

# JavaFX

By: Olenka Soko

# What is it?

- Graphic framework for creating GUIs in Java applications
- Replacement for Swing as a standard GUI library for Java SE
- Can create web applications and desktop applications
- Uses FXML for defining User Interfaces
- More powerful than Swing



# History of JavaFX

- Available (unofficially) since 2005
- At JavaOne 2007 conference - Sun Microsystems unveiled JavaFx
- Developed by Chris Oliver at SeeBeyond and originally called F3 (Form Follows Function)
- JavaFX 16 is the latest release of JavaFX.





## Key Features

- JavaFX is written in Java (starting from JavaFX 2.0)
- JavaFx is part of JDK (starting from JDK 8)
- Supports FXML
- User can use Swing UI in a JavaFx application
- Has WebView component ( HTML CSS, SVG, JavaScript)
- 2D and 3D graphics are available
- Audio (mp3, wav,aiff), video (flv), and images are available

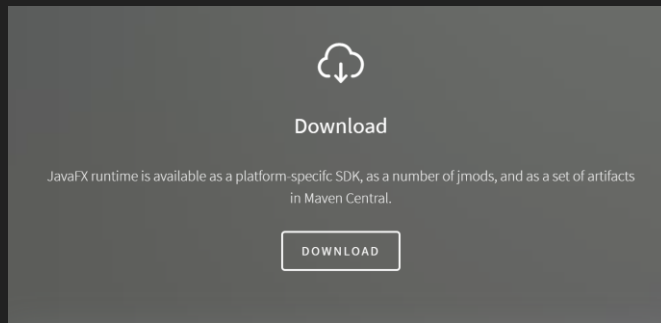
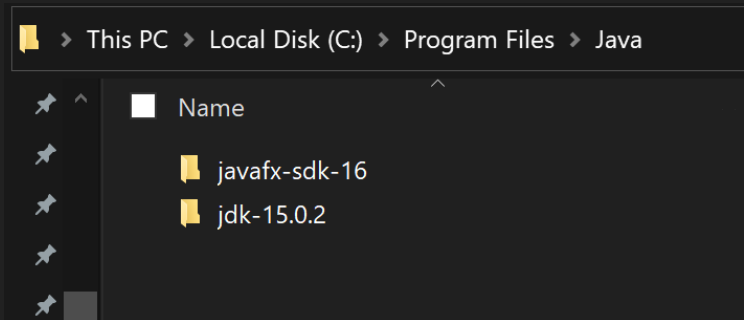
# Using JavaFx

- NetBeans
- Eclipse
- IntelliJ
- Really any IDE that you are comfortable with



# Let's start...

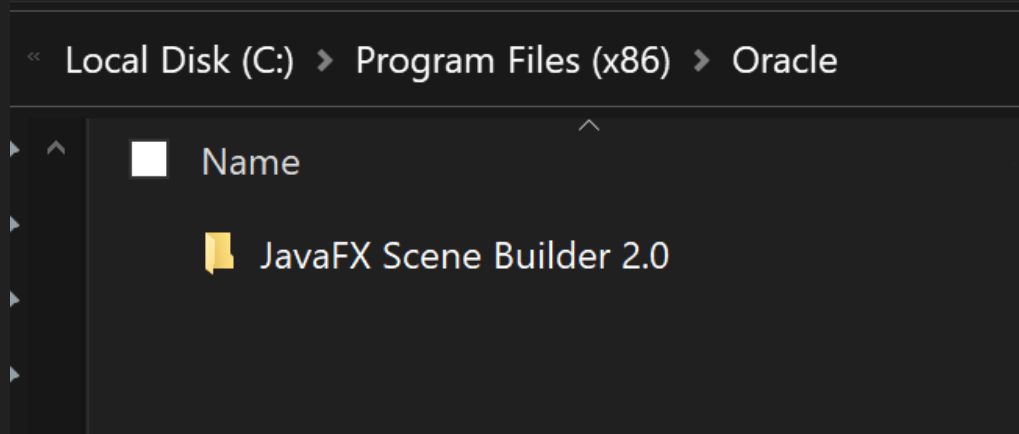
- openjfx.io
- Scroll to download
- Select product of choosing
- Extract to where your Java files are located



Product	Version	Platform	Download
JavaFX Windows x64 SDK	16	Windows x64	<a href="#">Download</a> <a href="#">[SHA256]</a>
JavaFX Windows x64 jmods	16	Windows x64	<a href="#">Download</a> <a href="#">[SHA256]</a>
JavaFX Windows x86 SDK	16	Windows x86	<a href="#">Download</a> <a href="#">[SHA256]</a>
JavaFX Windows x86 jmods	16	Windows x86	<a href="#">Download</a> <a href="#">[SHA256]</a>
JavaFX Mac OS X SDK	16	Mac	<a href="#">Download</a> <a href="#">[SHA256]</a>
JavaFX Mac OS X jmods	16	Mac	<a href="#">Download</a> <a href="#">[SHA256]</a>
JavaFX Linux SDK	16	Linux	<a href="#">Download</a> <a href="#">[SHA256]</a>
JavaFX Linux jmods	16	Linux	<a href="#">Download</a> <a href="#">[SHA256]</a>
JavaFX Documentation	16	Javadoc	<a href="#">Download</a> <a href="#">[SHA256]</a>

# Downloading JavaFX Scene Builder

- Allows you to layout UI controls, charts, shapes, containers, etc. for quicker use. Tool generates FXML code!
- <https://www.oracle.com/java/technologies/javafxscenebuilder-1x-archive-downloads.html>




# IDE - Eclipse

- <https://www.eclipse.org/downloads/>
- Extract to Program Files

Tool Platforms

The Eclipse Installer 2021-03 R now includes a JRE for macOS, Windows and Linux.



Get **Eclipse IDE 2021-03**

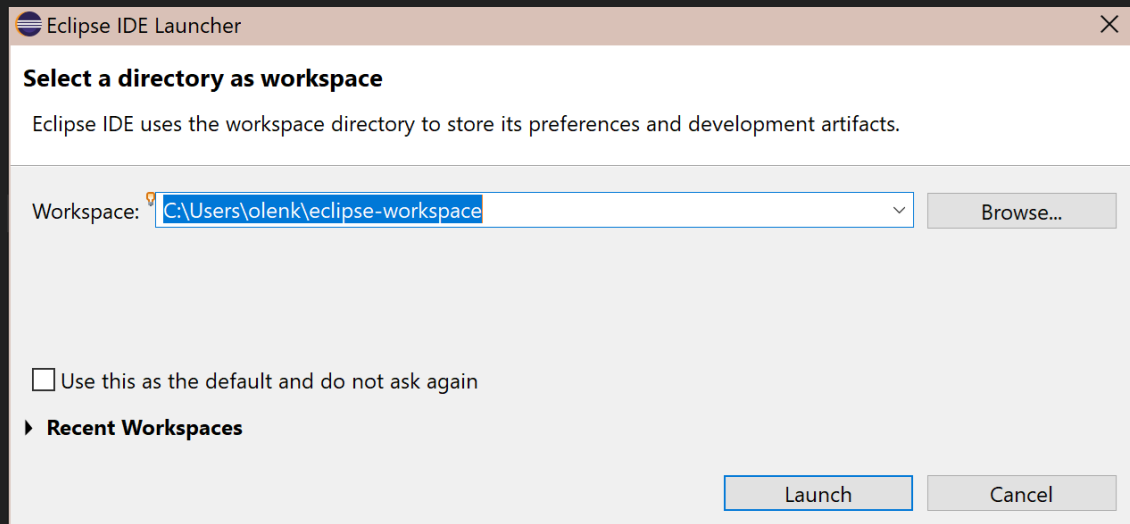
Install your favorite desktop IDE packages.

