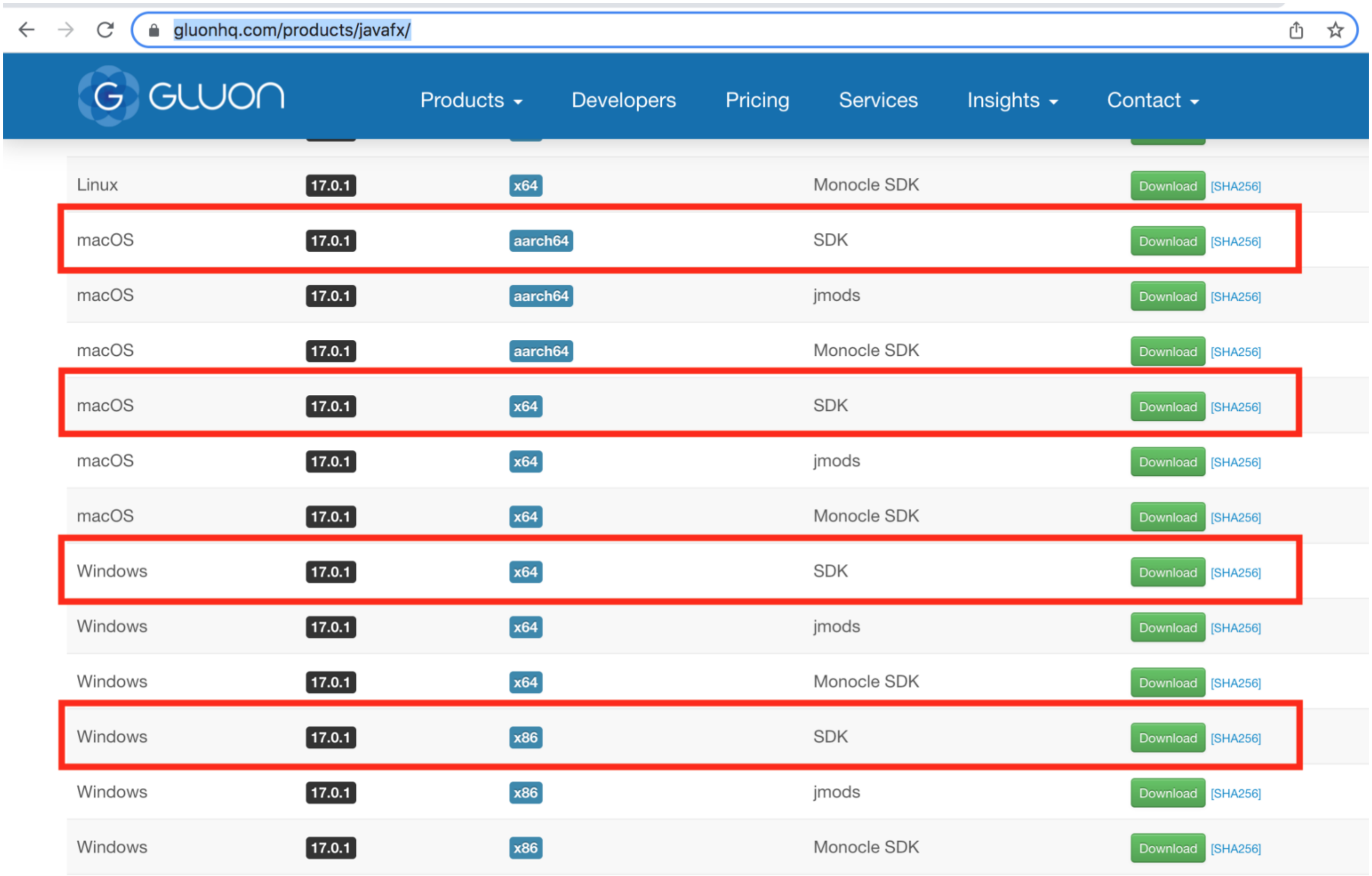
# How to add JavaFX to Eclipse (the easy way)

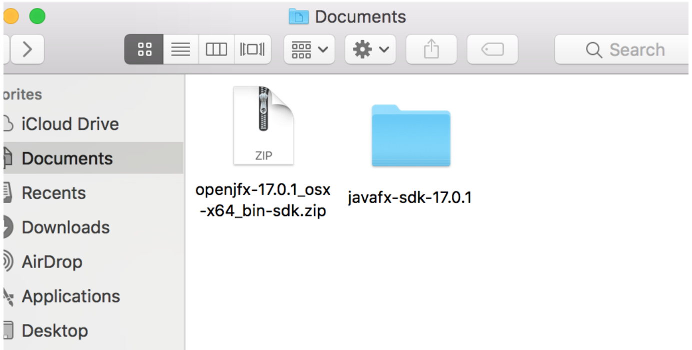
## STEP 1. DOWNLOAD JAVAFX

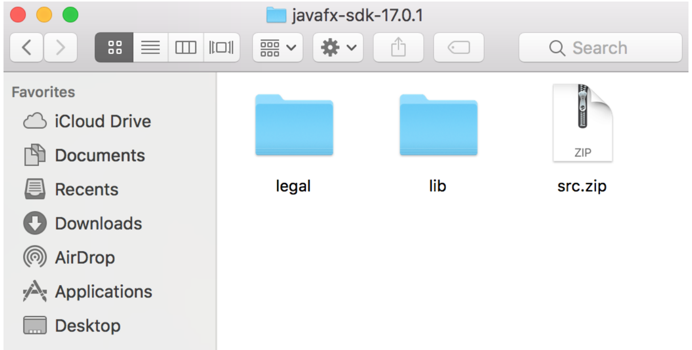
Go to [https://gluonhq.com/products/javafx/](https://gluonhq.com/products/javafx/) and download the appropriate **SDK** for your operating system. Make sure you download the SDK and make sure you choose the correct operating system and architecture for your computer. More than likely, you'll be downloading one of the outlined options below.



