

JavaFX

Using JavaFX Scene Builder with Java IDEs

Release 1.1

E36970-01

October 2013

JavaFX / Using JavaFX Scene Builder with Java IDEs, Release 1.1

E36970-01

Copyright © 2013 Oracle and/or its affiliates. All rights reserved.

Primary Author: Cindy Castillo

Contributor: Jerome Cambon

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

1 Using Scene Builder with NetBeans IDE

Downloading and Installing NetBeans IDE	1-1
Creating a New JavaFX FXML Project.....	1-1
Starting Scene Builder from NetBeans IDE.....	1-3
Synchronizing With the Controller Source Code	1-5
Where to Go From Here	1-6

2 Using Scene Builder with Eclipse IDE

Downloading and Installing e(fx)clipse	2-1
Creating a New JavaFX FXML Project.....	2-1
Editing an FXML File Using Scene Builder	2-6
Where to Go From Here	2-7
Using Scene Builder Samples From Eclipse IDE	2-7

3 Using Scene Builder with IntelliJ IDEA

Downloading and Installing IntelliJ IDEA.....	3-1
Creating a New JavaFX Application.....	3-1
Starting Scene Builder from IntelliJ IDE.....	3-4
Where to Go From Here	3-5
Running Scene Builder Samples from IntelliJ IDEA	3-5

Part I

About This Tutorial

JavaFX Scene Builder (Scene Builder) enables you to quickly design JavaFX application user interfaces by dragging a UI component from a library of UI components and dropping it into a content view area. The FXML code for the UI layout that you create in the tool is automatically generated in the background. To learn more about Scene Builder's features, see [JavaFX Scene Builder User Guide](#).

Scene Builder can be used as a standalone design tool, but it can also be used in conjunction with Java IDEs, so that you can use the IDE to write, build, and run the controller source code that you use with your application's user interface. Although Scene Builder is more tightly integrated with NetBeans IDE, it is also integrated with other Java IDEs described in this document. The integration enables you to start Scene Builder by opening an FXML document, run the Scene Builder samples, and generate a template for the controller source file.

This document gives information about how to configure the NetBeans, Eclipse, or IntelliJ IDEs to use with Scene Builder:

- [Using Scene Builder with NetBeans IDE](#)
- [Using Scene Builder with Eclipse IDE](#)
- [Using Scene Builder with IntelliJ IDEA](#)

Using Scene Builder with NetBeans IDE

This chapter describes how to download and install NetBeans IDE, use it to create a new JavaFX FXML project, start Scene Builder from within the IDE, and run Scene Builder sample applications.

The integration of JavaFX Scene Builder with NetBeans IDE provides optimal development workflow. It is assumed that you have already installed Scene Builder before continuing with the rest of this chapter. Go to the [JavaFX Scene Builder Installation Guide](#) for installation information.

Downloading and Installing NetBeans IDE

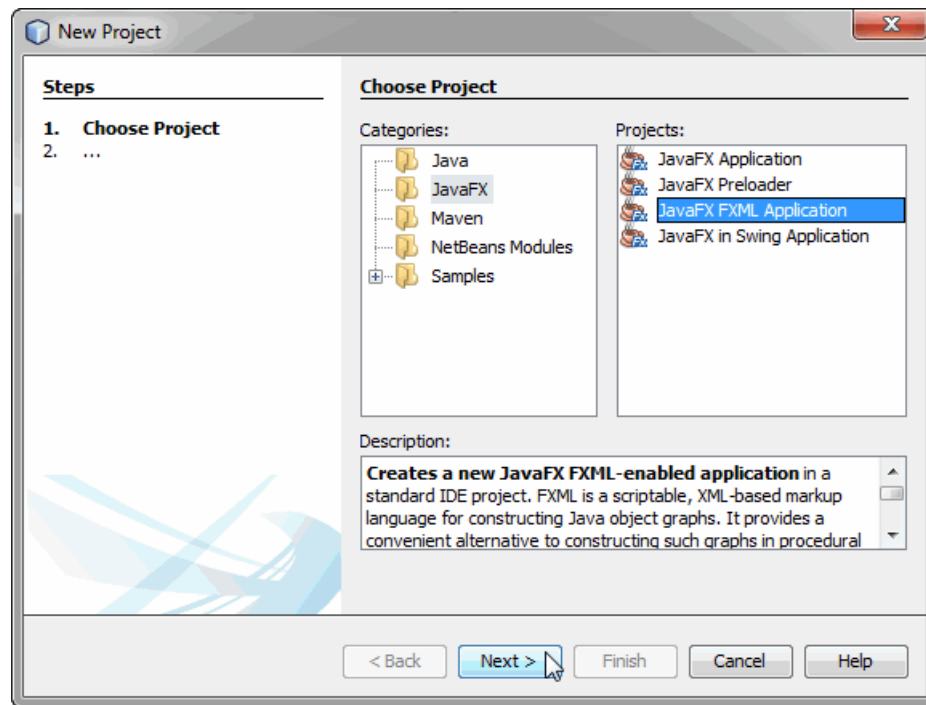
Go to <http://netbeans.org/downloads> for the latest NetBeans IDE 7.4 download and installation information.

Creating a New JavaFX FXML Project

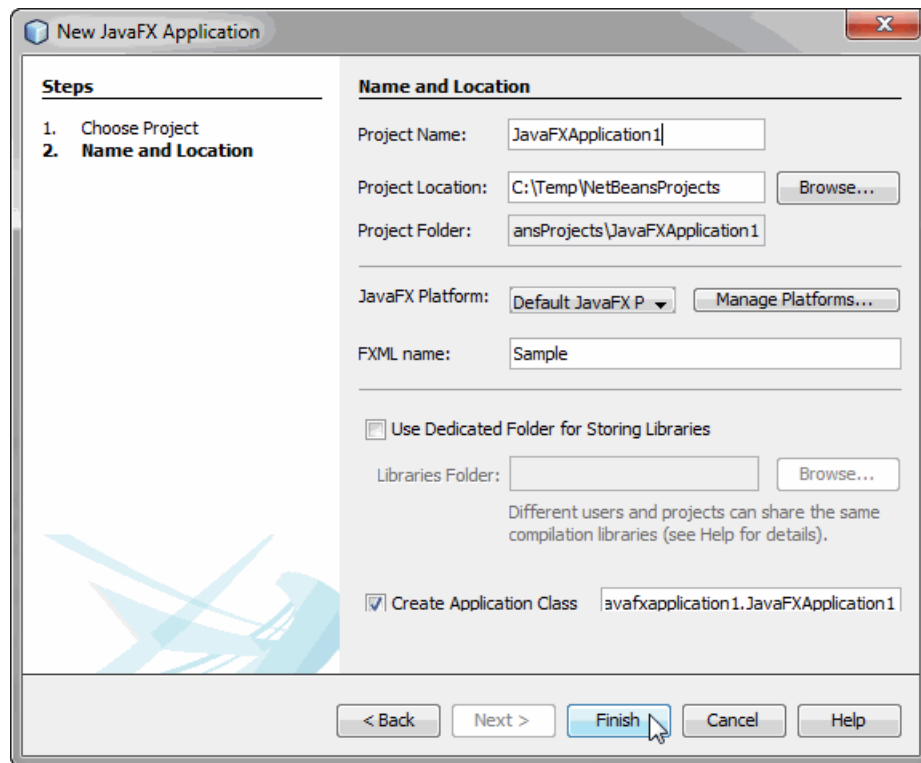
In NetBeans IDE, you use the New wizard to create a new JavaFX FXML Application, which is a JavaFX project that is based on an FXML layout. After the project is created, you can edit the FXML file using Scene Builder.

