Creating Custom Code Panels Using the Eclipse Visual Editor

InstallAnywhere allows developers to include custom code panels in their installers. Custom code panels extend com.zerog.ia.api.pub.CustomCodePanel and allow installation developers to design and employ custom panels in their installers.

Using the Eclipse Visual Editor

For developers familiar with the Eclipse development environment, the Eclipse Visual Editor provides the tools necessary to quickly prototype and easily refactor custom installer panels.

These instructions are intended to provide a blueprint for the general creation of a custom code panel, but screen captures included here reflect the creation of a Choose Install Drive panel.

Sample Code

Refer to the sample code for a complete, working example of this process.

Requirements

The following tasks require you have the Eclipse development environment (version 3.2 or newer) properly configured and running the Eclipse Visual Editor. For more information on these tools, visit

http://www.eclipse.org and http://www.eclipse.org/vep/

You must also ensure that the IAClasses.zip library is on your classpath. IAClasses.zip is installed with InstallAnywhere at \IAClasses.zip.

Tip: If your custom code panel uses the ISMP Service Layer APIs, remember to add services.jar to your classpath. You can find services.jar at \resource\services\services.jar.

Task: To create a custom code panel with Eclipse Visual Editor

1. Create a new **Visual Class**.

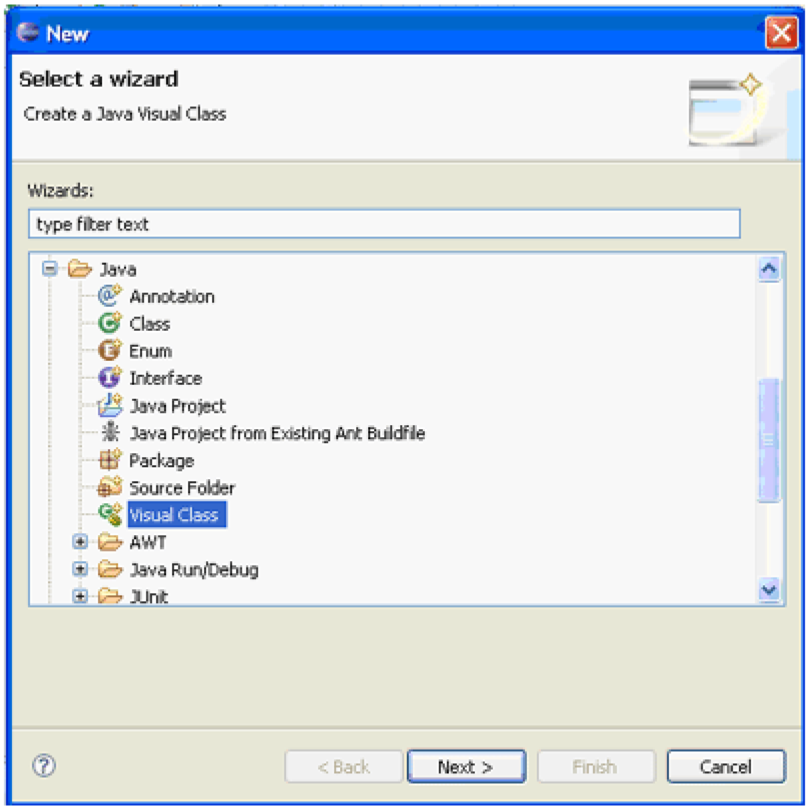
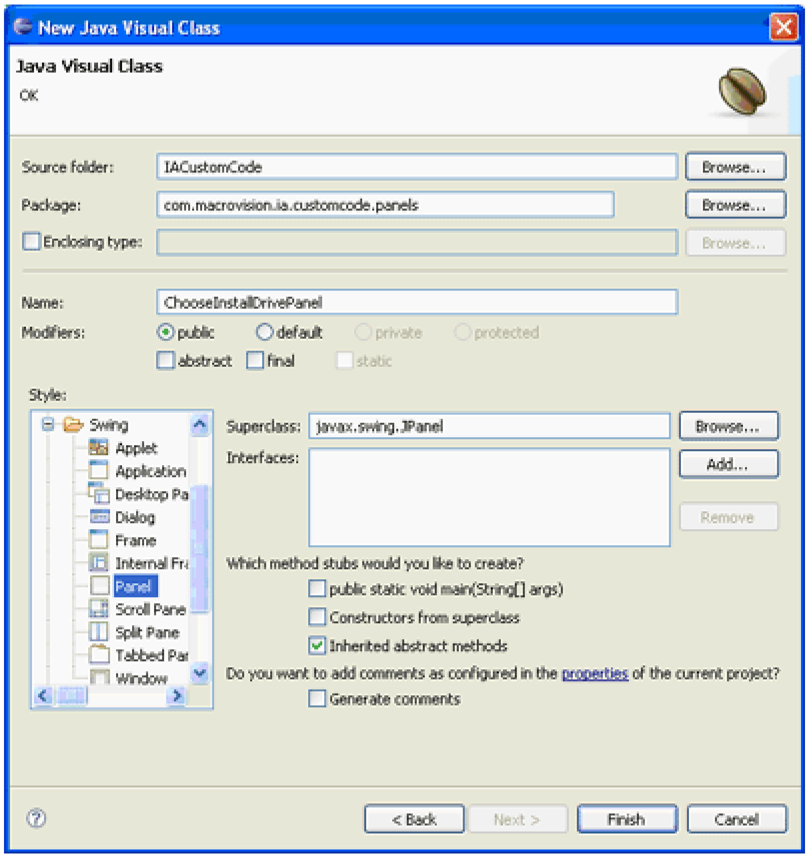


