Java on mobile devices

Building native apps for Android and iOS in Java

Thomas Künneth
MATHEMA Software GmbH

Spot the difference

The early years (1992 – 2002)

- Oak
- PersonalJava
- Java 2 Platform, Micro Edition (J2ME) and the K Virtual Machine for heavily resource-constrained devices

JavaFX Mobile

- Based upon JavaFX 1.x (JavaFX Script)
- Aimed at, among others, Android and Windows Mobile
- Demoed on Android at JavaOne 2008; general availability: February 2009
- Should be available on mobile devices through cooperation with device manufacturer

Mobile Java on the decline

- „Java's not worth building in [to the phone].
  Nobody uses Java anymore.
  It's this big heavyweight ball and chain.“
  (Steve Jobs in 2007)

- March 2008 Sun announces that JavaME would be available on the iPhone. That never happned

- With the rise of smartphones, the relevance of Java ME shrank

- As of today, no official Java version/distribution for iOS

Java 9

- In September 2015 Oracle suggested a new project:
  Mobile: JDK Ports to Modern Mobile Platforms

- Features:
  - JDK 9 based port (Headless)
  - Support at minimum the equivalent of JDK 8 compact2 profile (in module form)
  - iOS x64 and arm64 (arm64 will be provided via Zero interpreter)
  - Android x86 and arm (arm will be provided via Zero interpreter)
  - Windows 10 tablet apps (side loaded)
  - JavaLauncher helper interface to simplify the process of including Java in Mobile applications
  - Sample HelloWorld applications and/or project templates for each platform

- Apparently no installation bundles for end users on iOS and Android

Why use Java on mobile devices?

- Android and iOS divide the market of mobile platforms up between themselves
  - Gigantic potential user base
  - Microsoft heavily invested in buying Xamarin (other eco system, same story)
- Android uses Java anyway, so why content oneself with half of the cake?
- Java is still most used programming language
  - Huge amount of skilled developers
  - Widespread knowledge
  - Why learn yet another language?

A couple of options

- J2ObjC
- Gluon Mobile
- MobiDevelop's RoboVM Fork and BugVM (both based upon RoboVM)
- Multi-OS Engine
- ADF Mobile / Mobile Application Framework
- DukeScript
- JUniversal
- ...

- What parts of Java are used? Language, libraries, tool chain
- What target artefact is produced?
- What runtime environment is needed?
- What kind of app can be developed?

Advantages of building native applications

- Easily distribute in app stores
- Deep integration in the platform and eco system
- (Native user interfaces)
  - Even deeper integration
  - Less code reuse

- (In theory) not distinguishable from „real“ native apps

RoboVM (discontinued)

- Swedish startup (founded 2013)
- Goal: native Android and iOS apps in Java with native ui
- JVM bytecode is translated into machine language using LLVM
- Class library based on Android class library with additional bindings for all iOS apis
- RoboVM Studio: IDE based on IntelliJ IDEA; supports source level debugging

- October 2015: Xamarin buys RoboVM
- February 2016: Microsoft announces plans to by Xamarin
- April 2016: Microsoft announces to wind down RoboVM

Gluon

- Commercial sibling of JavaFXPorts
- Native apps for Android and iOS (as well as desktop and embedded)
- JavaFX for the user interface
- For iOS, currently relies on open source parts of RoboVM
- On Android, currently compiles and links against Androids native runtime

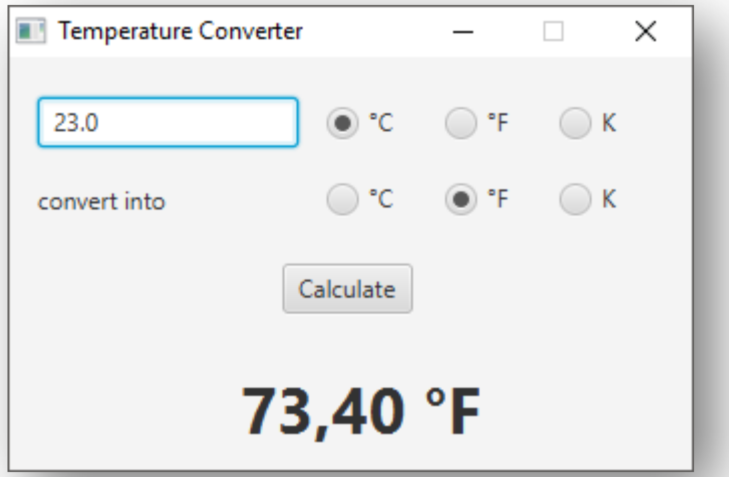
- Gluon has announced Gluon VM, which will be based upon Java 9