[Download x86\\_64](#)

[Download Packages](#) | [Need Help?](#)



# Launching

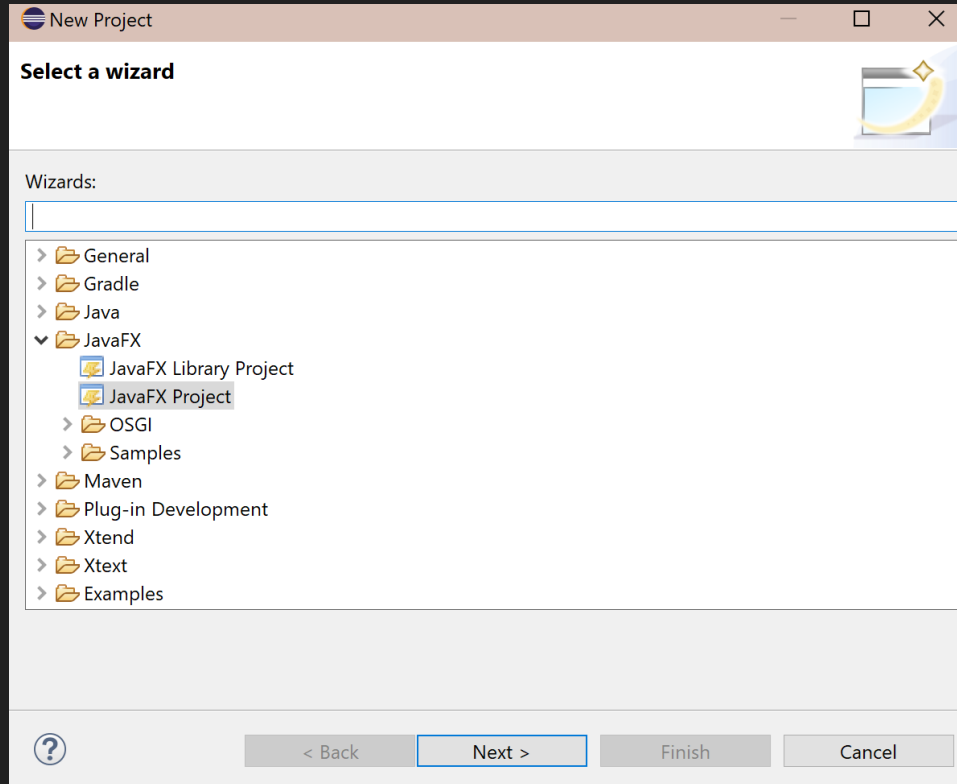


# In Eclipse

Select File

New

Project



- Name project
- Select where to store project
- Finish

New Java Project

### Create a Java Project

Create a Java project in the workspace or in an external location.

Project name:

☐ Use default location

Location:

JRE

☒ Use an execution environment JRE: JavaSE-15

☐ Use a project specific JRE: jre

☐ Use default JRE 'jre' and workspace compiler preferences

Project layout


☐ Use project folder as root for sources and class files


☒ Create separate folders for sources and class files

Working sets

☐ Add project to working sets

Working sets:

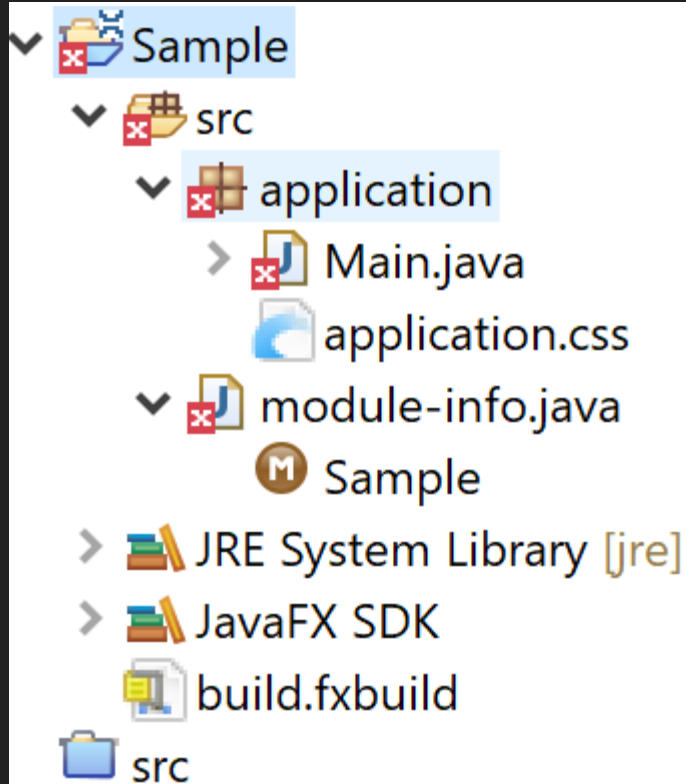
 The wizard will automatically configure the JRE and the project layout based on the existing source.

