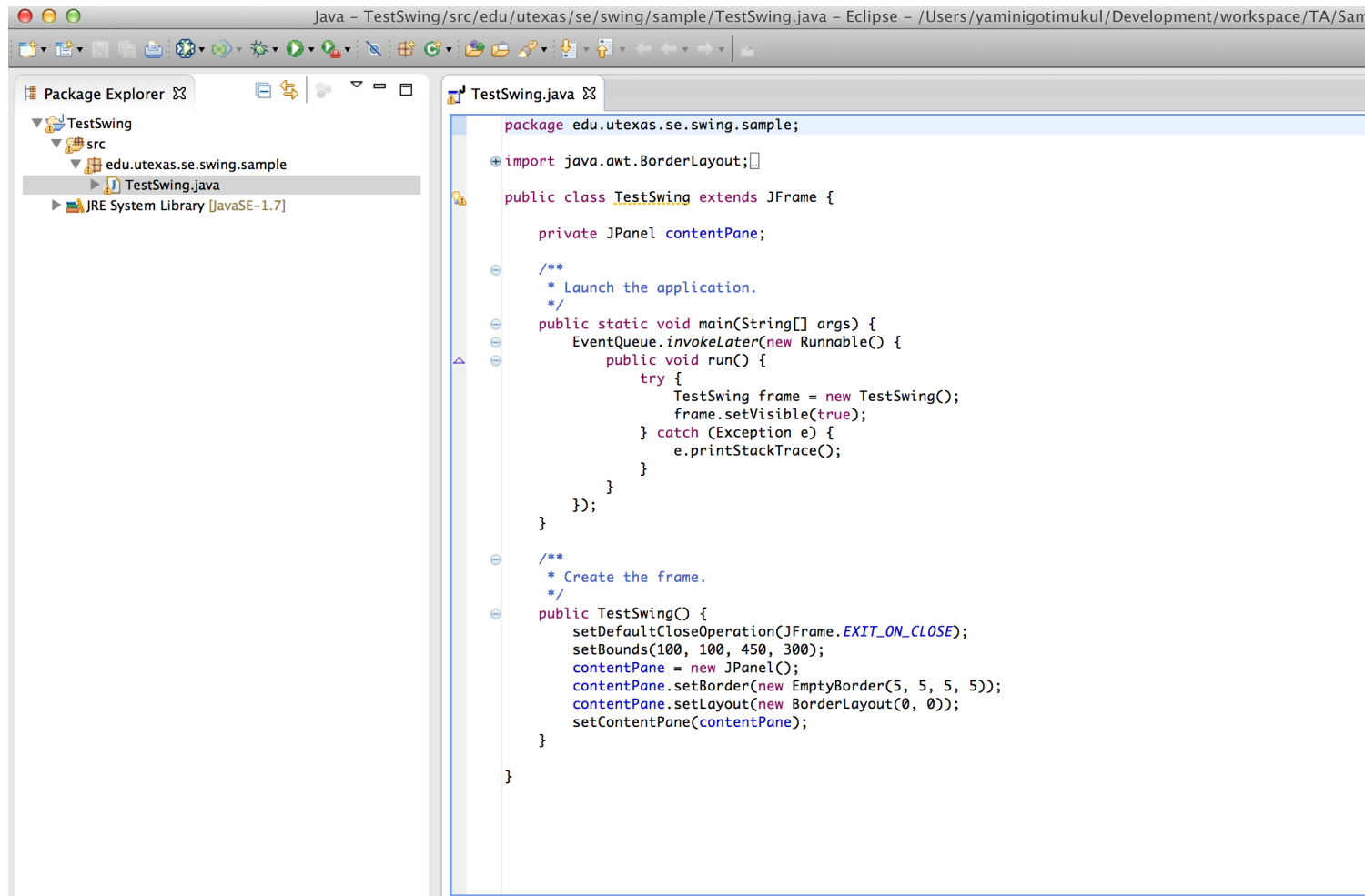
Short introduction for a quick start

Use class wizard to create your GUI window



Short introduction for a quick start

A class with some existing code is created.