Figure 1-1: Choosing Visual Class in the New dialog box.

2. Configure the new Java visual class. In the Style list, select **Swing > Panel**.


New Java Visual Class

Java Visual Class

OK

Source folder: IACustomCode

Package: com.macrovision.ia.customcode.panels

☐ Enclosing type:

Name: ChooseInstallDrivePanel

Modifiers: [✓ public] default private protected

☐ abstract ☐ final ☐ static

Style: Swing — Applet, Application, Desktop Pa, Dialog, Frame, Internal Fra, [✓ Panel], Scroll Pane, Split Pane, Tabbed Pa, Window

Superclass: javax.swing.JPanel

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

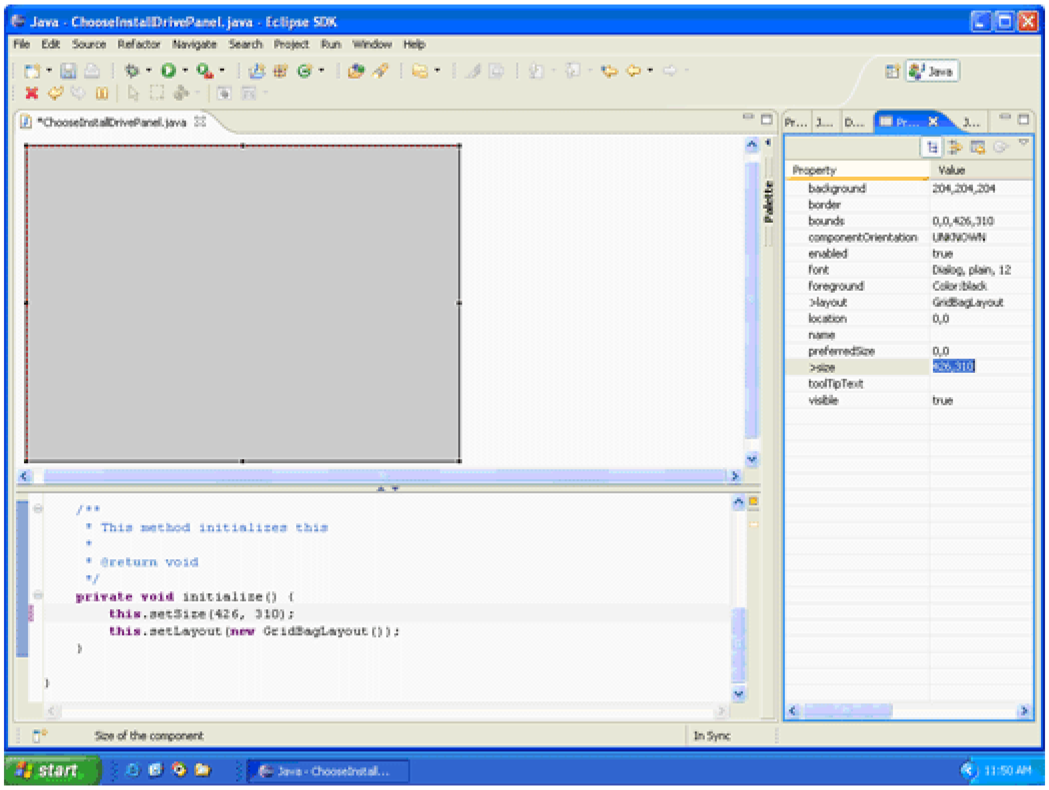
☑ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

