**Create a new JavaFX project﻿**

24 November 2020

<https://www.jetbrains.com/help/idea/javafx.html#vm-options>

JavaFX is a software platform that is used for developing desktop applications that can run across a wide variety of devices.

JavaFX support in IntelliJ IDEA includes code completion, search, navigation and refactoring in JavaFX-specific source files (including **.fxml** and JavaFX **.css** files), integration with JavaFX Scene Builder, JavaFX application packaging capabilities, and more.

Note that JavaFX is **no longer a part of the JDK** starting from JDK 11. That is why, if you use Java 11 and later, you need to [download the open-source JavaFX SDK](https://www.jetbrains.com/help/idea/javafx.html#download-javafx) in addition to the JDK.

If you use Java 10 and earlier, you can [create a new project right away](https://www.jetbrains.com/help/idea/javafx.html#create-project).

**Download the JavaFX SDK﻿**

1. Download the [JavaFX SDK](https://gluonhq.com/products/javafx/) package suitable for your operating system.
2. Unpack the archive and place the folder to a meaningful location, for example: **/Users/jetbrains/Desktop/javafx-sdk-12**.

**Create a new project﻿**

When you create a new JavaFX project, IntelliJ IDEA generates a fully configured sample application.

1. Launch IntelliJ IDEA.

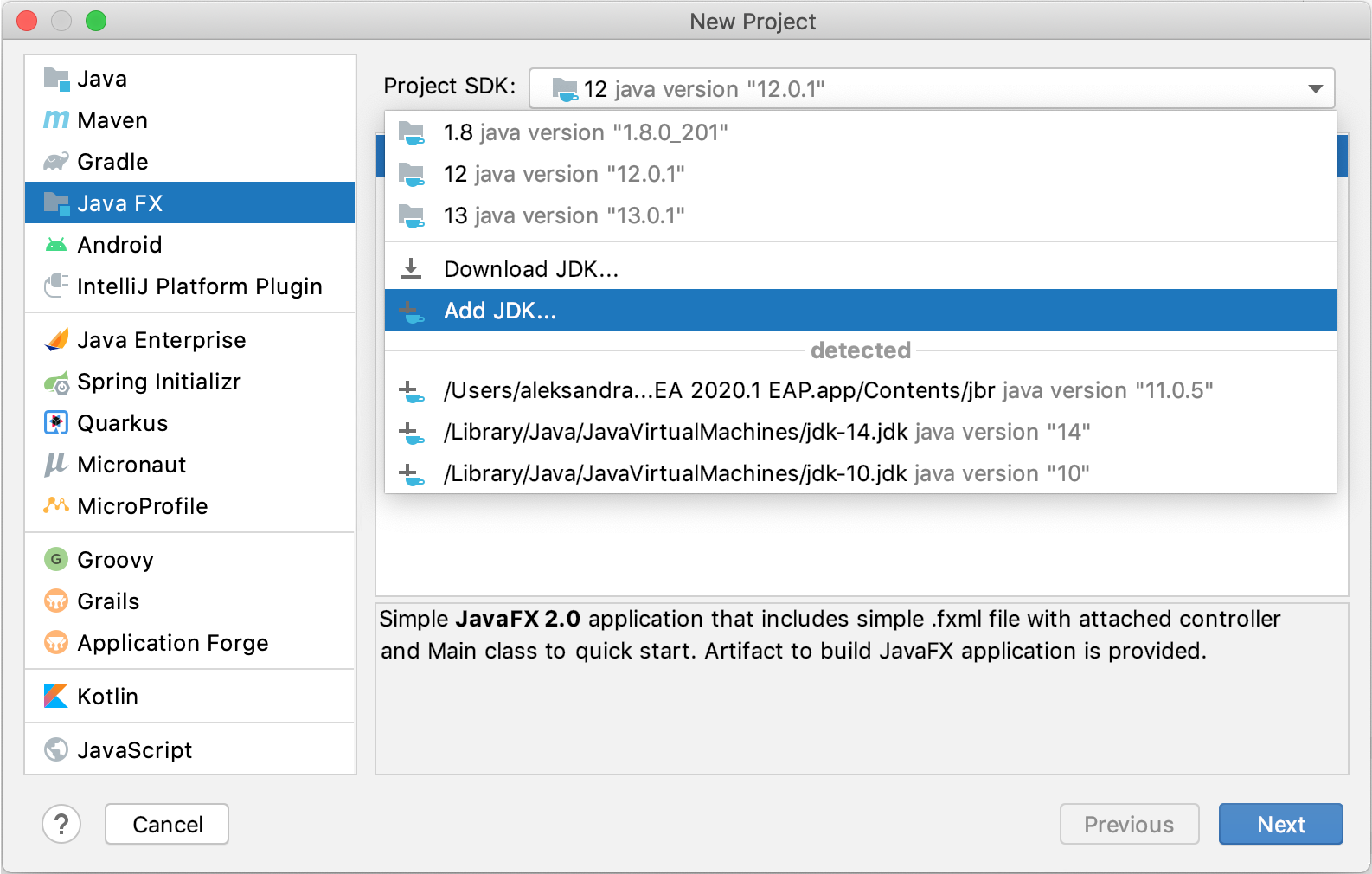
If the **Welcome** screen opens, click **New Project**.

Otherwise, from the main menu, select **File | New | Project**.

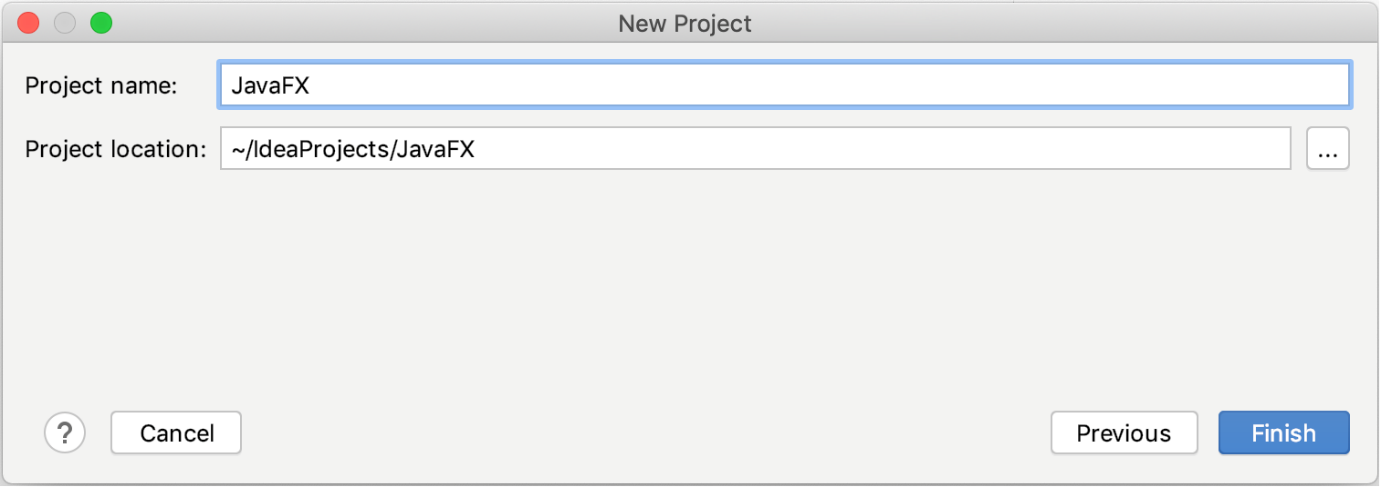
1. In the **New Project** wizard, select **JavaFX** from the list on the left.
2. From the **Project SDK** list, select the [JDK](https://www.jetbrains.com/help/idea/sdk.html#jdk) that you want to use in your project.