1. From the NetBeans IDE Main menu, select **File**, and then choose **New Project**.
2. In the New Project dialog box, choose the **JavaFX** category and **JavaFX FXML Application** project, as shown in [Figure 1-1](#). Click **Next**.

Figure 1–1 Create New JavaFX FXML Application Project



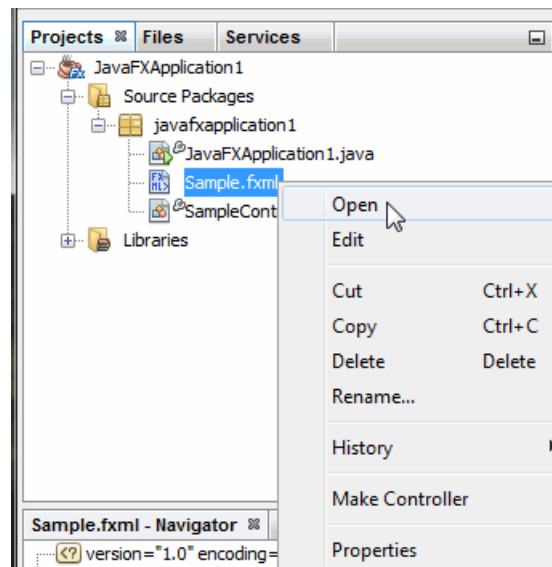
3. In the New JavaFX Application dialog box, enter the values you would like to use for the project name, project location, and FXML file name. Click **Finish** to complete the project creation. [Figure 1–2](#) gives an example with sample values.

Figure 1–2 Set the Name and Location of the New JavaFX Application

The IDE's Projects window is updated with the new JavaFX application you just created.

Starting Scene Builder from NetBeans IDE

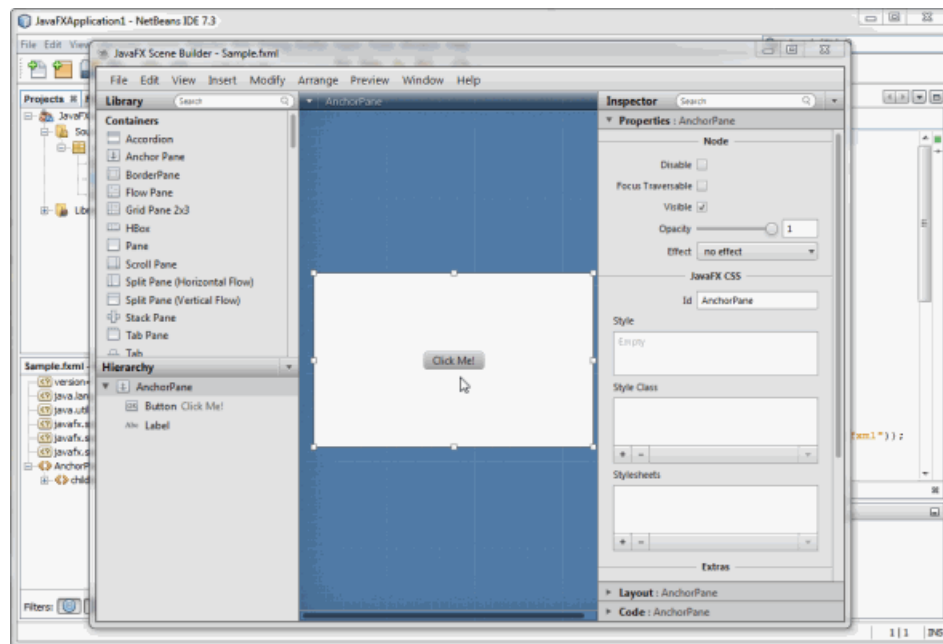
As shown in [Figure 1–3](#), you can either choose **Open** to edit the FXML file with the Scene Builder tool or choose **Edit** to edit the FXML file with the Netbeans FXML editor.

Figure 1–3 Open FXML File from NetBeans IDE

When you right-click the node for the FXML file and choose **Open**, then NetBeans IDE automatically locates the latest installed Scene Builder on your system and the Scene Builder window appears on top of the Netbeans window, as shown in [Figure 1–4](#).

If you want to use a different installation of Scene Builder or if NetBeans cannot locate the installed Scene Builder, use the following steps to set the location of the Scene Builder installation you would like to use with NetBeans IDE.

1. From the **Main** menu, select **Tools** and choose **Options**.
On the Mac OS platform, select **NetBeans** and choose **Preferences** from the **Main** menu.
2. In the Options window, click Java and then the **JavaFX** tab.
3. Specify the location of the Scene Builder installation folder and then click **OK**.

Figure 1–4 Open FXML File in Scene Builder (Click image to enlarge.)

To open the FXML file in the NetBeans FXML editor, from the **File** menu, right-click the file's node and choose **Edit**. The FXML source file contents are shown in the IDE's source editor window, as shown in [Figure 1–5](#).