The screenshot shows the Eclipse IDE with the 'TestSwing.java' file open. The Package Explorer on the left shows the project structure: 'TestSwing' package containing 'src' folder, which contains 'edu.utexas.se.swing.sample' package, which contains 'TestSwing.java'. The main editor shows the code for 'TestSwing.java'.

```
package edu.utexas.se.swing.sample;

import java.awt.BorderLayout;

public class TestSwing extends JFrame {

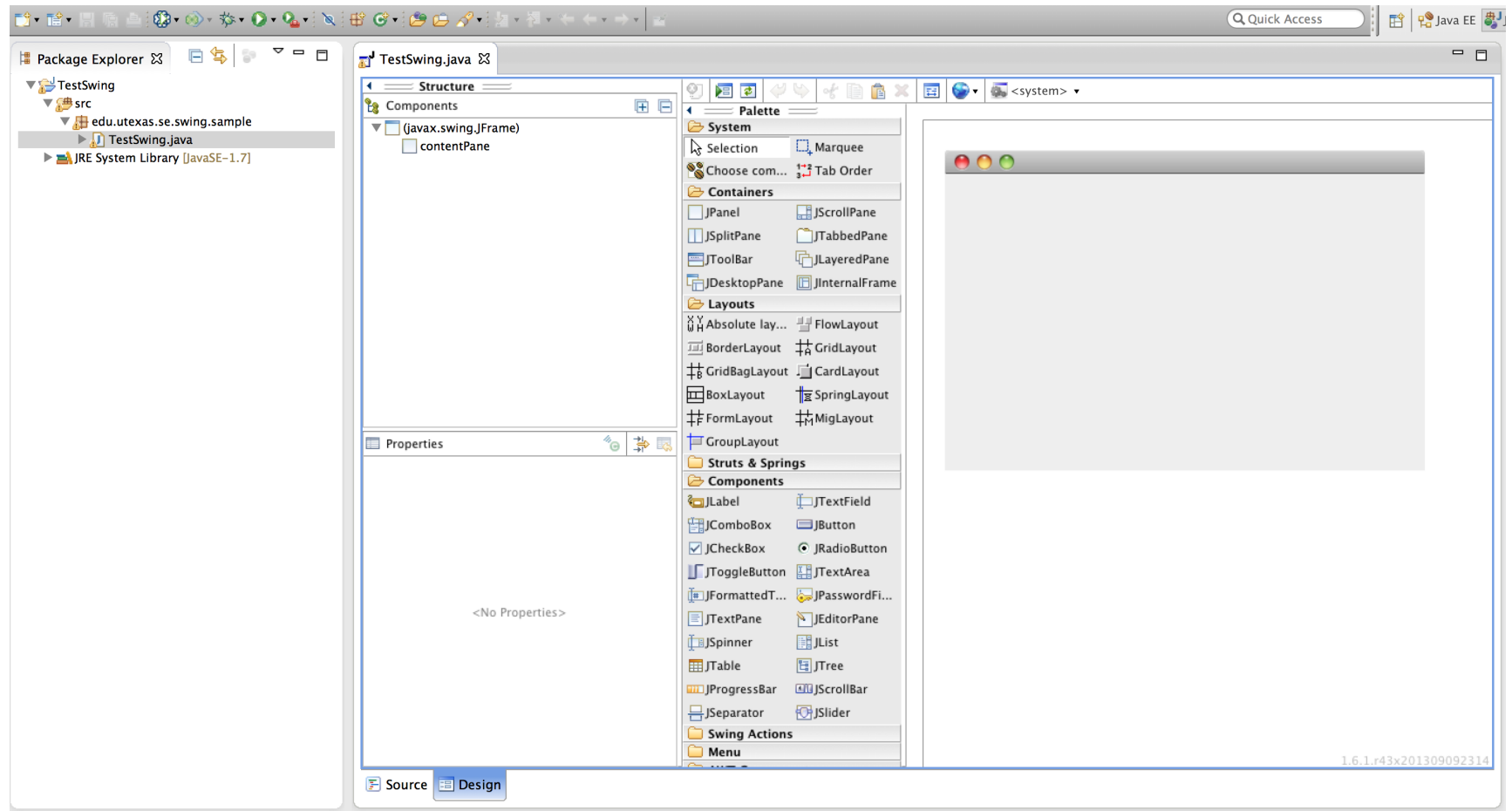
    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    TestSwing frame = new TestSwing();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public TestSwing() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        contentPane.setLayout(new BorderLayout(0, 0));
        setContentPane(contentPane);
    }
}
```