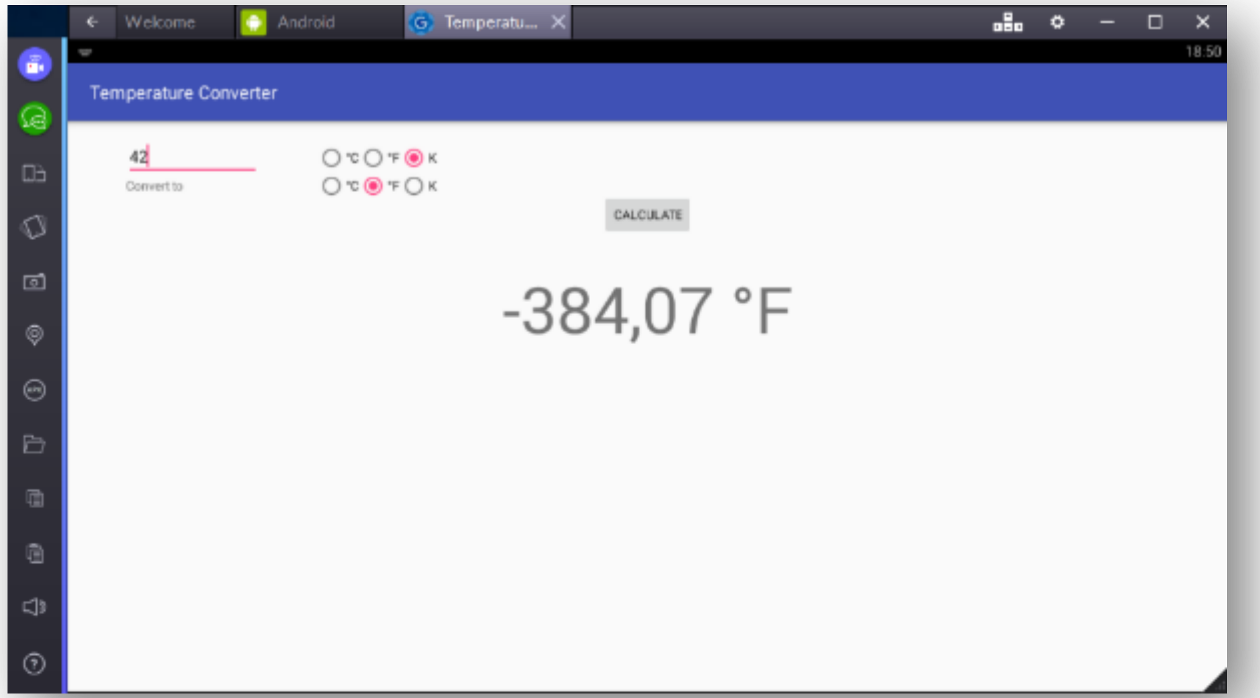
Multi-OS Engine

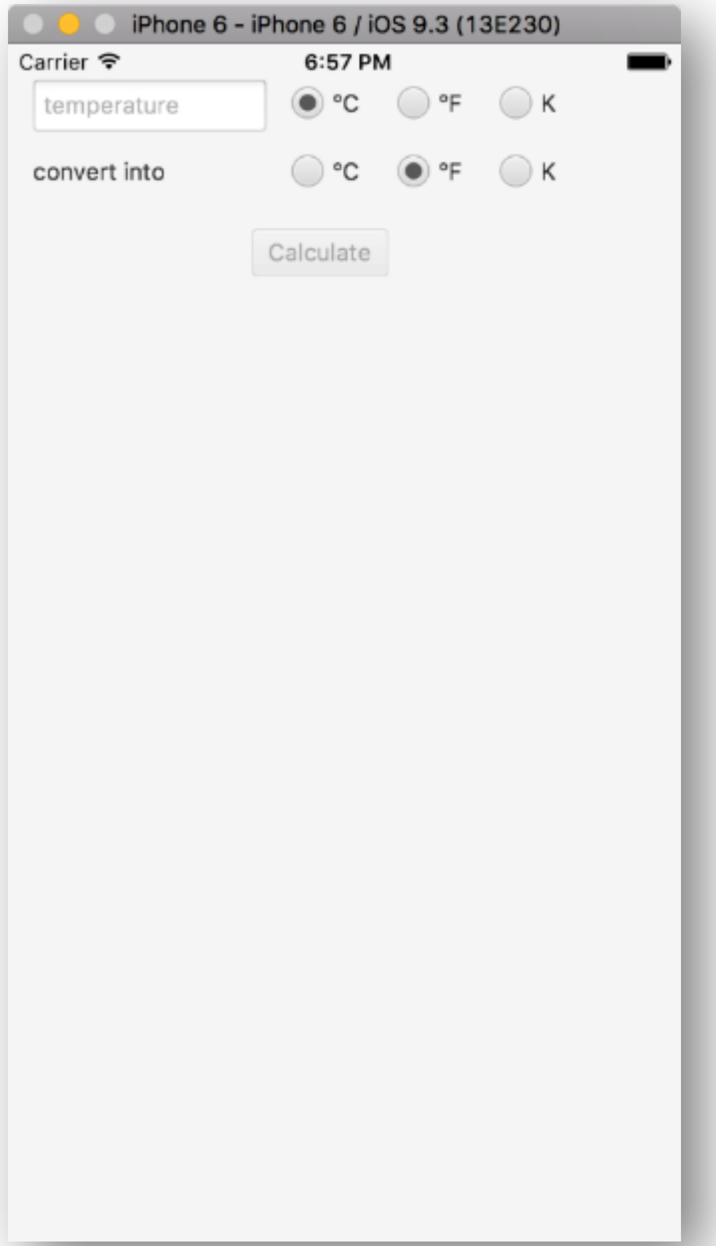
- Mobile apps for iOS and Android with native performance and native look and feel
- Develop on Macs or on Windows using remote build
- Plugins for Android Studio, IntelliJ IDEA and Eclipse

