

Building the OpenJFX embedded stack for Linux desktop

You can build the embedded OpenJFX sources for a Linux/x86 desktop. This has slightly different behavior to the regular OpenJFX for Linux:

- All windows are rendered onto a single surface, without decorations. This single surface can be either the framebuffer device `/dev/fb0` or an X11 window. Note that accelerated rendering to the framebuffer using EGL is generally not supported on desktop platforms. Accelerated rendering to an X11 window with EGL usually is supported, as long as you have the correct EGL and OpenGL ES 2.0 libraries
- The embedded input system is used, taking events directly from Linux input drivers instead of through GTK. This means that you get embedded multitouch support. On the other hand, keyboard input is currently limited to the US locale.
- You don't get a cursor. This is because the embedded hardware cursor implementation is used, which uses features of specific ARM chipsets (e.g. TI OMAP3, Broadcom BCM2835, Freescale i.MX6) to render a cursor on a separate layer. JavaFX Embedded doesn't currently have any cursor implementations for x86 chipsets.
- Media and web are excluded from the build by default

To build, prepare an Ubuntu build machine as you would for a desktop build. Then install the following additional packages:

- `libdirectfb-dev`
- `libudev-dev`
- `libegl1-mesa-dev`
- `libgles2-mesa-dev`

Build with

```
gradle -PCOMPILER_TARGETS=x86egl sdk
```

Run Java using the OpenJFX binaries that are created in the output directory `build/x86egl-sdk/rt`. The easiest way to do this is to run a regular Java 8 JRE with the command-line parameter:

```
-Djava.ext.dirs=build/x86egl-sdk/rt/lib/ext
```

To run using accelerated rendering to an X11 window, use the flags:

```
-Djavafx.platform=monocle -Dmonocle.platform=X11 -Dembedded=monocle
```

To run using software rendering to a framebuffer, use the flags:

```
-Djavafx.platform=monocle -Dmonocle.platform=Linux -Dprism.order=sw
```