If the JDK is installed on your computer, but not defined in the IDE, select **Add JDK** and specify the path to the JDK home directory.

If you don't have the necessary JDK on your computer, select **Download JDK**.



1. On the next step of the wizard, name the project, for example: JavaFX.
2. If necessary, change the default project location and click **Finish**.

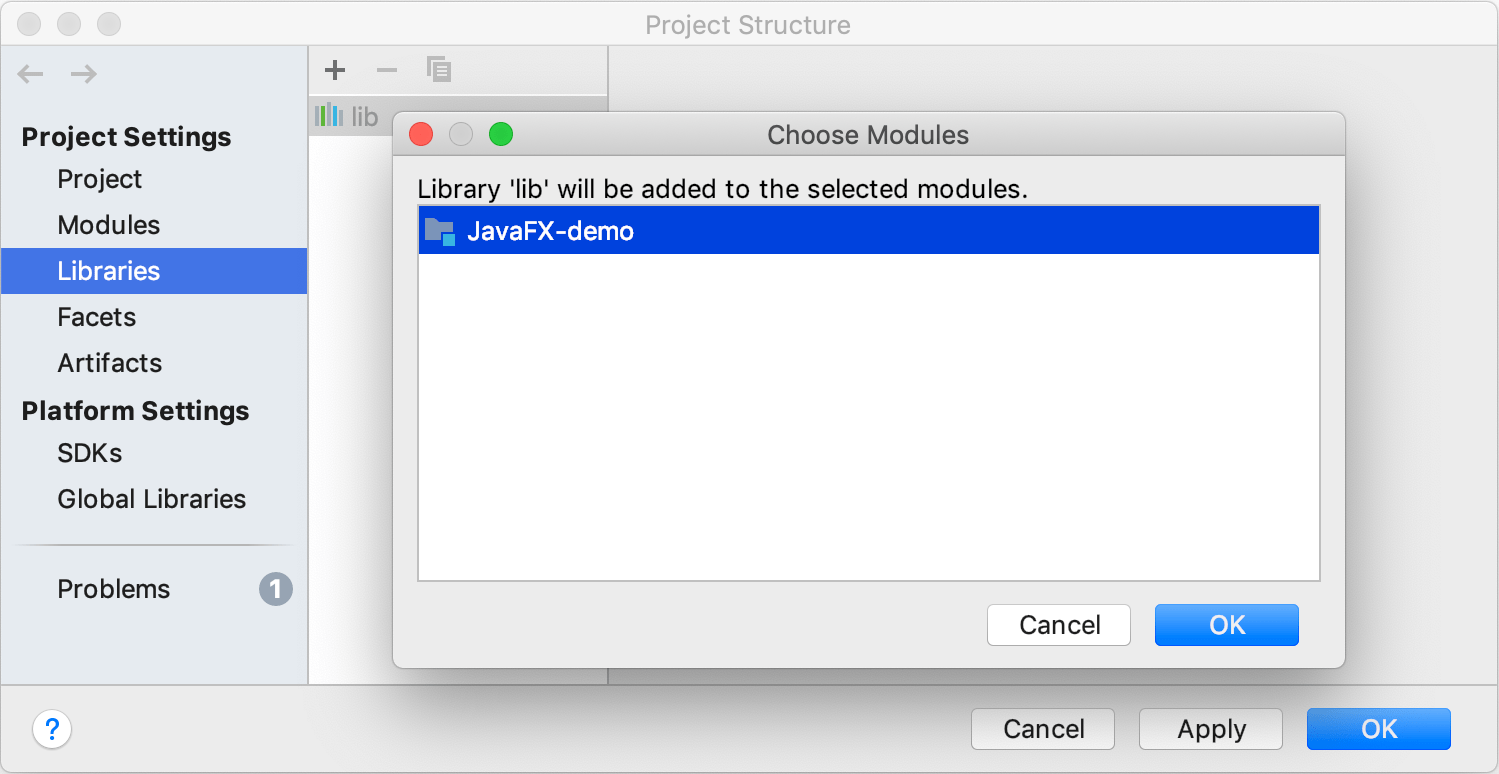


If you cannot find JavaFX in the **New Project** wizard, make sure that the bundled **JavaFX** plugin is enabled in **Settings/Preferences | Plugins**.

**Add the JavaFX library﻿**

These actions are required if you use Java 11 and later. If you use Java 10 and earlier, [run the sample application](https://www.jetbrains.com/help/idea/javafx.html#run) created together with the project to make sure that everything is working as intended.

1. From the main menu, select **File | Project Structure** Ctrl+Alt+Shift+S or click  on the toolbar.
2. Open the **Libraries** section, click , and select **Java**.
3. Specify the path to the **lib** folder in the JavaFX SDK package, for example: **/Users/jetbrains/Desktop/javafx-sdk-12/lib**.
4. In the **Choose Modules** dialog, select the necessary module and click **OK**.

