

Mit LiveCode unter Windows Android-Apps erstellen

LiveCode ist eine großartige Entwicklungsumgebung, mit der sich unkompliziert und extrem schnell Apps aller Art erstellen lassen. Solange diese als Windows- oder macOS-Programme auf dem PC oder Mac laufen sollen, ist es auch ebenso einfach und mit wenigen Klicks erledigt, Standalone-Versionen seiner Apps herzustellen, die man unabhängig von LiveCode verwenden und weiterverbreiten kann.

Der Reiz von LiveCode liegt allerdings auch darin, dass hier mit einem System und einem Quellcode Apps für die unterschiedlichsten Plattformen erstellt werden können – insbesondere für mobile Systeme wie Android und iOS.

Liest man aber die zahlreichen Beiträge hierzu im Forum fällt auf, dass es bei der Erstellung mobiler Apps mit der Unkompliziertheit oft vorbei ist. Während man sonst in LiveCode so schön intuitiv arbeiten kann und die Dinge eben gerade nicht so kompliziert sind wie in Xcode oder im Android Studio, wird es oft richtig haarig, wenn es daran geht, die unter Windows oder Mac hervorragend laufende App auf sein Handy oder Tablet zu bringen.

Dabei ist das Problem meist gar nicht die Programmierung, denn die bleibt unter LiveCode so einfach wie zuvor – man muss nur ein paar Dinge beachten, die mit den Bildschirmgrößen der Mobilgeräte zu tun haben – und um Spezialfeatures zu nutzen wie den Bewegungssensor, GPS usw. muss man sich mit den entsprechenden Modulen einarbeiten... so weit so gut.

Aber wie bringe ich diese Apps nun auf mein Gerät?

Auf Wunsch zahlreicher LiveCode-Einsteiger, die mich angeschrieben haben und als versprochenes Zusatzmaterial zu meinem Buch folgt hier eine aktuelle Anleitung, die Schritt für Schritt erklärt, wie man vorgehen muss, um Windows so einzurichten, dass LiveCode auf Knopfdruck komplette APK-Dateien erstellen kann, die dann auch auf dem Handy oder Tablet mit Android-System laufen.

Diese Fassung ist im Februar 2020 erstellt worden und funktioniert derzeit einwandfrei. Ich habe sie auf einem frischen Windows 10 System ausgiebig getestet. Es ist möglich, dass diese Anleitung in wenigen Wochen schon wieder teilweise aktualisiert werden muss. Ich werde sie also regelmäßig anpassen müssen. Eine regelmäßig aktualisierte Version dieser Anleitung befindet sich auch auf dem LiveCode-Blog

(<http://livecode-blog.de>)

Warum ist es denn so kompliziert?

Man würde sich sehr wünschen, dass LiveCode es einem leichter machen würde – dass man wie bei Windows oder Mac-Standalones einfach auf den Knopf drücken müsste und eine fertig kompilierte APK oder IPA herauskäme, ohne dass das System vorher aufwendig eingerichtet werden muss. Warum geht das nicht? Vermutlich liegt es an den Lizenzen und rechtlichen Einschränkungen im Zusammenhang mit Google und Oracle. LiveCode darf schlicht und einfach bei der Installation nicht die passenden SDK- und JDK-Dateien mitliefern und verwenden. Das muss also jeder Benutzer selbst und einzeln machen, damit es legal ist.

Aber immerhin: Einmal eingerichtet läuft das System am Ende problemlos und kann vermutlich jahrelang verwendet werden, um dann jederzeit schnell und einfach neue Android-Dateien zu produzieren. Die Mühe lohnt sich also – in ein bis zwei Stunden ist es eingerichtet, und dann läuft es!

Fangen wir also an!

Ich gehe davon aus, dass du LiveCode in der Version 9.5 STABLE auf deinem System installiert hast. Die Community-Version reicht aus.

Damit LiveCode APKs (Android Packages) erstellen kann, müssen zwei Dinge in deinem Windows-System vorhanden sein – ein Java-Compiler mit entsprechenden Bibliotheken (JDK), denn Android-Programme sind Java-Programme - und das Android-SDK (Software Development Kit), das die speziellen Funktionen und Bibliotheken für mobile Android-Geräte bereitstellt. Das Android SDK muss zudem noch für bestimmte Android-Versionen konfiguriert werden, bevor es funktioniert.

Installation des JDK (Java Development Kit)

Als erstes benötigst du (neuerdings) einen Oracle-Account, damit du das JDK überhaupt herunterladen kannst. Diesen kannst du kostenlos anlegen – und zwar hier:

<https://profile.oracle.com/myprofile/account/create-account.jsp>

Gib deine Daten ein, erstelle einen Oracle-Account, bestätige ihn mit dem Link, den du per E-Mail erhältst, und damit bist du registriert. Merke dir das Passwort.

Nun kannst du das JDK laden. Genauer gesagt brauchst du UNBEDINGT das **Java SE Development Kit 8**. Solltest du ein neueres JDK auf deinem System haben, rate ich dir, es zu deinstallieren, damit es keine Probleme mit mehrfachen Versionen gibt.

