Eclipse WindowBuilder - Creating user interfaces - Tutorial

Lars Vogel

Version 2.4

27.10.2015

QUICK LINKS

- 22 Feb - RCP Training
- 23 Nov - Android Training

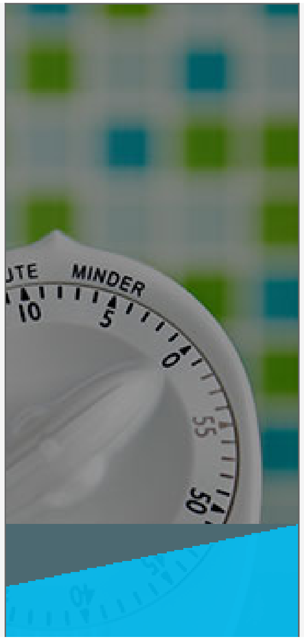
Lars Vogel
Foreword by Mike Milinkovich, Executive Director, Eclipse Foundation

Eclipse WindowBuilder

This tutorial describes the usage of WindowBuilder for creating user interfaces.

- vogella Training
- vogella Books

Table of Contents



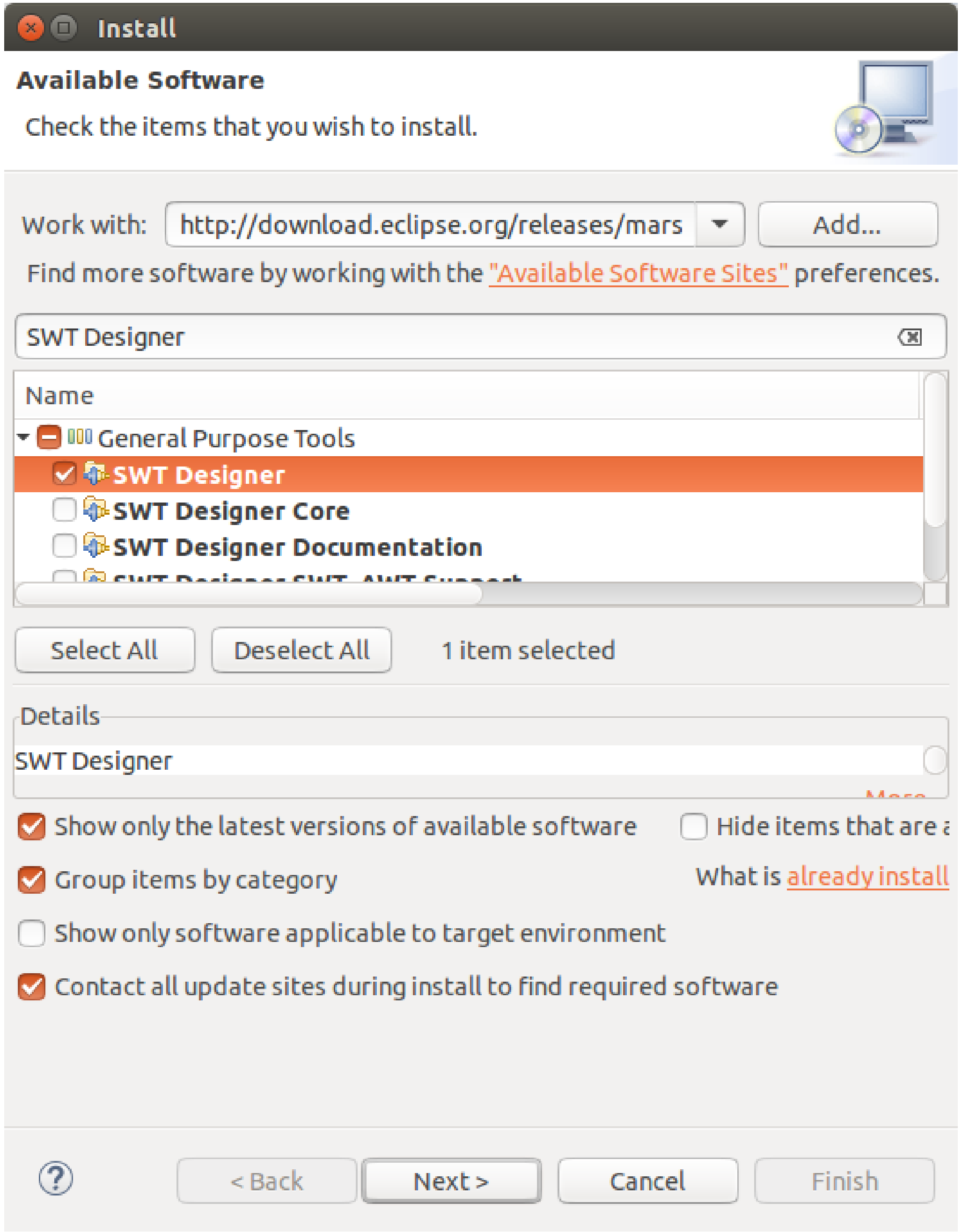
1. SWT Designer (WindowBuilder)