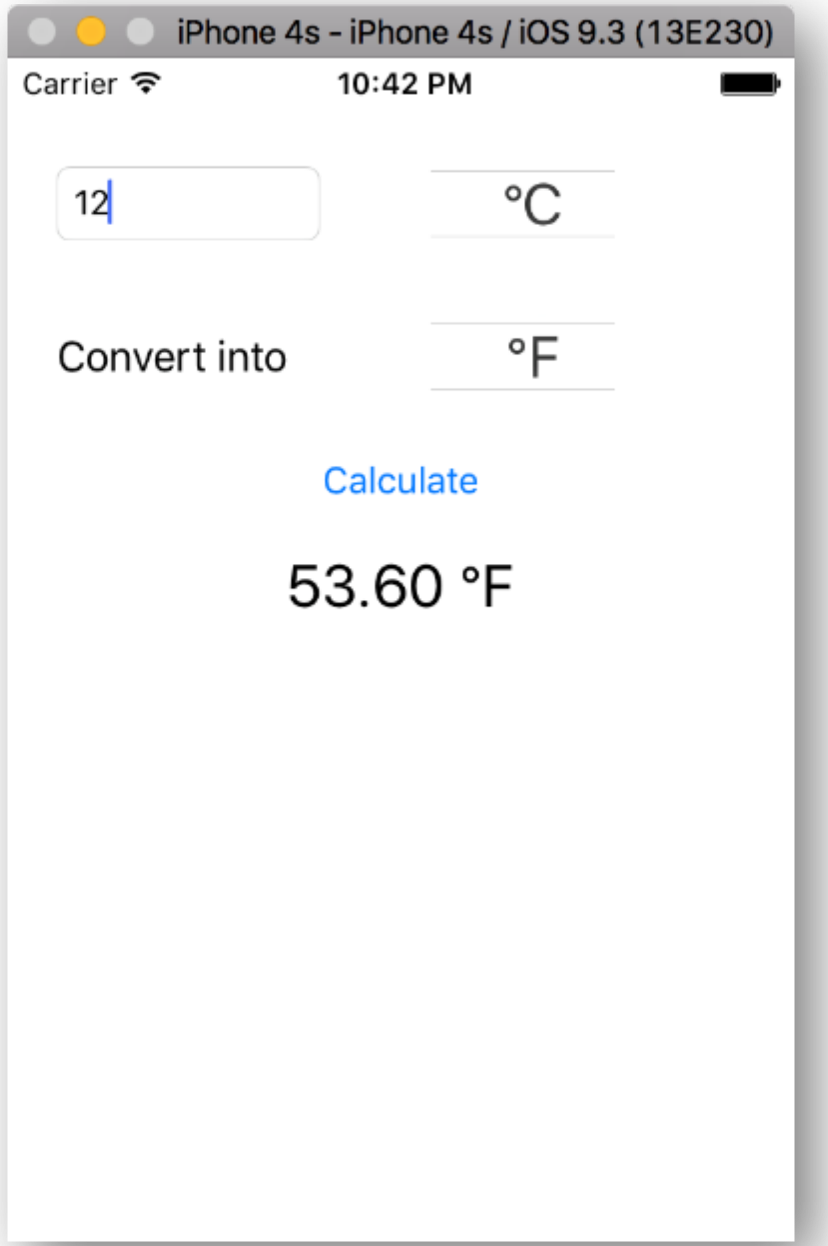
- Has been available as a technology preview; end of June 2016 Intel announced to open source Multi-OS Engine under the Apache License, Version 2.0
- Migeran is project lead and core developer
- Based upon Android Art Runtime environment and class libraries
- Java-Objective-C-bridge „Nat/J“ provides access to any Objective-C code