| | | | | | |
|---|---|---|---|---|---|
| Linux | 17.0.1 | x64 | Monocle SDK | Download | [SHA256] |
| macOS | 17.0.1 | aarch64 | SDK | Download | [SHA256] |
| macOS | 17.0.1 | aarch64 | jmods | Download | [SHA256] |
| macOS | 17.0.1 | aarch64 | Monocle SDK | Download | [SHA256] |
| macOS | 17.0.1 | x64 | SDK | Download | [SHA256] |
| macOS | 17.0.1 | x64 | jmods | Download | [SHA256] |
| macOS | 17.0.1 | x64 | Monocle SDK | Download | [SHA256] |
| Windows | 17.0.1 | x64 | SDK | Download | [SHA256] |
| Windows | 17.0.1 | x64 | jmods | Download | [SHA256] |
| Windows | 17.0.1 | x64 | Monocle SDK | Download | [SHA256] |
| Windows | 17.0.1 | x86 | SDK | Download | [SHA256] |
| Windows | 17.0.1 | x86 | jmods | Download | [SHA256] |
| Windows | 17.0.1 | x86 | Monocle SDK | Download | [SHA256] |

This will install a `.zip` file. Unzip this file and put the resulting folder in a memorable place, such as your `/Documents` folder. We'll need this later.

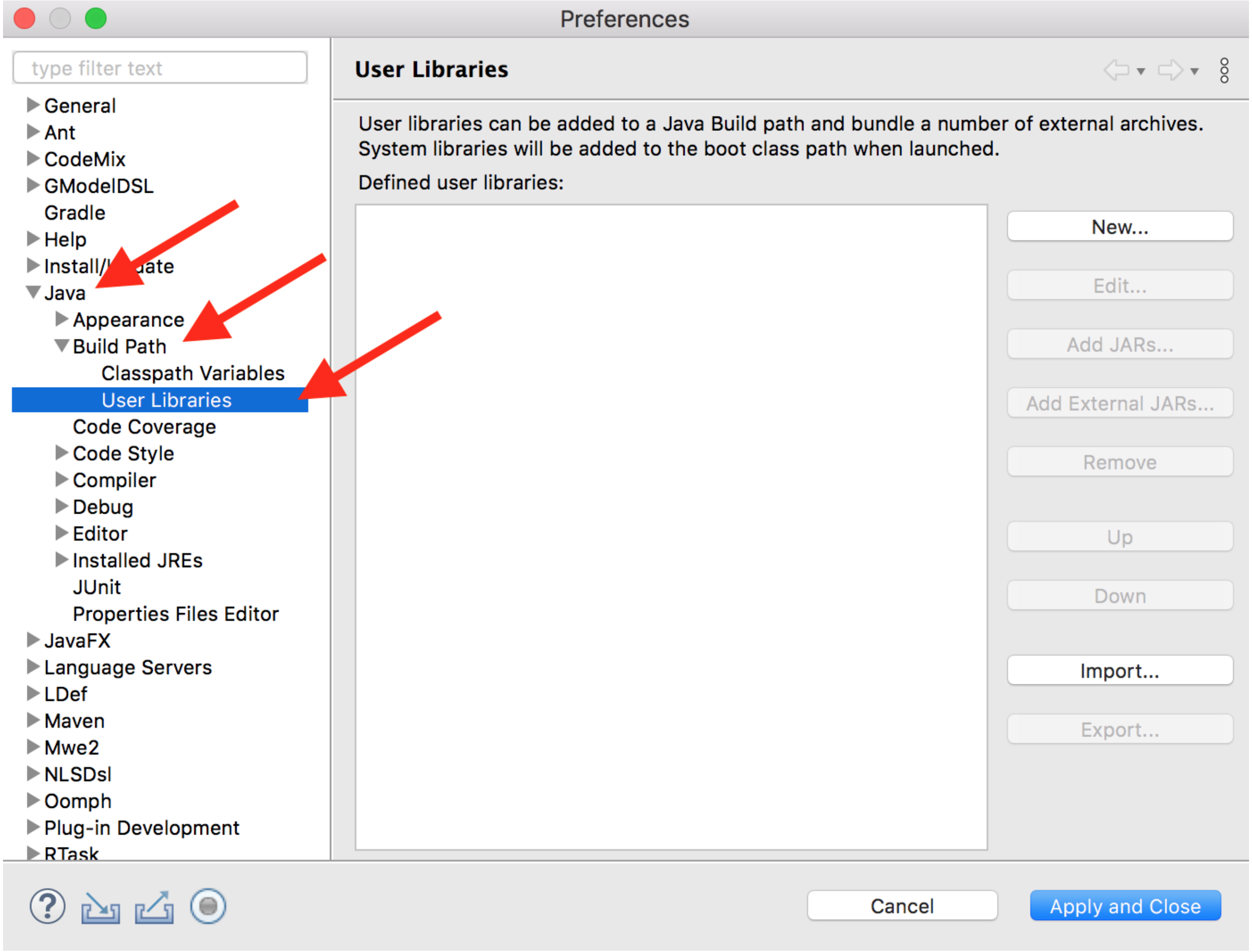




## STEP 2. CREATE A USER LIBRARY IN ECLIPSE

- Open up Eclipse and go to **Preferences**
- You should be able to get to Preferences from Eclipse > Preferences or Window > Preferences