SWT Designer is a visual editor used to create graphical user interfaces. It is a two way parser, e.g., you can edit the source code or use a graphical editor to modify the user interface and SWT Designer will synchronize between both representations.

SWT Designer is part of the WindowBuilder project. WindowBuilder provides the foundation and SWT Designer adds the support for working with SWT based applications. SWT Designer supports Eclipse 3.x and Eclipse 4 RCP applications.

2. Install SWT Designer

You can install SWT Designer via the Eclipse update manager from the main Eclipse update site.



Install

Available Software

Check the items that you wish to install.

Work with: http://download.eclipse.org/releases/mars

Find more software by working with the "Available Software Sites" preferences.

SWT Designer

Name
▾ ⊟ General Purpose Tools
☑ SWT Designer
☐ SWT Designer Core
☐ SWT Designer Documentation

1 item selected

Details

SWT Designer

☑ Show only the latest versions of available software

☐ Hide items that are a

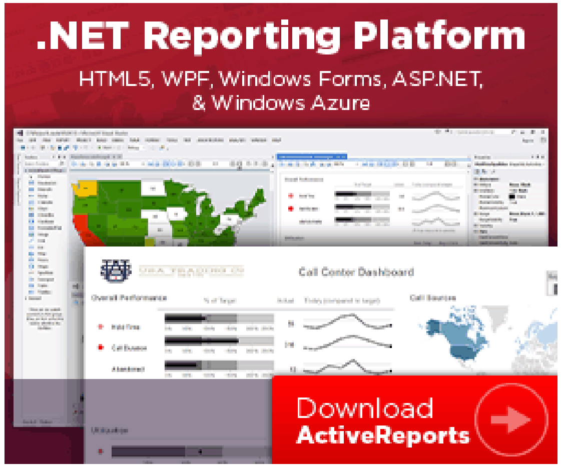
☑ Group items by category

What is already install

☐ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

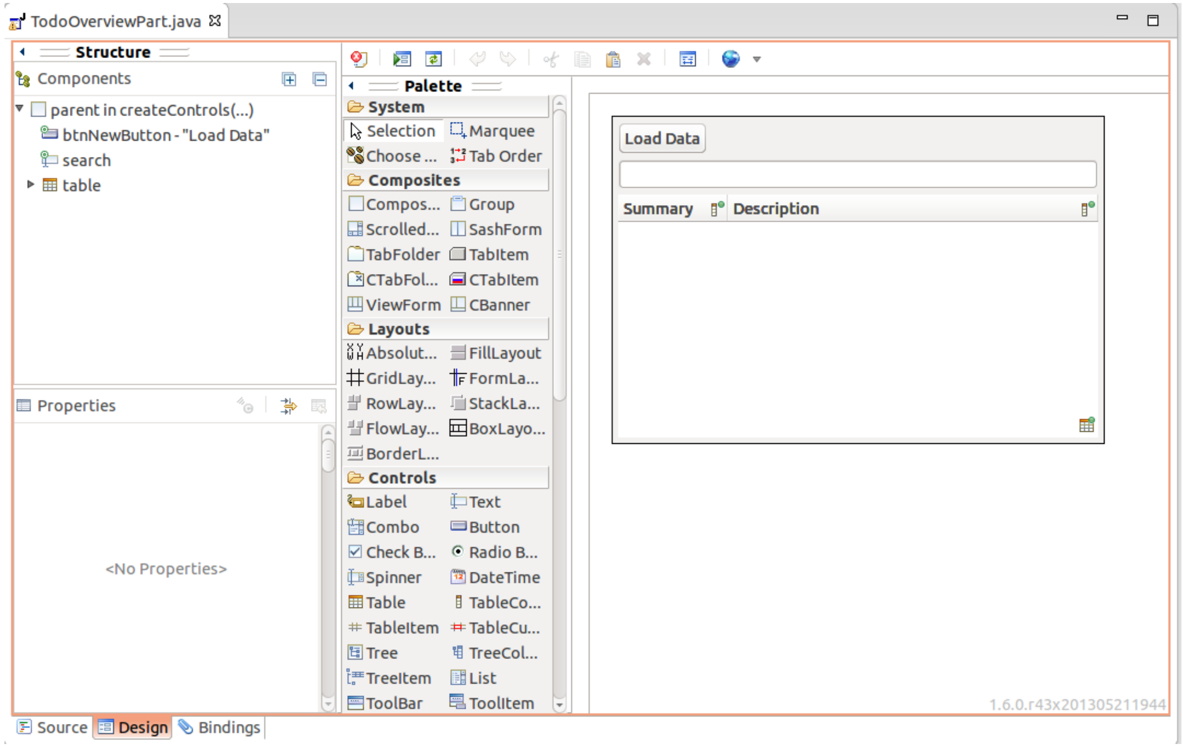
> **Tip:** The latest release of SWT Designer can be found on the following website: **WindowBuilder download**.