Figure 1–5 Open FXML File in NetBeans FXML Editor

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2
3  <?import java.lang.*?>
4  <?import java.util.*?>
5  <?import javafx.scene.*?>
6  <?import javafx.scene.control.*?>
7  <?import javafx.scene.layout.*?>
8
9  <AnchorPane id="AnchorPane" prefHeight="200" prefWidth="320" :
10     <children>
11         <Button layoutX="126" layoutY="90" text="Click Me!" on
12         <Label layoutX="126" layoutY="120" minHeight="16" minW
13     </children>
14 </AnchorPane>
15

```

Synchronizing With the Controller Source Code

The NetBeans IDE's Make Controller feature allows you to synchronize the modifications you make in the FXML file that is currently opened in Scene Builder and the controller source code opened in NetBeans IDE. To illustrate this feature, do the following:

1. In Scene Builder, drag a Button control from the Library panel to the Control panel.
2. In the Code panel, assign a new value for the new button's `fx:id` field and a new method name for the `onAction` method.
3. Select **File** in the main menu and then **Save**.
4. In NetBeans IDE 7.4 or later, right click the project node for the FXML file that you just edited and select **Edit** from the contextual menu.
5. From the main menu, select **Source** and then **Make Controller**.
The `@FXML` private variable and the new `onAction` method for the button you just added in Scene Builder are created in the controller source file. Use the Make Controller command if you delete an element in the Control panel or update an `fx:id` value or a method name in Scene Builder.

Where to Go From Here

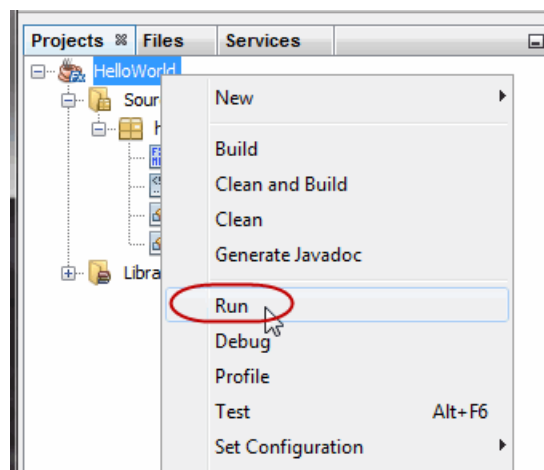
Now that you are familiar with the integration between the Scene Builder tool and NetBeans IDE, look at [JavaFX Scene Builder User Guide](#) to learn more about the available Scene Builder features. Use the steps described in [Getting Started with JavaFX Scene Builder](#) to create a simple issue tracking application.

You can also try the sample applications provided with the Scene Builder release. These samples are Netbeans projects, which means they can be compiled and ran directly after being opened in the NetBeans IDE. You can download the Scene Builder samples from