 < Back Next > Finish

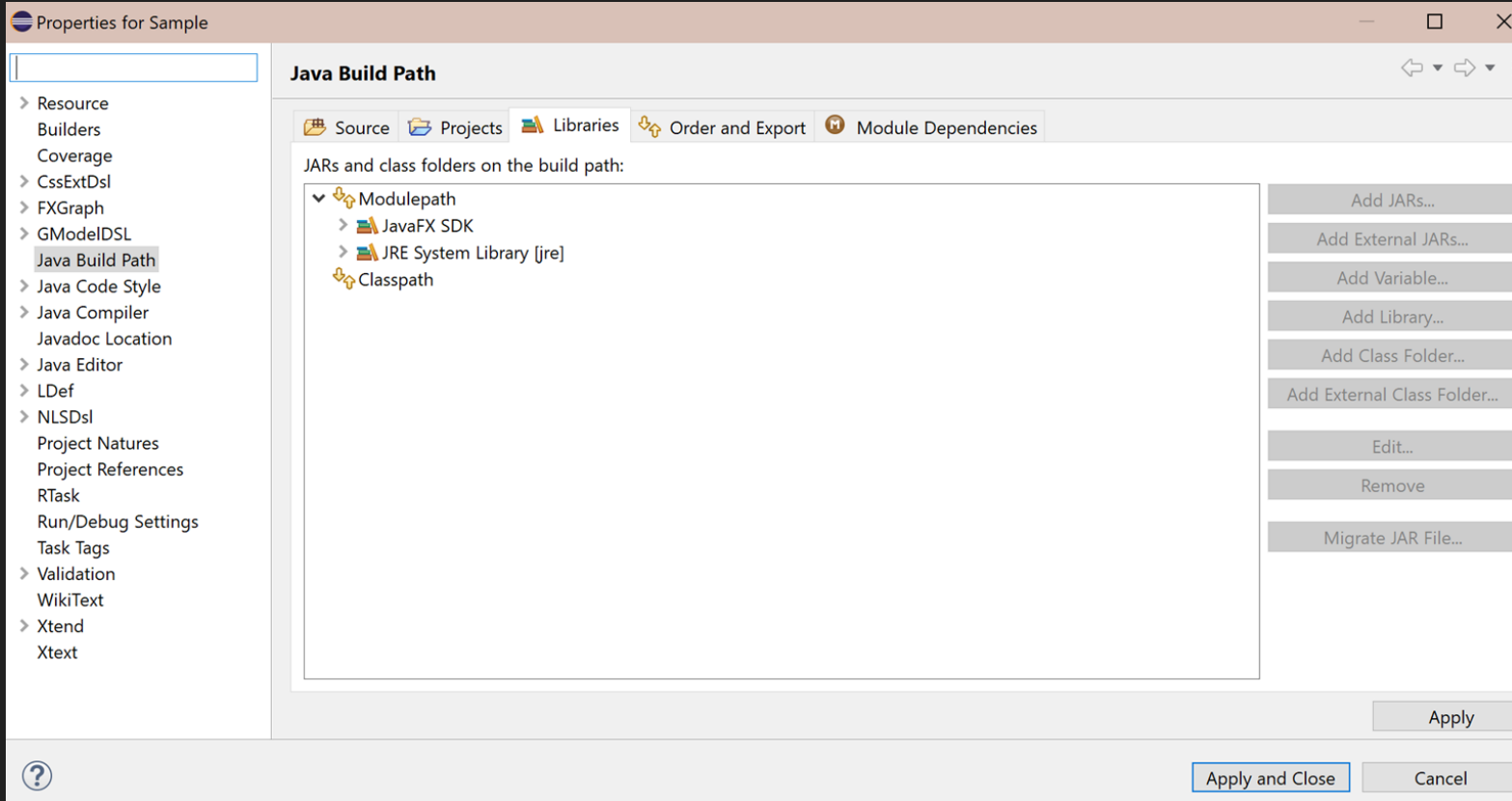
# Errors

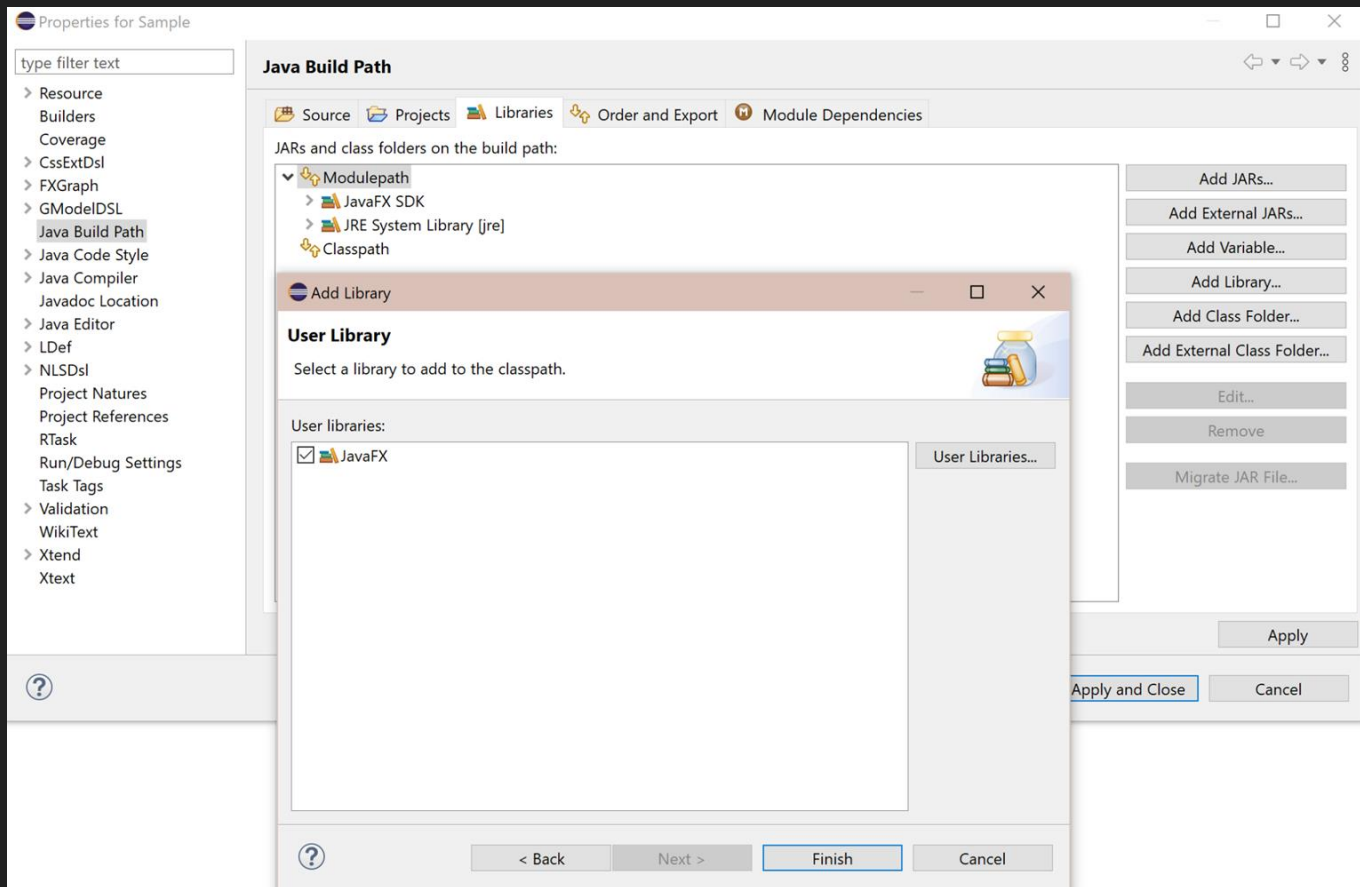
```
1 package application;  
2  
3 import javafx.application.Application;  
4 import javafx.fxml.FXMLLoader;  
5 import javafx.stage.Stage;  
6 import javafx.scene.Parent;  
7 import javafx.scene.Scene;  
8 import javafx.scene.layout.BorderPane;
```

```
1 package application;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.stage.Stage;
6 import javafx.scene.Parent;
7 import javafx.scene.Scene;
8 import javafx.scene.layout.BorderPane;
9 import javafx.fxml.FXML;
10
11
12
13 public class Main extends Application {
14     @Override
15     public void start(Stage primaryStage) {
16         try {
17             Parent root = FXMLLoader.load(getClass().getResource("/application/Example.fxml"));
18             Scene scene = new Scene(root, 400, 400);
19             scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
20             primaryStage.setScene(scene);
21             primaryStage.show();
22         } catch (Exception e) {
23             e.printStackTrace();
24         }
25     }
26
27     public static void main(String[] args) {
28         launch(args);
29     }
30 }
```



