

## Installing Custom Layout Managers in JBuilder

NOTE: This supplemental material is Example 6.2, "Creating a Custom Layout Manager," in Rapid Java Application Development Using JBuilder 4/5/6.

JBuilder uses the OpenTool technology to enable the users to customize JBuilder. You can add a new menu, a new file type, a new viewer, or a new layout manager. This section demonstrates installing customer layout managers in JBuilder.

Suppose you have created a custom layout manager named `DiagonalLayout` in Example 6.2, "Creating a Custom Layout Manager." Here are the steps to install it in JBuilder:

1. In the `customlayoutdemo` project, create a class named `DiagonalLayoutAssistant`, as shown in Listing 1. This class will be used to load the layout manager when JBuilder starts. The static method `registerAssistant` registers the layout assistant and the layout manager with JBuilder.

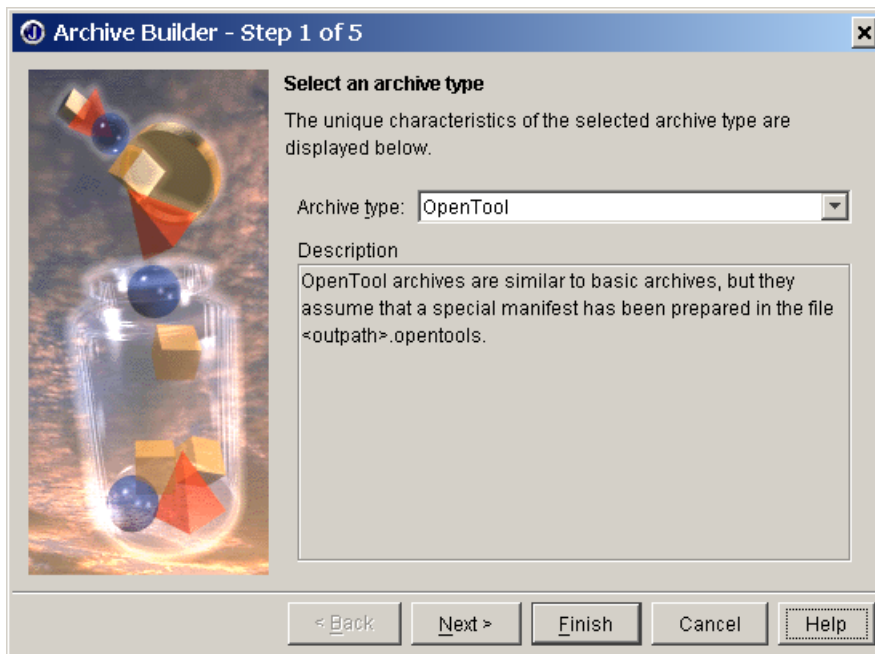
Listing 1: `DiagonalLayoutAssistant.java`

```
package customlayoutdemo;  
  
import com.borland.jbuilder.designer.ui.UIDesigner;  
import com.borland.jbuilder.designer.ui.BasicLayoutAssistant;  
  
public class DiagonalLayoutAssistant extends BasicLayoutAssistant {  
    /**Initializes and registers this OpenTool  
    */  
    public static void initOpenTool(  
        byte majorVersion, byte minorVersion) {  
        UIDesigner.registerAssistant(DiagonalLayoutAssistant.class,  
            "customlayoutdemo.DiagonalLayout", true);  
    }  
}
```

2. Create a JAR file for `DiagonalLayout.class` and `DiagonalLayoutAssistant.class` as follows:
  - 2.1. Choose Wizards, Archive Builder to display the Archive Builder - Step 1 of 5. Select OpenTool as the archive type, as shown in Figure 1.
  - 2.2. Click Next to display Archive Builder - Step 2 of 5, as shown in Figure 2. Click the Browse button to display a file dialog box, and type `c:\radjb4\customlayoutdemo.jar` in the file name field.
  - 2.3. Click Next to display Archive Builder - Step 3 of 5, as shown in Figure 3. Click the Add button to add `DiagonalLayout.class` and `DiagonalLayoutAssistant.class` to