☐ Generate comments

Figure 1-2: Configuring the new visual class.

3. Set the size of **JPanel**.


Java - ChooseInstallDrivePanel.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
*ChooseInstallDrivePanel.java
Palette
Size of the component
In Sync
start
Java - ChooseInstall...
11:50 AM

Property	Value
background	204,204,204
border	
bounds	0,0,426,310
componentOrientation	UNKNOWN
enabled	true
font	Dialog, plain, 12
foreground	Color.black
>layout	GridBagLayout
location	0,0
name	
preferredSize	0,0
>size	426,310
toolTipText	
visible	true

```
    /**
     * This method initializes this
     *
     * @return void
     */
    private void initialize() {
        this.setSize(426, 310);
        this.setLayout(new GridBagLayout());
    }

}
```

Figure 1-3: Setting the panel size.

Note: This size settings vary depending on the dimensions of your installer. (In the InstallAnywhere Advanced Designer, you set the installer dimensions on Installer UI > Look & Feel > General UI Settings.)

The recommended dimensions are determined by the following formulae:

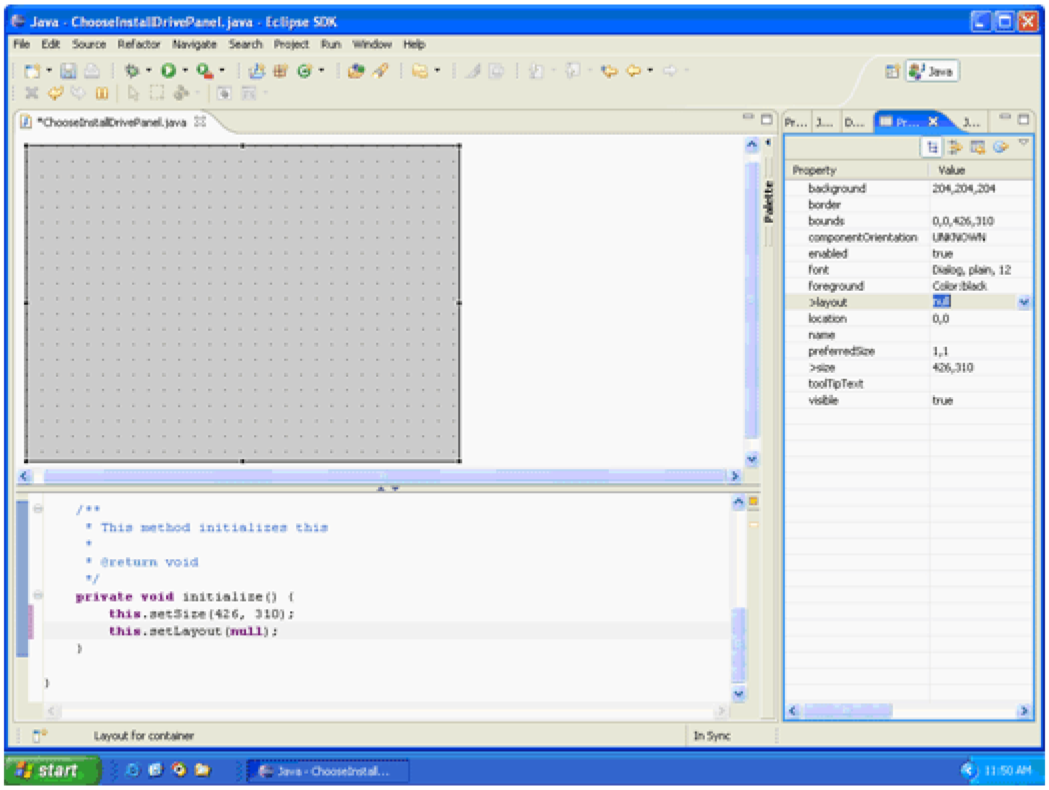
¥ PanelWidth=InstallerFrameWidth-174 pixels