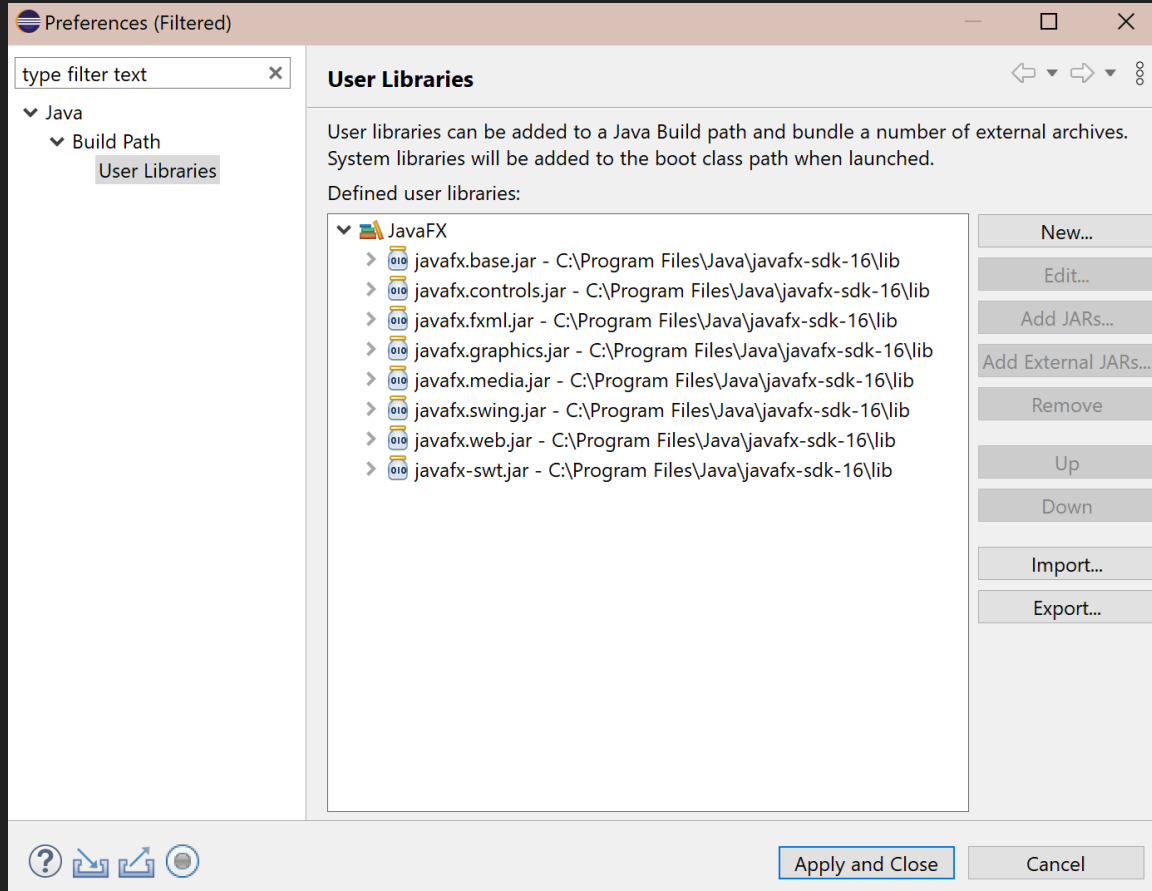
- Right click
- Properties

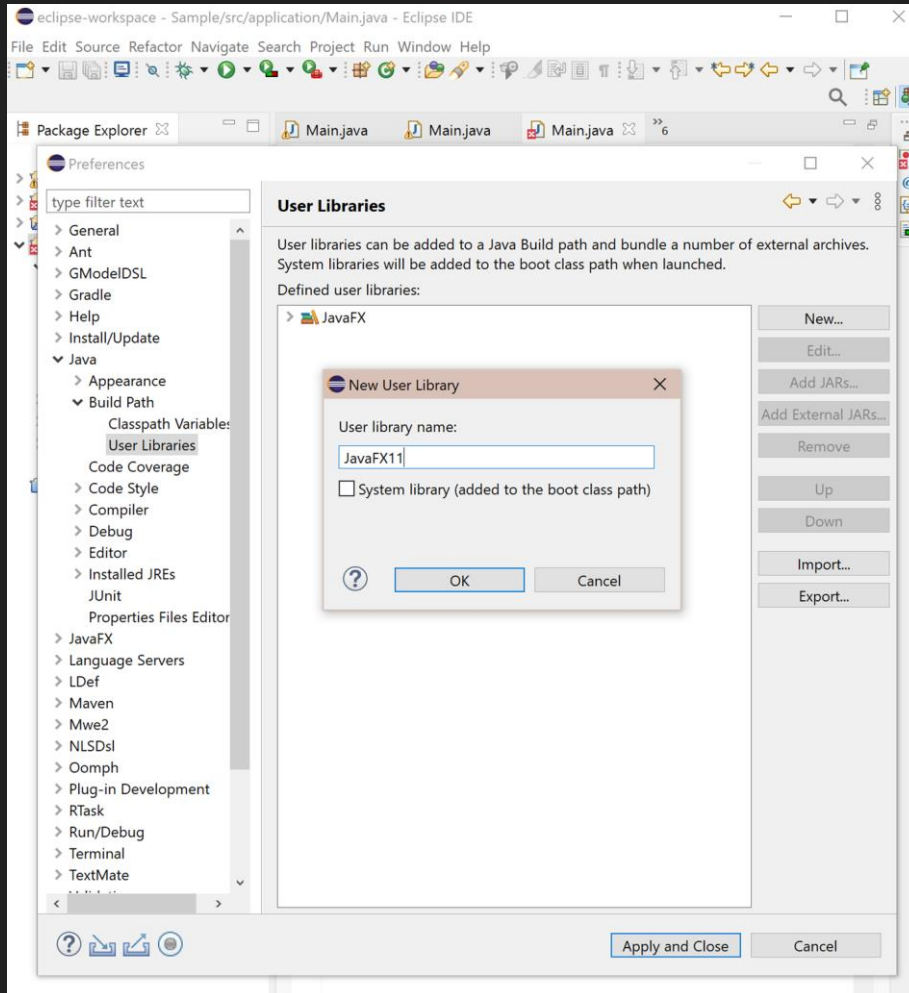






# If JavaFx isn't available





Select Window

Preferences

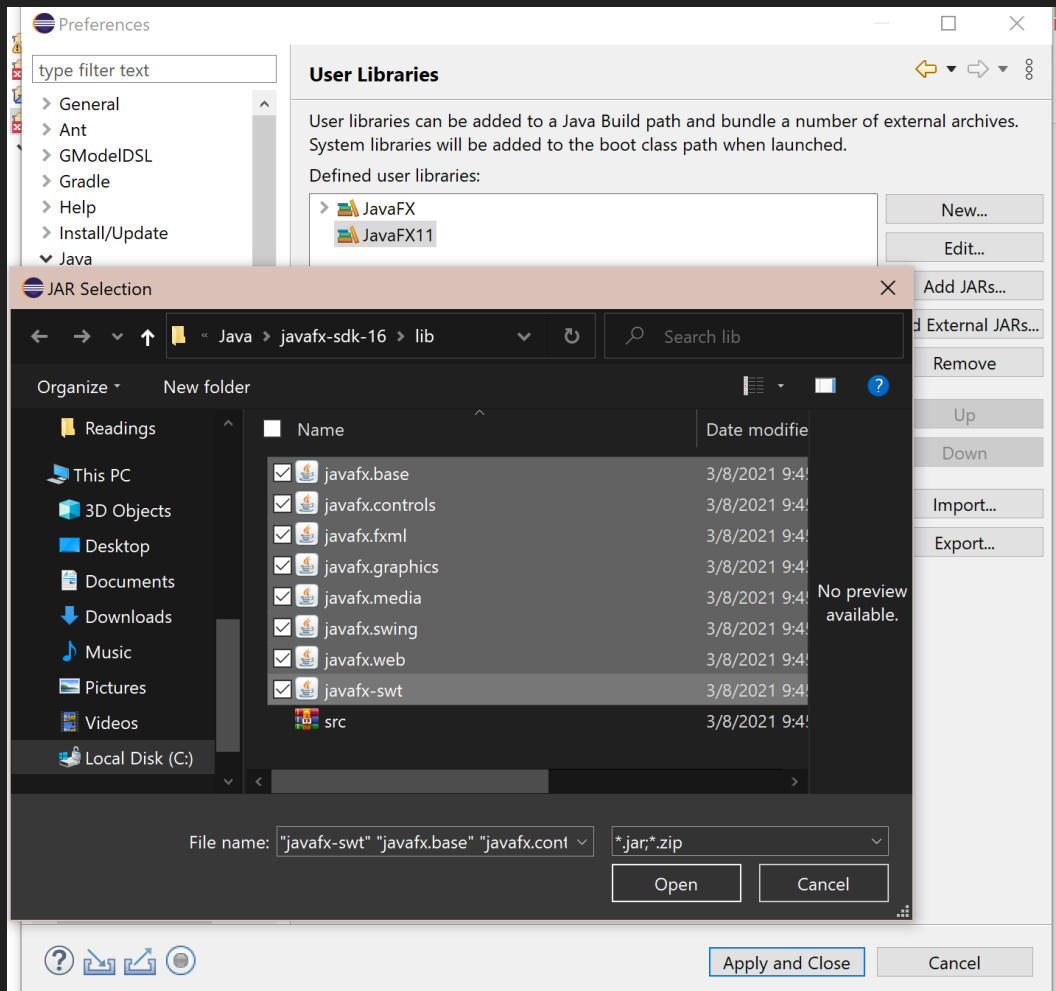
Java

Build Path

User Libraries

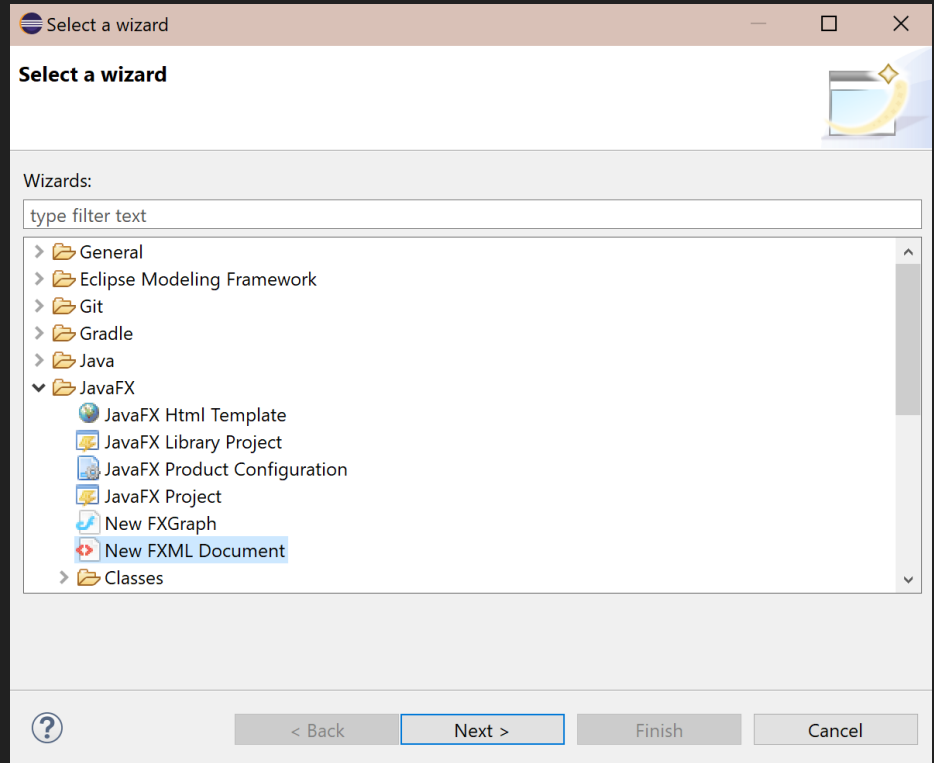
New.


