

Writing the Adapter Code

This chapter contains the following exercises:

- Defining the Class Name
 - Generating the Code
 - Programming the Method
 - Testing the Completed Application
 - Viewing the Page Outside the Layout Painter
-

Defining the Class Name

As a last step of your layout definition, you will now define the name of the class which is the logical counterpart of your page.

To define the class name

1. In the layout tree, select the top node **page**.

You can see the default entry "DummyAdapter" for the `model` property.

2. Specify the following property:

Properties	Value
<code>model</code>	HelloWorldAdapter

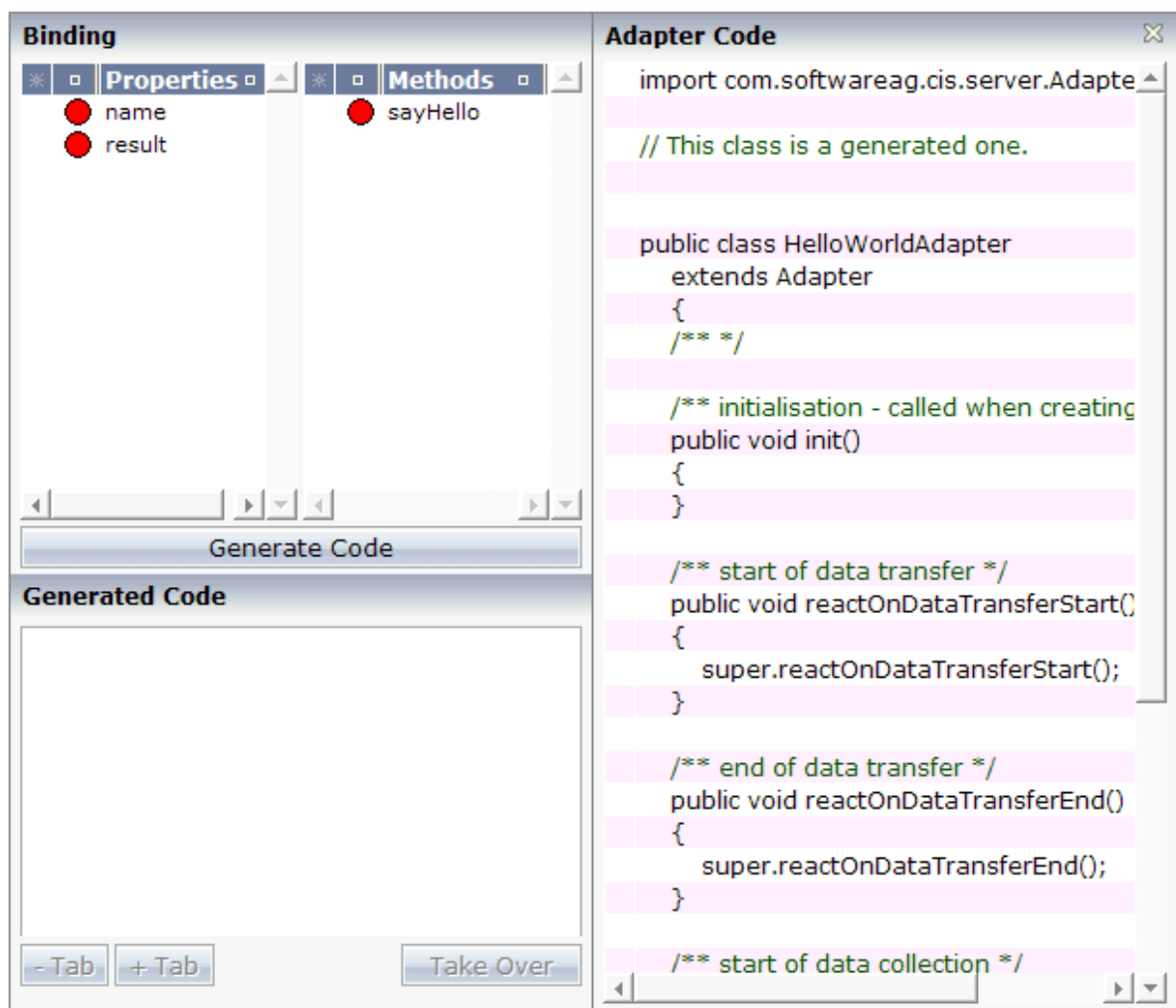
Generating the Code

You use the Code Assistant, which is part of the Layout Painter, to generate code for the referenced properties and methods.

To generate code

1. From the **Tools** tab of the Layout Painter, choose **Code Assistant**.

The following is shown in the preview area:



The properties and methods which cannot be found in the adapter code are indicated by red dots.