Start your engines





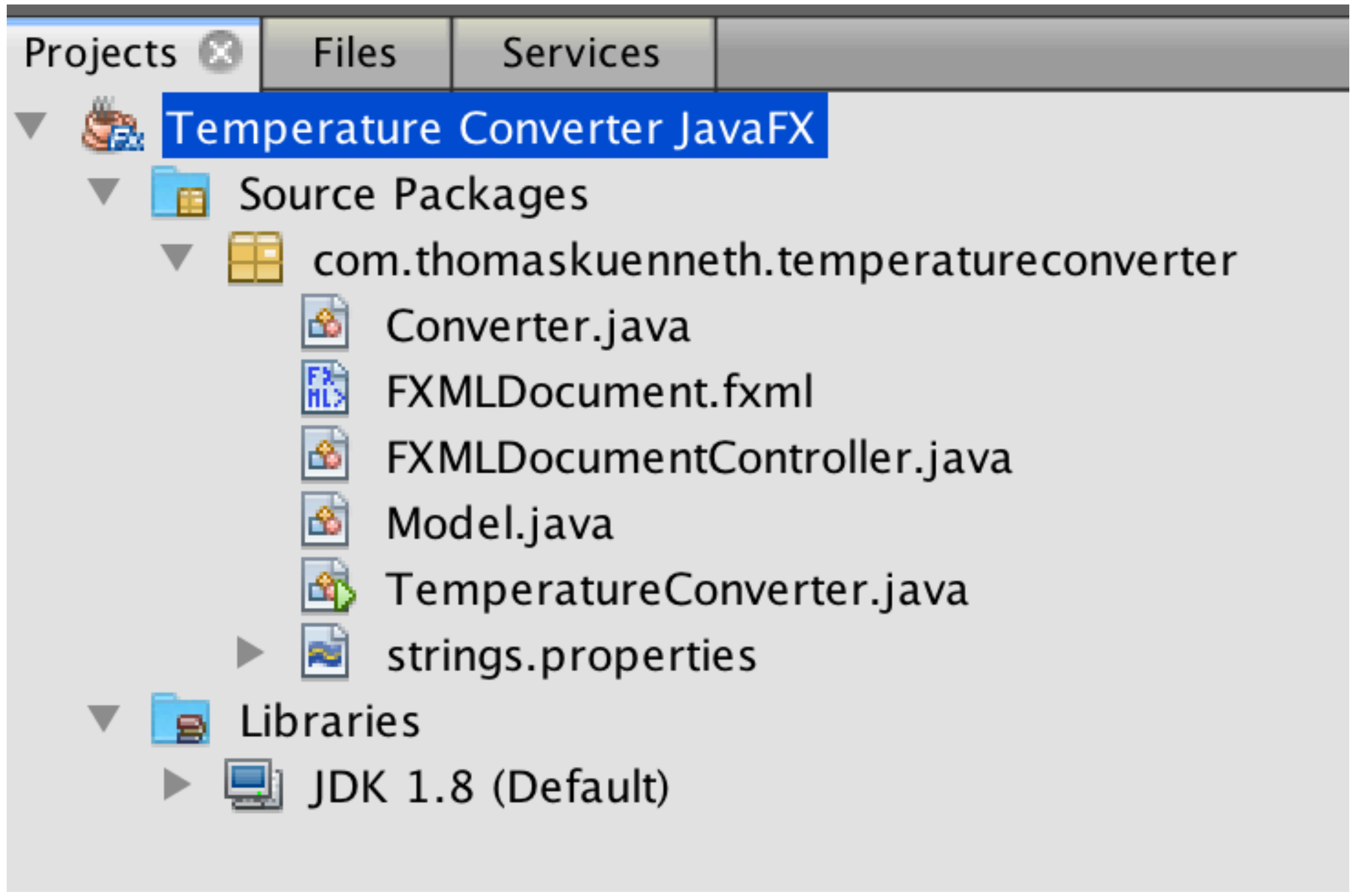




- https://github.com/tkuenneth/TemperatureConverter

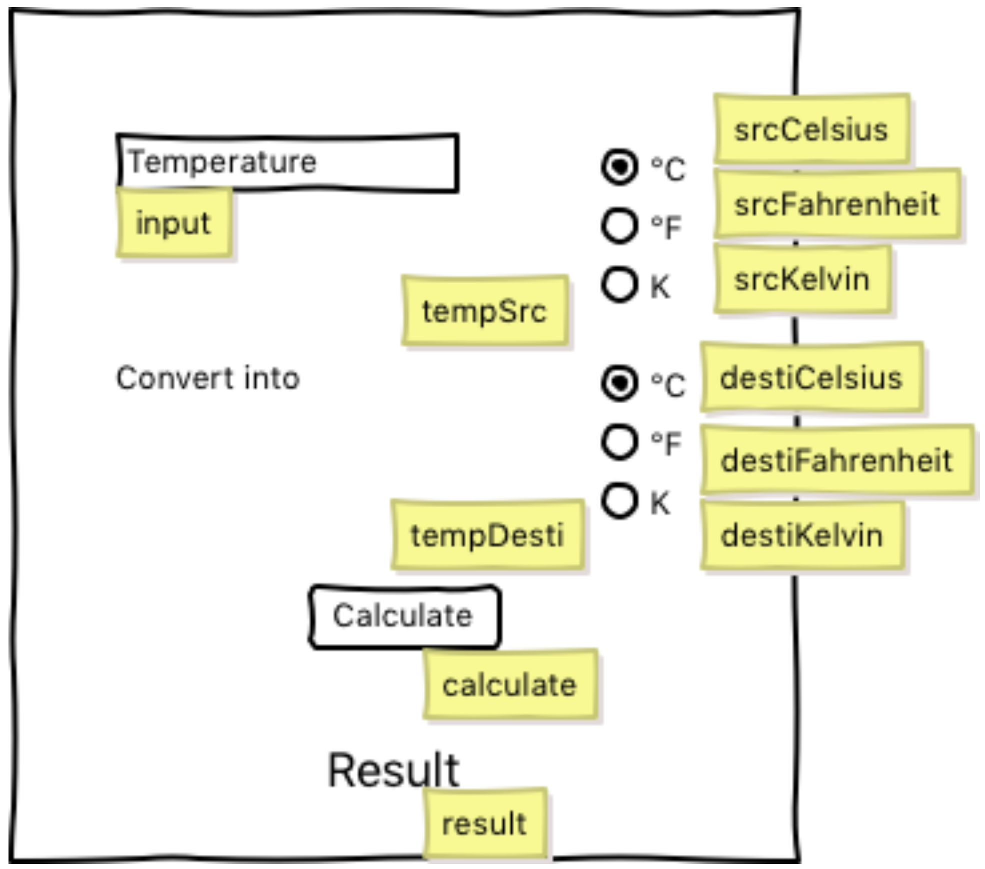
TOPCONF TALLINN

A glimpse at the JavaFX app

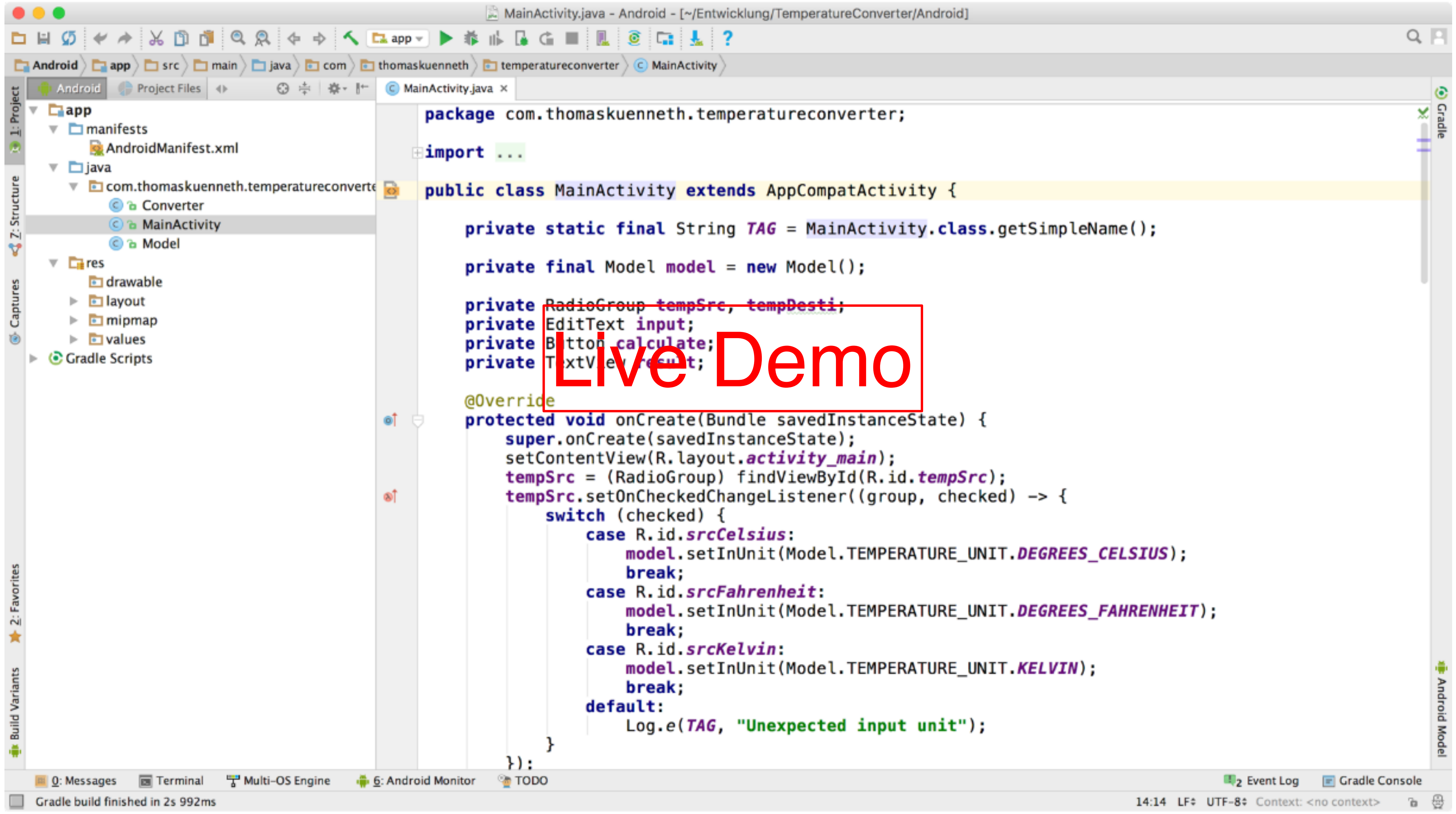


Projects | Files | Services

- Temperature Converter JavaFX
  - Source Packages
    - com.thomaskuenneth.temperatureconverter
      - Converter.java
      - FXMLDocument.fxml
      - FXMLDocumentController.java
      - Model.java
      - TemperatureConverter.java
      - strings.properties
  - Libraries
    - JDK 1.8 (Default)



Build a „pure Android“ app from scratch


MainActivity.java - Android - [~/Entwicklung/TemperatureConverter/Android]
Android > app > src > main > java > com > thomaskuenneth > temperatureconverter > MainActivity
app
manifests
AndroidManifest.xml
java
com.thomaskuenneth.temperatureconverte
Converter
MainActivity
Model
res
drawable
layout
mipmap
values
Gradle Scripts
0: Messages
Terminal
Multi-OS Engine
6: Android Monitor
TODO
Event Log
Gradle Console
Gradle build finished in 2s 992ms
Live Demo

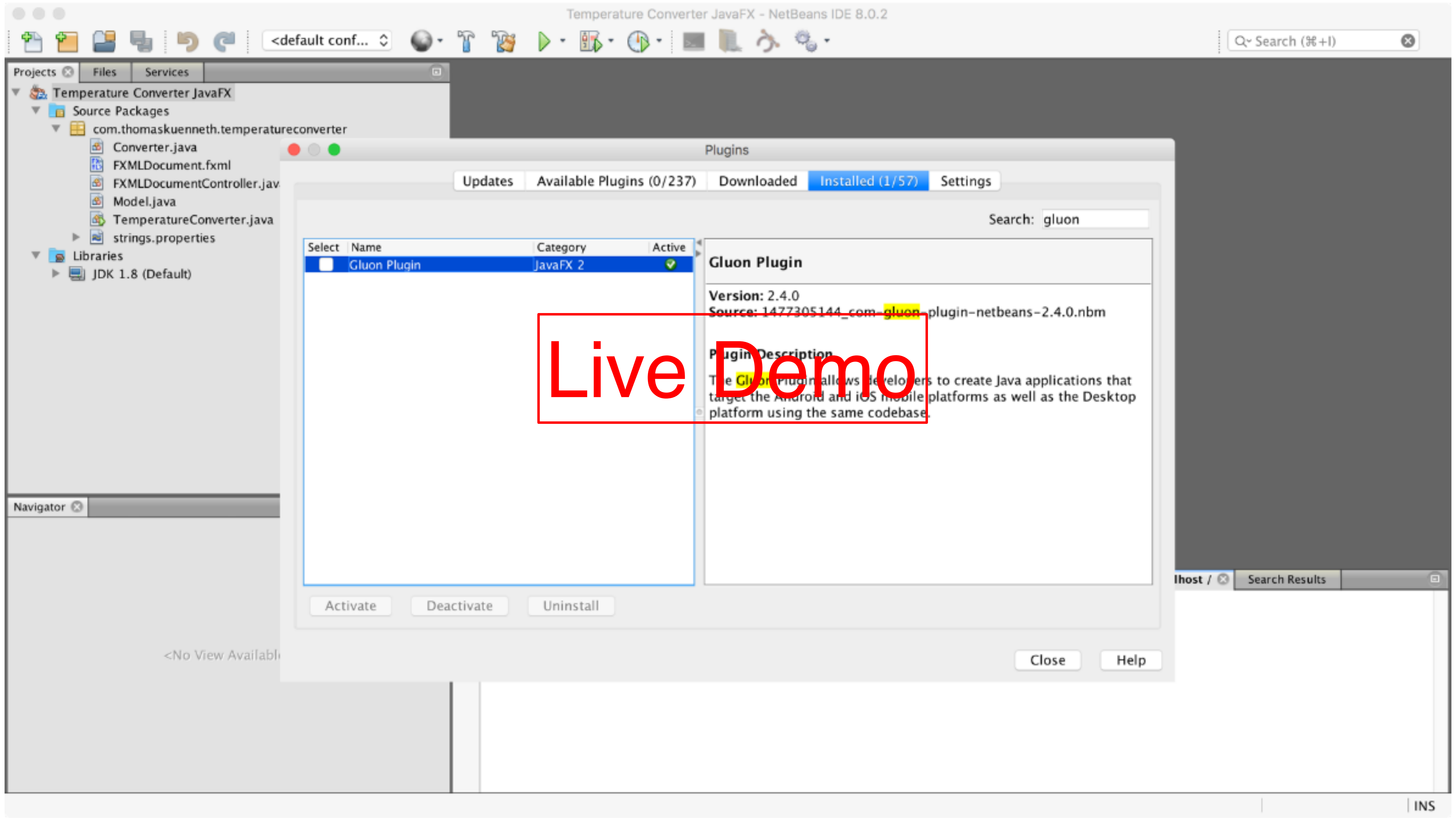
```
package com.thomaskuenneth.temperatureconverter;

import ...

public class MainActivity extends AppCompatActivity {

    private static final String TAG = MainActivity.class.getSimpleName();

    private final Model model = new Model();

    private RadioGroup tempSrc, tempDesti;
    private EditText input;
    private Button calculate;
    private TextView result;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tempSrc = (RadioGroup) findViewById(R.id.tempSrc);
        tempSrc.setOnCheckedChangeListener((group, checked) -> {
            switch (checked) {
                case R.id.srcCelsius:
                    model.setInUnit(Model.TEMPERATURE_UNIT.DEGREES_CELSIUS);
                    break;
                case R.id.srcFahrenheit:
                    model.setInUnit(Model.TEMPERATURE_UNIT.DEGREES_FAHRENHEIT);
                    break;
                case R.id.srcKelvin:
                    model.setInUnit(Model.TEMPERATURE_UNIT.KELVIN);
                    break;
                default:
                    Log.e(TAG, "Unexpected input unit");
            }
        });
```