Name it!




- Add External Jars
- Go to your JavaFX file
- Lib
- Open
- Apply and close

- Right click on name of file
- New
- Other



 □ ×

## FXML File

Create a new FXML File 


Source folder  Browse ...

Package  Browse ...

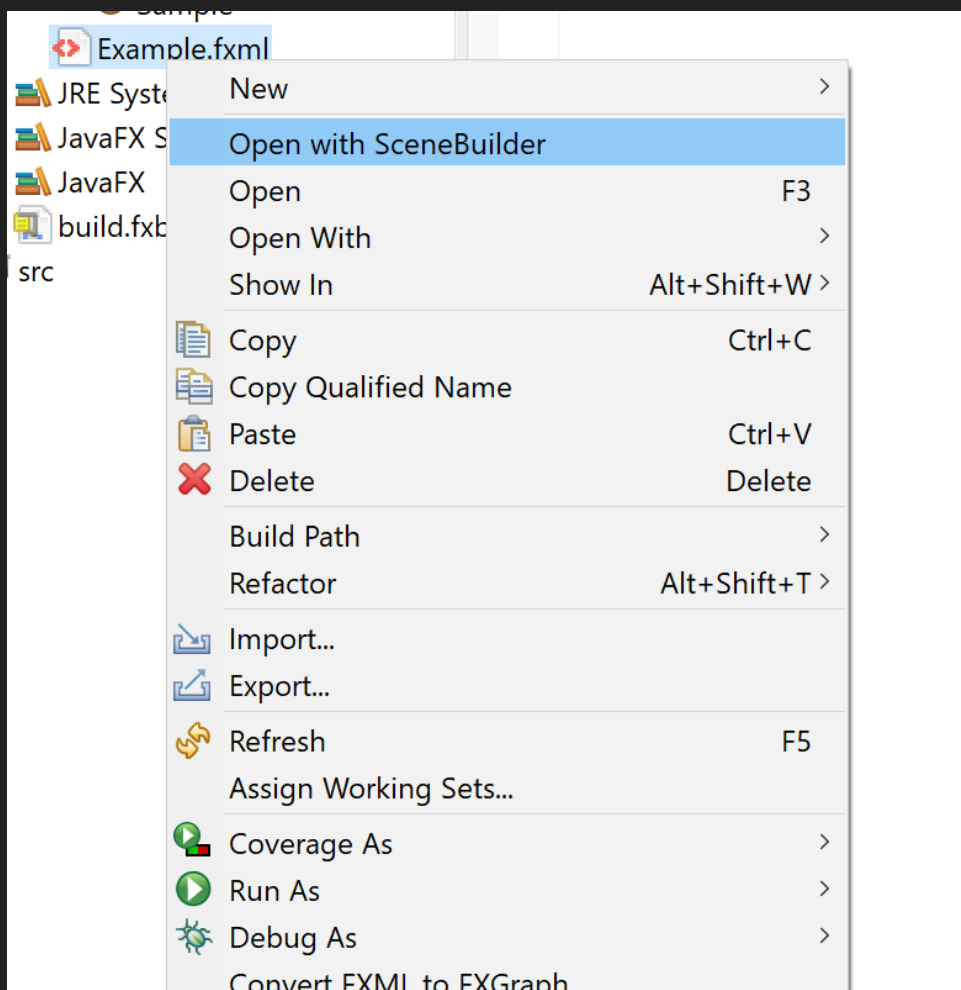
Name

Root Element  ▼ Browse ...