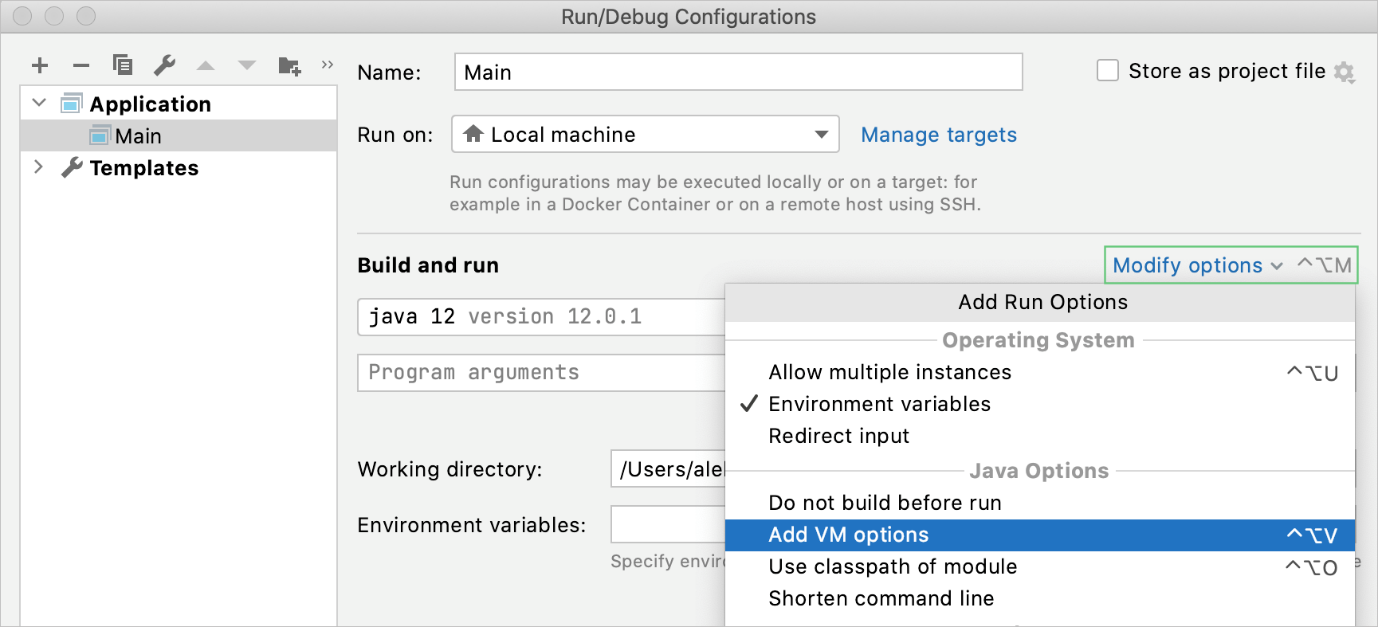


1. Apply the changes and close the **Project Structure** dialog.

**Add VM options﻿**

These actions are required if you use Java 11 and later. If you use Java 10 and earlier, you can [run the sample application](https://www.jetbrains.com/help/idea/javafx.html#run) created together with the project to make sure that everything is working as intended.

1. From the main menu, select **Run | Edit Configurations**.
2. Select **Application | Main** from the list on the left.
3. From the **More options** list, select **Add VM options**.



1. In the **VM options** field, specify the following options, but make sure to replace /path/to/javafx/sdk with the path the **lib** directory of the downloaded JavaFX SDK distribution (for example: **/Users/jetbrains/Desktop/javafx-sdk-12/lib**):

----module-path /path/to/javafx/sdk --add-modules javafx.controls,javafx.fxml

Copied!

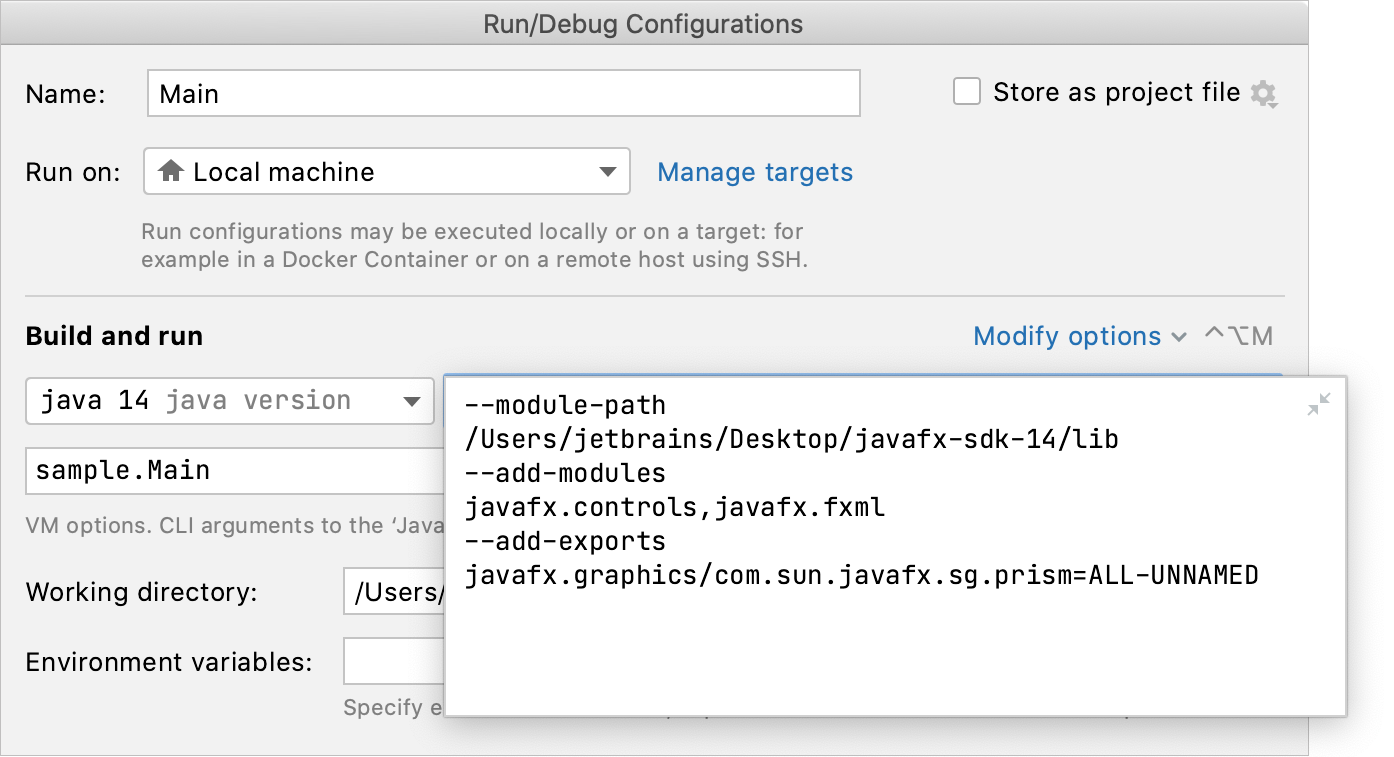
--module-path "D:\Software Development\Development Tools\sdk\javafx-sdk-11.0.2\lib" --add-modules javafx.controls,javafx.base,javafx.fxml,javafx.graphics,javafx.media,javafx.swing,javafx.web

If your path contains spaces, wrap it in double quotes, for example: **"/Users/jetbrains/Desktop/My Folder/javafx-sdk-12/lib"**.

If you're using JavaFX SDK 14 together with OpenJDK 14, add one more VM option:

--add-exports javafx.graphics/com.sun.javafx.sg.prism=ALL-UNNAMED

Copied!