Du findest die Downloadseite hier (sollte der Link nicht mehr funktionieren, google nach „JAVA SE Development Kit 8 Download“):

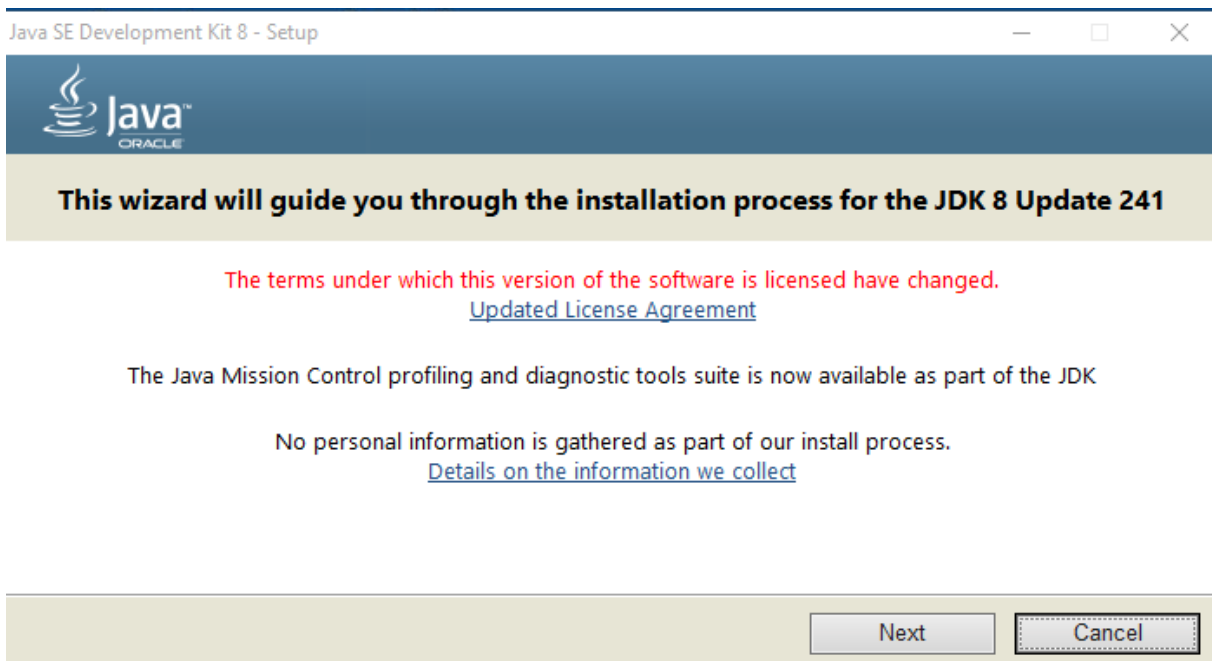
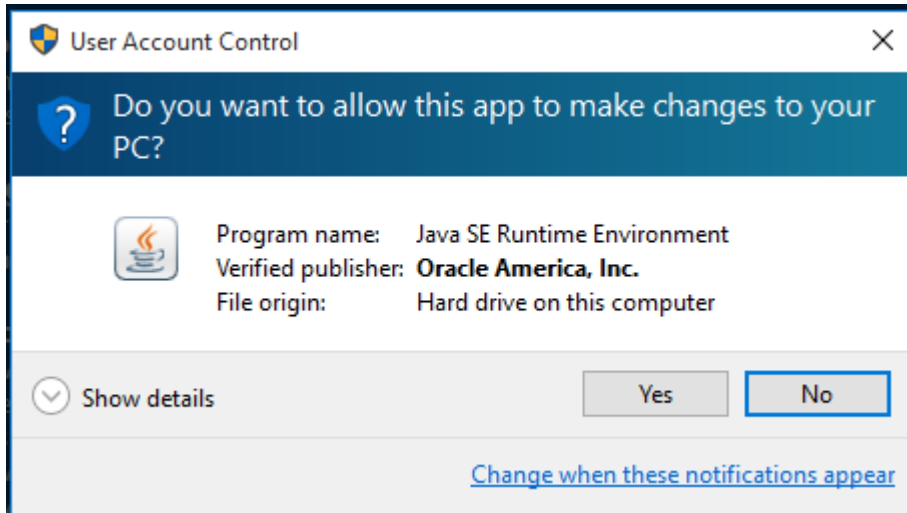
<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

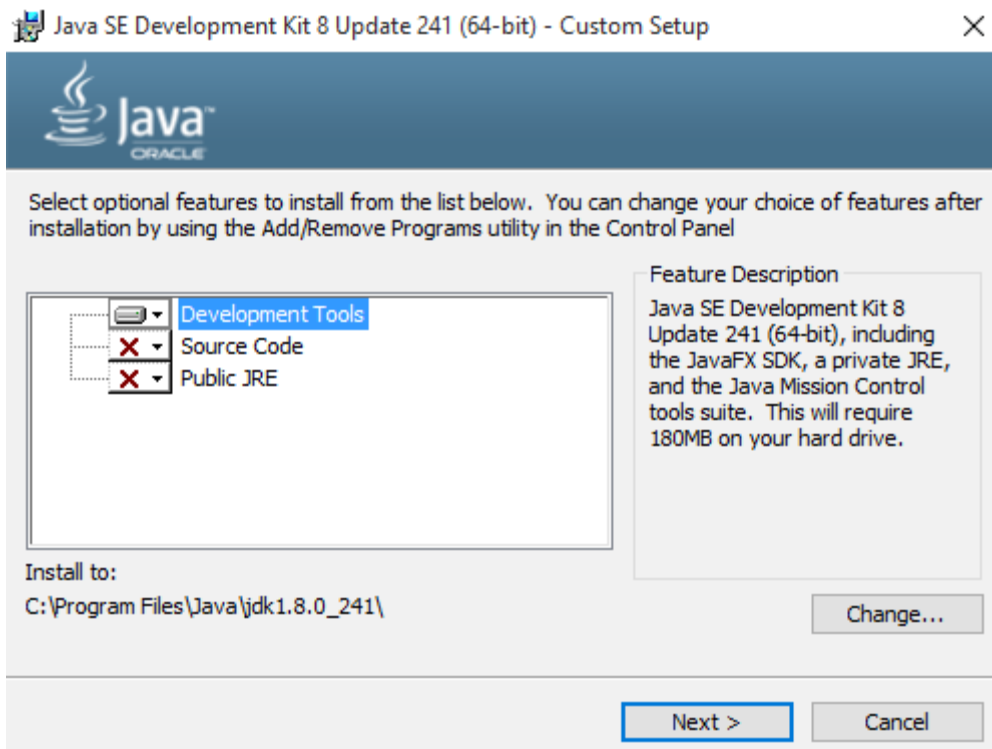
Auf der Webseite erscheint eine Liste der verfügbaren Versionen. Für Windows (64-bit) benötigst du die ganz unten auf der Liste stehende Fassung, Windows x64.