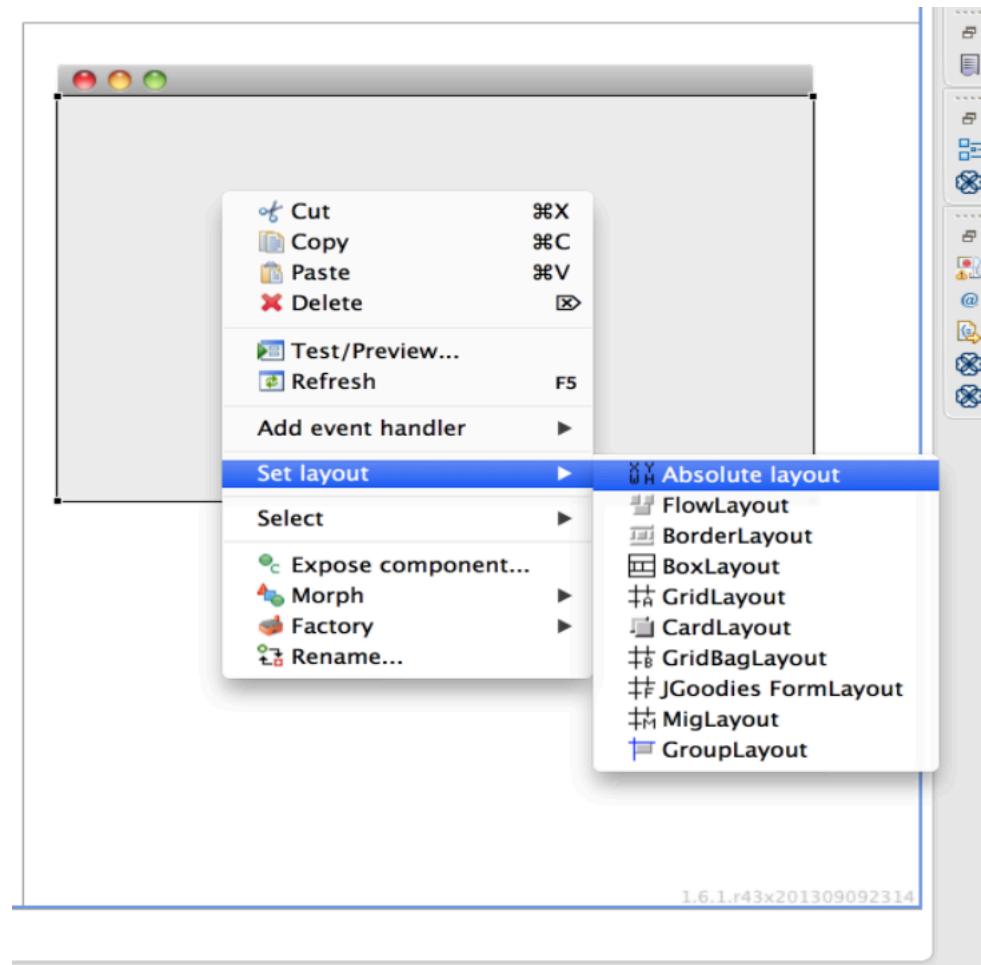

Short introduction for a quick start

To use the windowBuilder click on the "Design tab" at the bottom of the editor



