

OpenJFX Installation Guide for macOS

Prerequisites: Installing Java

Before installing OpenJFX, you need to have Java installed on your system.

⚠ IMPORTANT - Version Compatibility: The OpenJFX version must match your Java version. For example: - Java 11 → OpenJFX 11 - Java 17 → OpenJFX 17 - Java 21 → OpenJFX 21

Install Java via Homebrew

```
brew install openjdk
```

Configure the PATH by adding to your ~/.zshrc (or ~/.bash_profile if you use bash):

```
# For Apple Silicon (M1/M2/M3)
export PATH="/opt/homebrew/opt/openjdk/bin:$PATH"

# For Intel Mac
# export PATH="/usr/local/opt/openjdk/bin:$PATH"
```

Reload your configuration:

```
source ~/.zshrc
```

Verify the installation:

```
java -version
javac -version
```

Install a Specific Java Version

If you need a particular version:

```
# Java 11
brew install openjdk@11
```

```
# Java 17 (LTS)
brew install openjdk@17
```

```
# Java 21 (LTS)
brew install openjdk@21
```

Then configure the PATH with the desired version:

```
# For Java 17 for example
export PATH="/opt/homebrew/opt/openjdk@17/bin:$PATH"
```

Installing OpenJFX

Method 1: Via Homebrew (recommended)

```
brew install openjdk openjfx
```

This command installs Java (JDK) and OpenJFX.

If you already have Java installed, you can install only OpenJFX:

```
brew install openjfx
```

Method 2: Manual Installation from Official Website

1. Download OpenJFX from openjfx.io
2. Choose the macOS version (SDK)
3. Extract the archive to a directory, for example ~/javafx or /Library/Java/JavaFX
4. Note the path to the lib directory

Verifying the Installation

Verify Java

```
java -version  
javac -version
```

You should see the installed Java version (OpenJDK).

Verify OpenJFX

```
brew list openjfx
```

To find the installation path:

```
brew info openjfx
```

The typical path is /opt/homebrew/opt/openjfx/lib (Apple Silicon) or /usr/local/opt/openjfx/lib (Intel).

Verify Version Compatibility

```
# Check Java version  
java -version
```

```
# Check OpenJFX version  
brew info openjfx | grep openjfx:
```

The major version numbers must match (e.g., Java 17.x.x with OpenJFX 17.x.x).

JavaFX Application Launch Script

Create a file named launch-app.sh:

```
#!/bin/bash

# Move to script directory
cd "$(dirname "$0")"

# Automatic architecture detection
if [ -d "/opt/homebrew/opt/openjfx/lib" ]; then
    # Apple Silicon
    JAVAFX_PATH="/opt/homebrew/opt/openjfx/lib"
elif [ -d "/usr/local/opt/openjfx/lib" ]; then
    # Intel
    JAVAFX_PATH="/usr/local/opt/openjfx/lib"
else
    echo "OpenJFX not found. Install it with: brew install openjfx"
    exit 1
fi

CLASSPATH="lib/*:EspFx.jar:Thermoptim.jar" # Add lib folder and main jar
MAIN_CLASS="rg.espfx.EspFx"

# Required JavaFX modules
MODULES="javafx.controls,javafx.fxml,javafx.graphics,javafx.base"

# Launch application with macOS-specific options
java --module-path "$JAVAFX_PATH" \
    --add-modules "$MODULES" \
    -cp "$CLASSPATH" \
    -Xdock:name="EspFx" \
    -Dapple.laf.useScreenMenuBar=true \
    "$MAIN_CLASS"
```

Make the Script Executable

```
chmod +x launch-app.sh
```

Launch the Application

```
./launch-app.sh
```

Universal Script (Intel + Apple Silicon)

For a script that works on both architectures:

```
#!/bin/bash

# Automatic architecture detection
if [ -d "/opt/homebrew/opt/openjfx/lib" ]; then
    # Apple Silicon
    JAVAFX_PATH="/opt/homebrew/opt/openjfx/lib"
elif [ -d "/usr/local/opt/openjfx/lib" ]; then
```

```

# Intel
JAVAFX_PATH="/usr/local/opt/openjfx/lib"
else
    echo "OpenJFX not found. Install it with: brew install openjfx"
    exit 1
fi

APP_JAR="MyApplication.jar"
MODULES="javafx.controls,javafx.fxml,javafx.graphics,javafx.base"

java --module-path "$JAVAFX_PATH" \
    --add-modules "$MODULES" \
    -jar "$APP_JAR"