¥ PanelHeight=InstallerFrameHeight-90 pixels

For example, the default Installer frame size is 600 pixels wide by 400 pixels high. Therefore, the default panel size is 426 by 310 pixels.

4. Set LayoutManager to your preferred layout manager.

Note: Setting the Layout to null gives you the most freedom when you are laying out your controls.

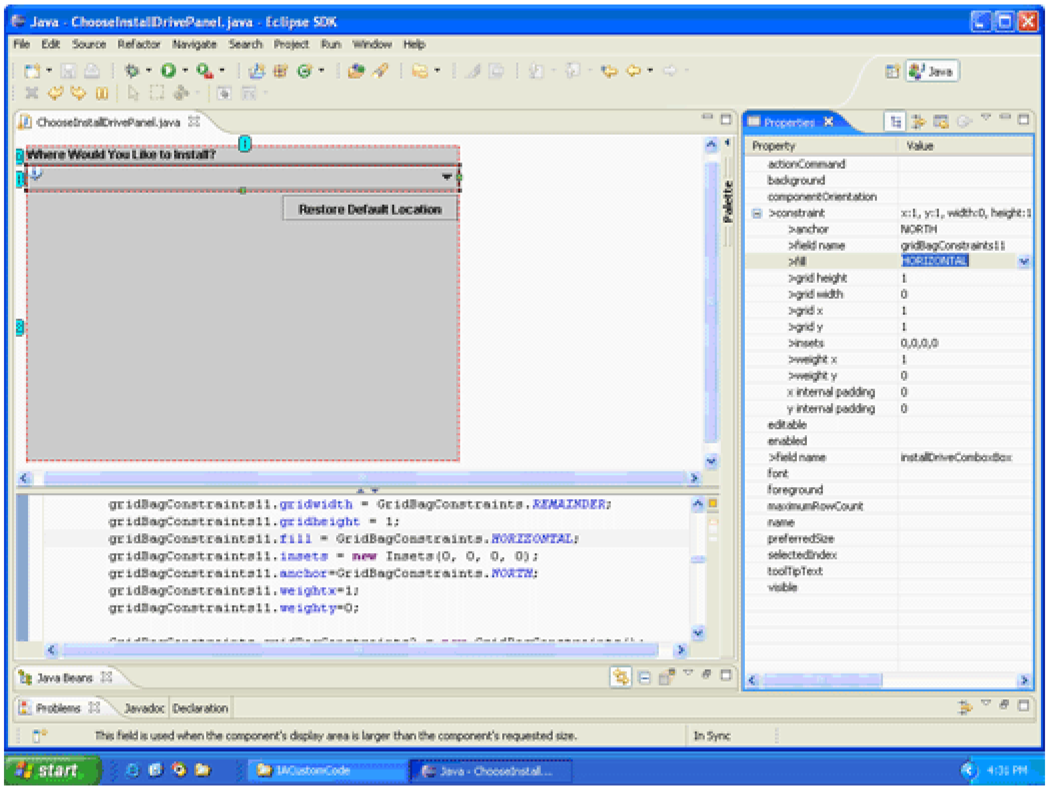

Java - ChooseInstallDrivePanel.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
*ChooseInstallDrivePanel.java
Palette
Layout for container
In Sync
start
11:50 AM

```
    /**
     * This method initializes this
     *
     * @return void
     */
    private void initialize() {
        this.setSize(426, 310);
        this.setLayout(null);
    }

}
```

Property	Value
background	204,204,204
border	
bounds	0,0,426,310
componentOrientation	UNKNOWN
enabled	true
font	Dialog, plain, 12
foreground	Color:black
>layout	null
location	0,0
name	
preferredSize	1,1
>size	426,310
toolTipText	
visible	true

Figure 1-4: Setting layout to null.

