



الجامعة اللبنانية
Lebanese University

Graphical Interface and Application(I3305)

Chapter 5-Part 2: JavaFX-Scene Builder

Lebanese University



Faculty of Science 1 - Department of Computer Science

Dr. Abed EL Safadi

What is a JavaFx Scene Builder?

JavaFX **Scene Builder** is a visual layout tool that lets users 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.

Getting set up to use Scene Builder

To download the Scene Builder tool at the following location:

<http://gluonhq.com/open-source/scene-builder/>

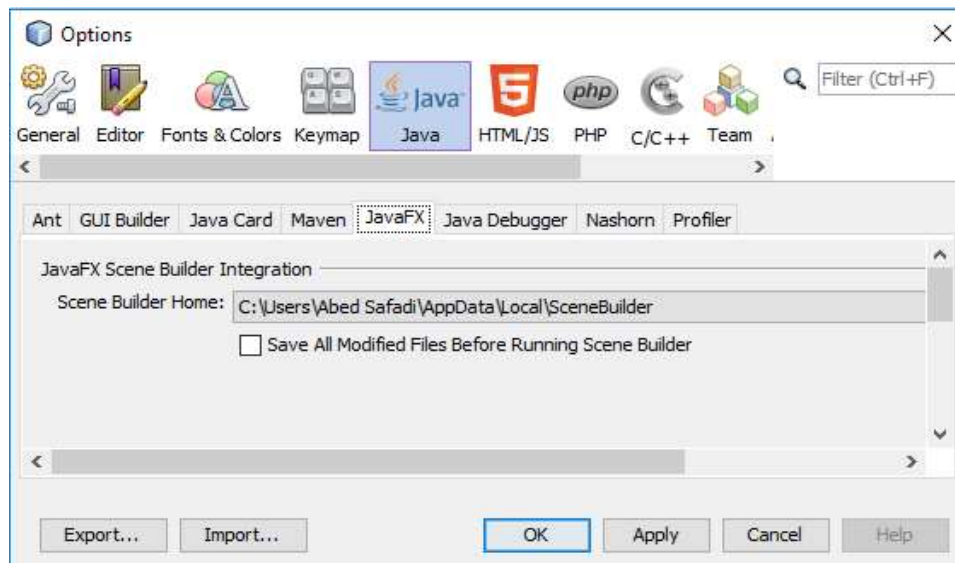
Or

<https://www.oracle.com/technetwork/java/javase/archive-139210.html>

The Scene Builder installer should also take care of automatically integrating Scene Builder into NetBeans.

Getting set up to use Scene Builder

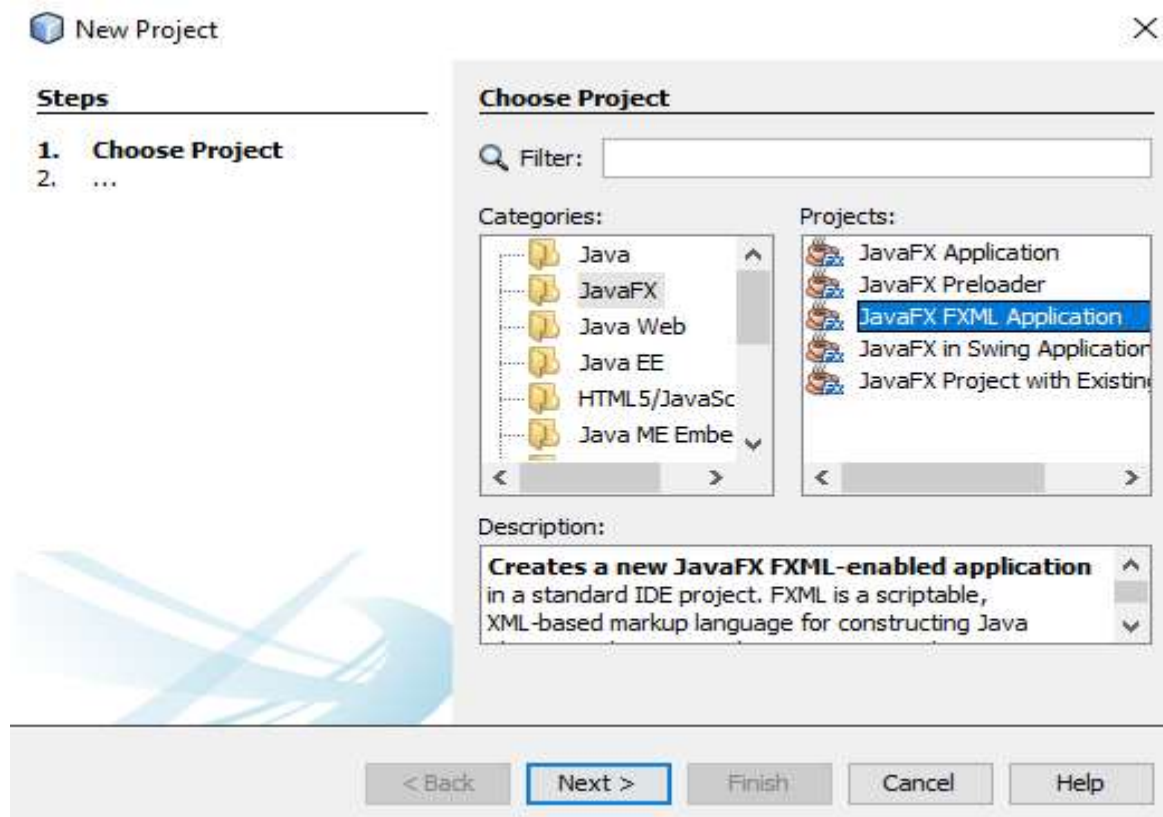
If Scene Builder does not launch automatically, select options from the Tools menu in NetBeans, check if a path to Scene Builder does not appear there