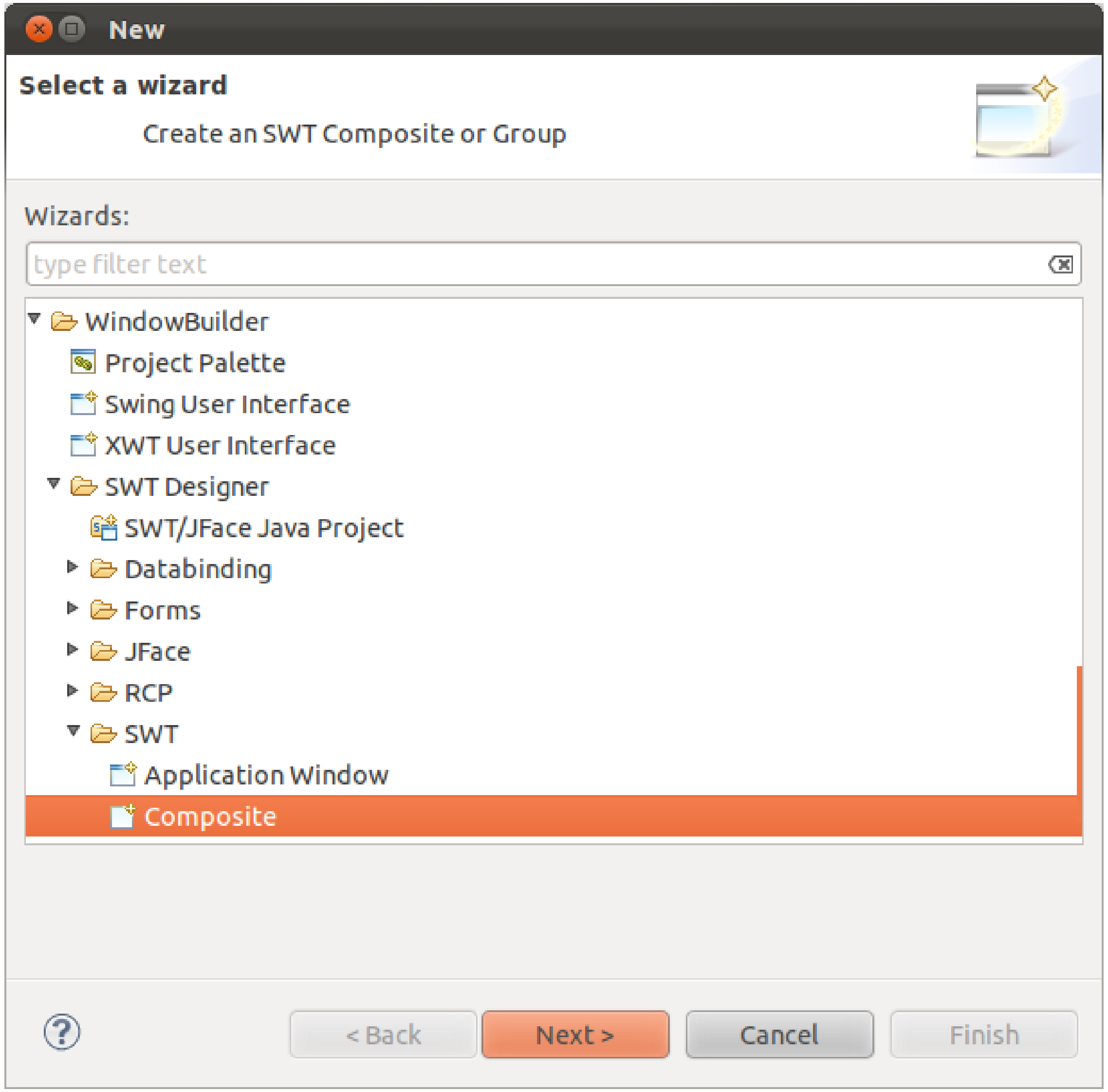
3. Using SWT Designer

SWT Designer allows you to open Java components in a special editor. This editor can work with parts, SWT and JFace dialogs, JFace wizards, etc.

SWT Designer allows to drag-and-drop SWT components into an existing layout, change layout settings and create event handlers for your widgets.



You can also use the SWT and JFace templates which SWT Designer contributes to the Eclipse IDE. For example you can use it to create `Composites` and add these to the