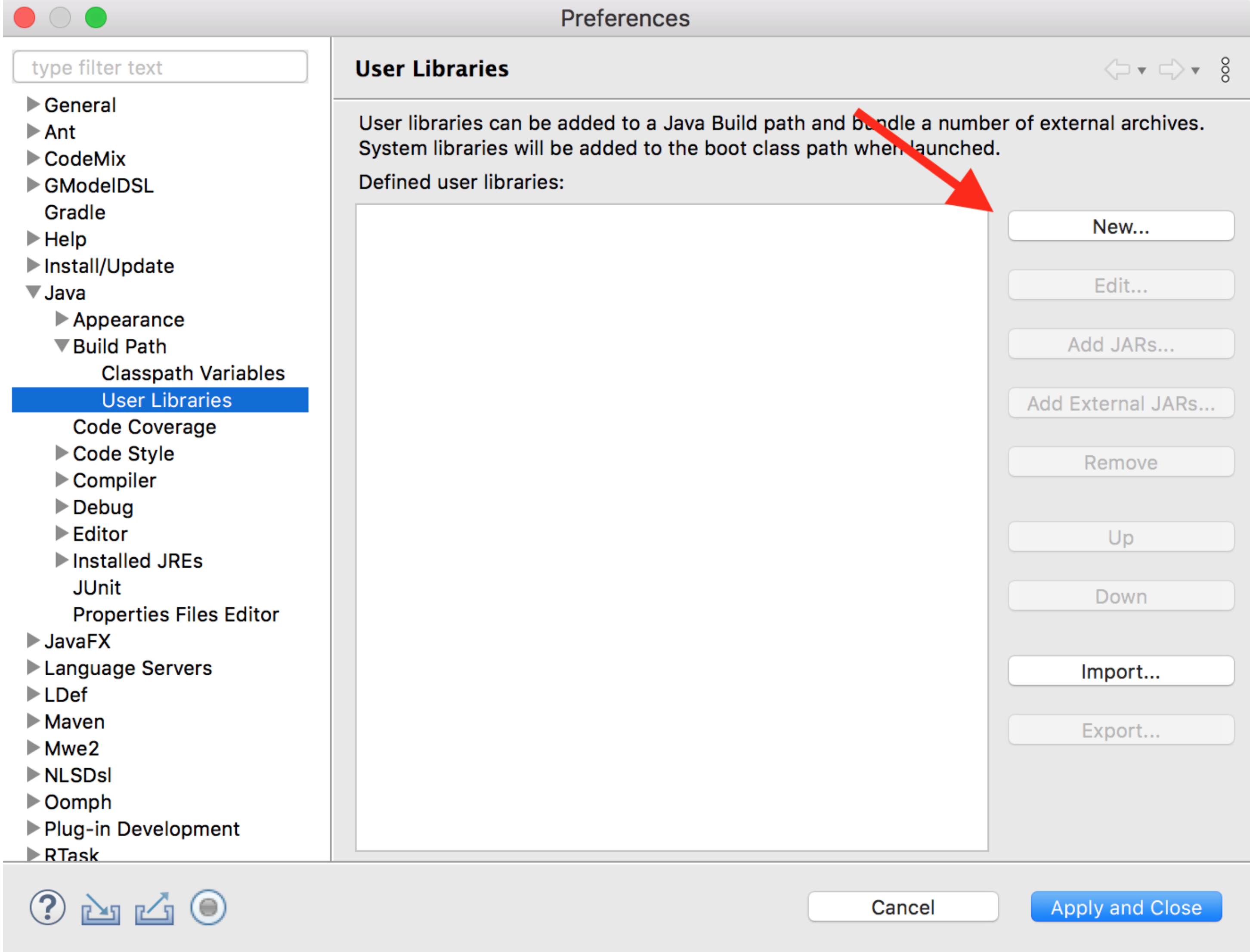
- Go to **Java** > **Build Path** > **User Libraries**



User libraries can be added to a Java Build path and bundle a number of external archives. System libraries will be added to the boot class path when launched.

Defined user libraries:

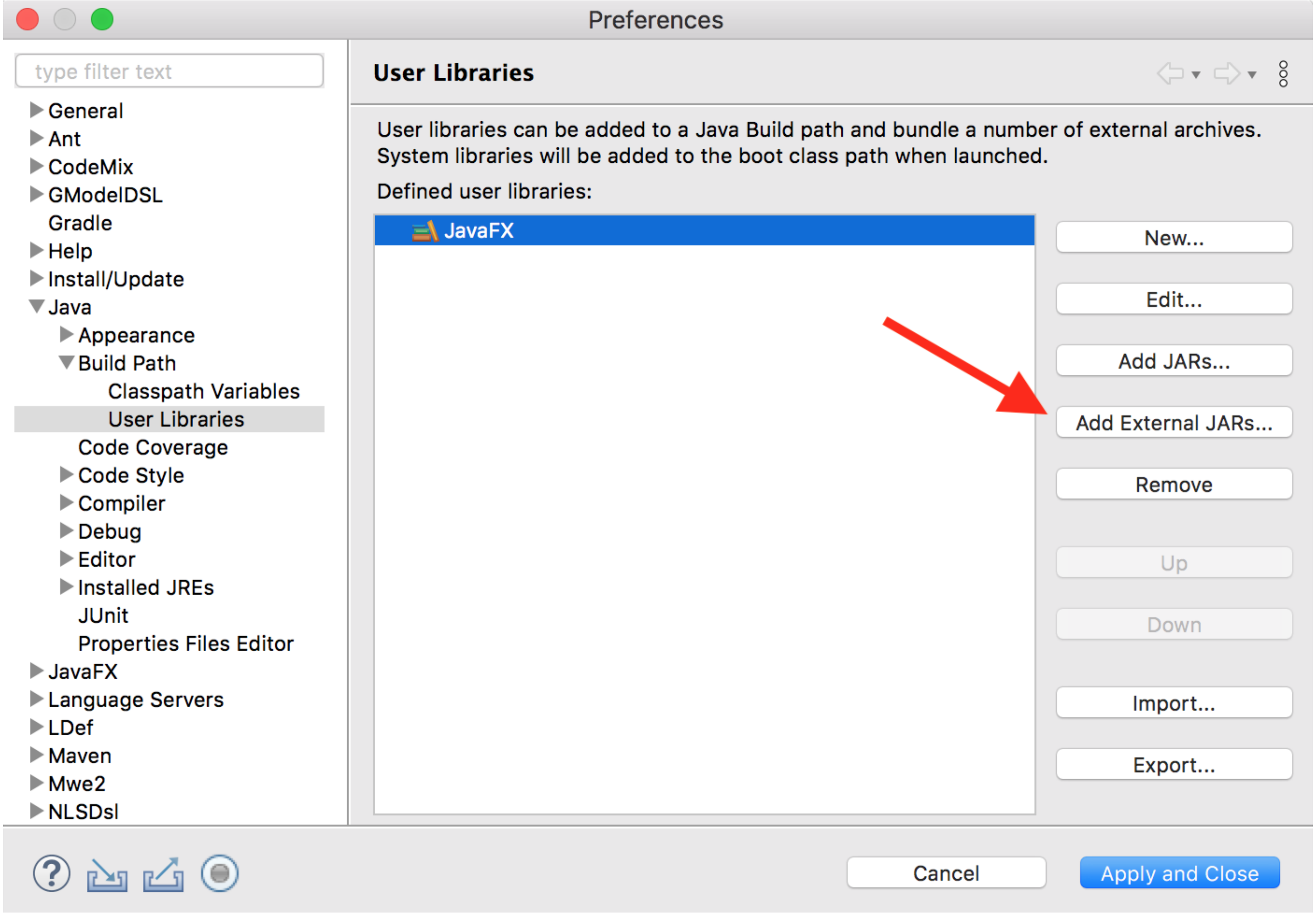
- Now click **New...**



User libraries can be added to a Java Build path and bundle a number of external archives. System libraries will be added to the boot class path when launched.

Defined user libraries:

- Name it **JavaFX** and click **Ok**
- With JavaFX highlighted, click **Add External JARs...**

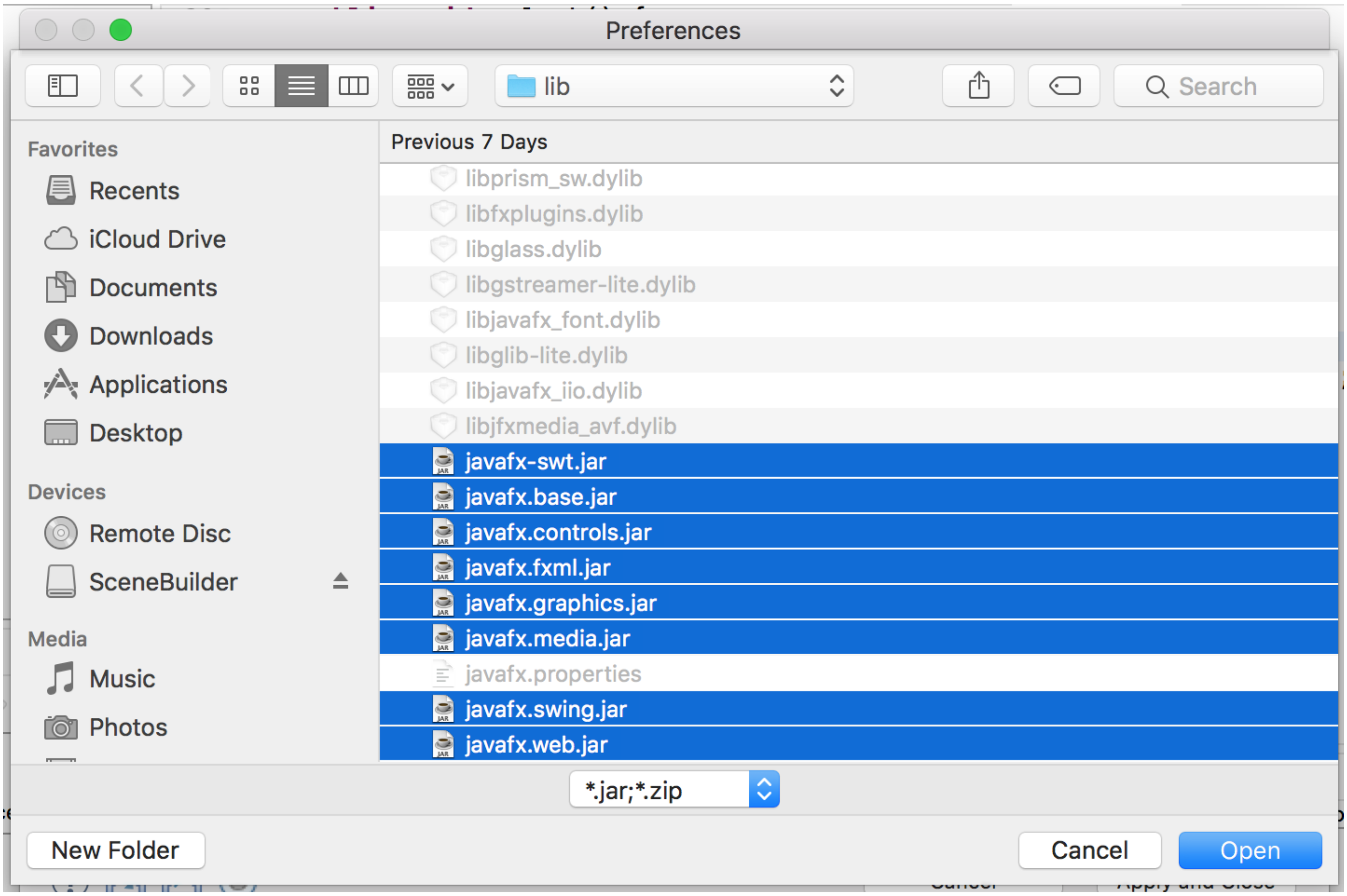


User libraries can be added to a Java Build path and bundle a number of external archives. System libraries will be added to the boot class path when launched.