<http://www.oracle.com/technetwork/java/javafx/downloads/index.html>.

Right-click the project's node in the IDE's Project window and choose **Run**, as shown in [Figure 1-6](#).

Figure 1-6 Run Scene Builder Sample Application



Using Scene Builder with Eclipse IDE

This chapter describes how to download and install the e(fx)clipse tool, which enables you to create a new JavaFX FXML project using Eclipse IDE, start Scene Builder from within the IDE, and run Scene Builder sample applications.

The JavaFX integration with the Eclipse IDE is provided with the e(fx)clipse tool. It is assumed that you have already installed Scene Builder before continuing with the rest of this section. Go to the [JavaFX Scene Builder Installation Guide](#) for installation information.

Downloading and Installing e(fx)clipse

Go to <http://www.eclipse.org/efxclipse/install.html> for information about the available options to download and install the e(fx)clipse tool to use with your JavaFX Scene Builder installation.

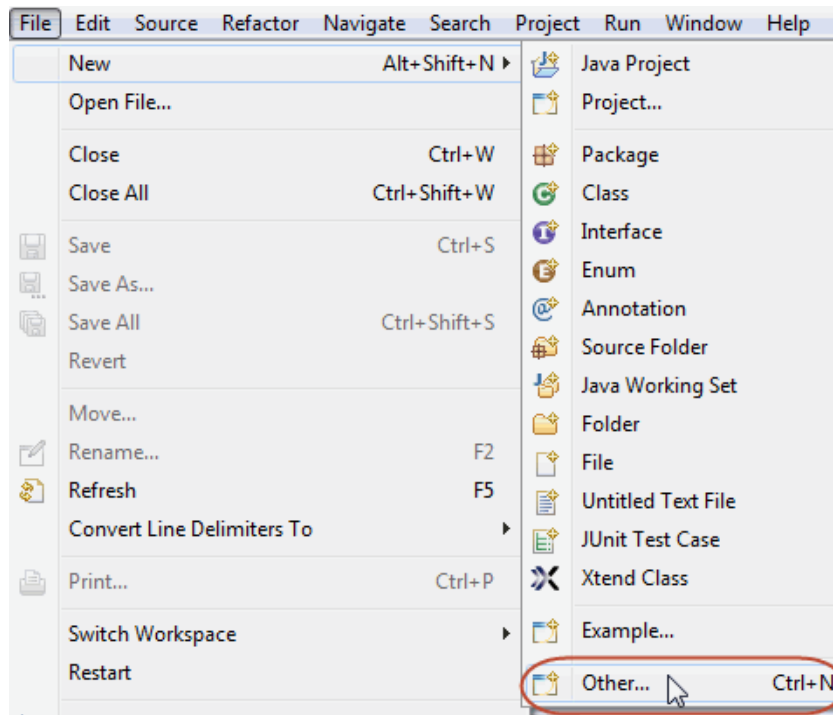
Note: The information in this chapter is based on the pre-configured distribution of Eclipse 4.2.2. SDK with e(fx)clipse 0.8.1

Creating a New JavaFX FXML Project

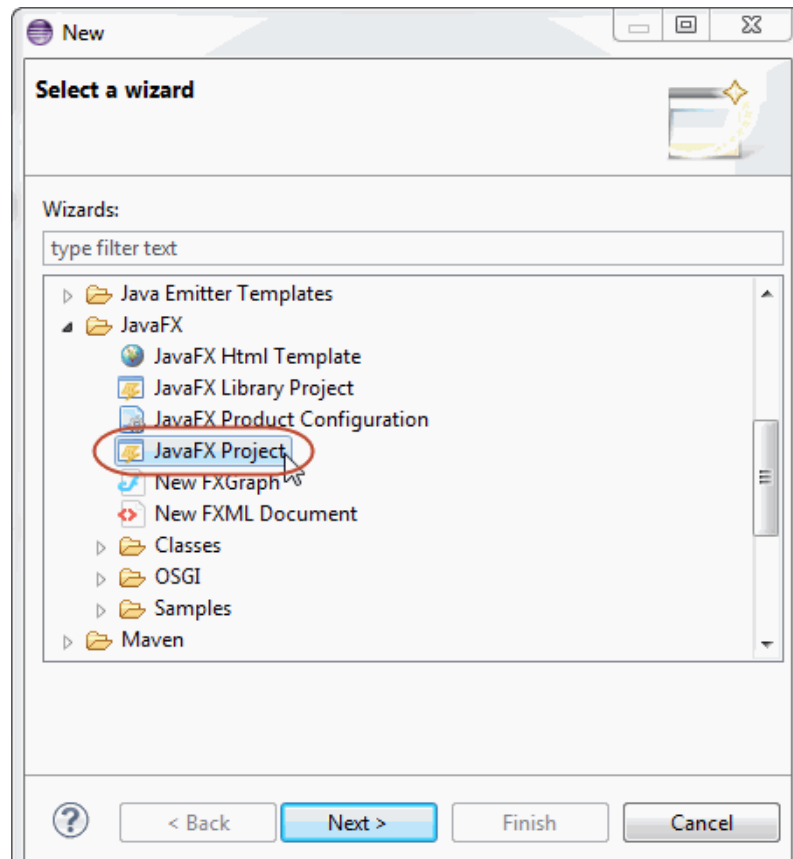
To create a new JavaFX FXML project using Eclipse, you must first create a new JavaFX project and then add a new FXML document to that JavaFX project.

1. From the Eclipse IDE Main menu, select **File**, choose **New**, and then **Other**, as shown in [Figure 2-1](#).

Figure 2-1 Create a New File

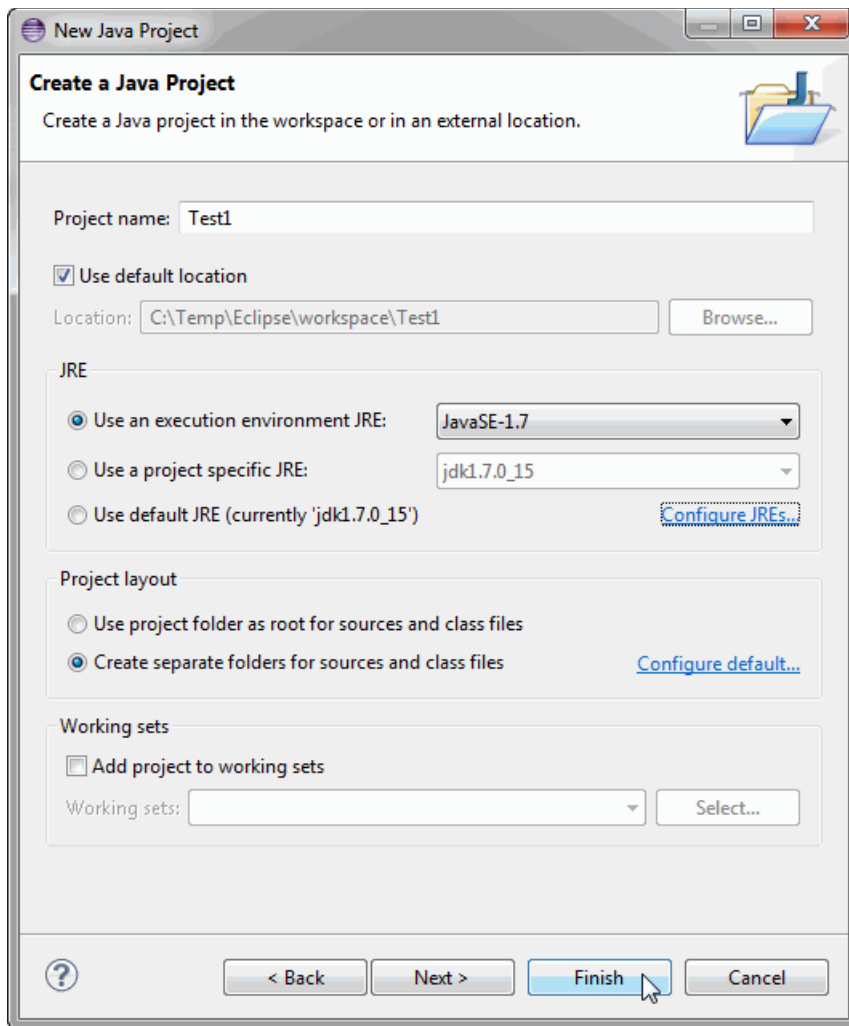


2. From the New dialog box, expand the JavaFX folder, and select the **JavaFX Project** wizard, as shown in [Figure 2-2](#).

Figure 2–2 Choose the JavaFX Project Wizard

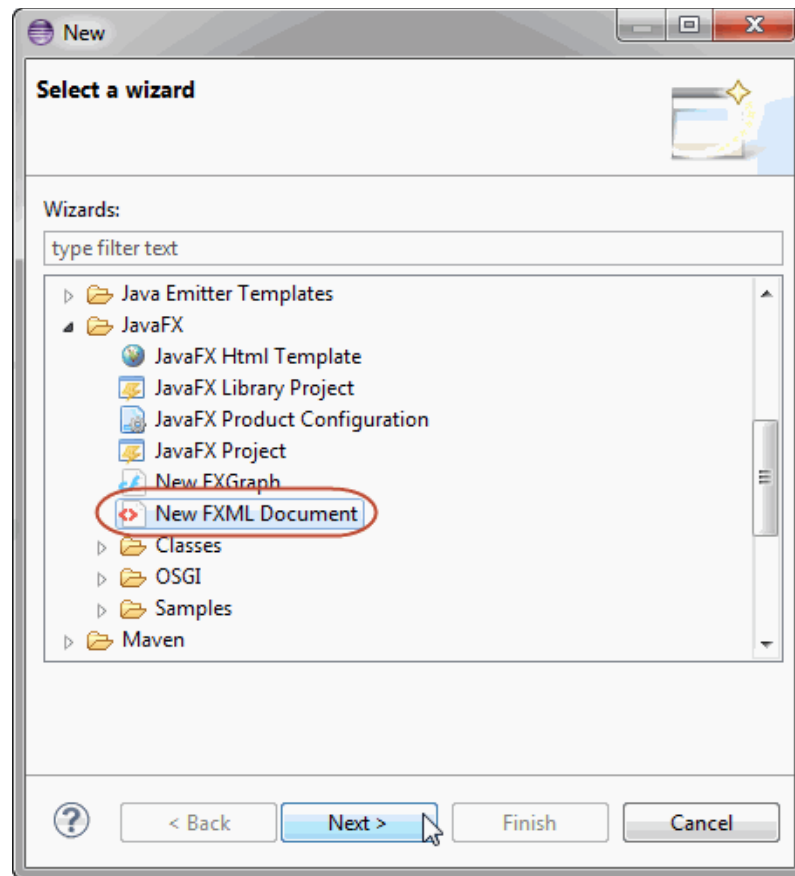
3. Enter the project name, specify the JRE to use, and select the Project layout option. Click **Finish** to complete the project creation, as shown in [Figure 2–3](#).

Figure 2–3 Create a New Java Project

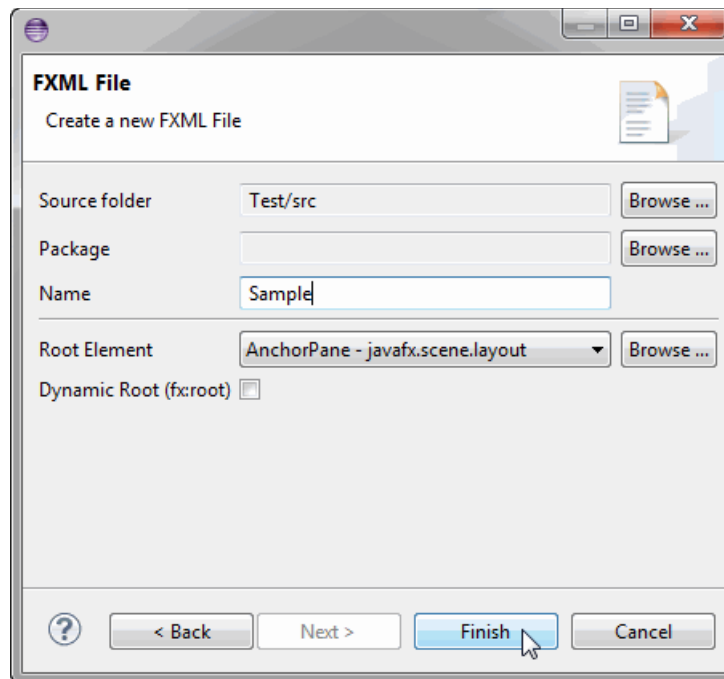


4. Now create a new FXML document
 - a. Right-click the **Test** folder in the Package Explorer.
 - b. Select **New** and then **Other**.
