



By Jerry Karasz  
Software Architect

# Building WindowBuilder

Once you decide to create a new BBJ® application with a graphical user interface (GUI), what is next? How do you get started? You probably already have a good idea of what you want the program to do and what you want your users to see. You may even know how you could create your GUI controls in BBJ code, but you are looking for an easier way to lay them out than manually creating the code. Wouldn't it be better if you could see how they look as you lay them out?

The BDT Eclipse development environment is now your best choice for GUI and BUI development. The BASIS NetBeans IDE is still a valid tool to use for the layout, but a better alternative is now under construction. Using the Business BASIC Development Tools (BDT) in Eclipse, you can download and install the BDT Eclipse plug-in, and you will be able to download the AppBuilder and fully functional WindowBuilder (WB) plug-ins with BBJ 15.0. These plug-ins support Java 1.8, whereas the BASIS NetBeans IDE only supports up to Java 1.7. Building on what Eclipse provides, the BDT plug-ins offer a number of user-friendly utilities that will make your development faster and easier.

To implement a BBJ GUI application fully, you need to perform these three tasks:

1. Lay out the graphical controls
2. Associate handlers with your controls' events
3. Write code to implement your business logic

This article focuses on the WB Eclipse plug-in as it exists today as the alpha release that accompanied BBJ 14.20. WB helps you lay out your graphical controls on one or more top level "windows" and "child windows" to match your design. For our purposes, we will assume that you have a BBJ project named **WindowBuilder1**, with an .arc file named **WindowBuilder1.arc** that is empty and open in a WindowBuilder editor window. If you don't have a project and .arc file, follow the directions at [Creating a WindowBuilder Project](http://links.basis.com/creatingwinbuild) ([links.basis.com/creatingwinbuild](http://links.basis.com/creatingwinbuild)) and create the BBJ project and .arc file. Delete all of the controls placed there when you created your .arc file, and start with just the "Composite" in the drawing area as shown in **Figure 1**.

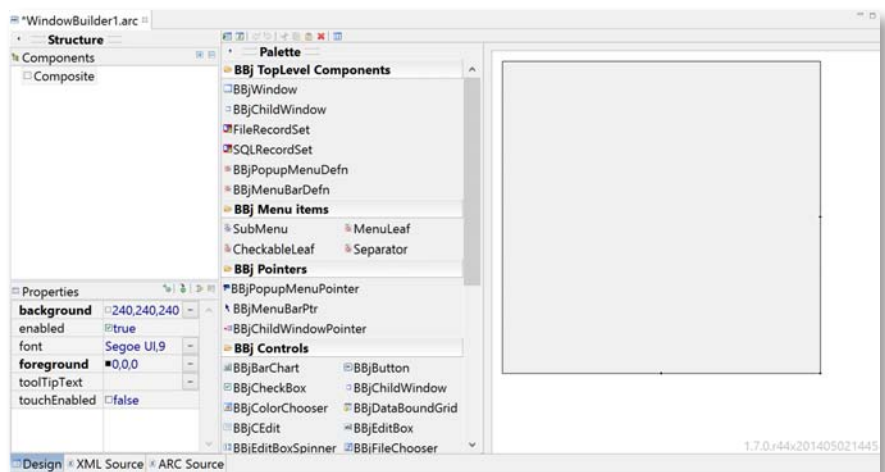


Figure 1. Starting with an empty .arc file

## Features Today

Let's take a moment and look at the BDT WindowBuilder features that are available in Eclipse with the release of BBJ 14.20.

### WindowBuilder Editor

When you open **WindowBuilder1.arc** for editing, Eclipse opens it in a WB editor. This editor displays as one tab labeled with the .arc file name that in turn has its own three tabs – 'Design', 'XML Source', and 'ARC Source'. By default the 'Design' tab displays, but if you click on the 'XML Source' or 'ARC Source' tab, you can view the XML or ARC file representation of the WB controls you have placed. The XML should be of no interest to you as it is just an intermediary format used by Eclipse. The ARC text is what WindowBuilder will actually write out to your .arc file when you save your changes.

For now, let's stick with the 'Design' tab and look at the three areas it offers: 1) 'Structure' made up of 'Components' and 'Properties' panes, 2) 'Palette', and 3) the unlabeled drawing area. The 'Palette' area shows all of the installed control palettes; **Figure 1** showed all of the default palettes disabled except for the four BBJ palettes (BBJ TopLevel Components, BBJ Menu items, BBJ Pointers, and BBJ Controls). To change which palettes display and how, use the Palette Manager (right-click any palette and choose [Palette Manager...]).