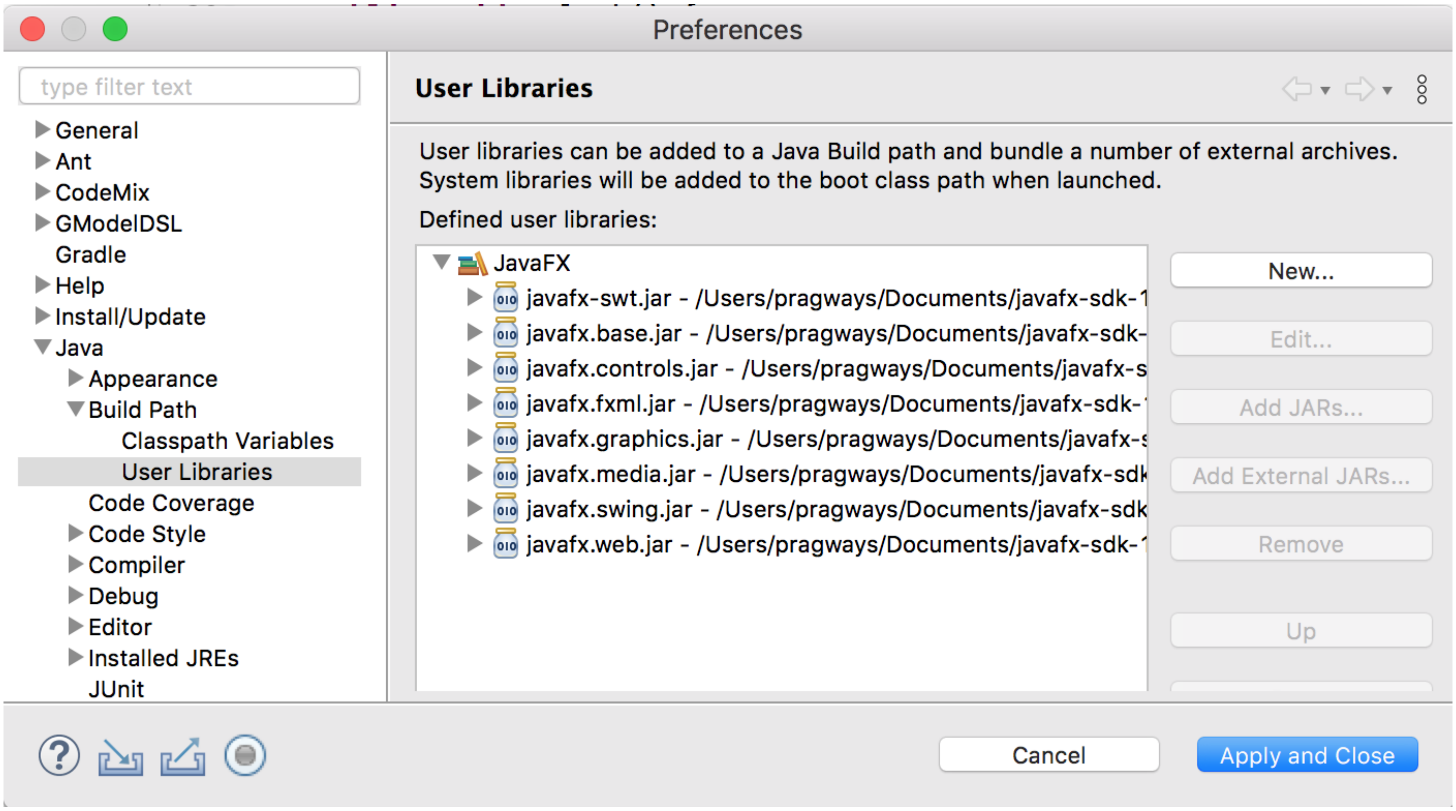
Defined user libraries:

JavaFX

- Select all the `.jar` files from the `/lib/` folder of the unzipped JavaFX folder we saved earlier.



- Click **Open** and your new User Library should look something like this:



**User Libraries**

User libraries can be added to a Java Build path and bundle a number of external archives. System libraries will be added to the boot class path when launched.

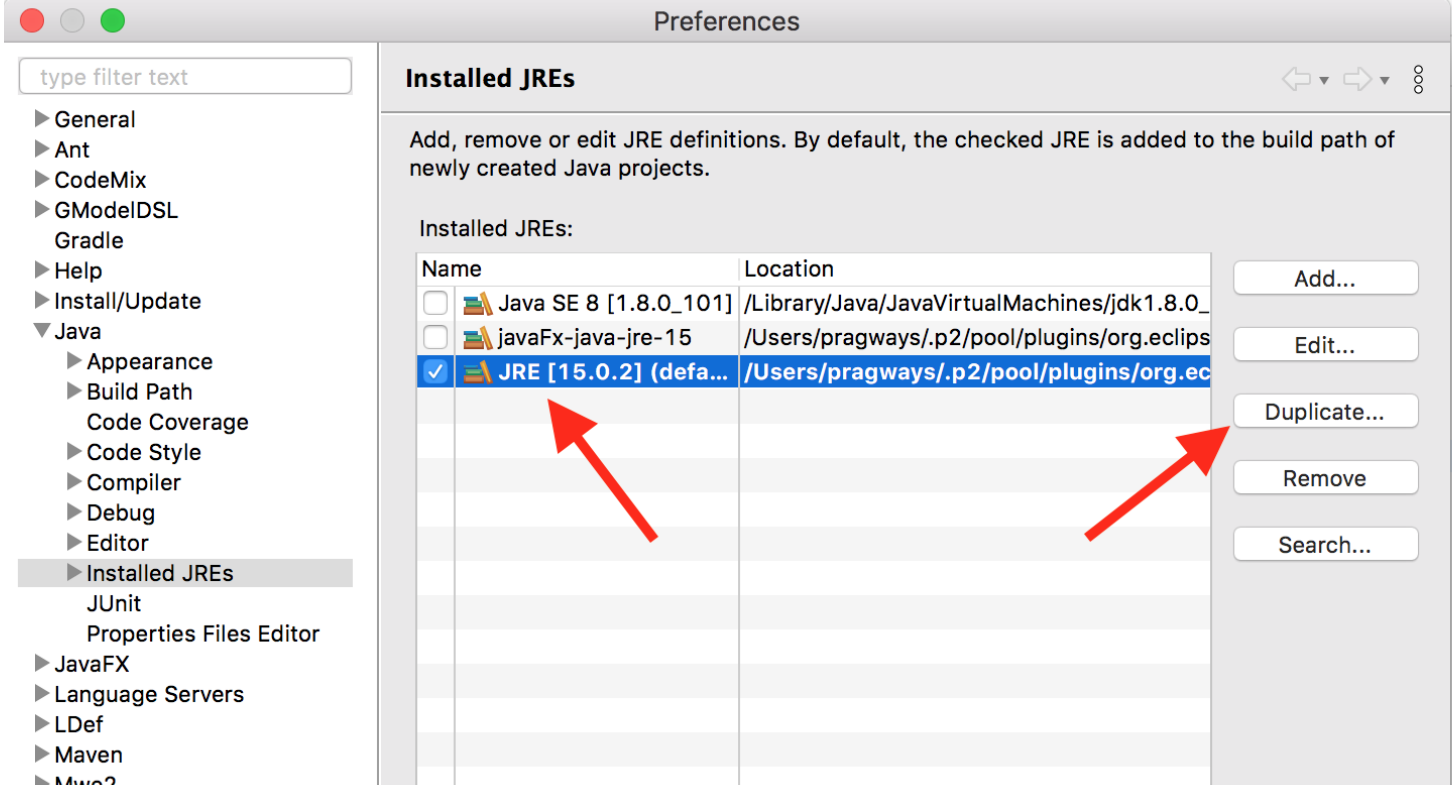
Defined user libraries:

```
▼ JavaFX
    ▶ javafx-swt.jar - /Users/pragways/Documents/javafx-sdk-1
    ▶ javafx.base.jar - /Users/pragways/Documents/javafx-sdk-
    ▶ javafx.controls.jar - /Users/pragways/Documents/javafx-s
    ▶ javafx.fxml.jar - /Users/pragways/Documents/javafx-sdk-
    ▶ javafx.graphics.jar - /Users/pragways/Documents/javafx-s
    ▶ javafx.media.jar - /Users/pragways/Documents/javafx-sdk
    ▶ javafx.swing.jar - /Users/pragways/Documents/javafx-sdk
    ▶ javafx.web.jar - /Users/pragways/Documents/javafx-sdk-1
```