Prerequisites for Android apps

- Android Studio
- Android SDK
- Android Emulator or real device

- To some extent know how to use these tools

Building the app using Gluon



Search: gluon

Select	Name	Category	Active
☐	Gluon Plugin	JavaFX 2	✓

Gluon Plugin

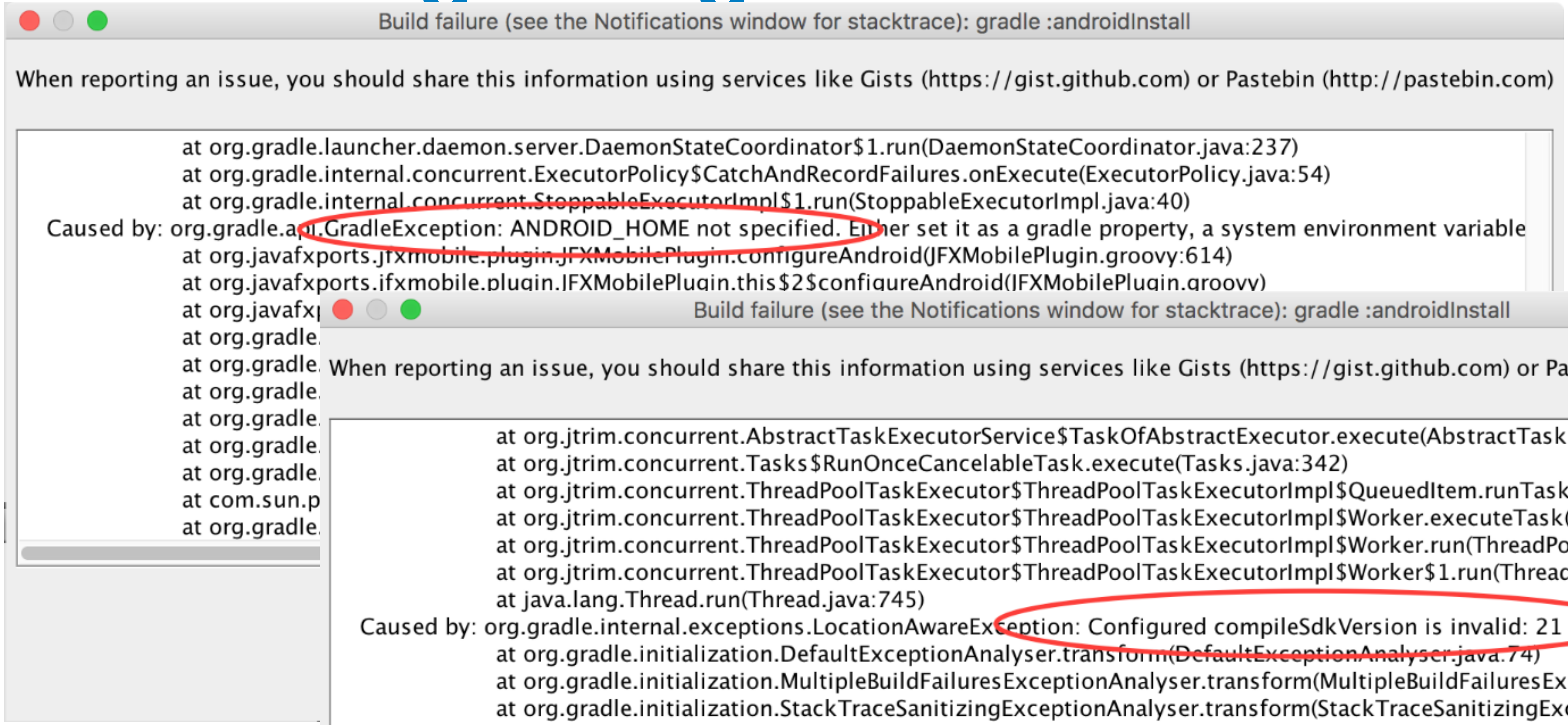
Version: 2.4.0
Source: 1477305144_com-gluon-plugin-netbeans-2.4.0.nbm

Plugin Description

The Gluon Plugin allows developers to create Java applications that target the Android and iOS mobile platforms as well as the Desktop platform using the same codebase.