Java SE Development Kit 8u241		
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.		
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.94 MB	jdk-8u241-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.83 MB	jdk-8u241-linux-arm64-vfp-hflt.tar.gz
Linux x86	171.28 MB	jdk-8u241-linux-i586.rpm
Linux x86	186.1 MB	jdk-8u241-linux-i586.tar.gz
Linux x64	170.65 MB	jdk-8u241-linux-x64.rpm
Linux x64	185.53 MB	jdk-8u241-linux-x64.tar.gz
Mac OS X x64	254.06 MB	jdk-8u241-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	133.01 MB	jdk-8u241-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.24 MB	jdk-8u241-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.8 MB	jdk-8u241-solaris-x64.tar.Z
Solaris x64	92.01 MB	jdk-8u241-solaris-x64.tar.gz
Windows x86	200.86 MB	jdk-8u241-windows-i586.exe
Windows x64	210.92 MB	jdk-8u241-windows-x64.exe

Als erstes klickst du auf „Accept License Agreement“ oben, dann wählst du die benötigte Version (Windows x64) und kannst sie nach einem Login in deinen zuvor erstellten Oracle-Account herunterladen.

Nach dem Laden startest du die heruntergeladene Datei per Doppelklick und installierst sie.





Bei der Auswahl der zu installierenden Komponenten kannst du „Source-Code“ und „Public JRE“ ruhig abwählen – du brauchst nur die Development Tools.

Am Schluss wird die Installation mit „Close“ beendet. Damit hast du Schritt 1 geschafft: Das Java Development Kit ist installiert.

Nun brauchst du noch das Android SDK.

Installation Android SDK (Software Development Kit)

Obwohl es theoretisch möglich ist, nur das SDK zu laden und zu konfigurieren, empfehle ich dir, das komplette **Android Studio** herunterzuladen und zu installieren. Die Ladezeit und der benötigte Platz auf der Festplatte sind ein bisschen höher, dein Stress verringert sich aber gleichzeitig.

Gehe auf folgende Seite:

<https://developer.android.com/studio>

android studio

Android Studio provides the fastest tools for building apps on every type of Android device.

DOWNLOAD ANDROID STUDIO

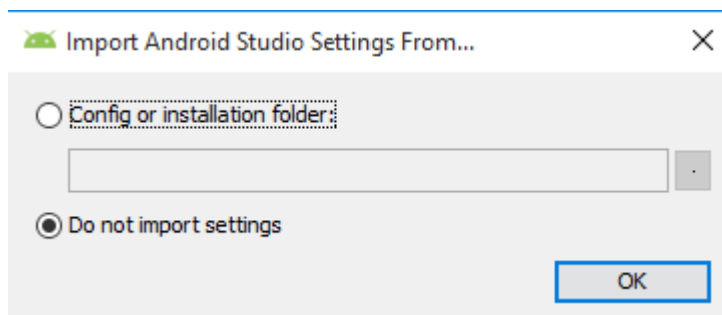
3.5.3 for Windows 64-bit (718 MB)