Your first task is to add a top-level window to the drawing area, so in the 'BBJ TopLevel Components' palette, click to select 'BBJWindow'. Then click in the gray portion of the drawing area to place your BBJWindow. Notice that several things happened when you did this. First, a rectangle appeared in the drawing area you selected that displayed resize points (small black squares) you could click and drag to resize it (see **Figure 2**). But more than that happened; a strange entry was selected in the 'Components' pane, **p1:BasisWindow** as illustrated in **Figure 3**.

The entry **p1:BasisWindow** is the XML element name for the BBJWindow. In fact, if you click on the 'XML Source' tab you will see a **p1:BasisWindow** XML element appear as shown in **Figure 4**. Correspondingly, if you click on the 'ARC Source' tab, you will see a WINDOW entry appear (see **Figure 5**).

Going back to the 'Design' tab, you may notice another change that took place. The 'Properties' pane now displays properties – in this case, for the selected item in the drawing area, the BBJWindow. If you expand the 'Bounds' property, you see the individual values: x, y, width, and height as **Figure 6** displays.

Your next task is to resize the BBJWindow to occupy most of the drawing area. You can do this by clicking and dragging the resize point in the lower-right corner of the BBJWindow in the drawing area, or you can click and edit the width and height properties directly in the 'Properties' pane.

It is now time to add a control in your design to your BBJWindow. To add any control, scroll in the 'Palette' pane and click to select it, and then click to place it where you would like it on top of the BBJWindow. Continue to add controls, setting their properties as you go, until your window is complete. Finally, save your work by selecting File > Save from the main menu, by using the [Ctrl]+S shortcut, or by clicking the [Save] toolbar button.

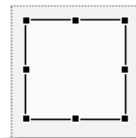
Now let's talk about some of the exciting new features that Eclipse offers us for WB.

### Previewing Your Window

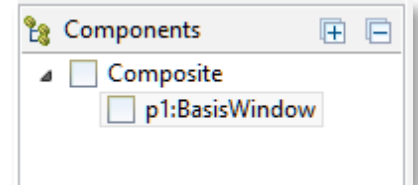
Locate the button on the WB editor toolbar shown in **Figure 7** that has the hover text 'Quickly see/preview the window without compiling or running it'. Click this button and a small popup window appears showing you what your screen will look like when you use your new .arc file in a BBJ program.

### Aligning Controls

WB assists you with aligning your controls as you drag them around the BBJWindow. If you drag a control to be near another control, WB will automatically show an alignment indicator (a line) connecting any



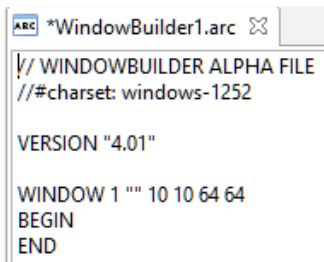
**Figure 2.** A BBJWindow selected in the drawing area



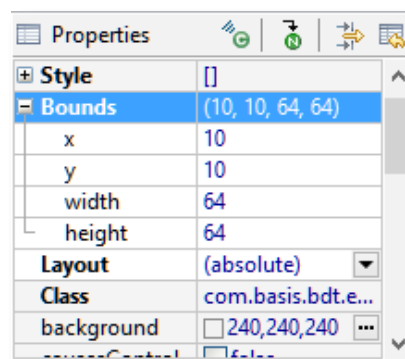
**Figure 3.** A BBJWindow in the 'Components' pane



**Figure 4.** BBJWindow in the XML Source as p1:BasisWindow



**Figure 5.** BBJWindow in the ARC Source as WINDOW

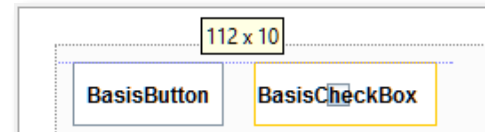


**Figure 6.** BBJWindow properties in the 'Properties' pane



**Figure 7.** Toolbar button for previewing your BBJWindow

side of the controls that is aligned (see **Figure 8** for an example when dragging a checkbox whose top aligns with the button). Notice also that the current X and Y coordinates of the control display as you drag it (112 x 10). This allows you to locate your control exactly where you want it. Remember, if you don't want to drag-and-drop your control, you can still select it and set the 'Bounds' values (x, y, width, and height) in the 'Properties' pane to get the same effect.