Eclipse user interface. To create a new `Composite` select *File → New → Other... → WindowBuilder → SWT Designer → SWT → Composite*.



SWT Designer has excellent support to establish data binding between your data model and your user interface components via the JFace Data Binding framework.

4. Exercise: Getting started with SWT Designer

4.1. Installation

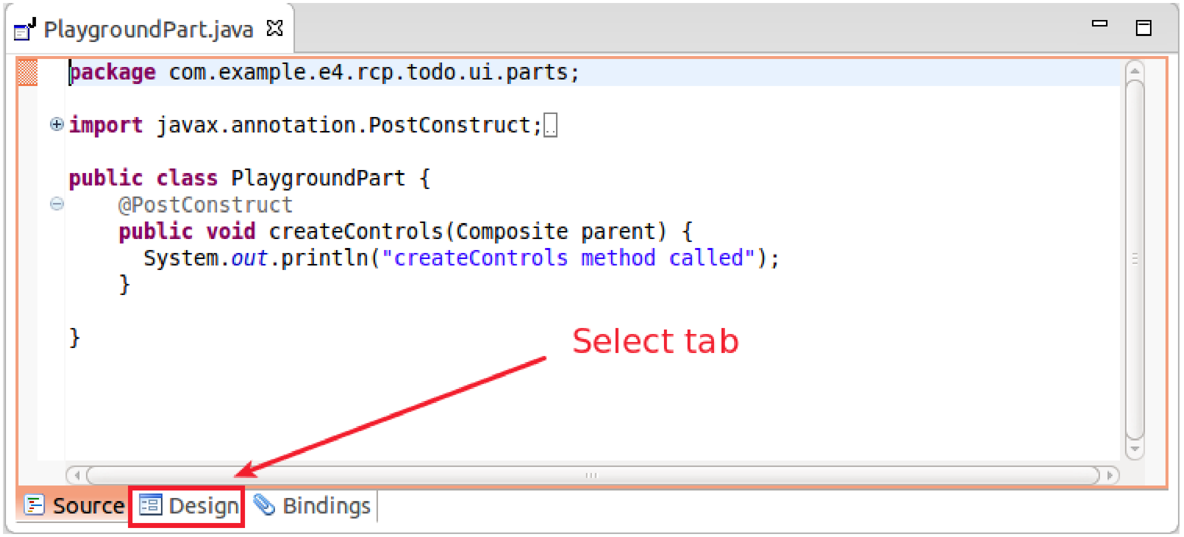
Install SWT Designer if you have not already done this. See **Section 2, "Install SWT Designer"** for a description.