DOWNLOAD OPTIONS

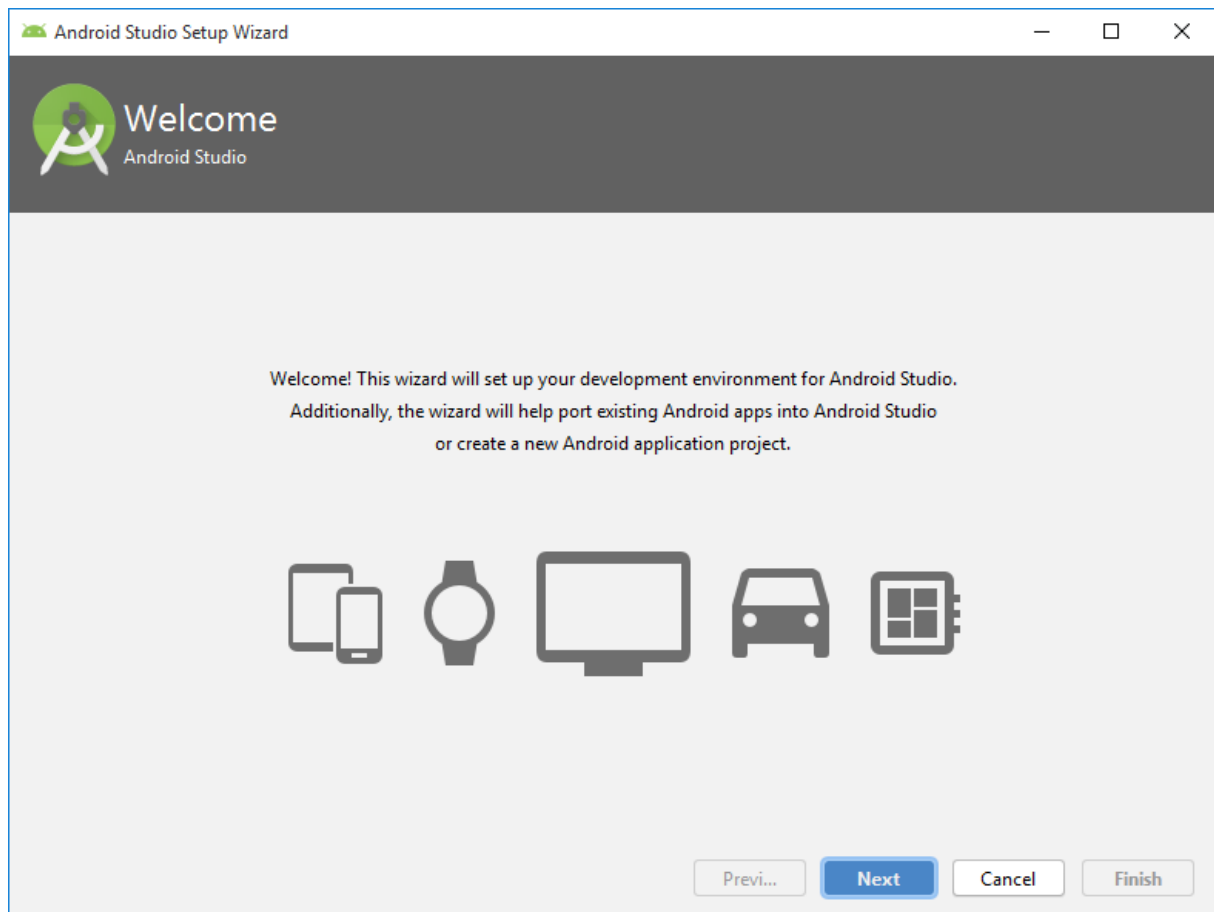
RELEASE NOTES

Klicke auf den grünen Button „Download Android Studio“, hier wird automatisch die zum Betriebssystem passende Version vorgeschlagen, und lade die Installationsdatei herunter. Je nach Internetverbindung kann das schnell gehen oder eine Weile dauern.

Anschließend startest du die Installation durch Doppelklick auf die Datei. Auch die Installation kann ein bisschen dauern. Danach startest du Android Studio.



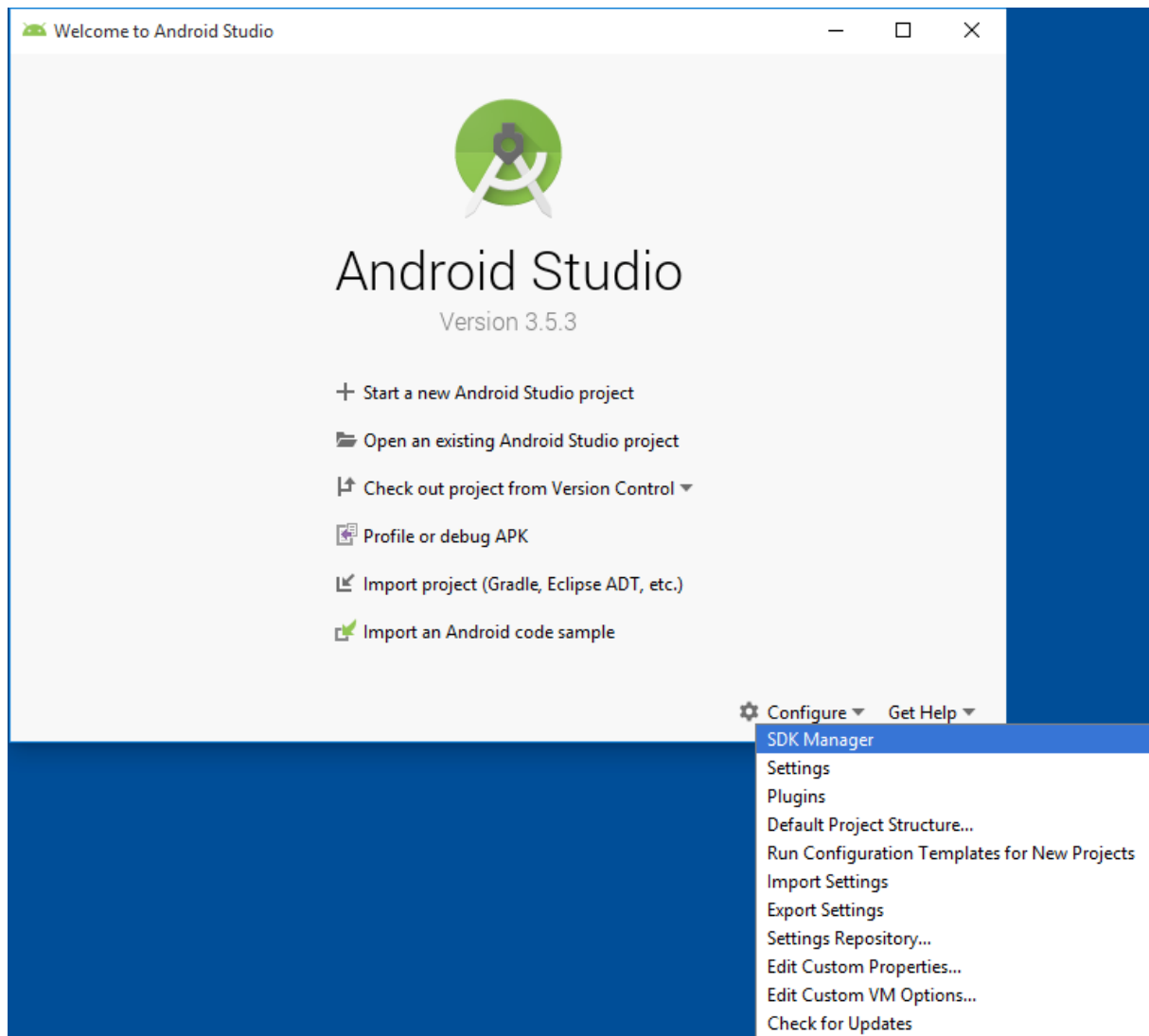
Nach dem Start des Programms wählst du aus, keine Einstellungen zu importieren.



Hier kannst du einmal auf Next klicken, nach der Grundeinrichtung beendest du Android Studio und startest es erneut, allerdings ohne ein Projekt zu beginnen.