select the Browse option in the pull down menu to browse to the location where Scene Builder is installed.

A first example

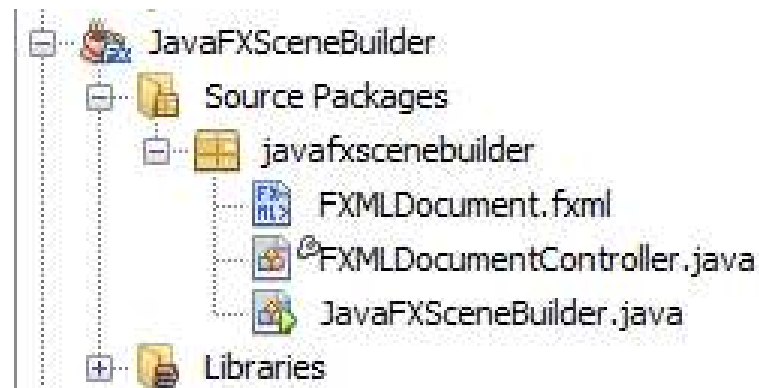
In the new project dialog select the option to make a JavaFX FXML application.



A first example-Contin

The project will have three components:

- Java file with a main method that sets up the application,
- Fxml file that you can use in combination with Scene Builder to set up the user interface,
- Controller class where you can put action methods.



A first example-Contin

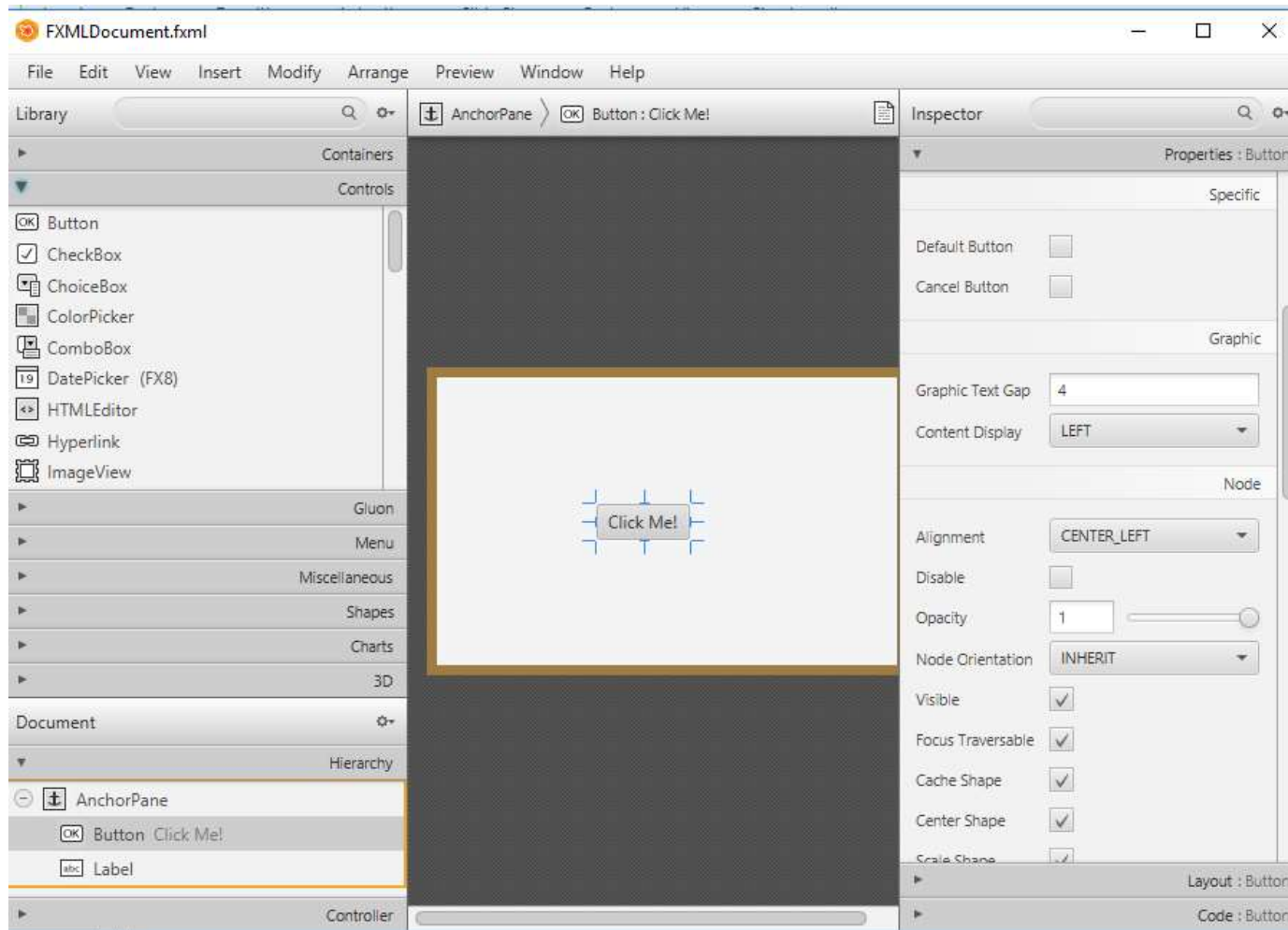
The class that contains the main method holds just enough code to set up the user interface:

```
public class JavaFXSceneBuilder extends Application {  
    @Override  
    public void start(Stage stage) throws IOException {  
        Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));  
  
        Scene scene = new Scene(root);  
  
        stage.setScene(scene);  
        stage.show();  
    }  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```

In this example all of the details of the scene contents are stored in an FXML file that the application reads and loads at the start of the start() method.

A first example-Contin

The FXML file contains all of the information about the contents and structure of the user interface.



A first example-Contin

FXML is an XML-based language designed to build the user interface for JavaFX applications.