2. Select the property name.
3. Choose the **Generate Code** button.

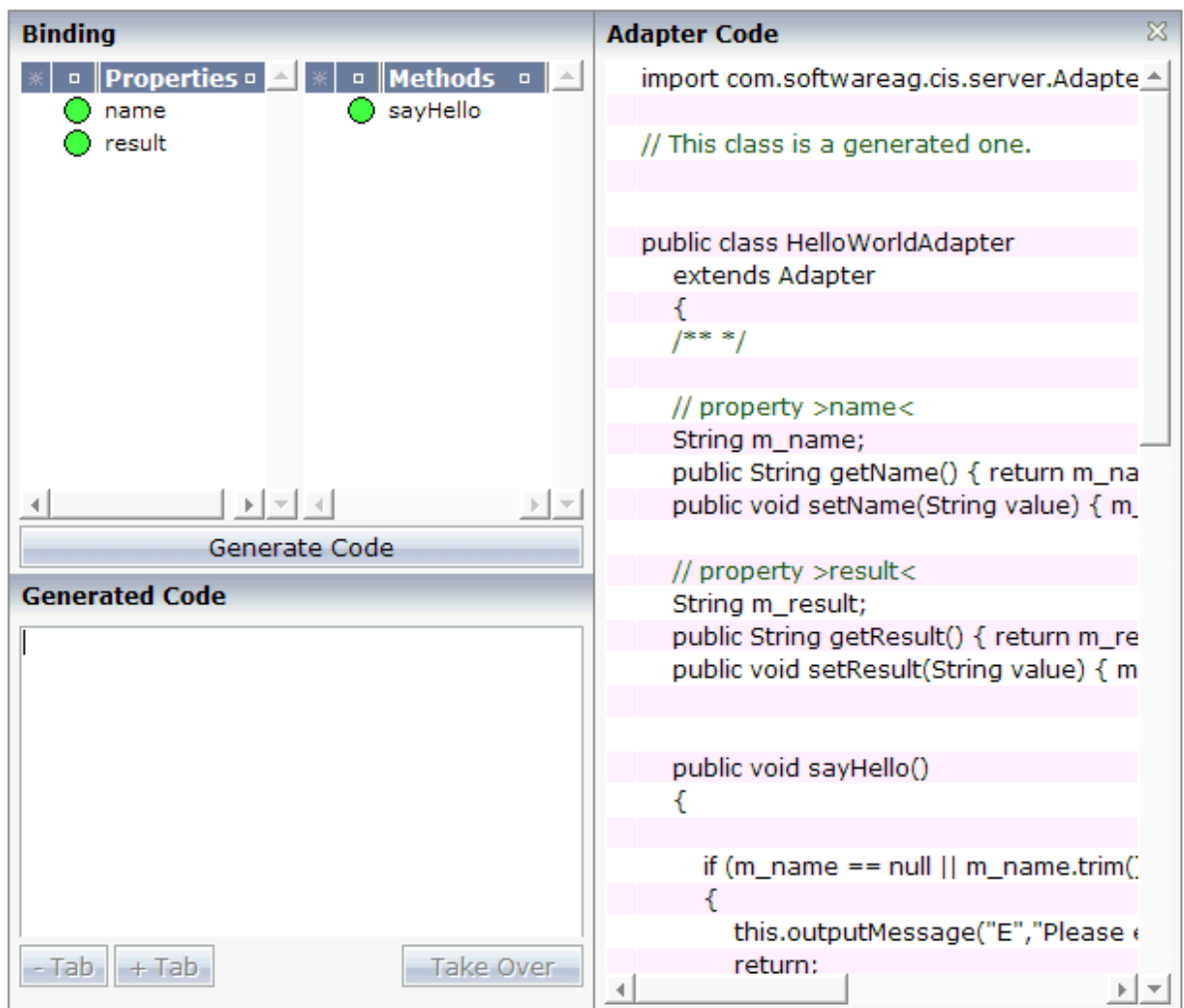
The generated code appears below this button.

4. Select the position in the adapter code at which you want to insert the code.
5. Choose the **Take Over** button to insert the code at the selected position.

The color of the dot changes from red to green. This indicates that the property's `get` and `set` methods are now available in the adapter code.

6. Repeat the above steps for the `result` property and the `sayHello` method.

When you have finished all required steps, the page should look as follows:



7. Save your changes and close the Code Assistant.

Programming the Method

Now that the framework for the `sayHello` method has been generated in the adapter code, you have to program the method itself.

▶ To program the method

1. Go to your development environment for Java.
2. Open the file `HelloWorldAdapter.java` which is located in the project directory `/cisyourfirstproject/src`.
3. Add the line which is indicated in bold:

```
/** */
public void sayHello()
{
    // TODO Auto-generated method stub
    m_result = "Hello World, " + m_name + "!";
}
```

4. Save your changes.

Testing the Completed Application

You will now check whether your application provides the desired result.

▶ To test the application

1. Go back to your layout in the Layout Painter.
2. Choose the following button which is shown at the top of the Layout Painter.



3. Enter your name and choose the **Say Hello** button.

The page should perform correctly now, and successfully "talk" to the adapter you have just created:

Input Area	
Your Name	<input type="text" value="Jo"/>
Output Area	
Result	<input type="text" value="Hello World, Jo !!!!"/>

Viewing the Page Outside the Layout Painter

The generated HTML file *helloworld.html* (which is updated each time you save your layout) can be found within the root of your application project, that is in `<installdir>/tomcat/webapps/cis/cisyourfirstproject`.

This HTML page has some prerequisites concerning the browser workplace in which it is running. Therefore, it is per se not usable as a directly accessible page but needs to be embedded into a frame providing a defined set of functions.

For viewing the page directly, enter the following URL inside your browser:

```
http://localhost:51000/cis/servlet/StartCISPage?PAGEURL=/cisyourfirstproject/helloworld.html
```

If the page is not displayed, check the following:

- URLs are case-sensitive. Double-check your input.
- Check whether the file *helloworld.html* is available in the directory *cisyourfirstproject*.

Inside the URL, a servlet `StartCISPage` is invoked. As parameter `PAGEURL`, a certain value is passed: this value is built from the application project name and the page name. Calling the servlet is the generic way of opening pages which are created with Application Designer directly inside the browser.

You have now completed this tutorial. See the remaining section of these *First Steps* for some background information.