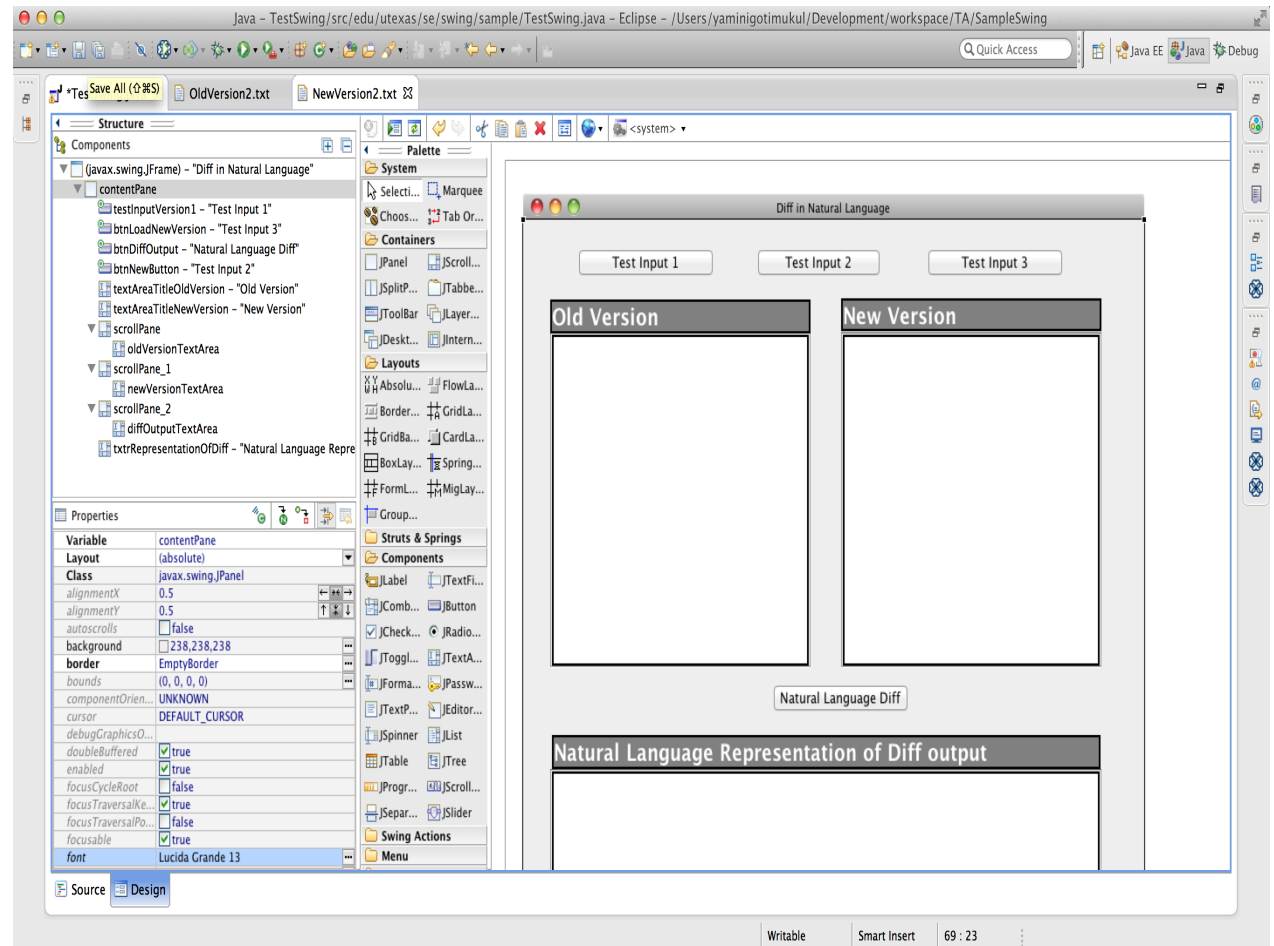
Short introduction for a quick start

To set layout, right click on the window frame and select " set layout " and select the appropriate layout