Dynamic Root (fx:root) ☐

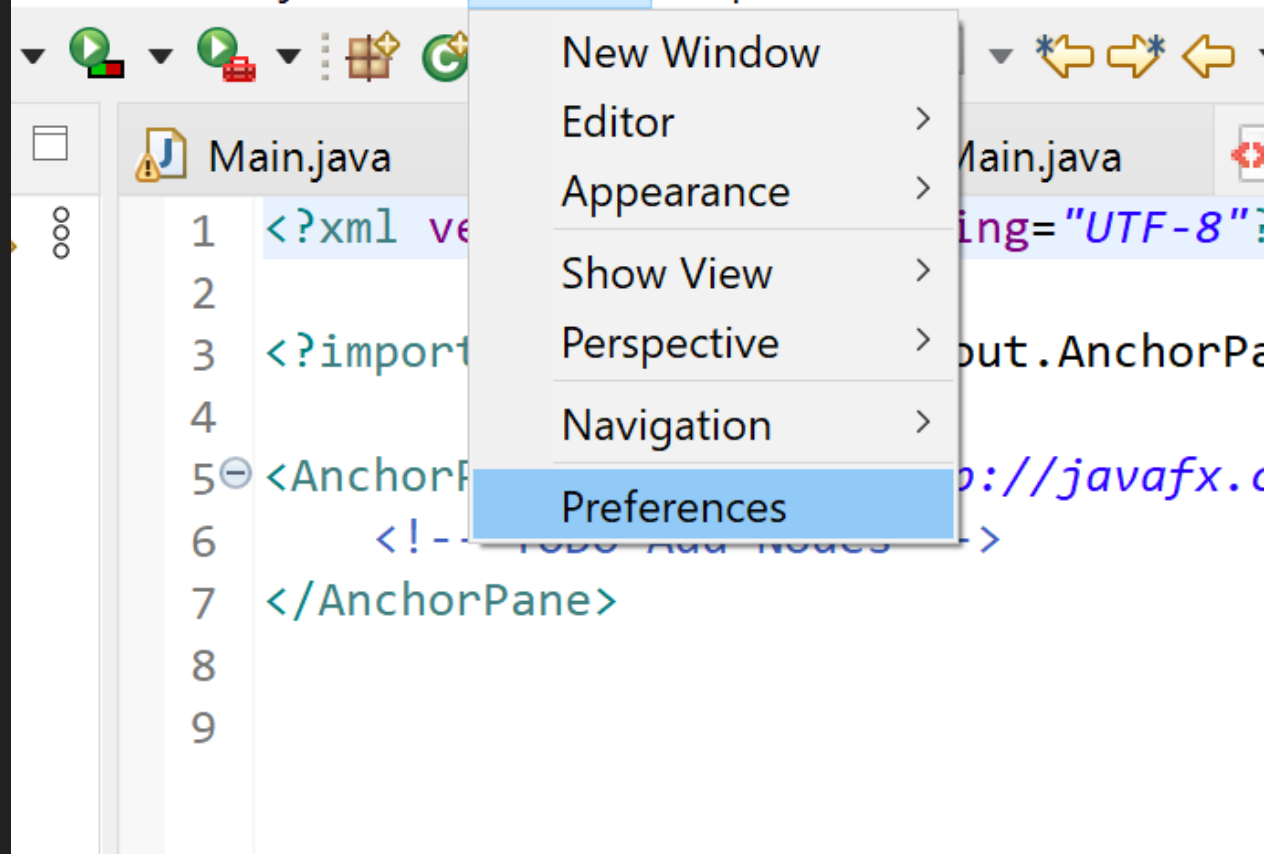
 < Back Next > Finish Cancel

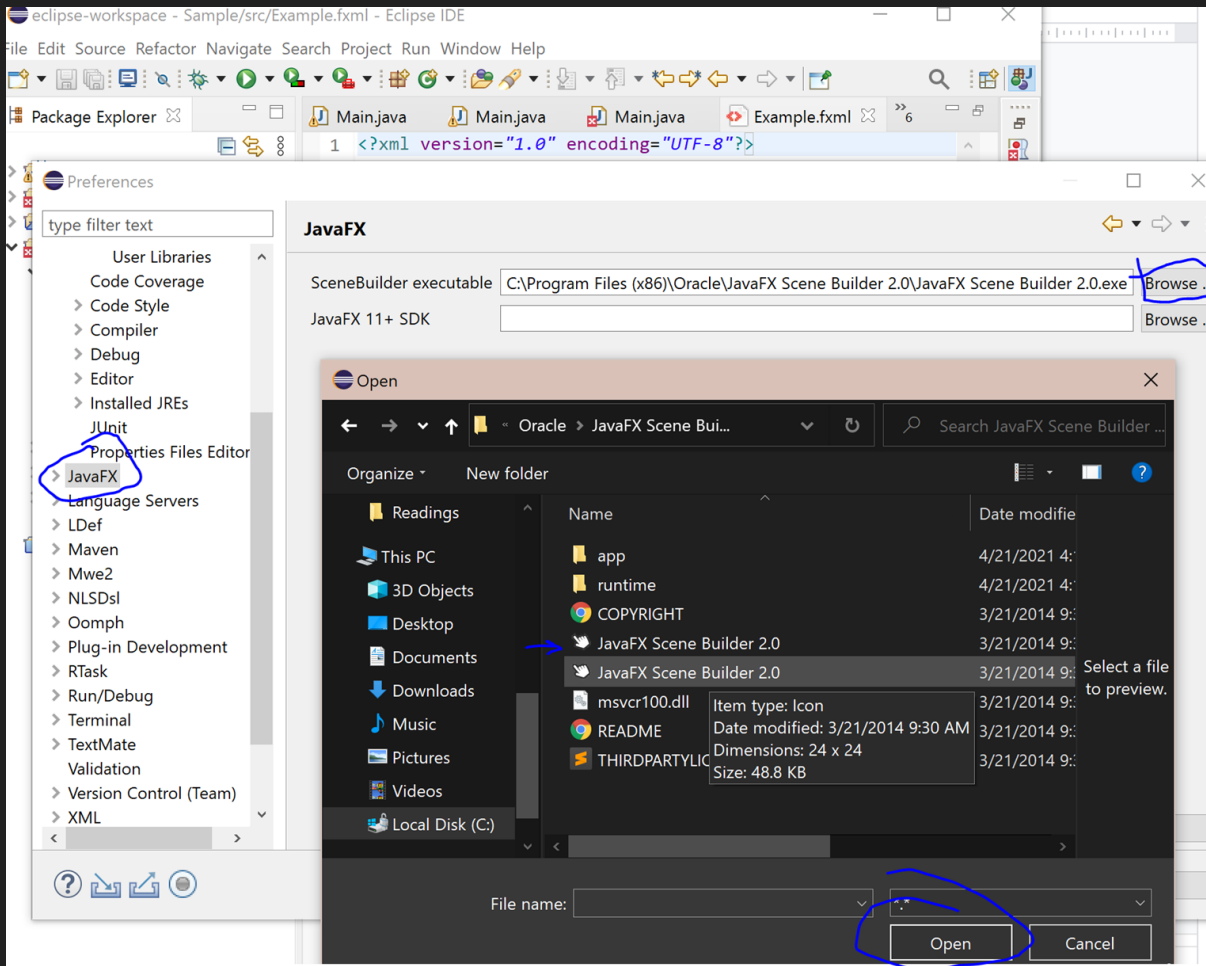
- Name file
- Choose Root Element



- If doesn't open then...