4.2. Building an user interface

Right-click on your `PlaygroundPart` class and select *Open With → WindowBuilder Editor*.

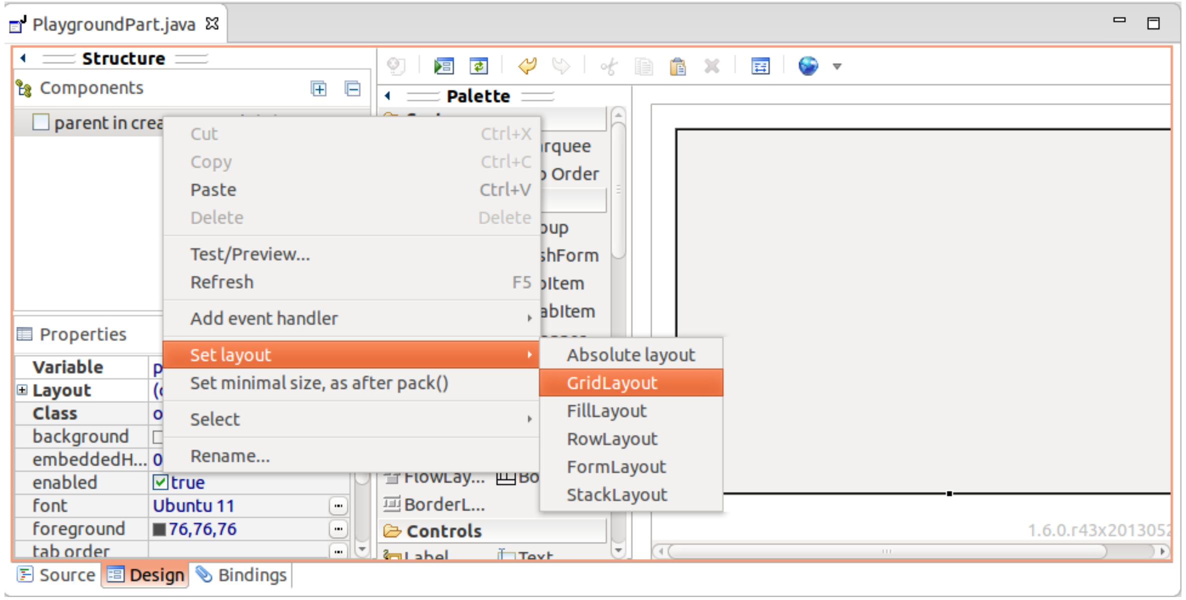
> **Note:** WindowBuilder uses the `@PostConstruct` method to identify that a class is an Eclipse part. This method needs to specify at least a `Composite` as parameter.
>
> ```
> // the WindowBuilder / SWTDesigner tooling
> // uses methods to figure out that the
> // class is am Eclipse 4 part
>
> // one method must be annotated with @PostConstruct and
> // must receive a least a Composite
>
> @PostConstruct
> public void createControls(Composite parent) {
>
> }
> ```

Switch to the *Design* tab in the WindowBuilder editor. This selection is highlighted in the following screenshot.