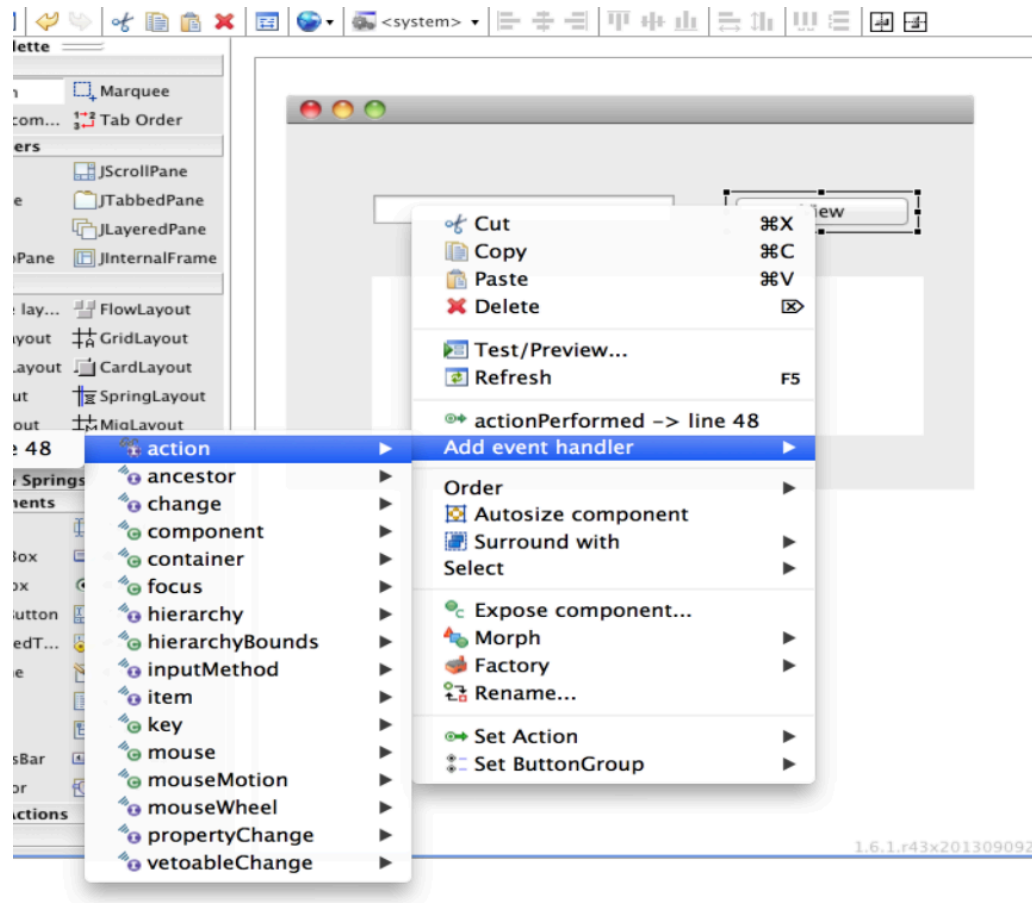
Short introduction for a quick start

Add various components from the Palette and edit their properties using the Property Pane.



Short introduction for a quick start

Add event handler to various widgets to provide behavior



Tips & Tricks

- Directly editing widget text in the design view
- Adding multiple widgets of the same type
- Editing properties of multiple widgets at the same time
- Configuring the palette
- Editing advanced widget properties
- Reverting a widget property to its default value
- Hiding code from the parser



Features

Bi-directional Code Generation

- Directly generates Java code
- The tool can read and write almost any format
- Reverse-engineer most hand-written Java GUI code.



Features

Internationalization (i18n)

- Directly generates Java code
- The tool can read and write almost any format
- Reverse-engineer most hand-written Java GUI code.



Features

Custom Composites & Panels

- Embed any Composite or Jpanel subclass in another window.
- All widgets defined in the are visible in the window they are used.
- Access any widget exposed via public accessors.
- Access exposed custom properties
- Exposed widgets appear in tree with the “exposed” decorator



Features

Factories

- Create a static factory method from any widget
- Factory methods may be parametrized with one or more arguments
- Morph any widget into a factory instance
- Add any factory component to the palette



Features

Visual Inheritance

- Inherit from any Shell, Composite, JFrame, Jdialog, Japplet or JPanel subclass.
- All inherited widgets are visible
- Access inherited widgets exposed via public or protected accesors
- Access inherited custom properties
- Inherited widgets appear in tree with the exposed decorator



Features

Event Handling

- Adding an Event Handler
 - First way is through the property pane
 - Second way is to simply right-click on a component, select add event handler.
- Deleting an Event Handler
 - Select the component tree
 - In the property pane, click the method you wish to delete and then press delete.