5. Layout components. For our Choose Install Drive example, we're using a **JLabel**, a **JComboBox**, and a **JButton** from the Swing Components palette.


Java - ChooseInstallDrivePanel.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
ChooseInstallDrivePanel.java
Where Would You Like to Install?
Restore Default Location
Palette
Properties
Java Beans
Problems
Javadoc
Declaration
start
IACustomCode
Java - ChooseInstall...
4:35 PM

```
gridBagConstraints11.gridwidth = GridBagConstraints.REMAINDER;
gridBagConstraints11.gridheight = 1;
gridBagConstraints11.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints11.insets = new Insets(0, 0, 0, 0);
gridBagConstraints11.anchor=GridBagConstraints.NORTH;
gridBagConstraints11.weightx=1;
gridBagConstraints11.weighty=0;
```

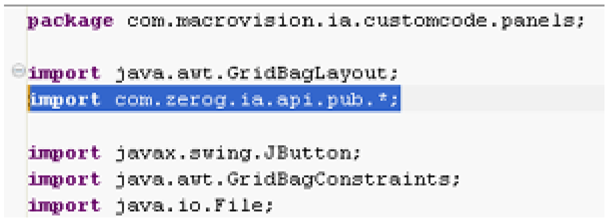
Property	Value
actionCommand	
background	
componentOrientation	
>constraint	x:1, y:1, width:0, height:1
>anchor	NORTH
>field name	gridBagConstraints11
>fill	HORIZONTAL
>grid height	1
>grid width	0
>grid x	1
>grid y	1
>insets	0,0,0,0
>weight x	1
>weight y	0
x internal padding	0
y internal padding	0
editable	
enabled	
>field name	installDriveComboBox
font	
foreground	
maximumRowCount	
name	
preferredSize	
selectedIndex	
toolTipText	
visible	

This field is used when the component's display area is larger than the component's requested size.

In Sync

Figure 1-5: Placing components on the panel.

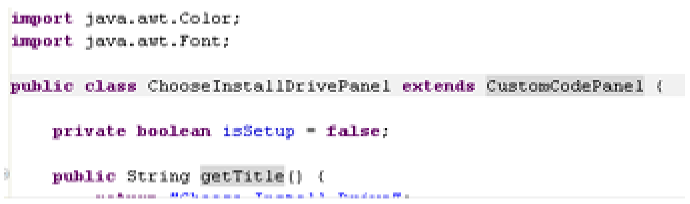
6. Add import com.zerog.ia.api.pub.* to the list of imports.



```
package com.macrovision.ia.customcode.panels;

import java.awt.GridBagLayout;
import com.zerog.ia.api.pub.*;

import javax.swing.JButton;
import java.awt.GridBagConstraints;
import java.io.File;
```