1. Apply the change and close the dialog.

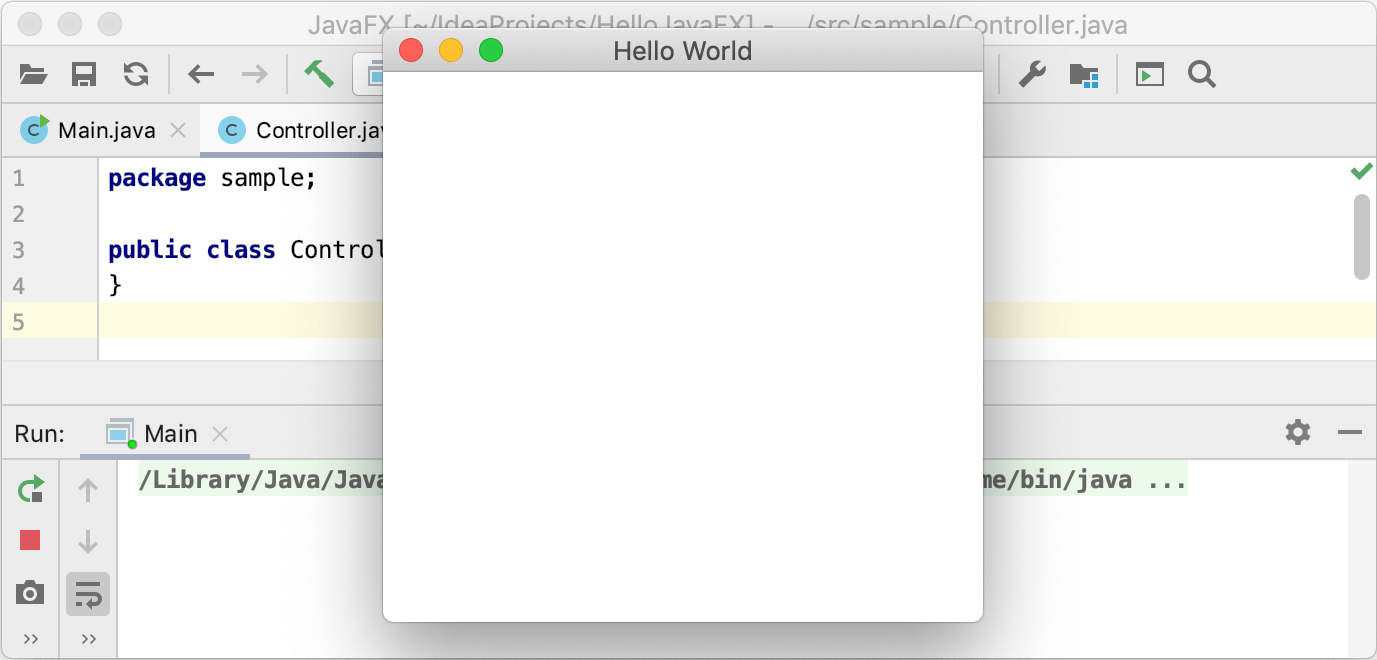
**Run the application﻿**

1. From the main menu, select **Run | Run 'Main'**Shift+F10.

The IDE starts compiling your code.

1. When the compilation is complete, the application window appears.

This means that the project is configured correctly and everything works as it should. The window is empty at the moment as we haven't added any elements yet.



For information on how to create modular and non-modular projects with Maven or Gradle, refer to [Getting Started with JavaFX](https://openjfx.io/openjfx-docs/#IDE-Intellij).

**Troubleshoot﻿**

* Error: JavaFX runtime components are missing, and are required to run this application

The Java launcher checks if the main class extends javafx.application.Application.

If you use Java 11 and later, make sure to specify [VM options](https://www.jetbrains.com/help/idea/javafx.html#vm-options) in the run configuration.

* Error occurred during initialization of boot layer java.lang.module.FindException: Module javafx.base not found

The [path to the JavaFX library](https://www.jetbrains.com/help/idea/javafx.html#path-vm-options) in the **VM options** field in the run configuration is likely to be incorrect. Make sure that the path is valid and try again.

* Process finished with exit code -1073740791 (0xC0000409)

The problem is caused by an error in the NVIDEA driver. Update your driver to the latest version. For more information, refer to the [community forum](https://intellij-support.jetbrains.com/hc/en-us/community/posts/115000060510-Process-finished-with-exit-code-1073740791-0xC0000409-JavaFX-Application).

* Error:Java FX Packager: Can't build artifact – fx:deploy is not available in this JDK

The **fx:deploy** task was a part of the Ant plugin that was formerly distributed in **ant-javafx.jar** as a part of Java Packager. The Ant plugin is not included in jpackage in the current JDK versions.

If you're using a JDK build of version 9 and later, use third-party solutions for packaging. For example, refer to section [Runtime images](https://openjfx.io/openjfx-docs/#modular) in the JavaFX official documentation.

**JavaFX 11 with Netbeans 11 and jdk 11**

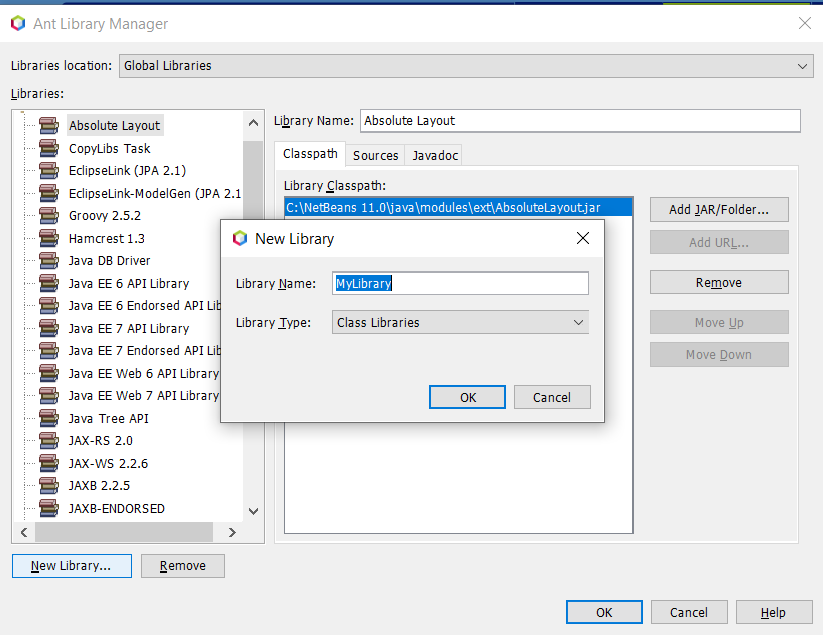
Go to Tools 🡪 Libraries

Make sure that the libraries location on top is “Global Libraries”

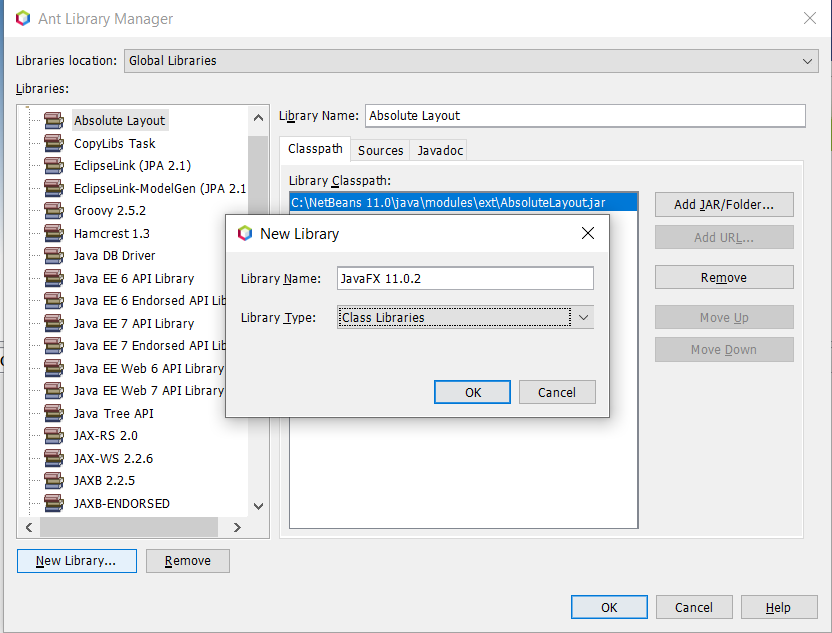
Click on New Library button (located at bottom left side panel) . Below pop up will be open.