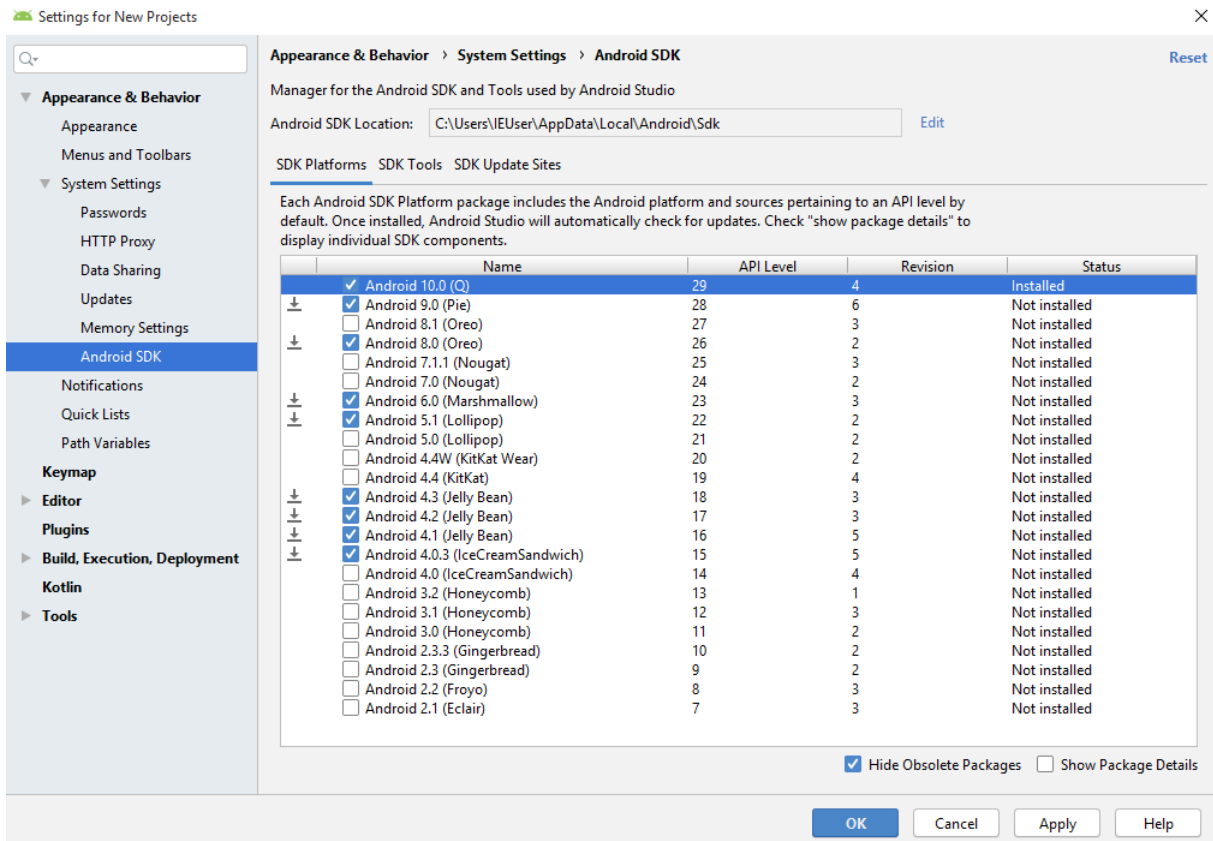
Den SDK-Manager verwenden, um die benötigten SDK-Versionen zu installieren.

Klicke auf dem Startbildschirm von Android Studio auf Configure (mit dem Zahnrad) und wähle dort „SDK Manager“.

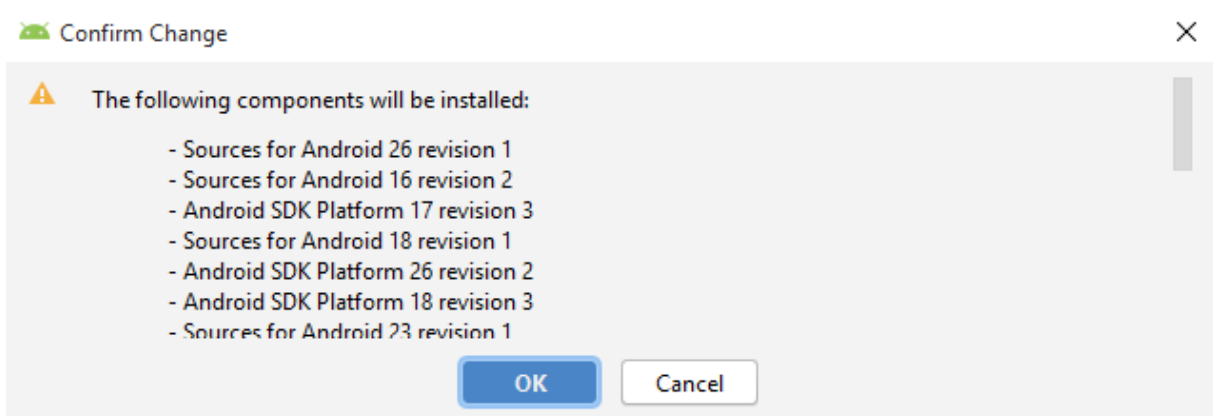


Dadurch kommst du auf ein Einstellungsfenster, in dem eine lange Liste von möglichen Android-SDK-Plattformen erscheint, die installierbar sind.