File Search Project Run Window Help







Example.fxml

FileEditViewInsertModifyArrangePreviewWindowHelp

Library

Containers

Controls

Menu

Miscellaneous

Shapes

Charts

3D

Document

Hierarchy

AnchorPane

Controller

No Selection

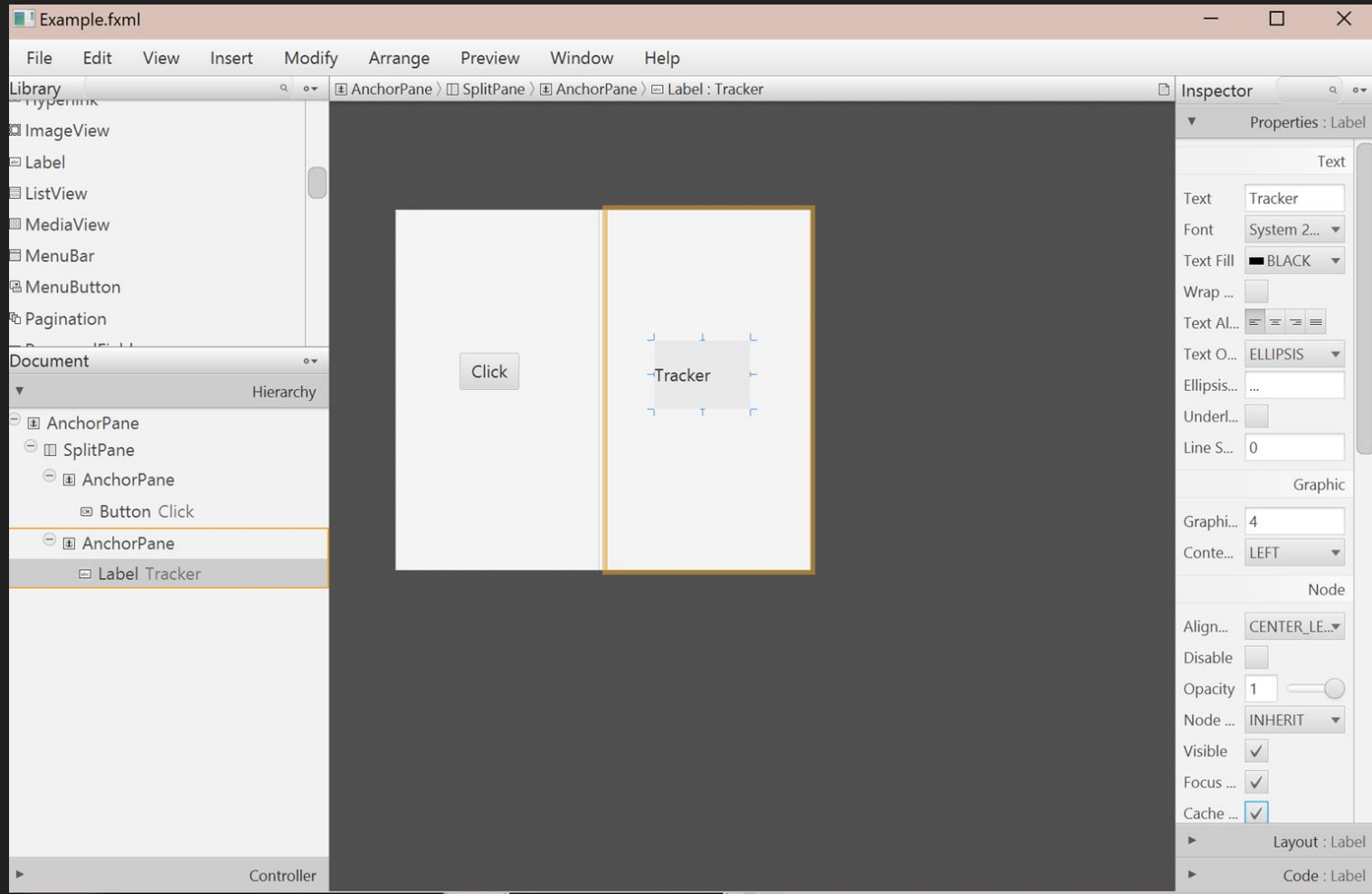
Inspector

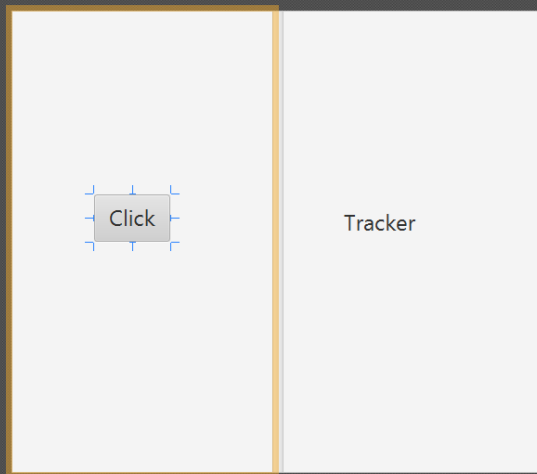
Properties

No Selection

Layout

Code





- Properties : Button
- Layout : Button
- Code : Button

### Identity

fx:id

### Main

#### On Action

#

### DragDrop

#### On Drag Detected

#

#### On Drag Done

#

#### On Drag Dropped

#

#### On Drag Entered

#

#### On Drag Exited

#

#### On Drag Over

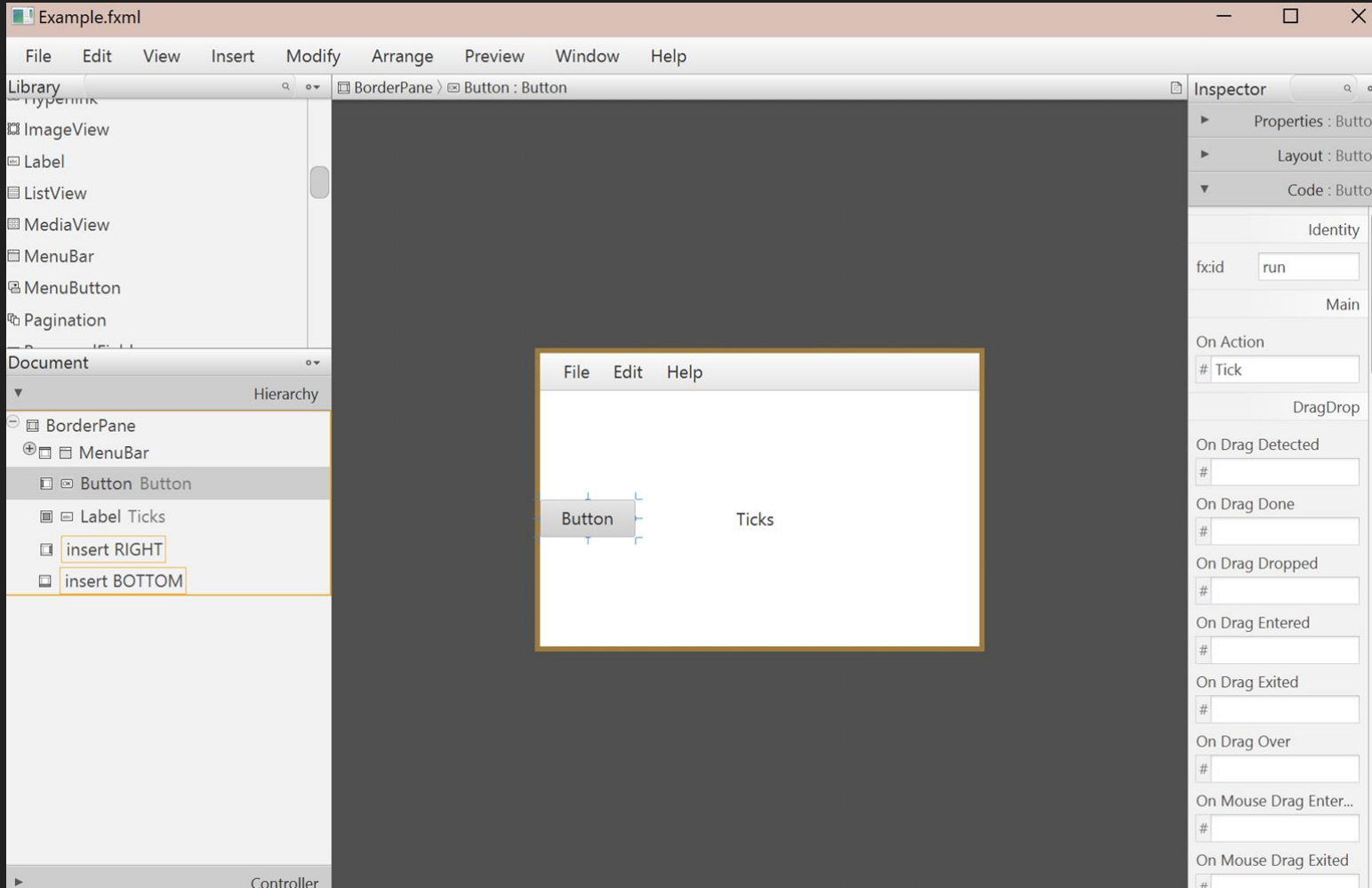
#

#### On Mouse Drag Enter...

#

#### On Mouse Drag Exited














#

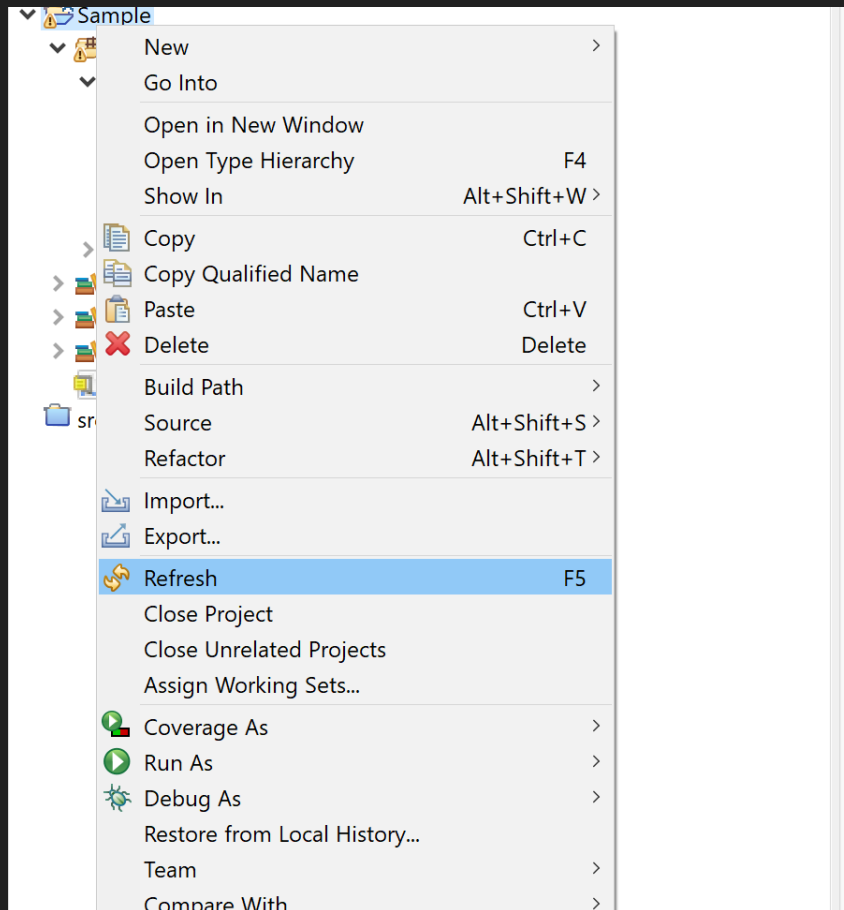


```
30 */
31 public class Controller implements Initializable{
32
33     @Override
34     public void initialize(URL location, ResourceBundle resources) {
35
36         // TODO Auto-generated method stub
37
38     }
39     /**
40      * Closes window
41      * @param event
42      */
43     public void Close(ActionEvent event) {
44         Platform.exit();
45         System.exit(0);
46     }
```