Change the library name from ‘MyLibrary’ to the version of javaFX being added. In this case, I am adding JavaFX 11.0.2 hence the library name is ‘JavaFX 11.0.2’ (shown in Figure2).

Library type is class library.

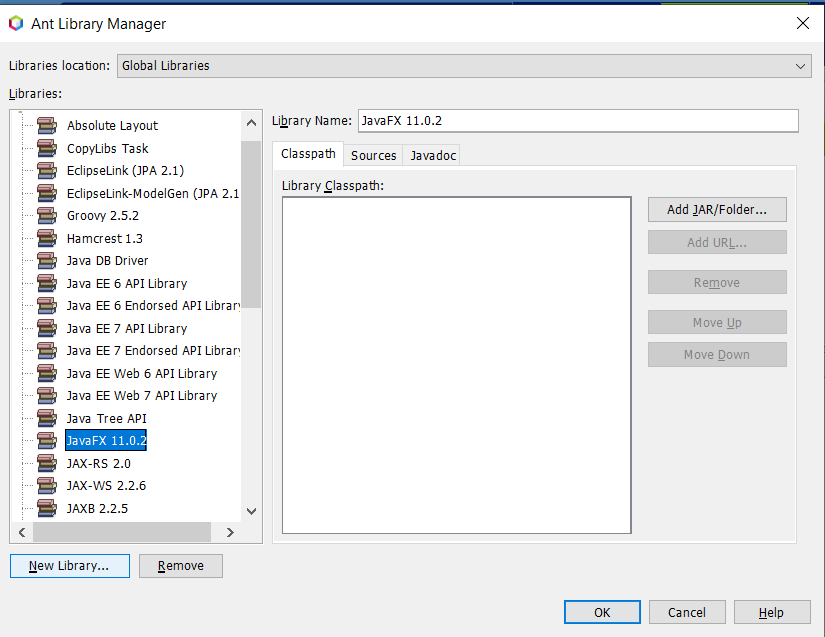


**Figure 1**



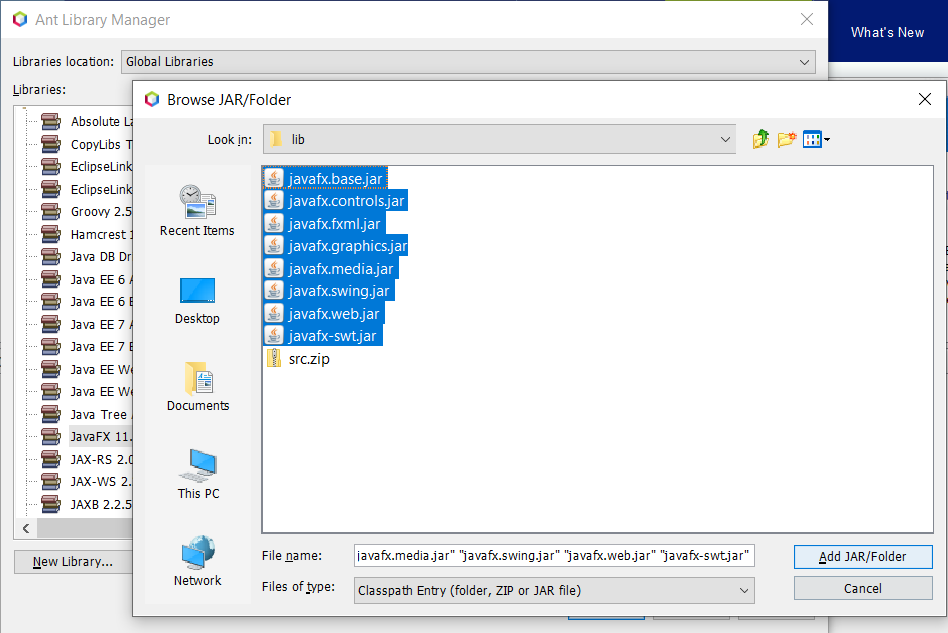
**Figure 2**

Click OK button shown in figure 2 and new library will be created at shown below.



**Figure 3**

Click on “Add JAR folder” button and pop up will open to select jar files from JavaFX SDK folder. (Shown in figure 4 below).



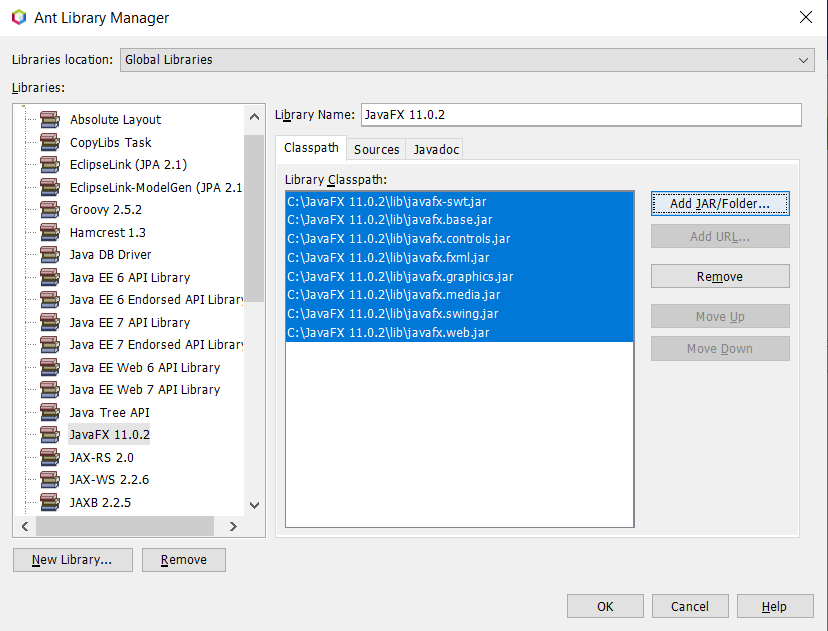
**Figure 4**

Go to ‘lib’ folder of the JavaFX SDK stored in the computer. In this case it is “C:\JavaFX 11.0.2\lib” (Shown in below figure 4).

Select all the jar files from the folder. DO NOT SELECT THE src.zip file.

Once selected click on the ‘Add JAR/Folder’ and the jar files will be added in the library as shown in below figure 5.