 - c. Expand the JavaFX folder in the New dialog box and then choose **New FXML Document**, as shown in [Figure 2–4](#). Click **Next**

Figure 2–4 Create a New FXML Document



5. In the FXML File dialog box, complete the creation of the new FXML document using the following steps:
 - a. If not automatically set to test/src, specify the Source folder value by clicking **Browse**. Navigate to the src folder of the JavaFX project that you just created in the previous steps. Leave the Package field as is.
 - b. In the Name field, enter `Sample` for the FXML file name, as shown in [Figure 2–5](#).
 - c. If not already set, specify the FXML document's root element by clicking **Browse** and selecting an item from the Find Preloader dialog box. The default value shown is `AnchorPane`.
 - d. Leave Dynamic Root unselected. It is for advanced users who want to manage custom types.
 - e. Click **Finish** to close the dialog box.
You can now complete the creation of your JavaFX FXML layout

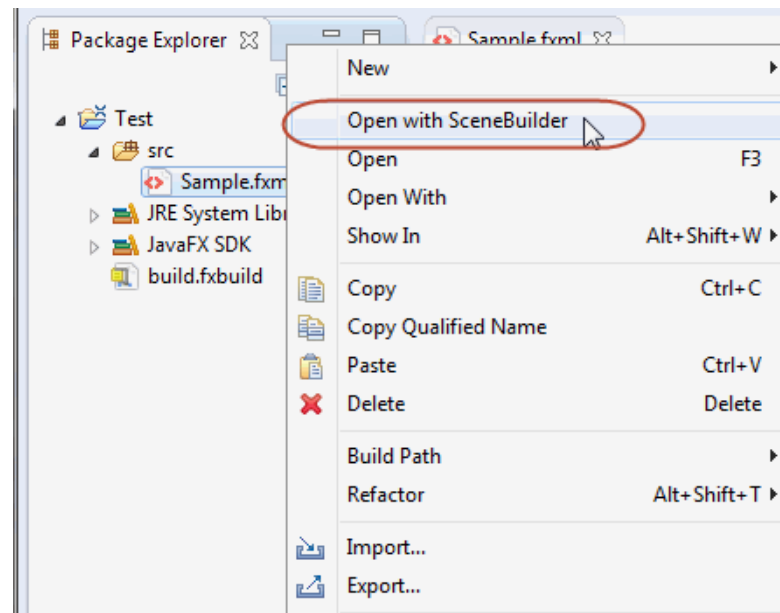
Figure 2-5 Finish Creating a New FXML File

Editing an FXML File Using Scene Builder

You can edit an FXML file using the Eclipse FXML editor or by opening the file using the JavaFX Scene Builder tool:

1. In the IDE's Package Explorer tab, expand the Test and src folders.
2. Right-click the `Sample.fxml` file and select **Open with Scene Builder**, as shown in [Figure 2-6](#).

A separate window for Scene Builder is opened and displays the `Sample.fxml` layout in the Content panel area. You may have to increase the size of the default AnchorPane that appears in the Content panel. You may start to add controls.

Figure 2–6 Open Sample.fxml File in Scene Builder

3. Use [JavaFX Scene Builder User Guide](#) to learn more about the available Scene Builder features and [Getting Started with JavaFX Scene Builder](#) to create a simple issue tracking application.

Where to Go From Here

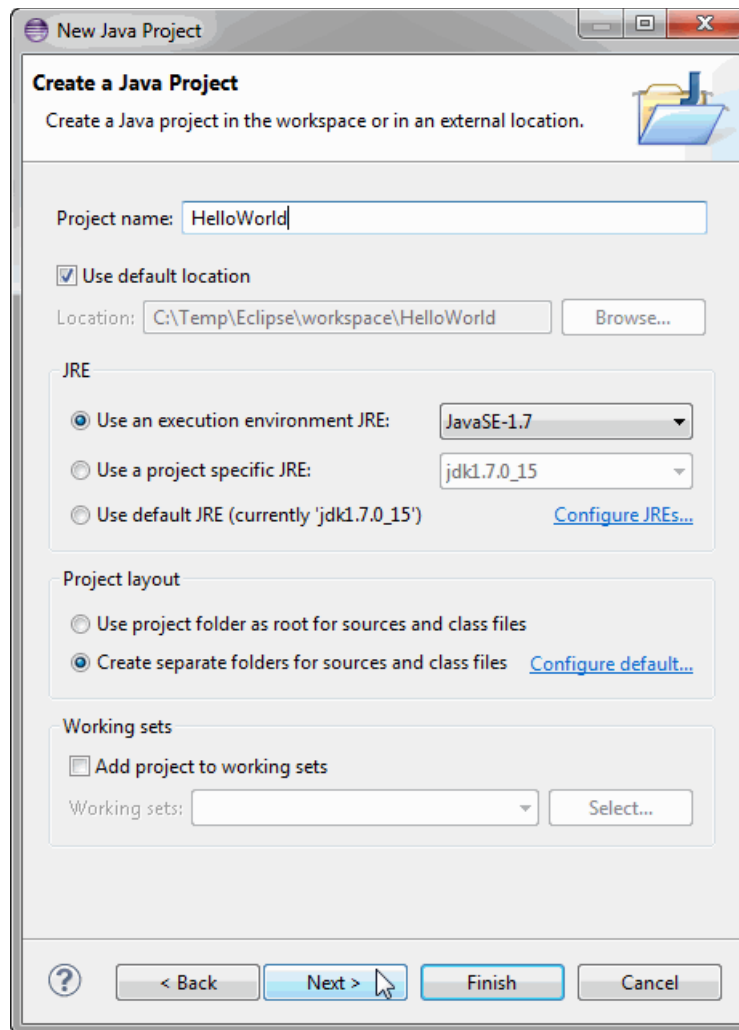
Now that you are familiar with the integration between the Scene Builder tool and Eclipse IDE, look at [JavaFX Scene Builder User Guide](#) to learn more about the available Scene Builder features. Follow the steps described in [Getting Started with JavaFX Scene Builder](#) to create a simple issue tracking application.