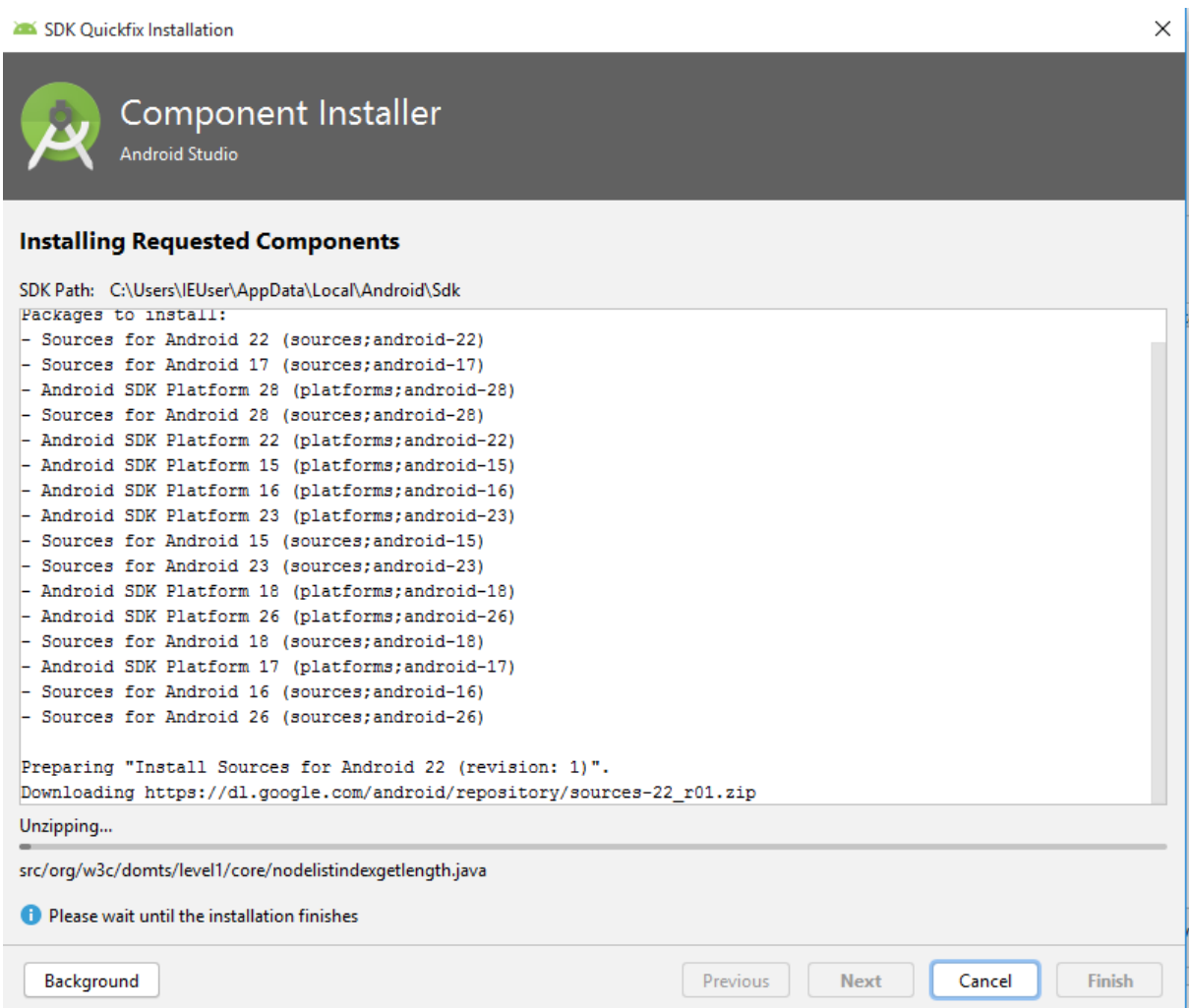
Wähle dort die Versionen aus, die du benötigst. Wichtig ist, dass du Android 4.03 (IceCreamSandwich) auswählst sowie Android 4.1 und gemäß Empfehlung von LiveCode zum Beispiel noch 4.2, 4.3, 5.1, 6.0, 7.0, 8.0... oder nach deinem Bedarf.



Nachdem du alle Versionen angewählt hast (hier nur beispielhaft), klickst du auf „OK“ oder „Apply“ – und der SDK-Manager lädt und installiert die entsprechenden Plattform-SDKs.



Mit OK bestätigen, und los geht's:

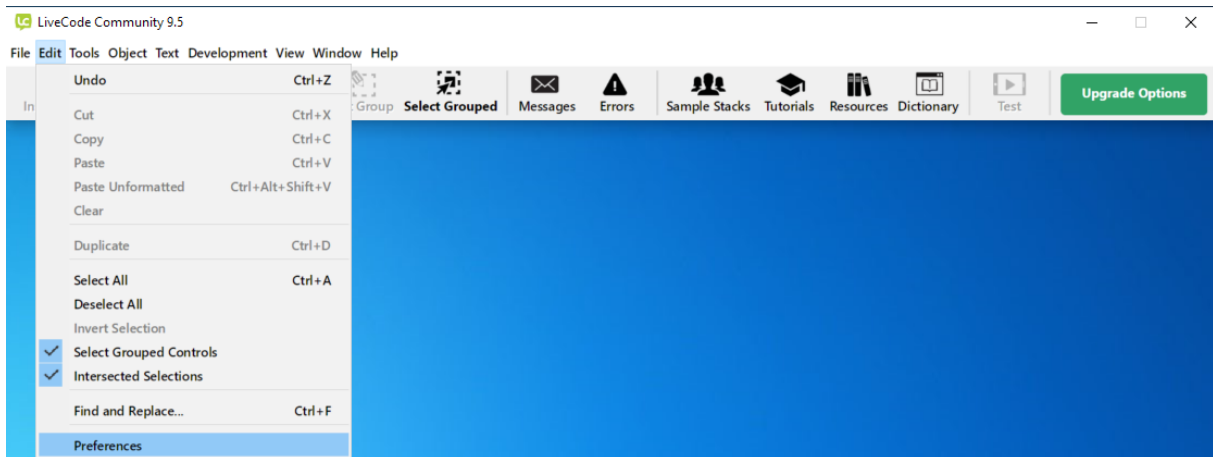


Das kann eine ganze Weile in Anspruch nehmen – also etwas Geduld.