**Figure 8.** Alignment indicator while dragging a checkbox control

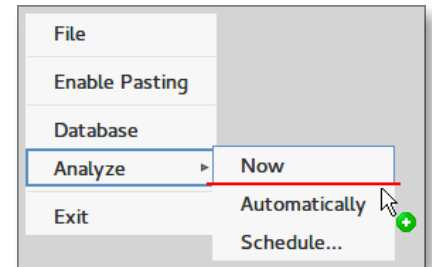
In addition, once you select more than one control, you have access to all of the standard alignment, movement, and spacing toolbar buttons shown in **Figure 9**, as well as the standard cut, copy, and paste functions.



**Figure 9.** WB alignment toolbar buttons

## Menus and PopUp Menus

We are currently finalizing the WYSIWYG model in WB that allows you to easily build menus and popup menus. The new mouse implementation for WB makes it simpler and more intuitive than ever. To create a menu, simply click a 'BBJMenuBarDefn' or 'BBJPopupMenuDefn' control to select it on the palette, and then click on either the 'Composite' object in the 'Structure' pane or on the gray portion of the drawing area to place it. Continue to build your menus by click-adding 'SubMenu', 'MenuLeaf', 'CheckableLeaf', or 'Separator' items onto a 'Menu' object in the 'Structure' pane or onto the expanded menu displayed in the drawing area (see **Figure 10**) where the red line appears.



**Figure 10.** Adding to a menu in the drawing area

Don't forget to place a menu pointer on every control that should offer the menu. If, for example, you have a 'BBJMenuBarDefn', you must place a 'BBJMenuBarPtr' control on the 'BBJWindow' that will hold the menu, and link them by setting the pointer's 'menuBarDefnID' to the 'controlID' value of the menu. Similarly, you must place a 'BBJPopupMenuPointer' control on the control that will offer the popup menu, and link them by setting the pointer's 'popupMenuID' to the 'controlID' value of the popup menu.

## Tab Controls

We have also enhanced your experience when placing a child window onto one of the tabs in a tab control – click a 'BBJChildWindowPointer' control to select it on the palette, and then click on either the 'TabFolder' object in the 'Structure' pane or on the top of the tab control in the drawing area (**Figure 11**) to place it. This adds a new tab with your pointer on it.



**Figure 11.** Adding a child window pointer to a tab control in the drawing area

## Productivity

Possibly the best benefit of using BDT WB is that your control layout and editing can now be together with your code editing in Eclipse ... one powerful IDE to enhance your productivity that gives you a boost in writing code and in laying out your controls.

## Features Coming Soon

You may have noticed that WB is still under construction. BASIS is working hard to make sure that all of the appropriate functionality in Eclipse is available for WB users. This is a big task and promises some exciting new enhancements for building GUIs in the near future.

## Multiple Top-level Windows

We are working to provide a better way for WB to manage the multiple top-level windows you may have in one .arc file. In the BASIS NetBeans IDE, you could open a display for each top-level window, but interaction with them was limited by only being able to open one shared support window of each type: BBJGUI Palette, BBJGUI Inspector, and Properties. This meant that much of the editing could only occur on one top-level window at a time. WB will soon allow you to view and edit multiple BBJWindows at one time, where each display has its own 'Palette' and 'Properties' displays so that you can easily compare values and edit them.

## .ARC File Properties

Another task is to extend the properties that are available for you to set and store in an .arc file. The format of an .arc file has been stable for many years, while the controls have continued to grow and offer more and more functionality. However, unless those properties can be stored in your .arc file, you cannot set them in WB and have them remain set when you run. Look for new properties to start appearing on many of the controls as we go forward – everything from setting the row gutter on a BBJGrid to setting the selected tab index on a BBJTabCtrl.

## Summary

The WB Eclipse plug-in is BASIS' newest user-friendly tool to help you develop GUI and BUI applications quickly and easily. It will carry you into the future as Java 1.8 releases occur, and it offers a number of user-friendly utilities to make your development faster and easier. To make it even better, BASIS plans several enhancements for WB to leverage the power of the Eclipse development environment, not the least of which is the integration into Eclipse with the BDT CodeEditor. WindowBuilder is your best choice for graphical control layout for your GUI and BUI applications. ■



- Review [Creating a WindowBuilder Project](#) in the online Help or within the EclipseHelp
- Check out the [list of WB features](#) already implemented and those yet to be implemented