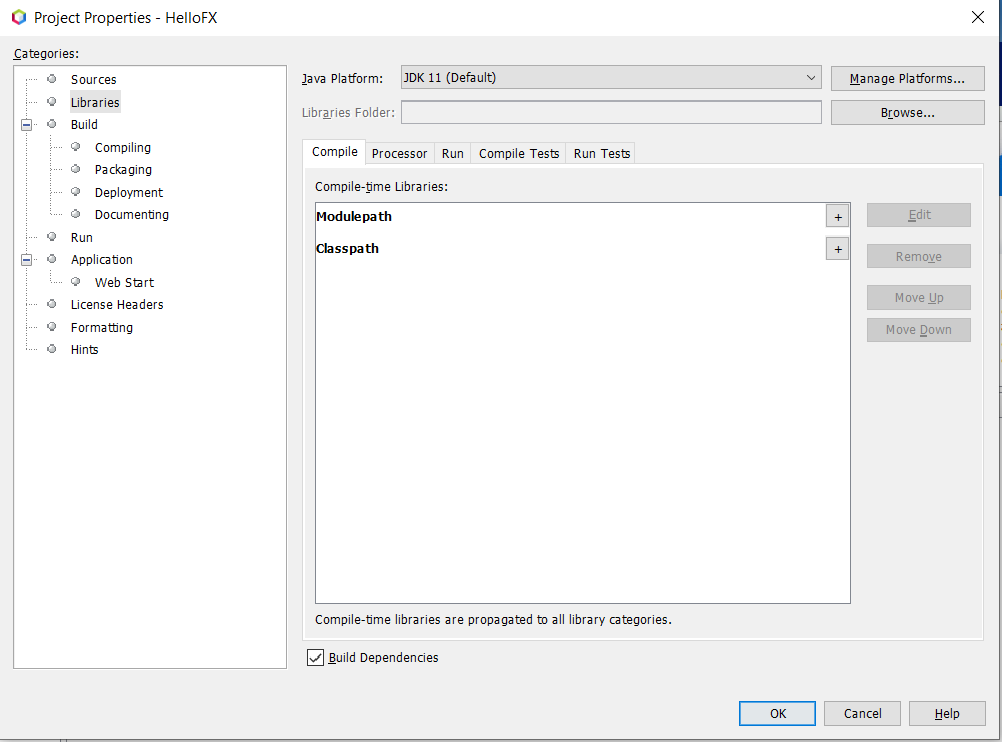
At last, click on OK button to come out of the libraries window.



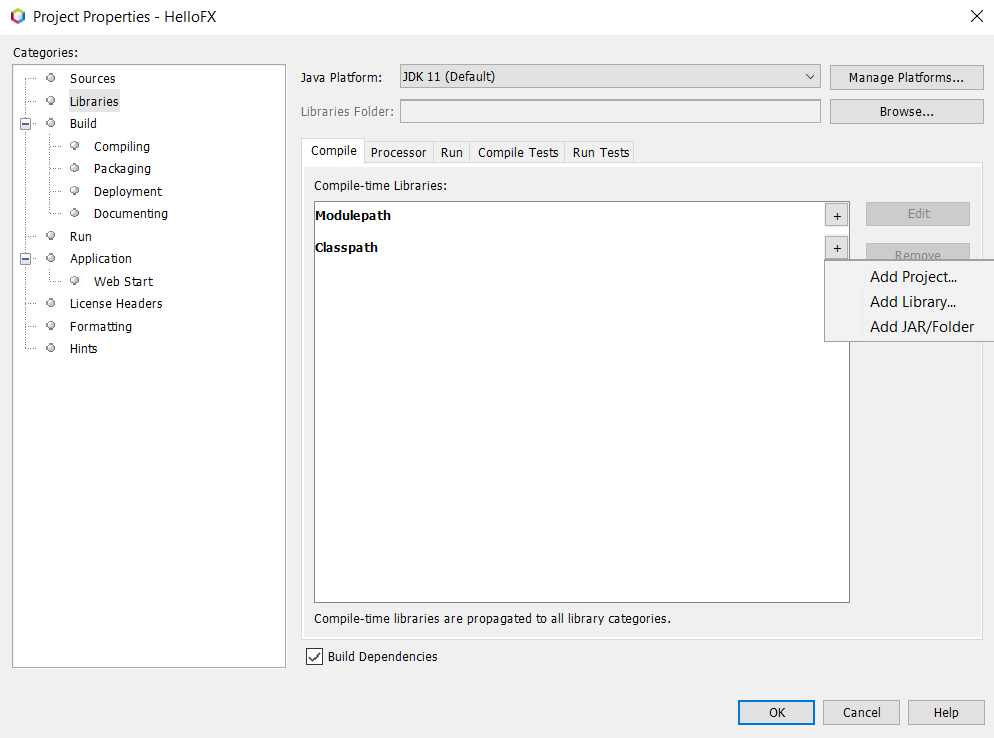
**Figure 5**

Set up for every project

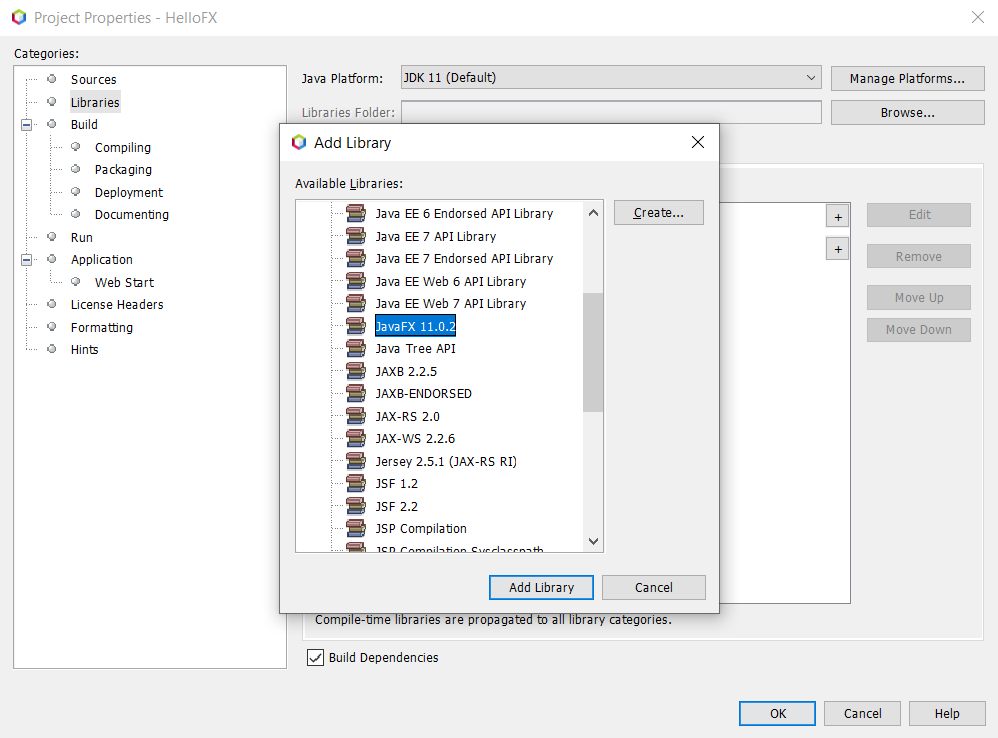
Right click on the NetBeans project and go to properties and select libraries from right side categories. Below window will open.