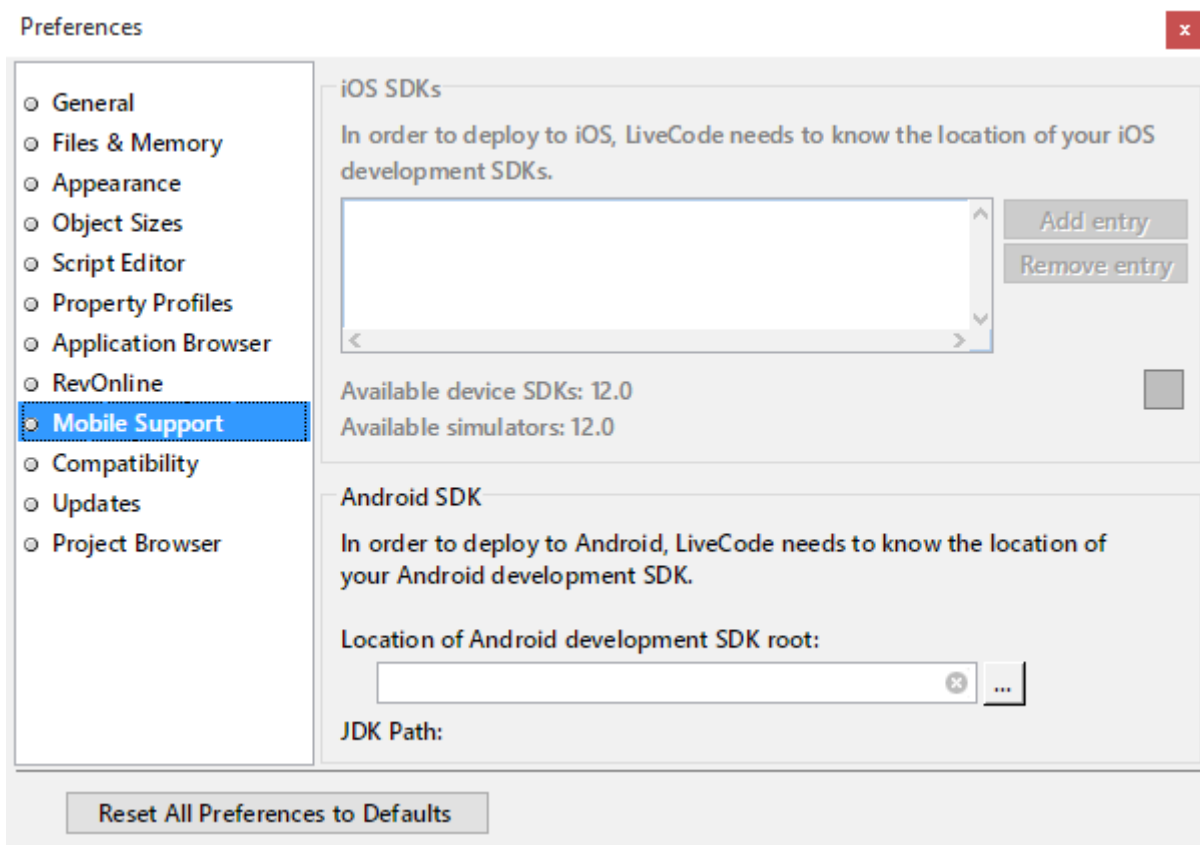
Anschließend hast du es im Grunde geschafft: Dein System ist bereit, APK-Dateien in LiveCode zu produzieren. Du musst es LiveCode nur noch sagen.

Pfad in LiveCode setzen

Starte LiveCode 9.5 und wähle aus dem Hauptmenü „Edit – Preferences“:



Im Einstellungsfenster, das nun erscheint, wählst du aus der linken Liste „Mobile Support“:



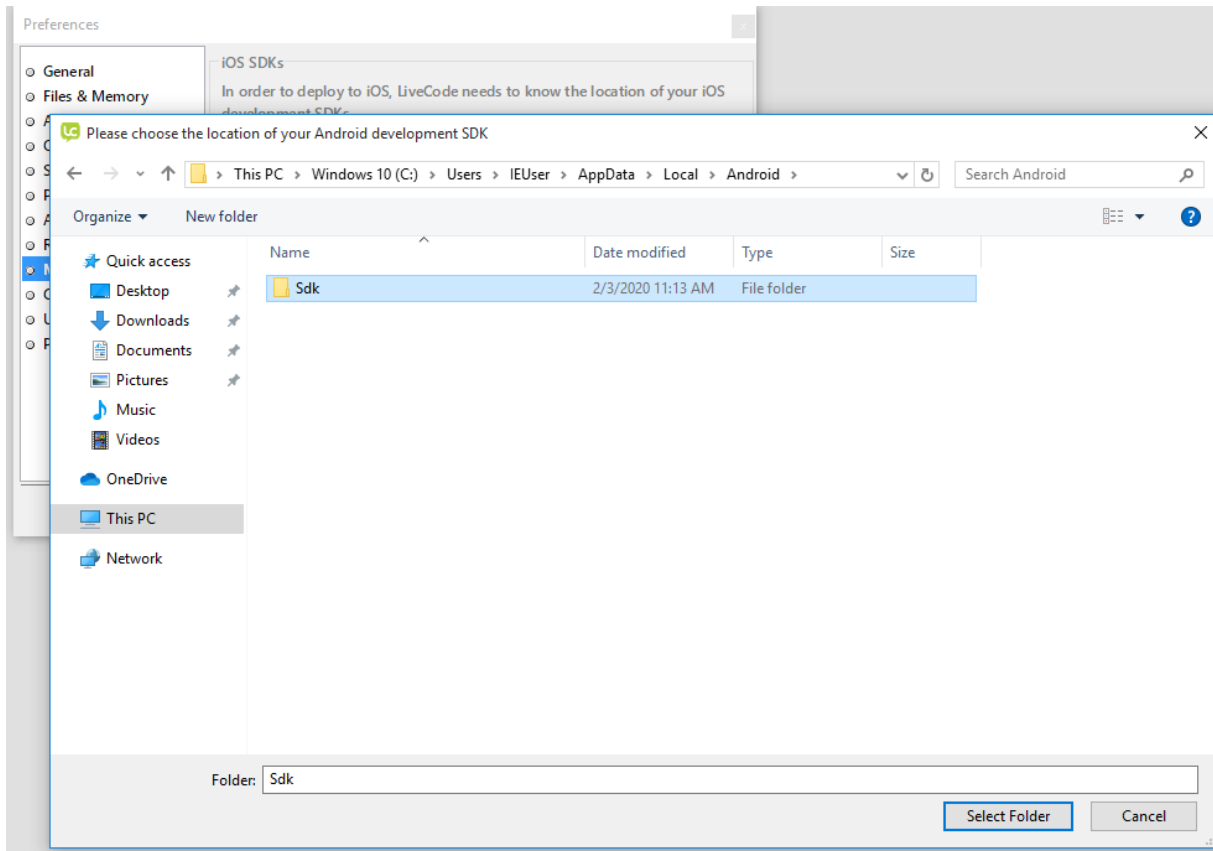
Hier ist die rechte untere Zeile wichtig, „Location auf Android development SDK root“ – hier muss nämlich der Dateipfad zu dem Ordner eingetragen werden, in dem sich das Android SDK befindet.