You can also try the sample applications provided with the Scene Builder release. The samples are Netbeans projects, so use the following information to learn how to use a sample using Eclipse IDE.

Using Scene Builder Samples From Eclipse IDE

Use the following information to use the Scene Builder samples. The HelloWorld application is used for illustrative purpose.

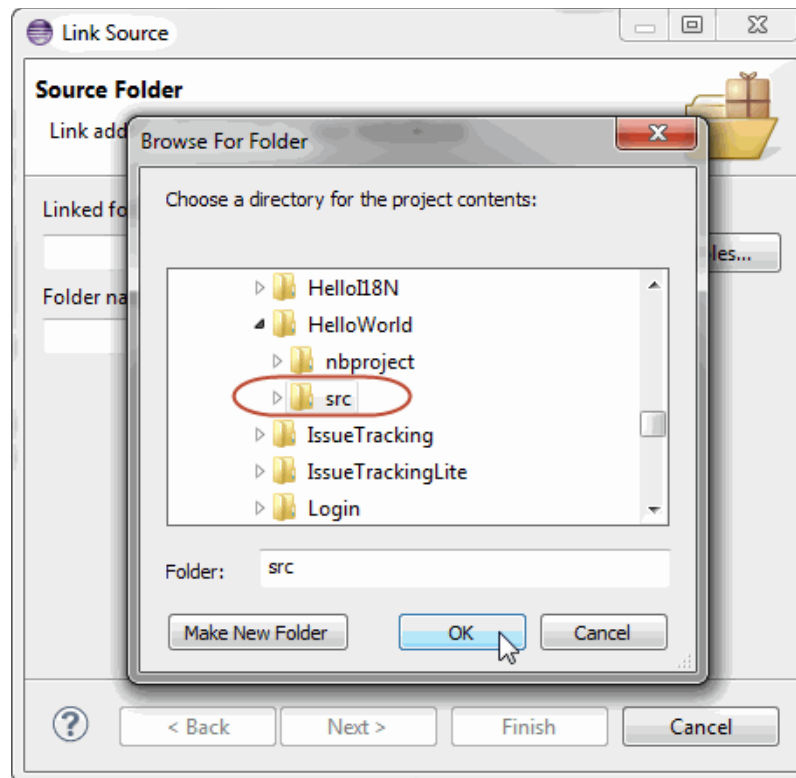
1. Ensure that you have downloaded JavaFX Scene Builder Samples bundle from <http://www.oracle.com/technetwork/java/javafx/downloads/index.html> and extracted its contents.
2. Create a new JavaFX Project by selecting **File**, and then choosing **New** and then **Other**.
3. In the New wizard, expand the JavaFX folder and choose **JavaFX Project**. Click **Next**.
4. In the New Java Project dialog box, enter HelloWorld for the **Project Name**, as shown in [Figure 2–7](#), and click **Next**.

Figure 2–7 Create a JavaFX Project for the HelloWorld Sample

5. In the Details section of the Java Settings dialog box, click **Link additional source** option. The Link Source dialog box is displayed.
6. Click **Browse** and navigate to the location of the src folder for the Scene Builder HelloWorld sample application and click **OK**, as shown in [Figure 2–8](#).

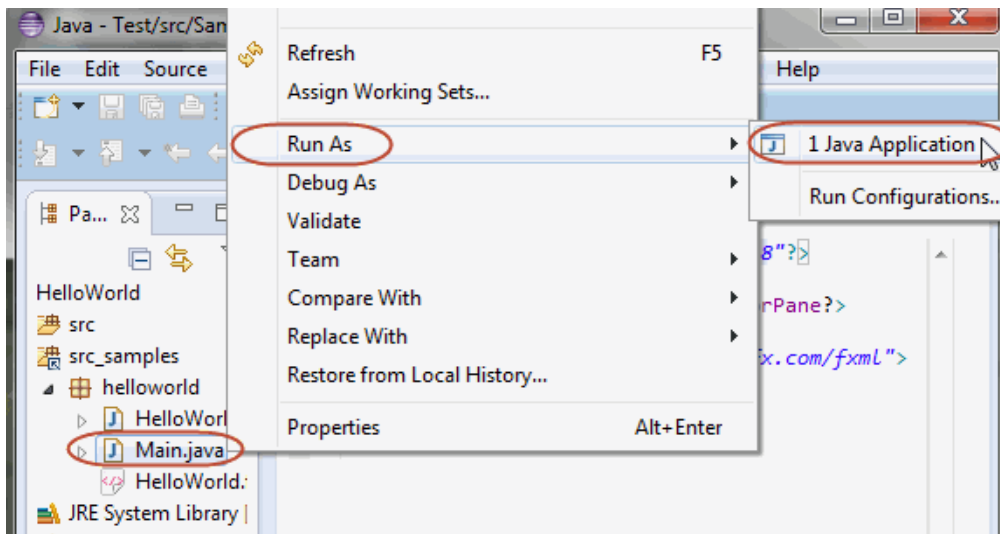
Note: Ignore the error “The folder is already a source folder.” because the Folder name field is automatically filled in

Figure 2–8 Open the HelloWorld Sample's src Folder



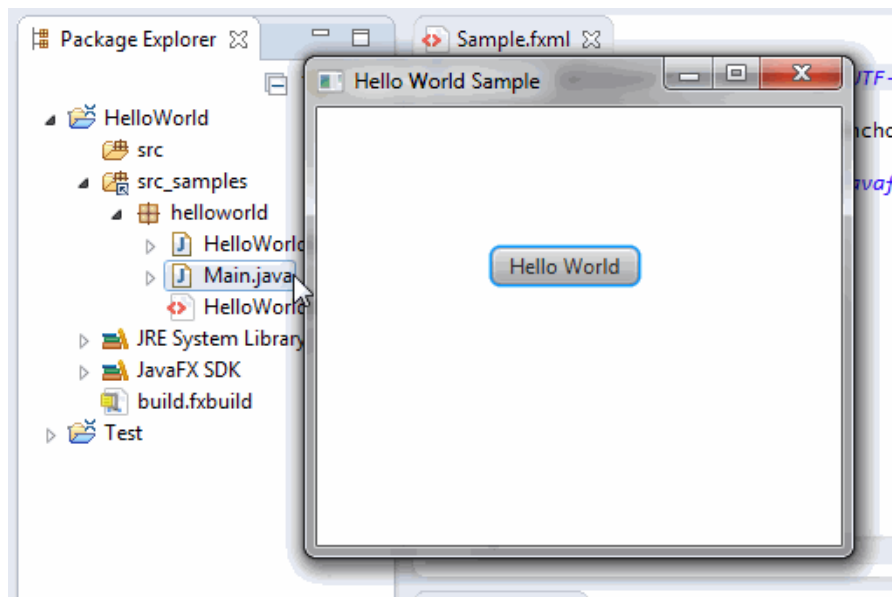
7. Change the default value in the Folder name text field to `src_samples`, or any folder name of your choice, and click **Finish**.
8. Back in the New Java Project wizard, click **Finish**.
9. Run the HelloWorld sample, as shown in [Figure 2–9](#):
 - a. In the Project Explorer window, expand the `src_samples` and `helloworld` folders for the HelloWorld project.
 - b. Right-click the node for the `Main.java` source file.
 - c. Choose **Run As** and then **Java Application**.

Figure 2–9 Run Scene Builder HelloWorld Sample



The HelloWorld application is displayed, as shown in Figure 2–10.

Figure 2–10 Hello World Sample Running