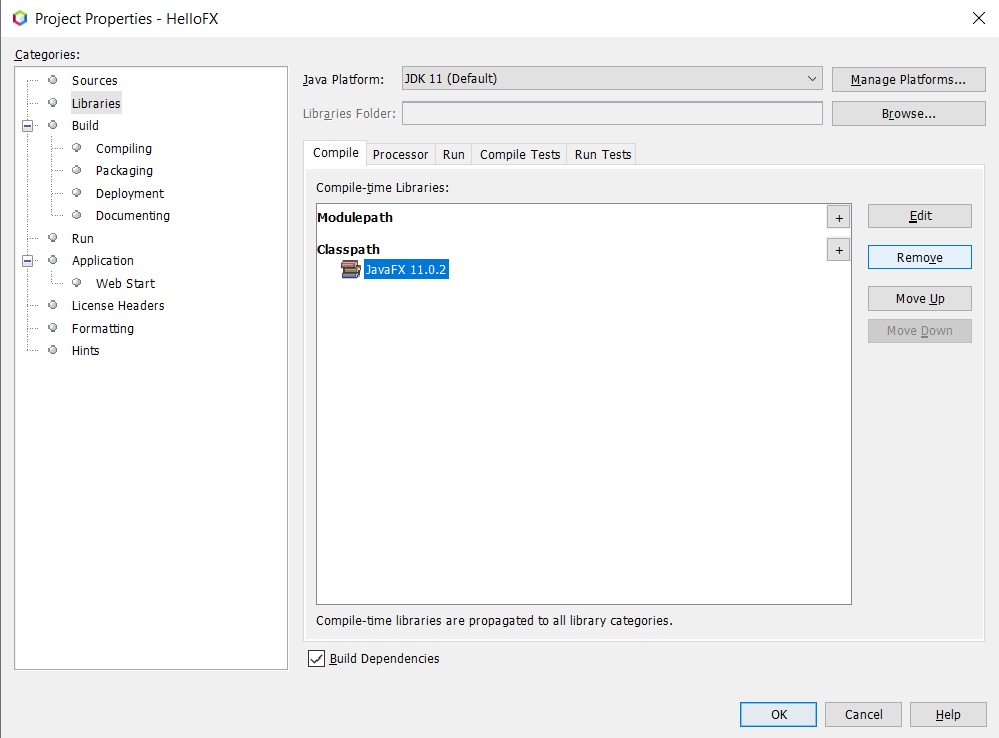
Under compile option, click on “+” sign in front of classpath and below window will open. Select “Add Library” from the option.



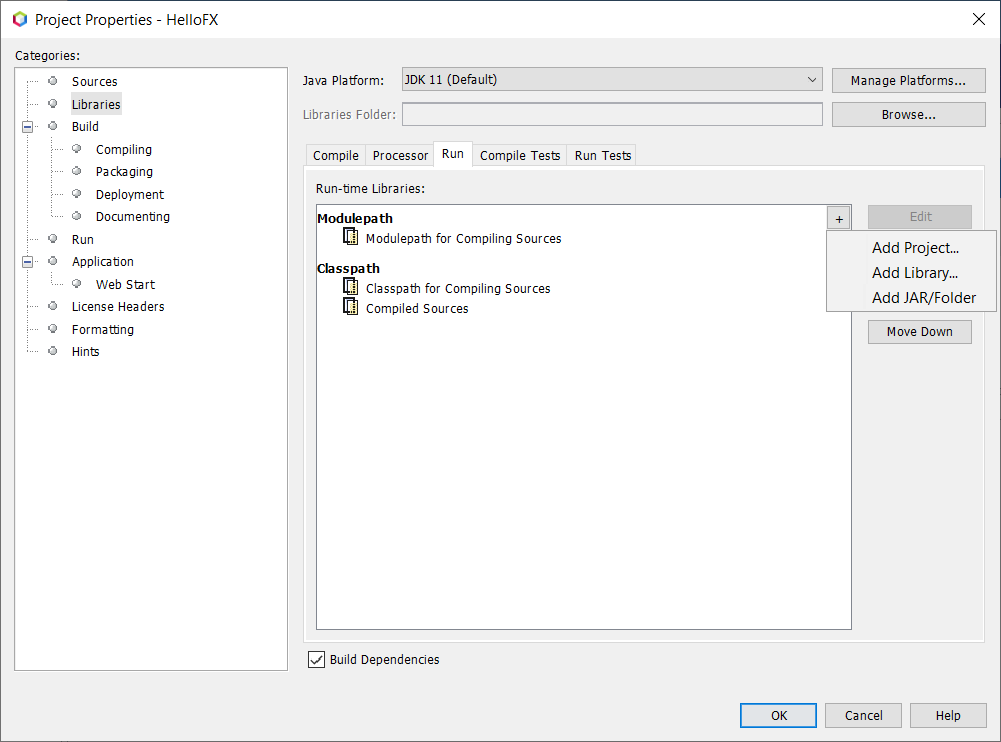
Select the “JavaFX 11.0.2” library from the list which we added in the first step and click on the “Add Library”.



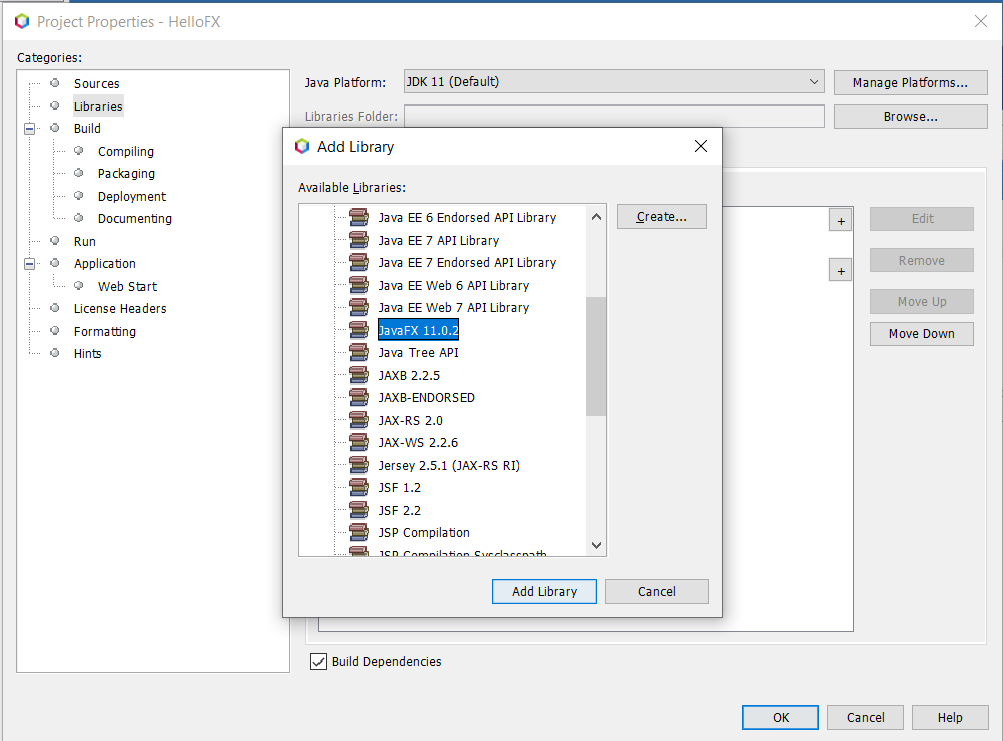
The library will be added under compile option as below.