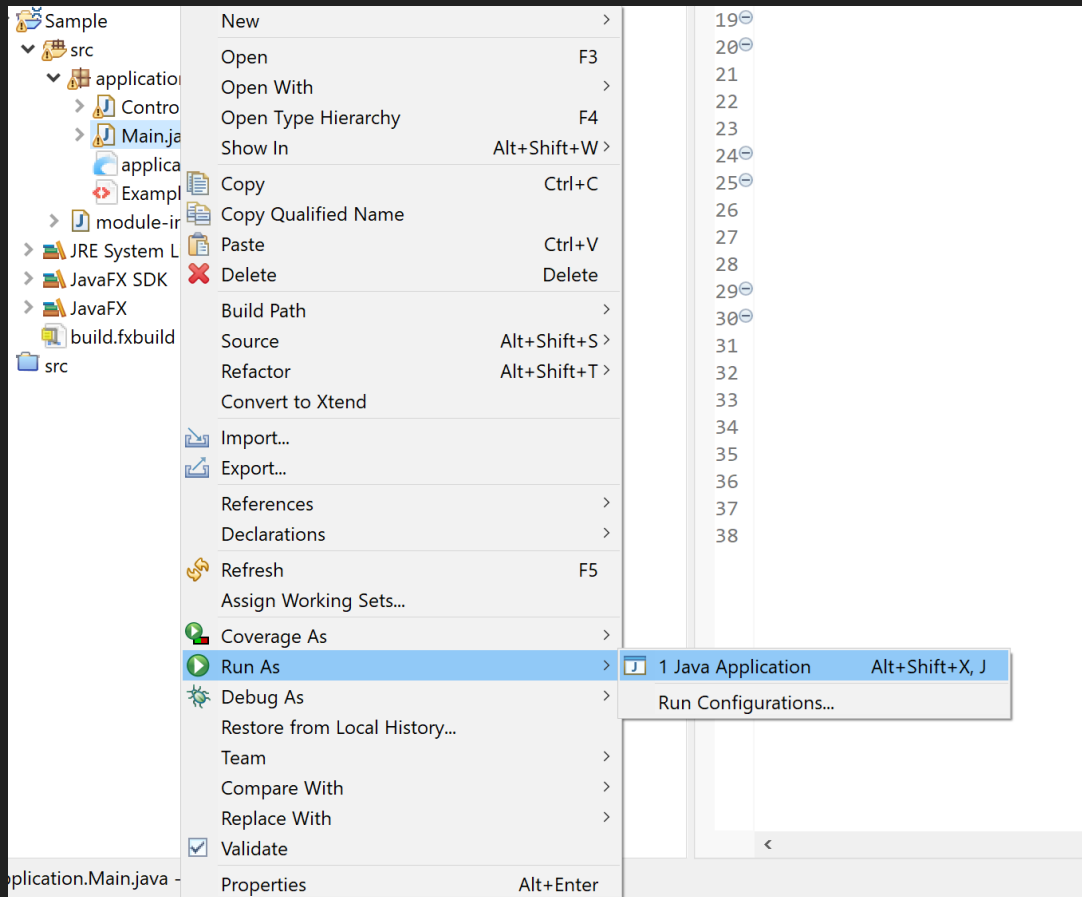
```
47
48
49
50
51     @FXML
52     private Button run;
53     @FXML
54     private Label lab;
55     int count = 0;
56
57     /**
58      * Represents the amount of times button is clicked
59      * @param event
60      */
61     @FXML
62     public void Tick(ActionEvent event) {
63         count++;
64         lab.setText("Count: "+count);
65
66     }
```

```
= "http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.Controller">  
ENTER" />
```

- ▼  Sample
  - ▼  src
    - ▼  application
      - >  Controller.java
      - >  Main.java
      -  application.css
      -  Example.fxml
      - >  module-info.java
    - >  JRE System Library [jre]
    - >  JavaFX SDK
    - >  JavaFX
    -  build.fxbuild
  -  src







```
* Main controller for javafx application
```

```
*
```

```
*/
```

```
public class
```

```
@Over
```

```
public
```

```
ces) {
```

```
}
```

```
/**
```

```
* CL
```

```
* @P
```

```
*/
```

```
public
```

```
P
```

```
S
```

```
}
```

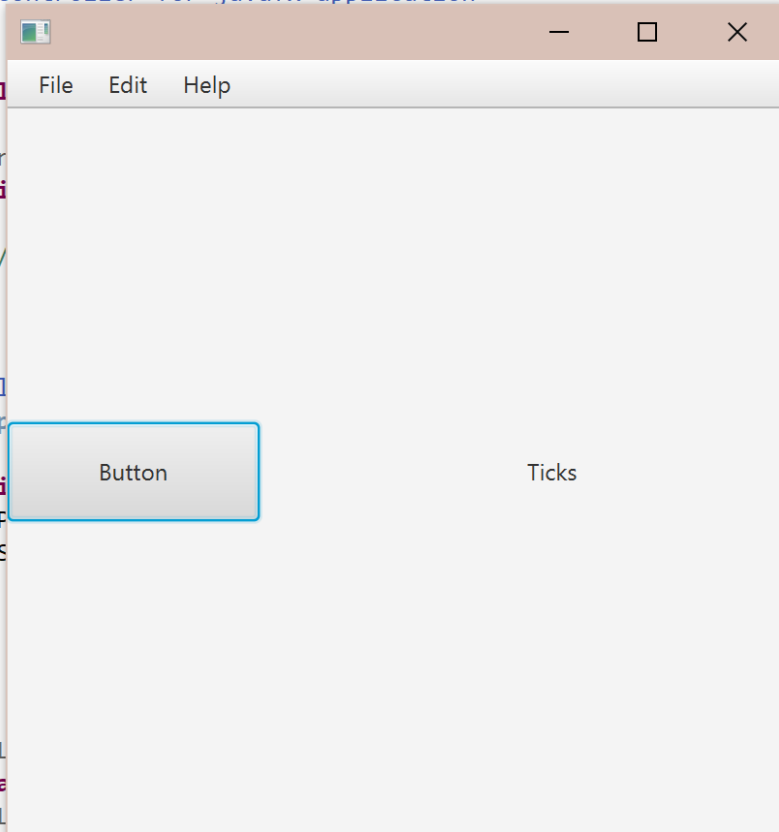
```
@FXML
```

```
private
```

```
@FXML
```

```
private Label lab;
```

```
int count = 0;
```



# The start of more

- Java APIs
- FXML and Scene Builder
- WebView
- Swing interoperability
- Built-in UI controls and CSS
- Canvas API
- Multitouch Support
- Hardware-accelerated graphics pipeline
- High-performance media engine
- Self-contained application deployment model