

Launch JavaFX Application JAR by Double Clicking

The 8th JavaFX User Group Seminar
on 2012-11-30

Toru Takahashi (@boochnich)

About Toru

Twitter: @boochnich

blog: <http://d.hatena.ne.jp/torutk/>

- Java GUI Programming Experience
 - in jdk1.0 age: AWT
 - in jdk1.1 age: AWT, Netscape IFC, and Swing
 - in jdk1.2 age and after: Swing, Java2D, (Java3D)
- Java Community Activity
 - Representative of Java Reading Circle in Japan
<http://www.javareading.com/bof/>

Agenda

- JavaFX 2.2.3 in the JDK 7u9
- Building JavaFX application with NetBeans IDE
- Creating an executable JAR with command
- (optional) creating OS native installer

JavaFX 2.2.3 in the JDK 7u9

C:\Program Files\Java\jdk1.7.0

```
+-- bin
|   +-- javafxpackager.exe
+-- jre
|   +-- lib
|       +-- javafx.properties
|       +-- jfxrt.jar
+-- lib
    +-- ant-javafx.jar
    +-- javafx-doclet.jar
```

(Case of Windows OS)

Java FX files are in
JDK's directory
tree!

(Note: Not all JavaFX files are described)

JavaFX application with NetBeans (1/5)

- In NetBeans, select New project – JavaFX Application, templates are generated.
- Build and execute on NetBeans



<http://ja.netbeans.org/nekobean/>

JavaFX application with NetBeans (2/5)

- Double-click the jar file on a file explorer
 - The program starts.



- 'java -jar foo.jar' on command prompt.
 - It launches.



JavaFX application with NetBeans (3/5)

- Compile the source file and create jar

```
>javac -cp "C:\Program Files\java\jdk1.7.0\jre\lib\jfxrt.jar"  
-d build\classes -encoding UTF-8 src\hellojfxug\HelloJfxug.java
```

```
>jar cvfm Hello.jar hello.mf -C build\classes .
```

```
added manifest
```

```
adding: hellojfxug/(in = 0) (out= 0)(stored 0%)
```

```
adding: hellojfxug/HelloJfxug$1.class(in = 1141) (out= 605)(deflated 46%)
```

```
adding: hellojfxug/HelloJfxug.class(in = 1740) (out= 981)(deflated 43%)
```

```
adding: hellojfxug/HelloJfxug.css(in = 302) (out= 214)(deflated 29%)
```

```
>java -jar Hello.jar
```

```
Exception in thread "main" java.lang.NoClassDefFoundError: javafx/app  
lication/Application
```

```
    at java.lang.ClassLoader.defineClass1(Native Method)
```

```
    at java.lang.ClassLoader.defineClass(ClassLoader.java:791)
```



Oh, error has occurred, missing classpath.

JavaFX application with NetBeans (4/5)

- What's the difference between NetBeans and me?
 - Comparing with the jar file generated by NetBeans.



```
com/javafx/main/Main$1.class  
com/javafx/main/Main$2.class  
com/javafx/main/Main.class  
com/javafx/main/NoJavaFXFallback.class
```

These classes are only included in the jar file generated by NetBeans

```
Manifest-Version: 1.0  
JavaFX-Version: 2.2  
implementation-vendor: torutk  
implementation-title: HelloJjfxug  
implementation-version: 1.0  
JavaFX-Application-Class: hellojjfxug.HelloJjfxug  
JavaFX-Class-Path: HelloJjfxug.jar  
Created-By: JavaFX Packager  
Main-Class: com/javafx/main/Main
```

MANIFEST.MF in the jar file generated by NetBeans

JavaFX application with NetBeans (5/5)

- add `-Djavafx.verbose=true` option and execute the jar file generated by NetBeans

```
>java -Djavafx.verbose=true -jar dist\HelloJjfxug.jar
```



```
...
```

```
1) Try existing classpath...
```

```
==== URL list
```

```
file:/C:/work/HelloJjfxug/dist/HelloJjfxug.jar
```

```
====
```

```
2) Try javafx.runtime.path property...
```

```
3) Look for cobundled JavaFX ... [java.home=C:\Program Files\Java\jdk1.7.0\jre
```

```
==== URL list
```

```
file:/C:/work/HelloJjfxug/dist/HelloJjfxug.jar
```

```
file:/C:/Program%20Files/Java/jdk1.7.0/jre/lib/jfxrt.jar
```

```
file:/C:/Program%20Files/Java/jdk1.7.0/jre/lib/deploy.jar
```

```
file:/C:/Program%20Files/Java/jdk1.7.0/jre/lib/plugin.jar
```

```
file:/C:/Program%20Files/Java/jdk1.7.0/jre/lib/javaws.jar
```

```
====
```

```
Try calling Class.forName(hellojjfxug.HelloJjfxug) using classLoader = java.net.
```

```
URLClassLoader@06f404c51
```

```
found class: class hellojjfxug.HelloJjfxug
```

```
...
```

Creating an executable JAR with command

- Use `javafxpackager` command to create executable jar for JavaFX application.



```
>javafxpackager -createjar -nocss2bin  
-appclass hellojfxug.HelloJfxug  
-srcdir build\classes  
-outdir .  
-outfile HelloJfxug-0.1
```

Specify a class, extends Application and has a main method.

Specify a base directory to include a jar file.
Not java source files!

```
>java -jar HelloJfxug-0.1.jar
```

Specify the name of jar file without extension name(.jar)

javafxpackager command reference.

<http://docs.oracle.com/javafx/2/deployment/javafxpackager001.htm>

Reference

- Oracle Blogs [JavaFX] Native packaging for JavaFX
 - https://blogs.oracle.com/talkingjavadeployment/entry/native_packaging_for_javafx
- Oracle JavaFX Documents "Deploying JavaFX Applications"
 - <http://docs.oracle.com/javafx/2/deployment/jfxpub-deployment.htm>
 - 5.5 Create the Main Application JAR File

Conclusion

- JavaFX is not only GUI toolkit but also an application deployment toolkit.
- NetBeans IDE provides JavaFX deployment.
- JDK provides `javafxpackager` command, Ant task.
- There is the `e(fx)clipse` plugin for eclipse user.

(additional1) create MSI

- javapackager can create a Windows MSI Installer.
 - Needs WiX(Windows Installer XML): Free
- NetBeans 7.3 can create by native build
- I introduced creating msi installer on my blog (Japanese only)
 - <http://d.hatena.ne.jp/torutk/20121119>

(additional2) Creating RPM

- javapackager can create Linux RPM installer.
 - Needs rpmbuild command
- NetBeans 7.3 can create by native build
- May be create deb installer.

(additional3) Note for sample program

To read a CSS file which is in the same directory with source file.

```
Scene scene = new Scene(root, 300, 250);
scene.getStylesheets().add(
    this.getClass().getResource("HelloJjfxug.css").toExternalForm());
```

Sample CSS for specifying a font for Button class

```
.button {
    -fx-font-size: 24;
    -fx-font-family: Consolas sans-serif;
}
```