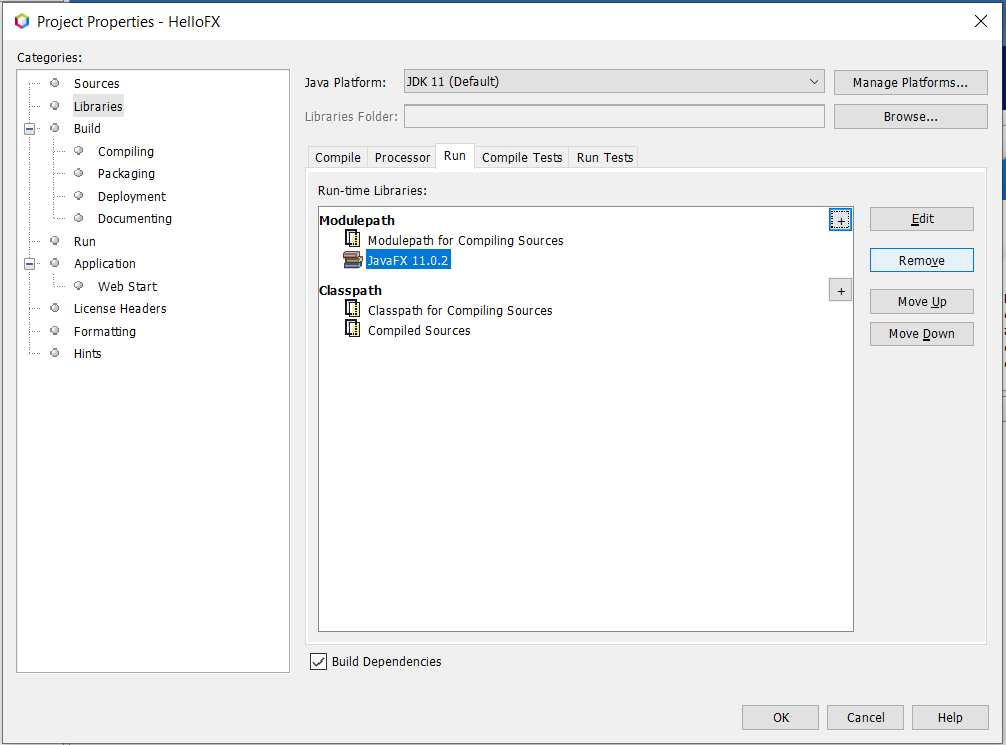
Next, click on Run option then click on “+” sign next to the ModulePath. Then select Add library option.



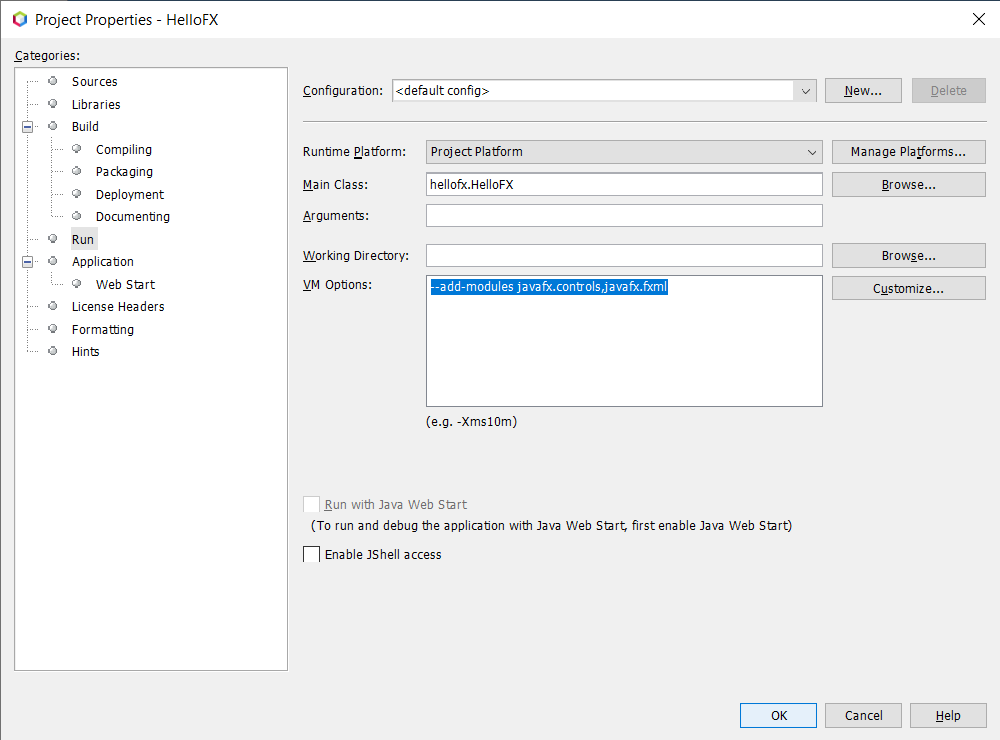
Choose “JavaFX 11.0.2” from the library as we did previously and then click on add “Add Library” button as shown below.



JavaFX library will added as below.



Now select “Run” category from the left side categories option (As shown below)



In “VM options:”, type the below command to add controls and fxml modules for the project.

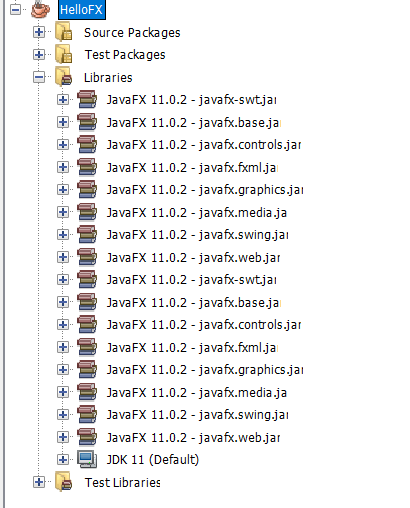
--add-modules javafx.controls,javafx.fxml

**OR**

--module-path "D:\Software Development\Development Tools\sdk\javafx-sdk-11.0.2\lib" --add-modules javafx.controls,javafx.base,javafx.fxml,javafx.graphics,javafx.media,javafx.swing,javafx.web

**Note**: You can add many more modules which requires for the project under VM options.

Once added, click on OK button. JavaFX libraries will be added under Libraries folder in the project as shown below.



Now, we can create and run the JavaFX project.