Features

Menu Editing

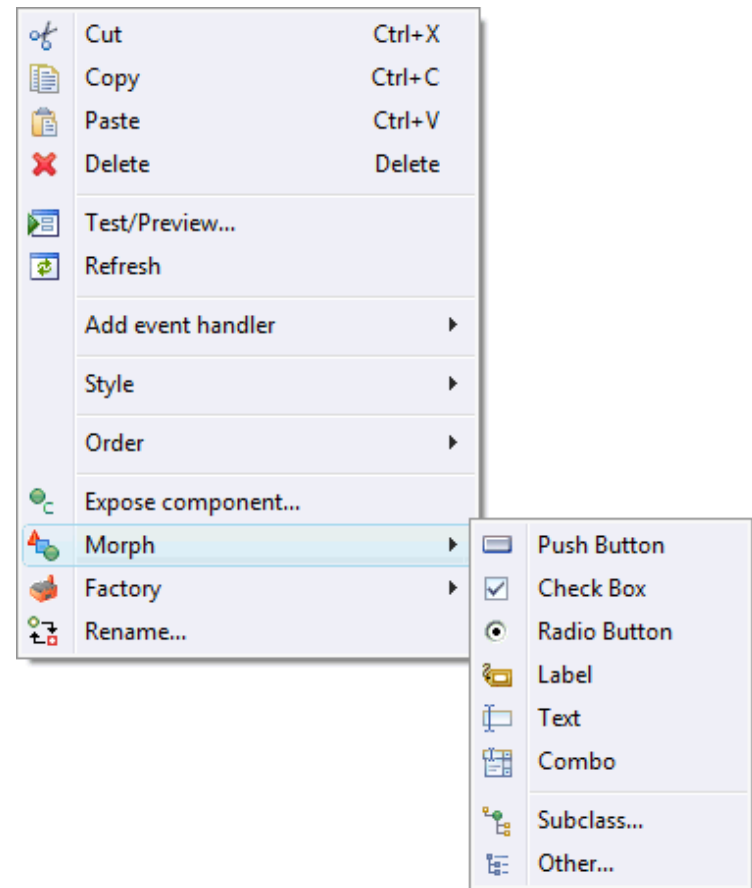
- Menus are a must for just about all GUIs.
- Only SWT Shells, SWT Applications Windows, and Wing JFrames supports menubars.
- Each window can have at most one Menu Bar, and the only place that you can put a Menu Bar is directly on a window.



Features

Widget Morphing

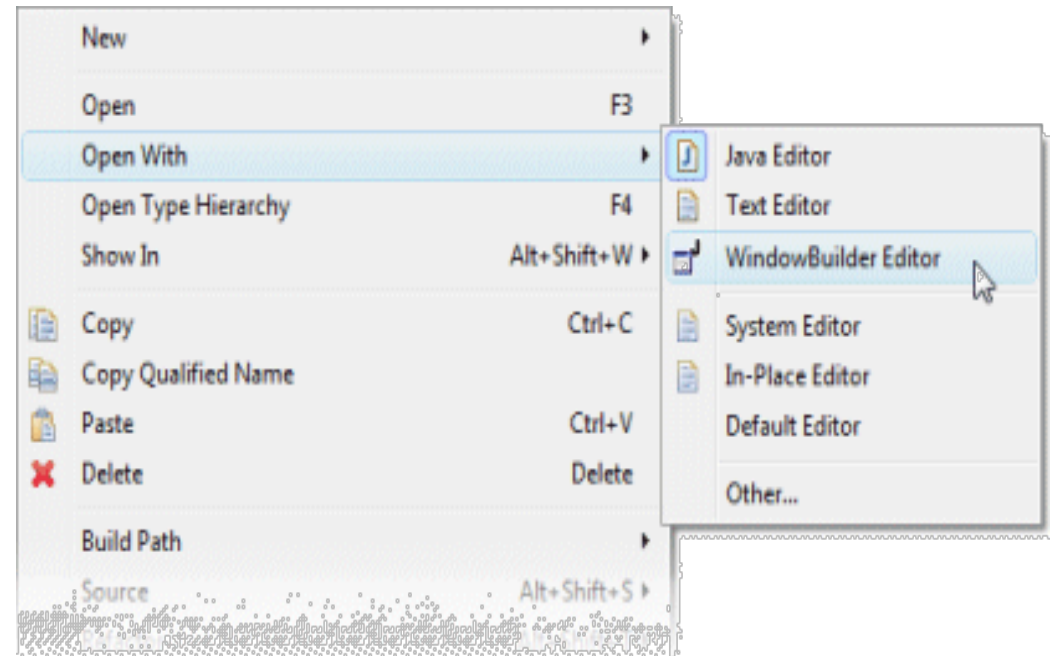
- The editor allows you to morph similar components from one type to another.