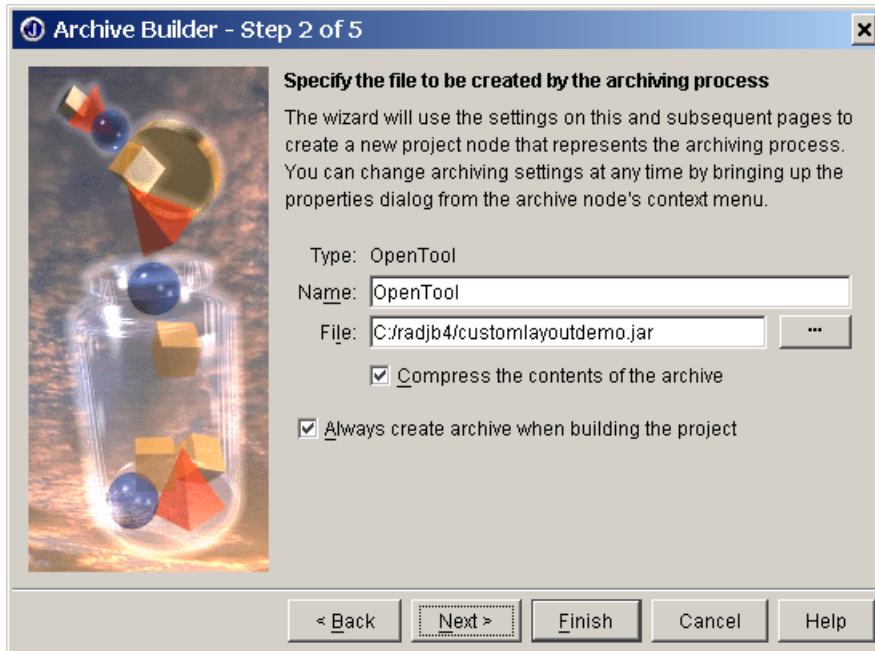
the Required classes and Resources section. Check Include required classes and resources.

- 2.4. Click Next to display Archive Builder - Step 4 of 5, as shown in Figure 4. Choose Never include classes and resources to exclude JBuilder library classes.
  - 2.5. Click Next to display Archive Builder - Step 5 of 5, as shown in Figure 4. Specify radjb4.opentools as the path for the manifest file.
  - 2.6. Click OK to close the Archive Builder.
  - 2.7. Rebuild the project to generate the archive. You can view the contents of the archive file using WinZip. It should contain three files as shown in Figure 6.
3. Move c:\radjb4\customlayoutdemo.jar to c:\jbuilder4\lib\ext. Restart JBuilder to load the layout manager to JBuilder.



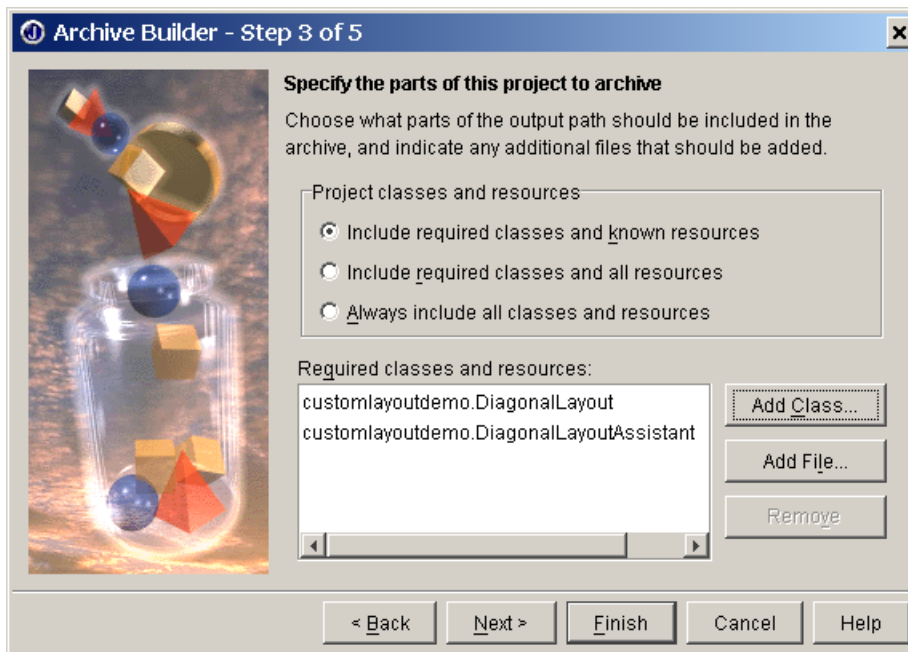
**Figure 1**

*The OpenTool prepares a special manifest file for use in the JBuilder OpenTool.*