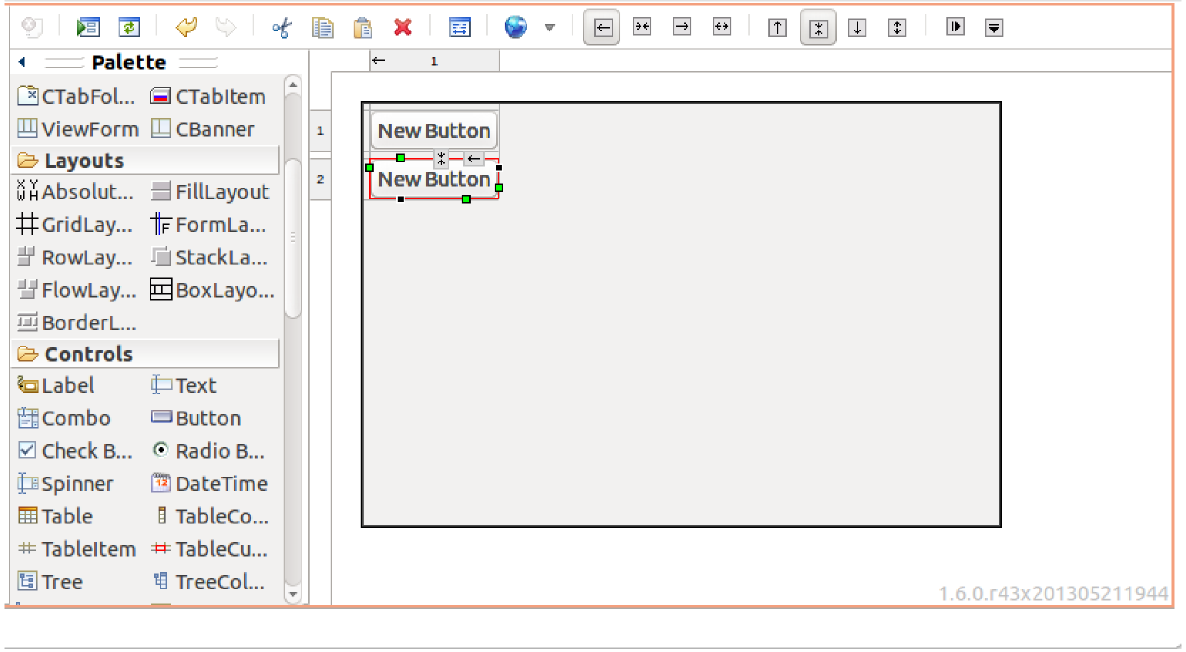


```
package com.example.e4.rcp.todo.ui.parts;

import javax.annotation.PostConstruct;

public class PlaygroundPart {
    @PostConstruct
    public void createControls(Composite parent) {
      System.out.println("createControls method called");
    }

}
```

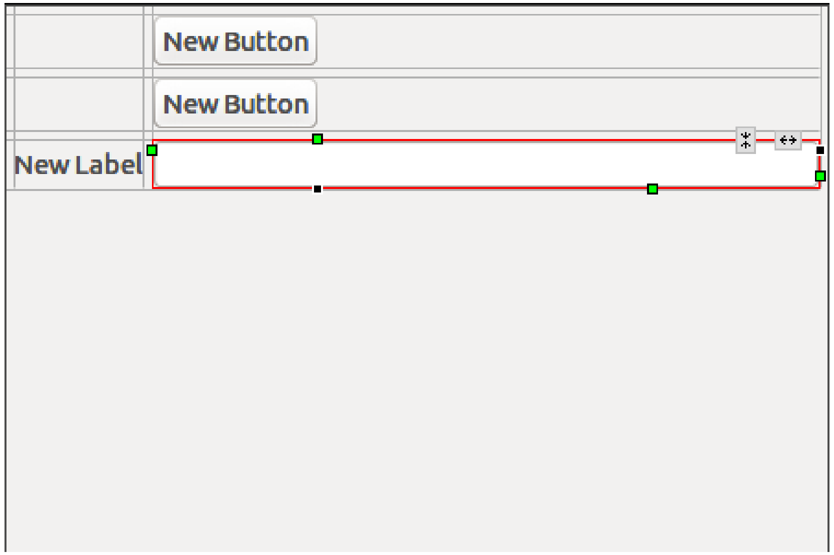
Use SWT Designer to change the layout of the `Composite` of the part to a `GridLayout`.



Click in the Palette on *Button* and add a few buttons to your user interface.