- Click **Apply and Close** to save your new User Library

## STEP 3. CREATE A JRE CLONE WITH THE REQUIRED VM ARGUMENTS

- Go back to **Preferences** in Eclipse
- Go to **Java** > **Installed JREs**
- Select your default JRE and click **Duplicate**



Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

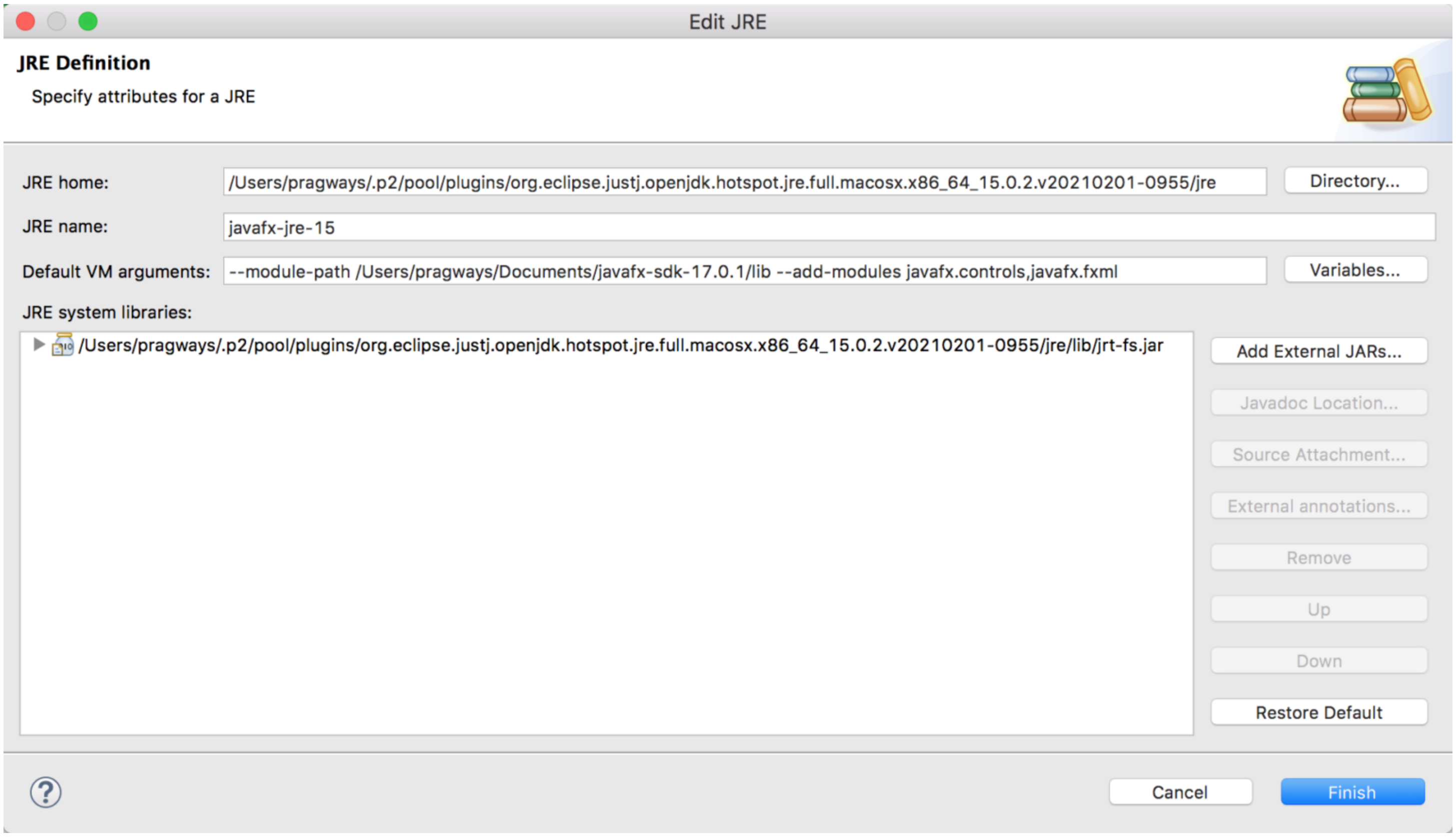
| | Name | Location |
|---|---|---|
| ☐ | Java SE 8 [1.8.0_101] | /Library/Java/JavaVirtualMachines/jdk1.8.0_ |
| ☐ | javaFx-java-jre-15 | /Users/pragways/.p2/pool/plugins/org.eclips |
| ☑ | JRE [15.0.2] (defa... | /Users/pragways/.p2/pool/plugins/org.ec |