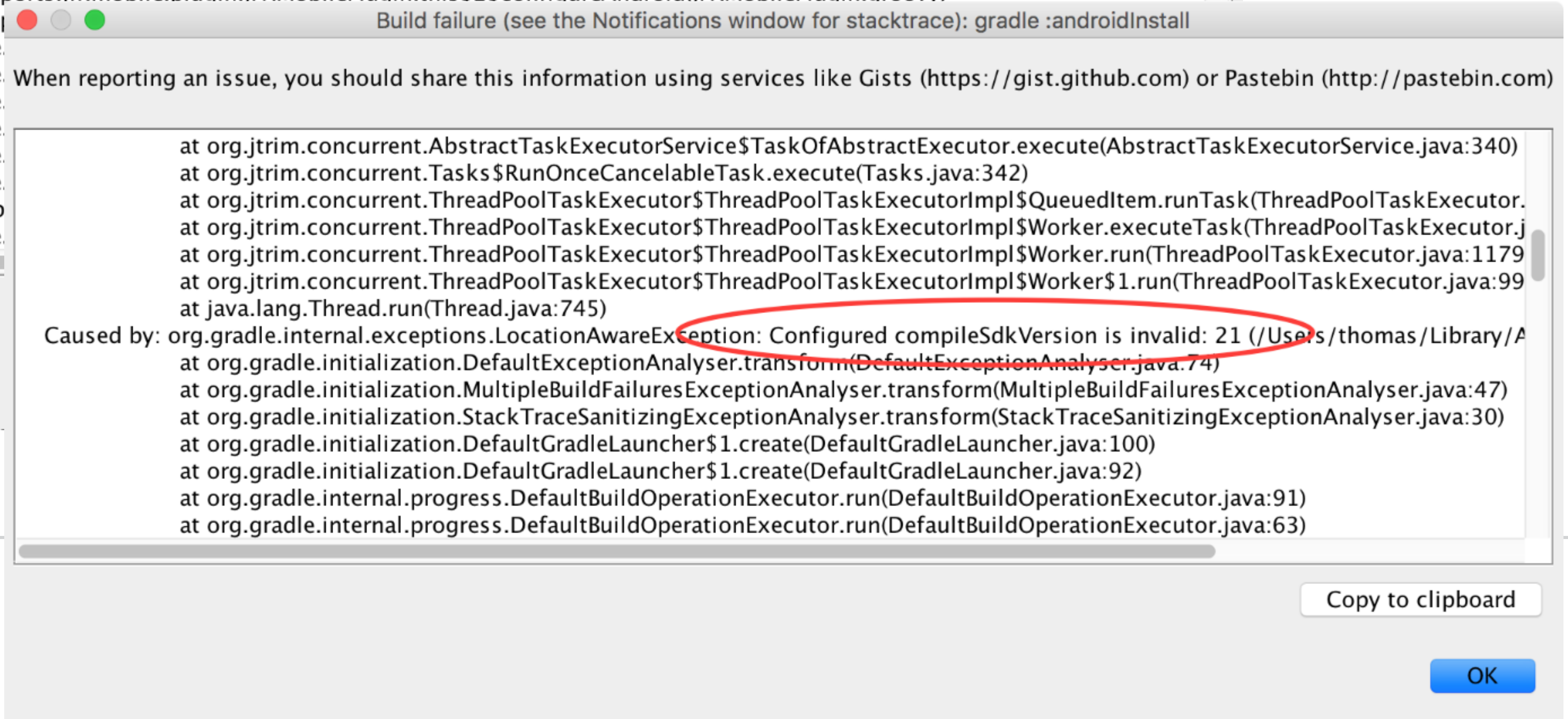
TOPCONF TALLINN

Errors regarding Android SDK


Build failure (see the Notifications window for stacktrace): gradle :androidInstall

When reporting an issue, you should share this information using services like Gists (https://gist.github.com) or Pastebin (http://pastebin.com)

```
        at org.gradle.launcher.daemon.server.DaemonStateCoordinator$1.run(DaemonStateCoordinator.java:237)
        at org.gradle.internal.concurrent.ExecutorPolicy$CatchAndRecordFailures.onExecute(ExecutorPolicy.java:54)
        at org.gradle.internal.concurrent.StoppableExecutorImpl$1.run(StoppableExecutorImpl.java:40)
Caused by: org.gradle.api.GradleException: ANDROID_HOME not specified. Either set it as a gradle property, a system environment variable
        at org.javafxports.jfxmobile.plugin.JFXMobilePlugin.configureAndroid(JFXMobilePlugin.groovy:614)
        at org.javafxports.jfxmobile.plugin.JFXMobilePlugin.this$2$configureAndroid(JFXMobilePlugin.groovy)
        at org.javafx
        at org.gradle.
        at org.gradle.
        at org.gradle.
        at org.gradle.
        at org.gradle.
        at org.gradle.
        at com.sun.p
        at org.gradle.
```


Build failure (see the Notifications window for stacktrace): gradle :androidInstall

When reporting an issue, you should share this information using services like Gists (https://gist.github.com) or Pastebin (http://pastebin.com)

```
        at org.jtrim.concurrent.AbstractTaskExecutorService$TaskOfAbstractExecutor.execute(AbstractTaskExecutorService.java:340)
        at org.jtrim.concurrent.Tasks$RunOnceCancelableTask.execute(Tasks.java:342)
        at org.jtrim.concurrent.ThreadPoolTaskExecutor$ThreadPoolTaskExecutorImpl$QueuedItem.runTask(ThreadPoolTaskExecutor.
        at org.jtrim.concurrent.ThreadPoolTaskExecutor$ThreadPoolTaskExecutorImpl$Worker.executeTask(ThreadPoolTaskExecutor.j
        at org.jtrim.concurrent.ThreadPoolTaskExecutor$ThreadPoolTaskExecutorImpl$Worker.run(ThreadPoolTaskExecutor.java:1179
        at org.jtrim.concurrent.ThreadPoolTaskExecutor$ThreadPoolTaskExecutorImpl$Worker$1.run(ThreadPoolTaskExecutor.java:99
        at java.lang.Thread.run(Thread.java:745)
Caused by: org.gradle.internal.exceptions.LocationAwareException: Configured compileSdkVersion is invalid: 21 (/Users/thomas/Library/A
        at org.gradle.initialization.DefaultExceptionAnalyser.transform(DefaultExceptionAnalyser.java:74)
        at org.gradle.initialization.MultipleBuildFailuresExceptionAnalyser.transform(MultipleBuildFailuresExceptionAnalyser.java:47)
        at org.gradle.initialization.StackTraceSanitizingExceptionAnalyser.transform(StackTraceSanitizingExceptionAnalyser.java:30)
        at org.gradle.initialization.DefaultGradleLauncher$1.create(DefaultGradleLauncher.java:100)
        at org.gradle.initialization.DefaultGradleLauncher$1.create(DefaultGradleLauncher.java:92)
        at org.gradle.internal.progress.DefaultBuildOperationExecutor.run(DefaultBuildOperationExecutor.java:91)
        at org.gradle.internal.progress.DefaultBuildOperationExecutor.run(DefaultBuildOperationExecutor.java:63)
```