Figure 1-6: Importing com.zerog.ia.api.pub.*

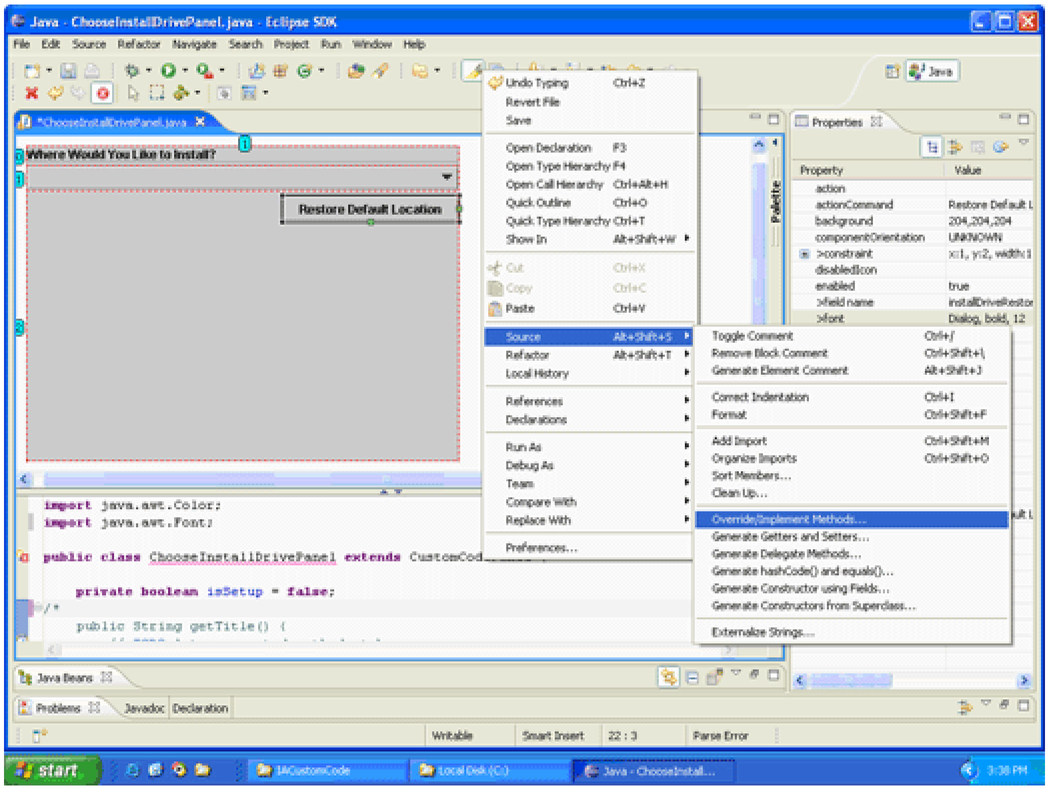
7. Make **ChooseInstallDrivePanel** a subclass of the CustomCodePanel class instead of the JPanel class. Change extends JPanel to extends CustomCodePanel.



```
import java.awt.Color;
import java.awt.Font;

public class ChooseInstallDrivePanel extends CustomCodePanel {

    private boolean isSetup = false;

    public String getTitle() {
```

Figure 1-7: Changing ChooseInstallDrivePanel.

8. Right-click **CustomCodePanel** and select **Override Methods**.

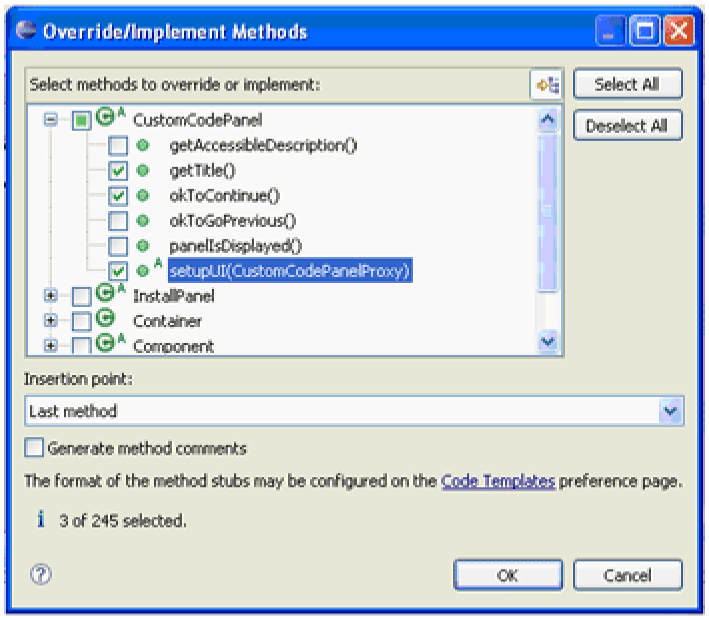

Java - ChooseInstallDrivePanel.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
*ChooseInstallDrivePanel.java
Where Would You Like to Install?
Restore Default Location
Undo Typing Ctrl+Z
Revert File
Save
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Quick Outline Ctrl+O
Quick Type Hierarchy Ctrl+T
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Source Alt+Shift+S
Refactor Alt+Shift+T
Local History
References
Declarations
Run As
Debug As
Team
Compare With
Replace With
Preferences...
Toggle Comment Ctrl+/
Remove Block Comment Ctrl+Shift+\
Generate Element Comment Alt+Shift+J
Correct Indentation Ctrl+I
Format Ctrl+Shift+F
Add Import Ctrl+Shift+M
Organize Imports Ctrl+Shift+O
Sort Members...
Clean Up...
Override/Implement Methods...
Generate Getters and Setters...
Generate Delegate Methods...
Generate hashCode() and equals()...
Generate Constructor using Fields...
Generate Constructors from Superclass...
Externalize Strings...
Properties
Property
Value
Palette
Java Beans
Problems
Javadoc
Declaration
Writable
Smart Insert
22 : 3
Parse Error
start