- Copy this line if you're on **Windows**
- `--module-path "\path\to\javafx-sdk-17\lib" --add-modules javafx.controls,javafx.fxml`
- ~~Copy this line if you're on **Mac** or **Linux**~~
- ~~`--module-path /path/to/javafx-sdk-17/lib --add-modules javafx.controls,javafx.fxml`~~

- **Important:** Make sure you replace the `/path/to/javafx-sdk-17/lib` with the path to where you placed your unzipped JavaFX folder from before. You will want the entire path all the way to the /lib/ folder. For instance, mine on a Mac looks like this:
- `--module-path /Users/pragways/Documents/javafx-sdk-17.0.1/lib --add-modules javafx.controls,javafx.fxml`

- Paste that line in the **Default VM arguments:** field

- Rename the **JRE name:** field to something memorable, such as **javafx-jre-15**



Edit JRE

**JRE Definition**

Specify attributes for a JRE

JRE home: /Users/pragways/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_15.0.2.v20210201-0955/jre

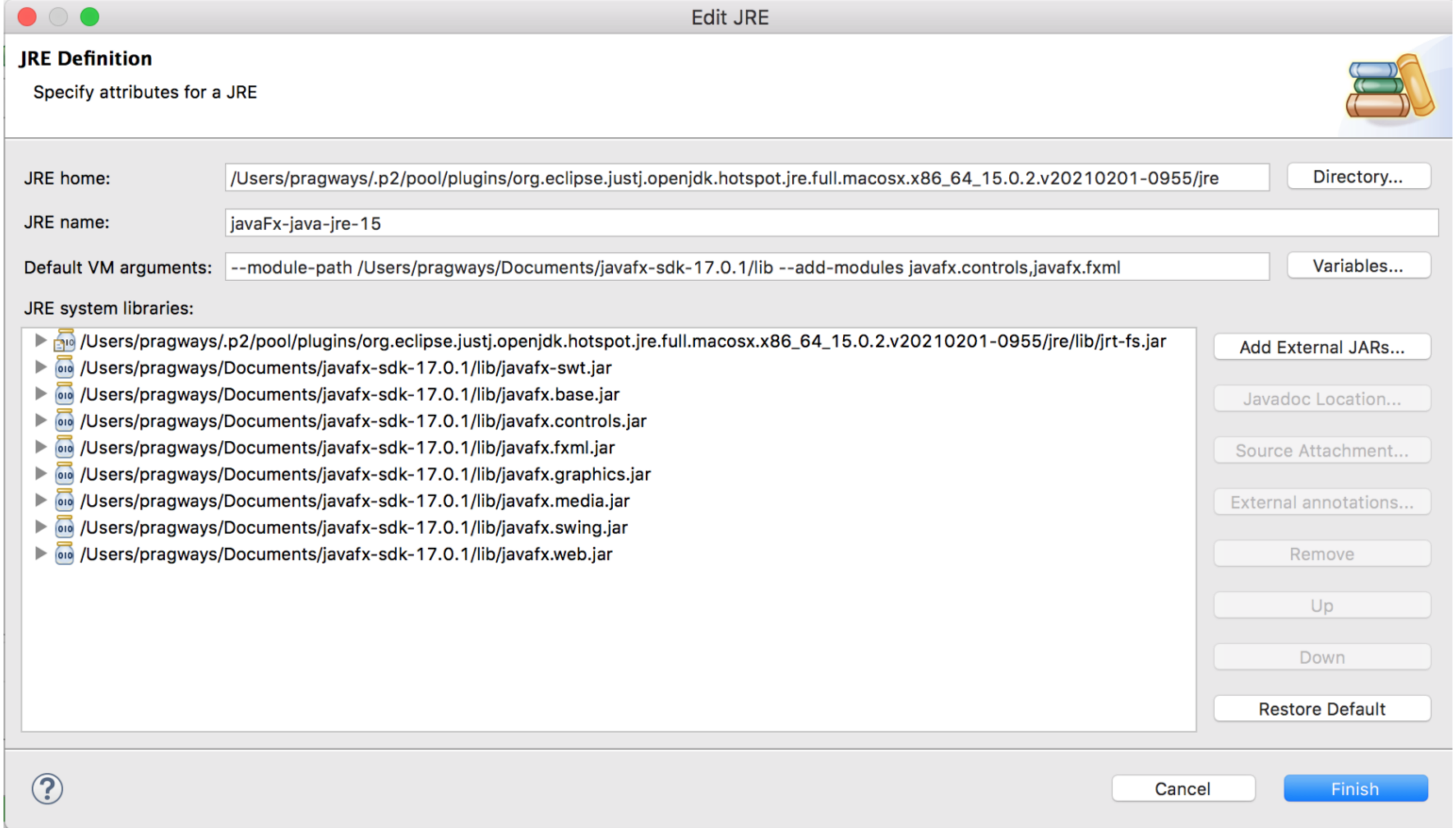
JRE name: javafx-jre-15

Default VM arguments: --module-path /Users/pragways/Documents/javafx-sdk-17.0.1/lib --add-modules javafx.controls,javafx.fxml

JRE system libraries:

/Users/pragways/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_15.0.2.v20210201-0955/jre/lib/jrt-fs.jar

- Click **Add External JARs...**
- Select all the `.jar` files from the `/lib/` folder of the unzipped JavaFX folder we saved earlier and then click **Open**



Edit JRE

**JRE Definition**

Specify attributes for a JRE

JRE home: /Users/pragways/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_15.0.2.v20210201-0955/jre

JRE name: javaFx-java-jre-15

Default VM arguments: --module-path /Users/pragways/Documents/javafx-sdk-17.0.1/lib --add-modules javafx.controls,javafx.fxml