Easily fixed

- Set environment variable ANDROID_HOME to the Android SDK:
  export ANDROID_HOME=/Users/thomas/Library/Android/sdk/
- OR: put androidSdk=... in build.gradle

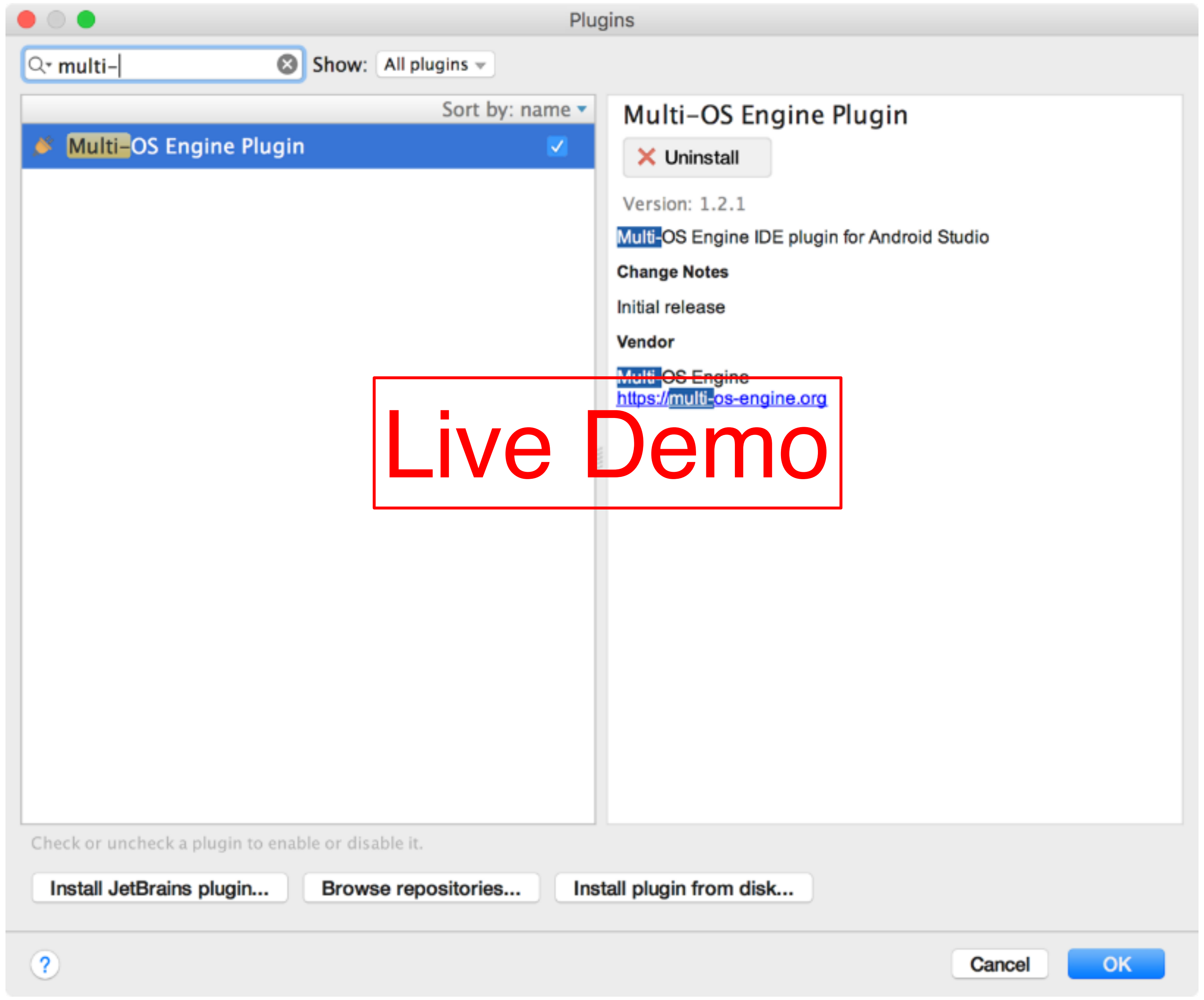
- Download appropriate Android platform
- OR: put compileSdkVersion=... in build.gradle

Prerequisites for iOS apps

- computer running macOS
- Xcode and Xcode commandline tools
- iOS Simulator or a real device

- To some extent know how to use thesm

Building the app with Multi-OS Engine



Multi-OS Engine Plugin

Multi-OS Engine Plugin

Version: 1.2.1

Multi-OS Engine IDE plugin for Android Studio

Change Notes

Initial release

Vendor

Multi-OS Engine
https://multi-os-engine.org

Check or uncheck a plugin to enable or disable it.

Live Demo

Wrap up

- Apps are truely native
- UI may be depending on the solution being used
- Heavy code-reuse possible
- Existing knowledge can be leveraged
- Fair amount of knowledge regarding native tools needed
- To build native ui, the underlying concepts must be mastered

Thomas Künneth

MATHEMA Software GmbH

@tkuenneth (Twitter)

thomas.kuenneth@mathema.de

TOPCONF TALLINN