```

Configuration for Manual Installation

If you installed OpenJFX manually, modify the path in the script:

```
JAVAFX_PATH="$HOME/javafx/lib" # Adjust according to your installation
```

Available JavaFX Modules

Adjust the MODULES variable according to your needs:

- javafx.base - Base classes
- javafx.controls - UI components (buttons, tables, etc.)
- javafx.fxml - FXML support
- javafx.graphics - Graphics and rendering
- javafx.media - Multimedia support
- javafx.swing - Swing integration
- javafx.web - Embedded web browser

Complete Example with JVM Options

```

#!/bin/bash

# Detect OpenJFX path
if [ -d "/opt/homebrew/opt/openjfx/lib" ]; then
    JAVAFX_PATH="/opt/homebrew/opt/openjfx/lib"
elif [ -d "/usr/local/opt/openjfx/lib" ]; then
    JAVAFX_PATH="/usr/local/opt/openjfx/lib"
else
    echo "Error: OpenJFX not found"
    exit 1
fi

APP_JAR="MyApplication.jar"
MODULES="javafx.controls,javafx.fxml"

```

```
# macOS-specific options
java --module-path "$JAVAFX_PATH" \
  --add-modules "$MODULES" \
  -Xmx512m \
  -Xdock:name="My Application" \
  -Xdock:icon="icon.icns" \
  -Dapple.laf.useScreenMenuBar=true \
  -Dfile.encoding=UTF-8 \
  -jar "$APP_JAR" "$@"
```

macOS-specific options: - -Xdock:name: Name displayed in the Dock - -Xdock:icon: Custom icon in the Dock - -Dapple.laf.useScreenMenuBar=true: Menu in the macOS menu bar

Creating a .app Application

To create a real macOS application, create the following structure:

```
MyApp.app/
├── Contents/
│   ├── Info.plist
│   ├── MacOS/
│   │   └── launcher.sh
│   └── Resources/
│       ├── MyApplication.jar
│       └── icon.icns
```

Info.plist file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>CFBundleName</key>
  <string>My Application</string>
  <key>CFBundleExecutable</key>
  <string>launcher.sh</string>
  <key>CFBundleIconFile</key>
  <string>icon.icns</string>
  <key>CFBundleIdentifier</key>
  <string>com.example.myapp</string>
  <key>CFBundleVersion</key>
  <string>1.0</string>
</dict>
</plist>
```

launcher.sh file in MacOS/:

```
#!/bin/bash
cd "$(dirname "$0")/../Resources"

if [ -d "/opt/homebrew/opt/openjfx/lib" ]; then
    JAVAFX_PATH="/opt/homebrew/opt/openjfx/lib"
else
    JAVAFX_PATH="/usr/local/opt/openjfx/lib"
fi

java --module-path "$JAVAFX_PATH" \
    --add-modules javafx.controls,javafx.fxml \
    -Xdock:name="My Application" \
    -jar MyApplication.jar
```

Make the script executable:

```
chmod +x MyApp.app/Contents/MacOS/launcher.sh
```

Troubleshooting

Error “java: command not found”

Java is not installed or not in PATH.

Install Java:

```
brew install openjdk
```

Then configure the PATH in your ~/.zshrc:

```
export PATH="/opt/homebrew/opt/openjdk/bin:$PATH" # Apple Silicon
# or for Intel: export PATH="/usr/local/opt/openjdk/bin:$PATH"
```

Reload: source ~/.zshrc

Version Incompatibility Error

If you get strange errors on launch, check that versions match:

```
java -version # Note the major version (11, 17, 21, etc.)
brew info openjfx # Should have the same major version
```

If versions don’t match, install a specific version:

```
# Example for Java 17
brew install openjdk@17
brew install openjfx@17
```

Error “JavaFX runtime components are missing”

Verify that: - OpenJFX is properly installed: `brew list openjfx` - The JAVAFX_PATH is correct - Required modules are properly specified

Find Installation Path

```
brew --prefix openjfx
```

Then add /lib to the returned path.

macOS Security Issue

If macOS blocks the application, go to: **System Preferences > Privacy & Security** and authorize execution.

Versions

To check the installed version:

```
brew info openjfx
```

To update:

```
brew upgrade openjfx
```

Alternative: jpackage

To create a professional macOS bundle, use jpackage (included with JDK 14+):

```
jpackage --type app-image \  
  --name "MyApplication" \  
  --input target \  
  --main-jar MyApplication.jar \  
  --main-class com.example.MainApp \  
  --module-path /opt/homebrew/opt/openjfx/lib \  
  --add-modules javafx.controls,javafx.fxml \  
  --icon icon.icns
```