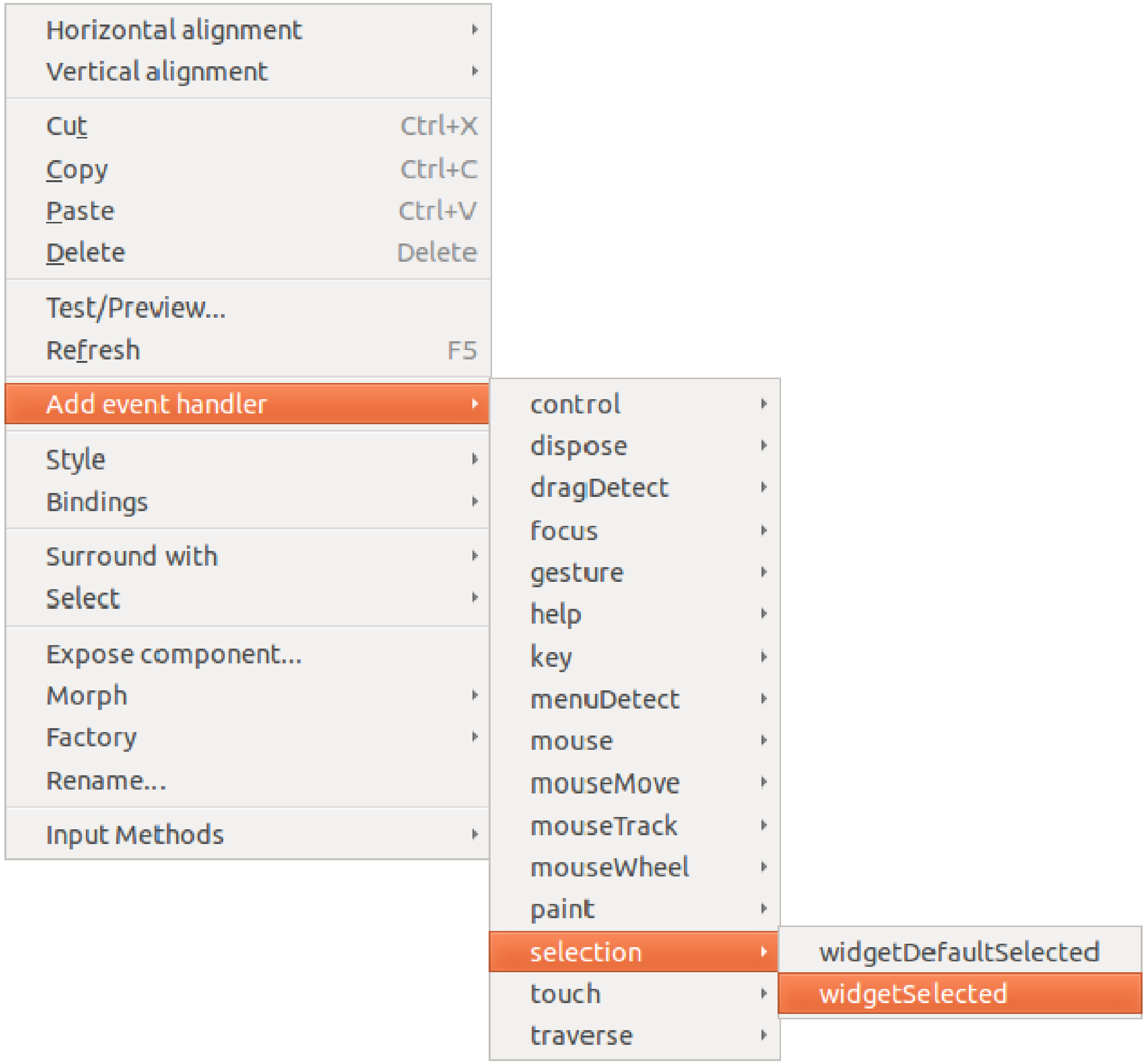


Add a `Label` and a `Text` field.



4.3. Creating an event handler

Assign an event handler to one of your buttons via a right-click on the button. Select *Add event handler → selection → widgetSelected*.

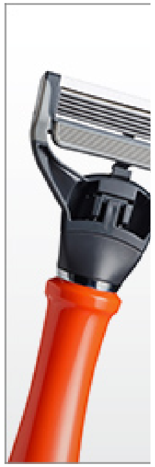


4.4. Review the generated code

Switch to the *Source* tab and review the code generated by the SWT Designer.

5. Learn more about Eclipse 4 RCP development

I hope you enjoyed this tutorial. You find this tutorial and much more information also in the **Eclipse 4 RCP book** from this author.



6. About this website

Support free content

Questions and discussion

Tutorial & code license

Get source code