JRE system libraries:

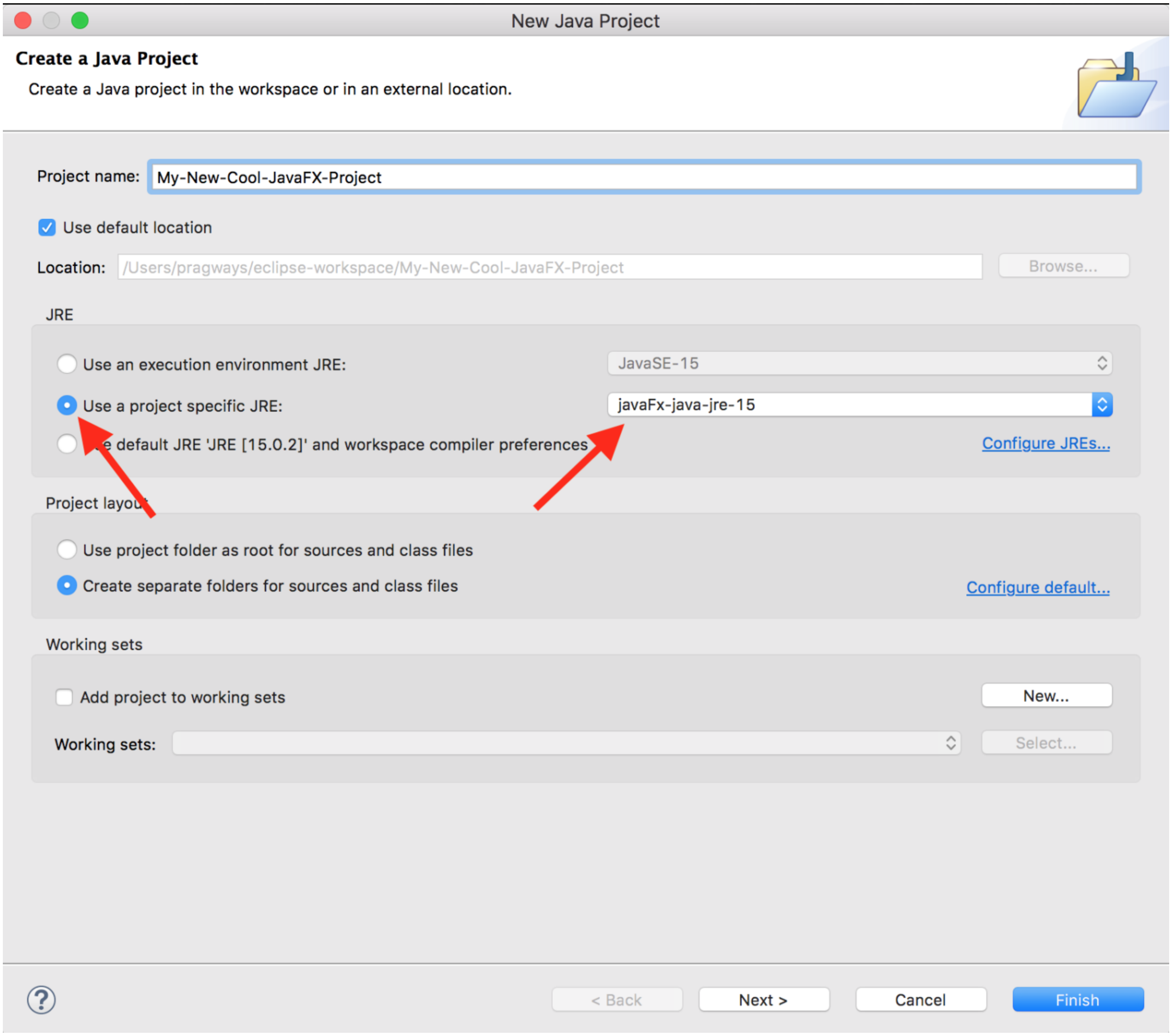
```
/Users/pragways/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_15.0.2.v20210201-0955/jre/lib/jrt-fs.jar
/Users/pragways/Documents/javafx-sdk-17.0.1/lib/javafx-swt.jar
/Users/pragways/Documents/javafx-sdk-17.0.1/lib/javafx.base.jar
/Users/pragways/Documents/javafx-sdk-17.0.1/lib/javafx.controls.jar
/Users/pragways/Documents/javafx-sdk-17.0.1/lib/javafx.fxml.jar
/Users/pragways/Documents/javafx-sdk-17.0.1/lib/javafx.graphics.jar
/Users/pragways/Documents/javafx-sdk-17.0.1/lib/javafx.media.jar
/Users/pragways/Documents/javafx-sdk-17.0.1/lib/javafx.swing.jar
/Users/pragways/Documents/javafx-sdk-17.0.1/lib/javafx.web.jar
```

- Your JRE Definition window should now look something like this
- Click **Finish**

## STEP 4. CREATE A NEW PROJECT WITH YOUR NEW JRE

Now whenever you want to create a new JavaFX project, it will be a lot quicker and easier!

All you need to do is make sure you select **Use a project specific JRE** when creating a new project, and then select your new JavaFX specific JRE that you just created.



**Create a Java Project**

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

Project name: My-New-Cool-JavaFX-Project

☑ Use default location

Location: /Users/pragways/eclipse-workspace/My-New-Cool-JavaFX-Project

JRE

○ Use an execution environment JRE: JavaSE-15

● Use a project specific JRE: javaFx-java-jre-15

○ Use default JRE 'JRE [15.0.2]' 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:

## Adam Allard

Hi, I'm Adam Allard. I'm a Full Stack Software Engineer for Northrop Grumman creating web applications for the DoD. At this time I'm primarily working with Java and Angular based applications, although I have some years of experience in various languages and frameworks from previous jobs where I dabbled with Python & Django, C# & Xamarin, PHP, and Bootstrap. My hobbies include time with my family, wondering when the Green Bay Packers will win their next Super Bowl, drinking over-priced beer, and of course learning and teaching.