Property	Value
action	
actionCommand	Restore Default L
background	204,204,204
componentOrientation	UNKNOWN
>constraint	x:1, y:2, width:1
disabledIcon	
enabled	true
>field name	installDriveRestor
>font	Dialog, bold, 12

```
import java.awt.Color;
import java.awt.Font;

public class ChooseInstallDrivePanel extends CustomCod

    private boolean isSetup = false;
/*
    public String getTitle() {
```

Figure 1-8: Opening the Override/Implement Methods dialog box.

9. In the **Override/Implement Methods** dialog box, select **setupUI**, **okToContinue**, and **getTitle**. (The other methods are optional.)



Select methods to override or implement:

- [■] CustomCodePanel
  - [ ] getAccessibleDescription()
  - [✓] getTitle()
  - [✓] okToContinue()
  - [ ] okToGoPrevious()
  - [ ] panelIsDisplayed()
  - [✓] setupUI(CustomCodePanelProxy)
- [ ] InstallPanel
- [ ] Container
- [ ] Component

Insertion point: Last method

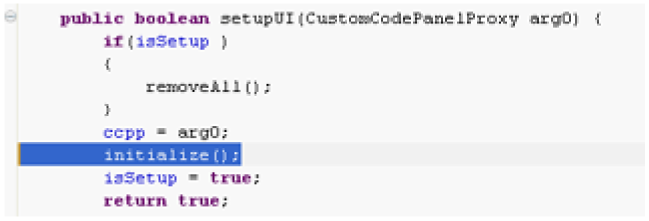
[ ] Generate method comments

The format of the method stubs may be configured on the Code Templates preference page.

3 of 245 selected.

Figure 1-9: Choosing methods to override/implement.

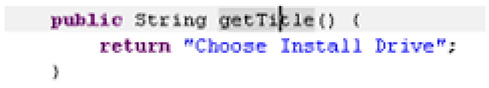
10. Move **initialize()** method call to **setupUI**, and set **setupUI** to return true.



```
    public boolean setupUI(CustomCodePanelProxy arg0) {
        if(isSetup )
        {
            removeAll();
        }
        ccpp = arg0;
        initialize();
        isSetup = true;
        return true;
```

Figure 1-10: Modifying setupUI.

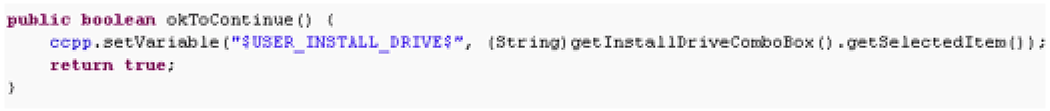
11. Implement the **getTitle()** method.



```
    public String getTitle() {
        return "Choose Install Drive";
    }
```

Figure 1-11: Implementing getTitle.

12. Implement the **okToContinue()** method.



```
public boolean okToContinue() {
    ccpp.setVariable("$USER_INSTALL_DRIVE$", (String)getInstallDriveComboBox().getSelectedItem());
    return true;
}
```

Figure 1-12: Implementing okToContinue.

13. Add a Custom Code panel to an InstallAnywhere project.

    In InstallAnywhere's Advanced Designer,

    a. Click **Choose JAR or ZIP**, browse to your JAR, and click **Open**.

    b. In the Class field, enter a fully-qualified class name. In this example, we use **com.macrovision.ia.customcode.panels.ChooseInstallDrivePanel**.

Note: On the Custom Code customizer, remember to add swing-layout-1.0.jar to the Dependencies list. You can find swing-layout-1.0.jar at \platform6\modules\ext. Choose Add jar or zip, browse to swing-layout-1.0.jar, and click Open.