Using Scene Builder with IntelliJ IDEA

This chapter gives information on how to download and use the JavaFX 2 plugin that provides the JavaFX integration with the IntelliJ IDEA 12.1.

The JavaFX 2 plugin enables you to create a new JavaFX Application project, start Scene Builder from within the IDE, and run the Scene Builder HelloWorld sample application. It is assumed that you have already installed Scene Builder before continuing with the rest of this chapter. Go to the [JavaFX Scene Builder Installation Guide](#) for installation information.

Downloading and Installing IntelliJ IDEA

Use the information from <http://www.jetbrains.com/idea/download/> to download and install the latest IntelliJ IDEA 12.1 version that includes the JavaFX 2 plugin.

Creating a New JavaFX Application

The JavaFX 2 plugin for IntelliJ IDEA 12.1 enables you to create a new JavaFX application template.

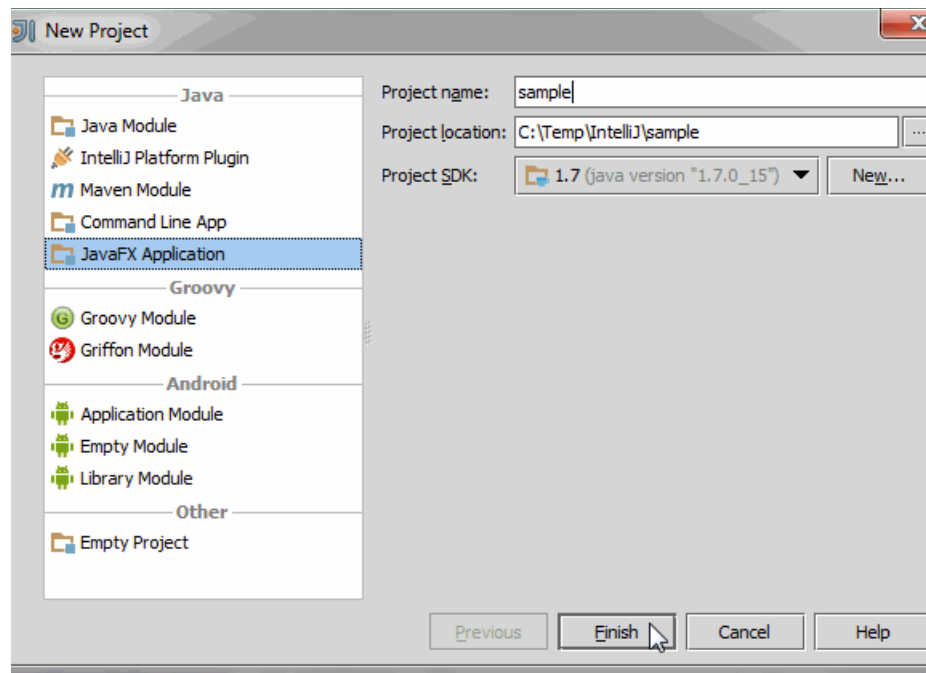
1. From the IntelliJ IDEA welcome window, click the **Create New Project**, as shown in [Figure 3-1](#).

Figure 3–1 Select Create New Project in the Welcome Window

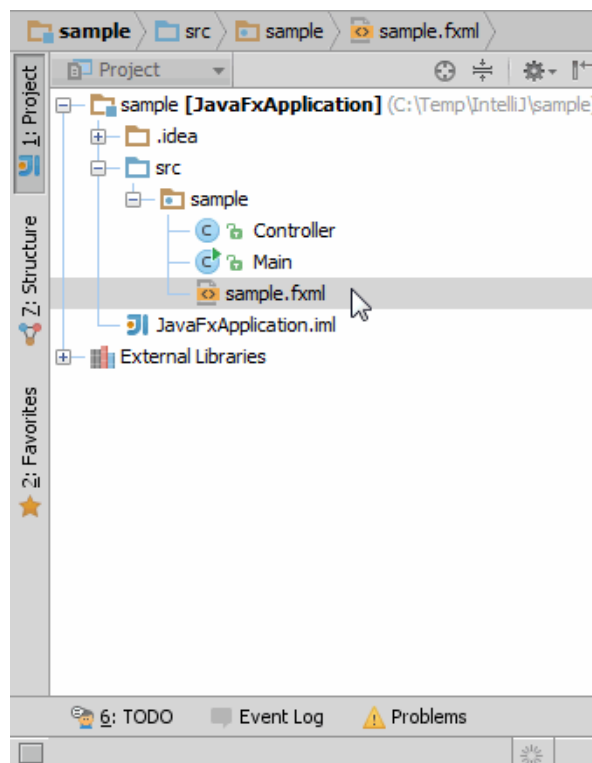


2. In the New Project dialog box, select **JavaFX Application**, specify the project information, as shown in [Figure 3–2](#), and click **Finish**.

The node for a new JavaFX application template is added to the Project tab in the main window.

Figure 3–2 Specify the Project Information for the New JavaFX Application Project

3. Expand the nodes for the project, as shown in [Figure 3–3](#).

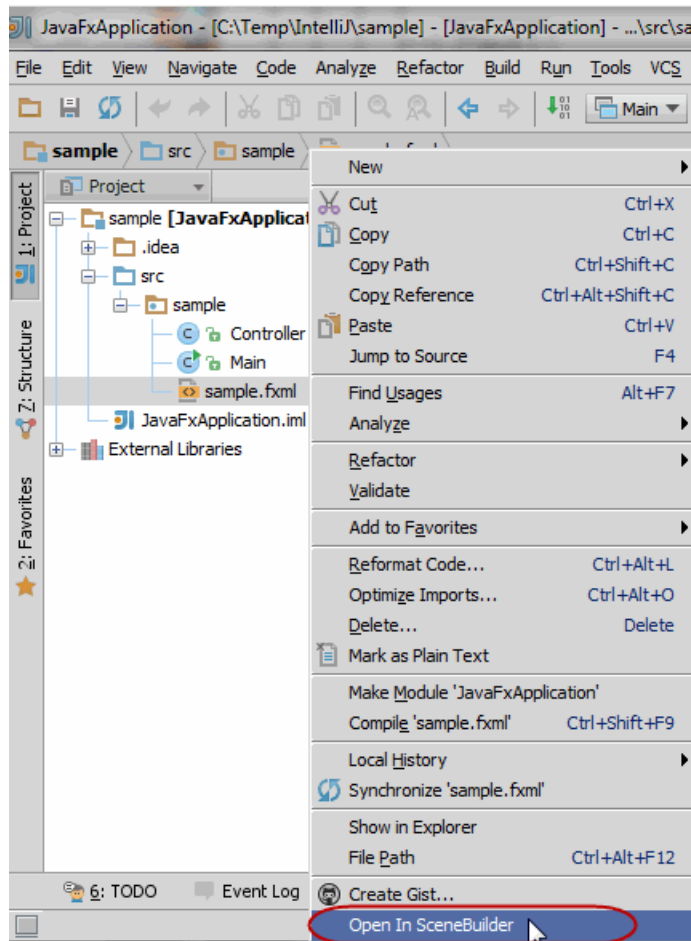
Figure 3–3 Expand the Nodes in the Project Window

Starting Scene Builder from IntelliJ IDE

Included in a newly created JavaFX Application template is the `sample.fxml`, which you can open and edit using Scene Builder.

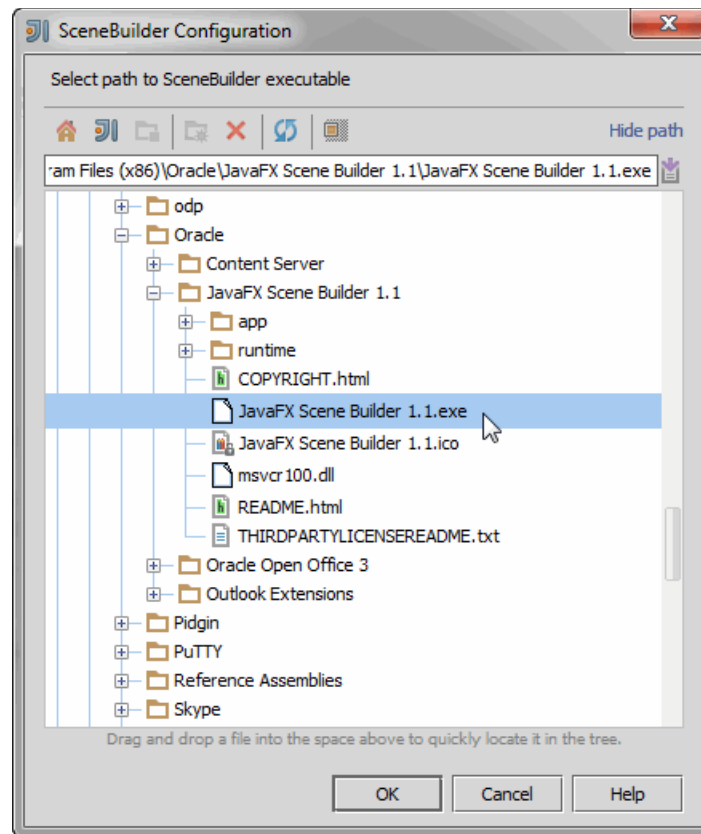
1. In the Projects tab, right-click the `sample.fxml` file and select **Open In Scene Builder**, as shown in [Figure 3-4](#).

Figure 3-4 Open JavaFX Scene Builder to `sample.fxml` File



2. Specify the path to your Scene Builder installation the first time you try to open Scene Builder from within IntelliJ IDEA.

Figure 3-5 Select Path to the Scene Builder Installation Folder