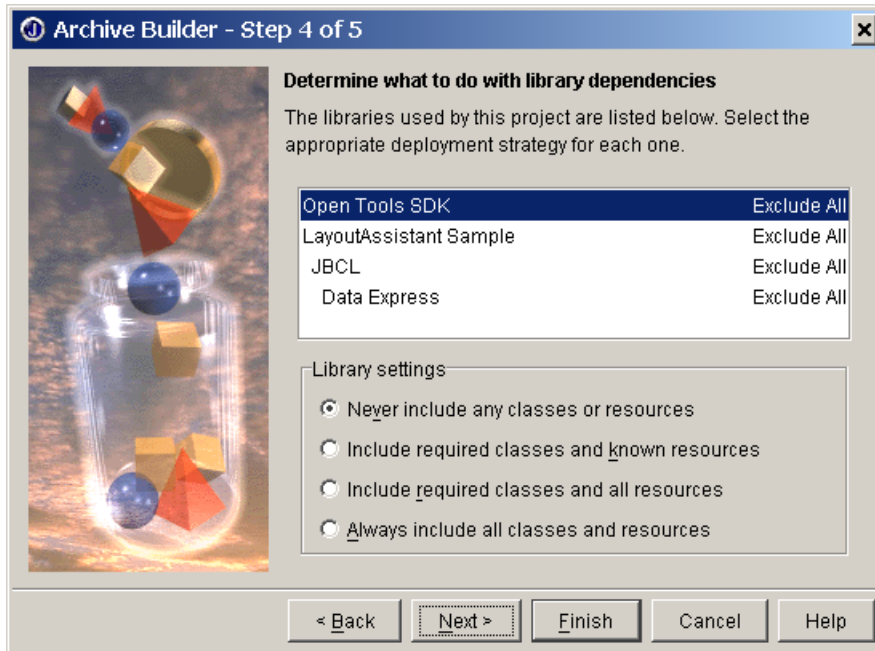
**Figure 2**

*The Archive Builder lets you specify an archive file name.*



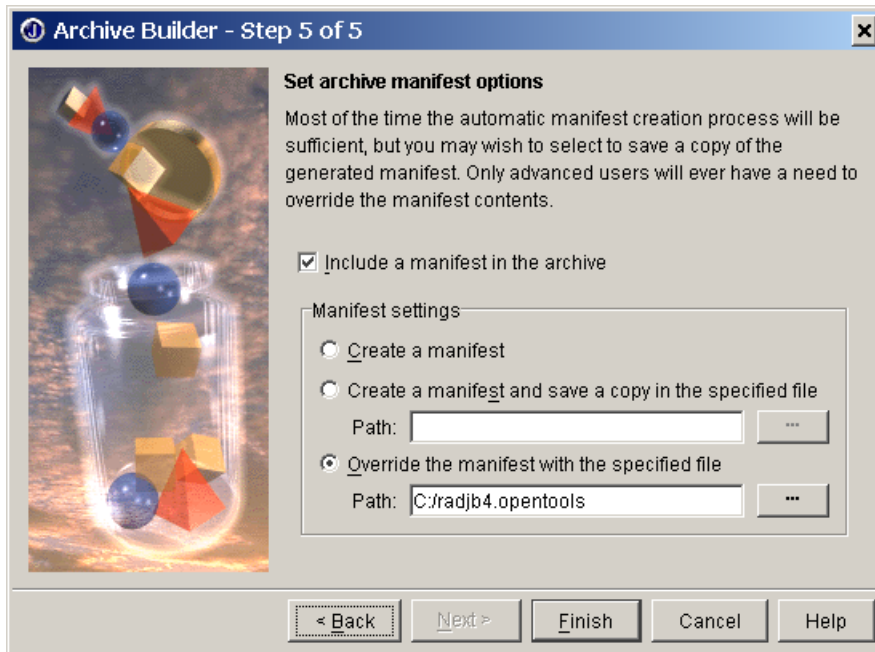
**Figure 3**

*Include only required classes and resources for the archive and specify the classes to be archived.*



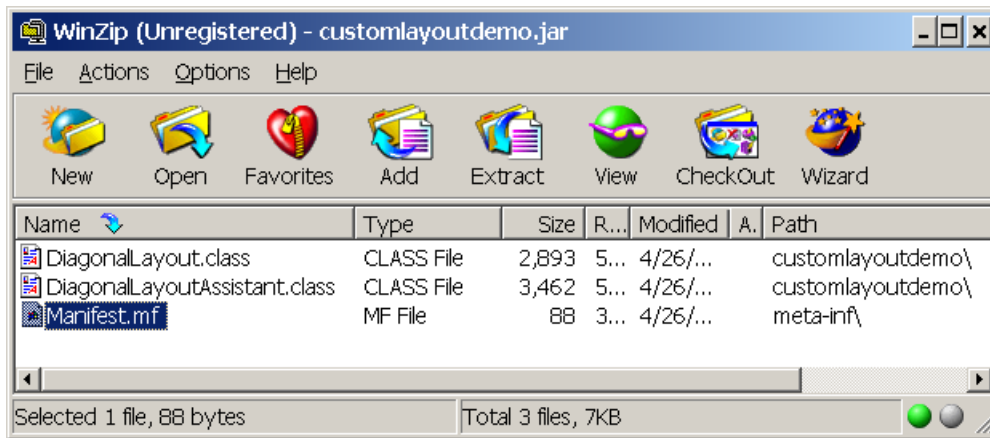
**Figure 4**

*Don't include any classes from JBuilder library.*



**Figure 5**

*Include manifest in the archive.*



**Figure 6**

*The archive file should contain DiagonalLayout.class, DiagonalLayoutAssistant.class and Manifest.mt.*