14. Compile, package, and execute.

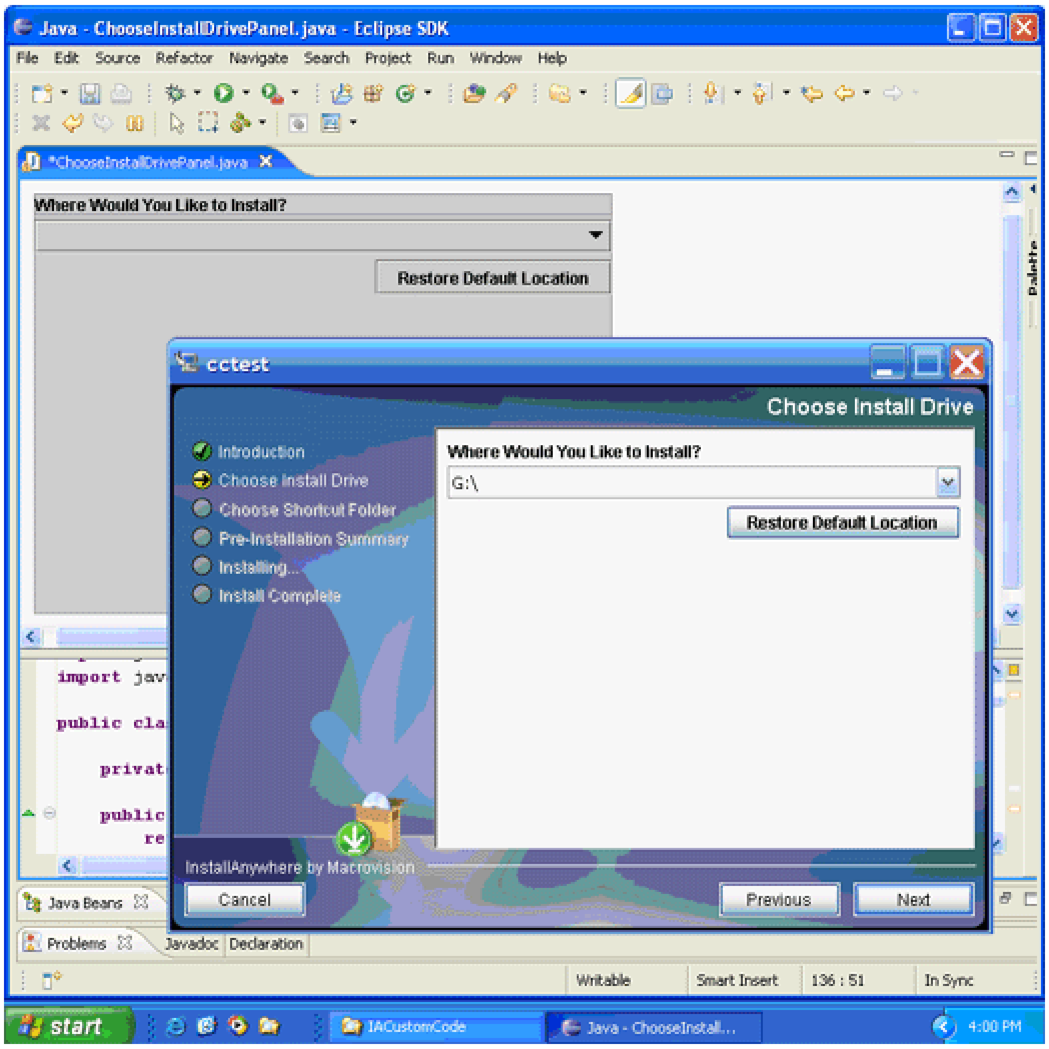


Figure 1-14: Testing the Choose Install Drive panel.

For more information

Additional information about CustomCodePanel methods, such as **getTitle**, **okToContinue**, and **setupUI**, is available as Javadocs in your InstallAnywhere folder.

Note: You can also open Javadocs from InstallAnywhere by clicking Open Javadocs on the Custom Code panel customizer.

For more information about using InstallAnywhere, open your local copy of the

InstallAnywhere Help Library or go to http://helpnet.flexearasoftware.com and click the link for InstallAnywhere.