3. Use the [JavaFX Scene Builder User Guide](#) document to learn more about the available Scene Builder features and the [Getting Started with JavaFX Scene Builder](#) tutorial to create a simple issue tracking application.

Where to Go From Here

Now that you are familiar with the integration between the Scene Builder tool and IntelliJ IDEA, look at [JavaFX Scene Builder User Guide](#) to learn more about the available Scene Builder features. Also, use the [Getting Started with JavaFX Scene Builder](#) to create a simple issue tracking application.

You can also try running the sample applications provided with the Scene Builder release. These samples are Netbeans projects, so use the information in the following section to learn how to use one of the samples using the IntelliJ IDEA.

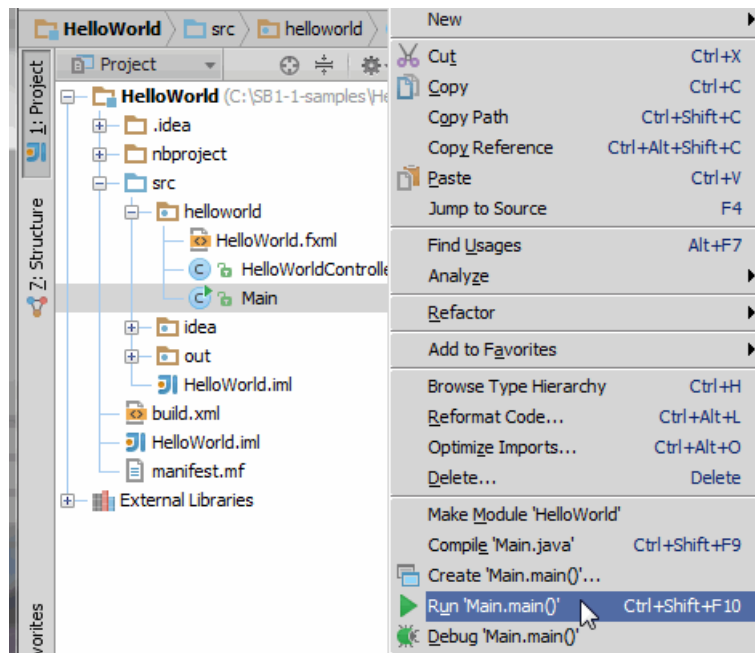
Running Scene Builder Samples from IntelliJ IDEA

Use the following steps to run the HelloWorld sample that is included in the JavaFX Scene Builder samples bundle:

1. Ensure that you have downloaded JavaFX Scene Builder Samples bundle from <http://www.oracle.com/technetwork/java/javafx/downloads/index.html> and extracted its contents.
2. From the IntelliJ IDEA main window, click **File** and choose **Import Project**.

3. Navigate to the location where you extracted the JavaFX Scene Builder samples and expand the **HelloWorld** sample folder. Select the **src** folder and click **OK**.
4. In the Import Project wizard, keep the “Create project from existing sources” option selected and click **Next**.
5. Type `HelloWorld` in the Project Name field and click **Next**.
6. Leave the rest of the default choices and values specified in the Import Project wizard and continue to click **Next** until you see the Finish button. Click **Finish**.
7. Choose to open the new project in the current window or open a new window. The new HelloWorld project is opened in the Project tab.
8. In the Project tab, right-click the node for the HelloWorld project’s Main class and choose **Run Main.main()** as shown in [Figure 3–6](#).

Figure 3–6 Run the Main Class File for HelloWorld Sample Application



The window for the HelloWorld sample application appears on top of the IDEA window.