Features

Editing Existing Windows

Open a file that has not been created by the tool, simply right-click on the file and select Open With > WindowBuilder Editor



Wizards

- Swing
 - JFrame
 - JPanel
 - JDialog
 - JApplet
 - JInternalFrame
 - Application Window
 - Automatic Data Binding



Data Binding

- How WindowBuilder help us?
- Relieves the user from writing and registering listeners himself.
- Provides a common infrastructure for validation and conversion.
- It makes it easy to connect data sources to widgets.



Data Binding

- We have to import a JAR file to use dataBinding.



Beans Binding Org.jdesktop » 1.2.1

Mavenized construir para Beans Binding de "https://beansbinding.dev.java.net".

Fecha	(19 de mayo de 2010)
archivos	Descargar (JAR) (616 KB)
repositorios	Central
Usado por	12 artefactos

<https://mvnrepository.com/artifact/org.jdesktop/beansbinding/1.2.1>



Data Binding

Create Data Binding

Properties
Choose properties for the target and the model.

UpdateStrategy: READ

Target: **m_phoneTextField.text**
text

Model: **m_emailTextField.text**
text

Converter: N/S

Validator: N/S

Binding name:

▼ Binding

☐ Assign Binding to field

Cancel OK

Data Binding

Bound Properties:

Target	Model	Strategy	Binding
m_groupList	m_groups.groups	READ	jListBinding
m_groupList.detail	m_groups[\${name}] (\${personCount		jListBinding.setDetailBinding()
m_personTable	m_groupList.selectedElement.persc	READ	jTableBinding
m_personTable.column - Name -	m_groupList.name		

Target (Widgets): (javax.swing.JFrame).m_contentPane.rightPanel.topP
Model (Beans): m_deletePersonButton

type filter text

- (javax.swing.JFrame) - "Phone Book"
 - m_contentPane
 - leftPanel
 - groupToolBar
 - gbc
 - m_newGroupButton - "New..."
 - m_editGroupButton - "Edit"
 - m_deleteGroupButton - "Delete"

- autoBinding_2 - AutoBinding
- jTableBeanProperty_2 - BeanProperty
- jTextFieldBeanProperty_2 - BeanProperty
- autoBinding_3 - AutoBinding
- jTableBeanProperty_3 - BeanProperty
- jTextFieldBeanProperty_3 - BeanProperty
- autoBinding_4 - AutoBinding
- iTableBeanPropertyv_4 - BeanProperty

Properties:

<Self Object>
X+Y
= ?
<EL Expression>

Properties:

<Self Object>
X+Y
= ?
<EL Expression>

- background
- enabled
- font
- foreground
- height
- icon
- inputMap

Source Design Bindings

Bibliography

- [Help eclipse](#)
- [Eclipse windowBuilder](#)
- [Eclipse proposals windowBuilder](#)



END

THANKS!

You can find as alu0100881677 & alu0100898293

Slides and code at [Github repo](#)

