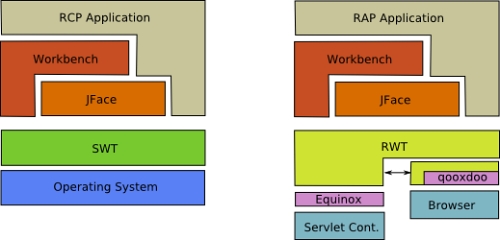
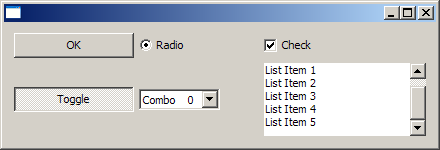
2- RCP (Rich Client Platform)

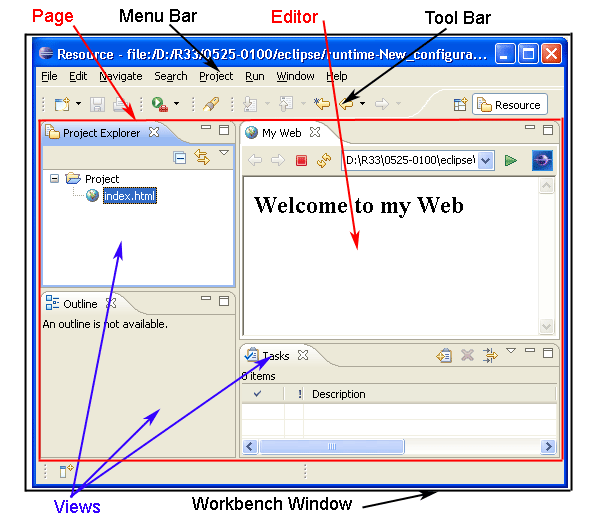
RCP (Rich Client Platform) - As a platform based on SWT, used for programming Desktop applications, so far it has built a platform that allows you to develop desktop-style applications Workbench, like Eclipse IDE, or programmers can integrate the plugin to Eclipse IDE.  
  
But even if you want to use SWT to programming, and do not need to use those provided by the RCP you should also create an RCP application.



SWT:

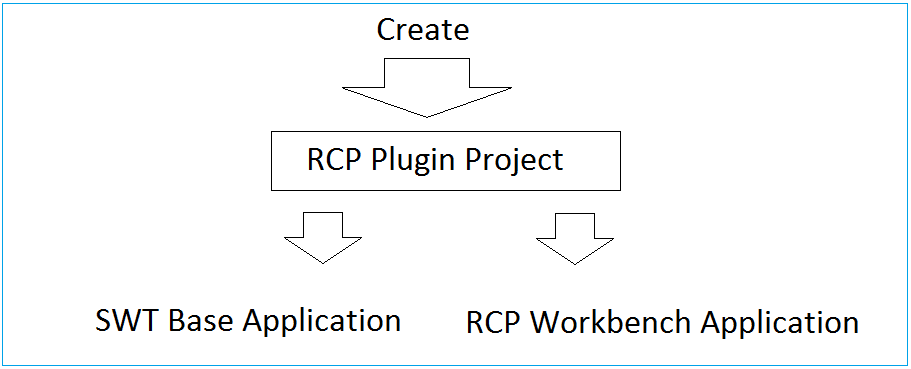


Workbench Application:



To create a desktop application using SWT. On the Eclipse, we will create an RCP Plugin Project. You have 2 options.

* Only use the features of the SWT
* Using the platform provided by the RCP to RCP Application Workbench programming

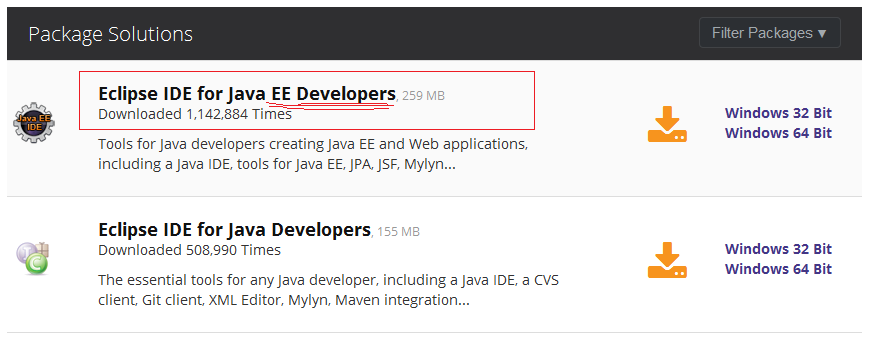


In this document, I will guide you to become familiar with basic programming SWT, using WindowBuilder to drag and drop components into the interface.

3- The settings required before starting

Some required settings before you start:  
  
You need the latest version of Eclipse. There currently is Eclipse 4.7 (Codes OXYGEN).

In my opinion you to download package: "Eclipse IDE for Java EE Developers". The only different is number of Plugins, for the purpose of different programming. You can install additional plugins for other purposes if desired.



WindowBuilder, this is a plugin that allows you to design SWT GUI applications using drag and drop convenience.  
See installation instructions at:

* Install WindowBuilder for Eclipse



4- Some concepts of SWT.

4.1- Display & Shell

The Display and Shell classes are key components of SWT applications.  
  
- org.eclipse.swt.widgets.Shell class represents a window.  
  
- org.eclipse.swt.widgets.Display class is responsible for managing event loops, fonts, colors and for controlling the communication between the UI thread and other threads. Display is the base for all SWT capabilities.  
  
Every SWT application requires at least one Display and one or more Shell instances. The main Shell gets, as a default parameter, a Display as a constructor argument. Each Shell is constructed with a Display and if none is provided during construction it will use either the Display which is currently used or a default one.

Example:

Display display = **new** **Display**();

Shell shell = **new** **Shell**(display);

shell.open();

// run the event loop as long as the window is open

**while** (!shell.isDisposed()) {

// read the next OS event queue and transfer it to a SWT event

**if** (!display.readAndDispatch())

{

// if there are currently no other OS event to process

// sleep until the next OS event is available

display.sleep();

}

}

// disposes all associated windows and their components

display.dispose();

4.2- SWT Widgets

SWT widgets are located in the packages org.eclipse.swt.widgets and org.eclipse.swt.custom. Widgets extend either the Widget or the Control class. Several of these widgets are depicted in the following graphic.