```
<?xml version="1.0" encoding="UTF-8"?>

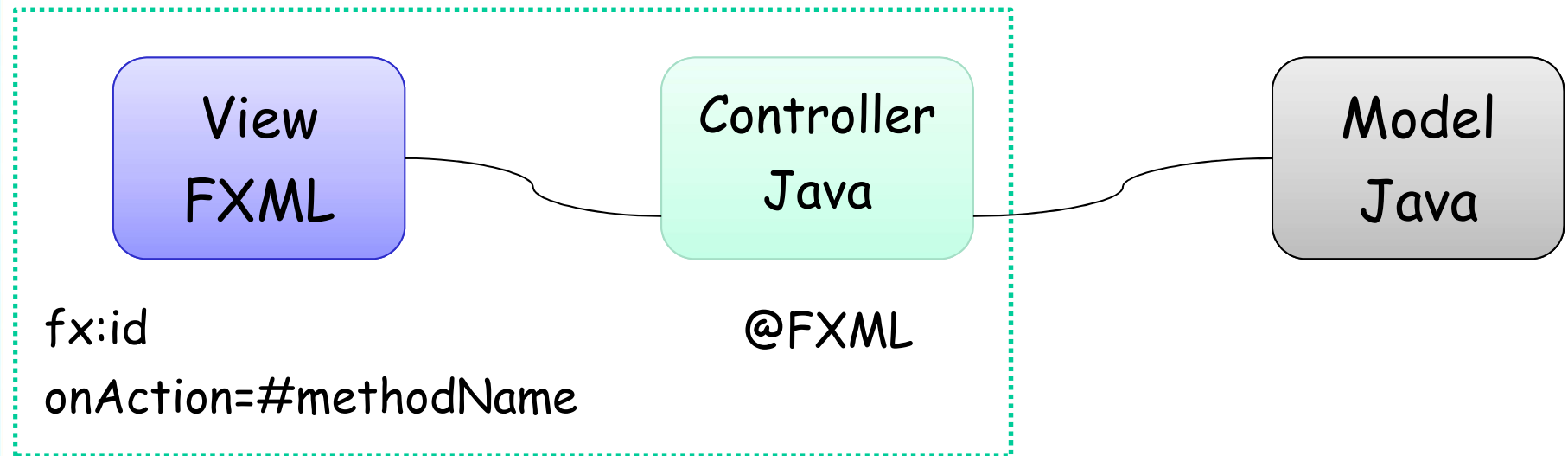
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<AnchorPane id="AnchorPane" prefHeight="200" prefWidth="320" xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="firstexample.FXMLDocumentController">
    <children>
        <Button layoutX="126" layoutY="90" text="Click Me!" onAction="#handleButtonAction" fx:id="button" />
        <Label layoutX="126" layoutY="120" minHeight="16" minWidth="69" fx:id="label" />
    </children>
</AnchorPane>
```



```
Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));
```

A first example-Contin



A first example-Contin

```
<children>
  <Button layoutX="126" layoutY="90" text="Click Me!" onAction="#handleButtonAction" fx:id="button" />
  <Label layoutX="126" layoutY="120" minHeight="16" minWidth="69" fx:id="label" />
</children>
```

The controller class contains action methods that you will link to various controls in the interface.

```
public class FXMLDocumentController implements Initializable {

    @FXML
    private Label label;

    @FXML
    private void handleButtonAction(ActionEvent event) {
        System.out.println("You clicked me!");
        label.setText("Hello World!");
    }

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

}
```