Wo befindet sich das SDK denn?

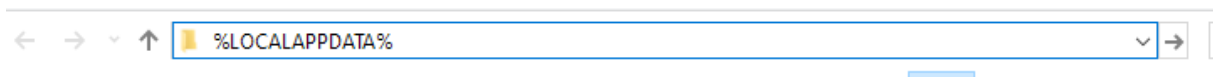
Wenn du alles so installiert hast wie vorher beschrieben, dann sollte sich dein SDK-Ordner hier befinden:

C:\Benutzer\deinName\AppData\Local\Android\Sdk

„deinName“ muss dabei natürlich durch deinen Benutzernamen ersetzt werden. Beachte auch, dass AppData ein Systemordner und damit ein standardmäßig unsichtbarer Ordner ist, der eventuell erst einmal sichtbar gemacht werden muss.

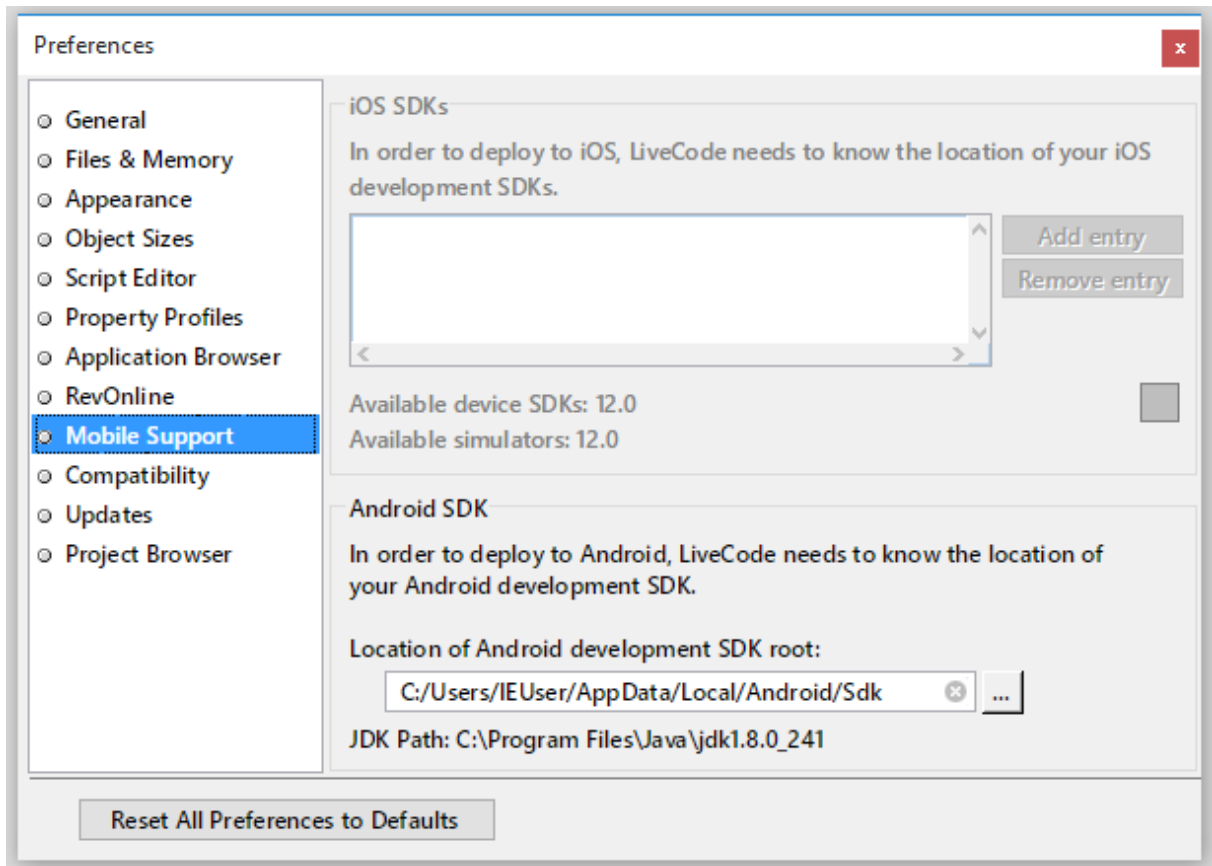


Ein einfacherer Weg geht so: Klicke auf den Ordnerauswahl-Button „...“. Gib in der oberen Zeile des Explorers ein: %LOCALAPPDATA% (mit den Prozentzeichen am Anfang und Ende).



Drücke Return, und der AppData/Local Ordner für deinen Benutzer öffnet sich. Wähle dort den Ordner „Android“, doppelklicke und markiere dann anschließend den Ordner „Sdk“.

Wenn du diesen Ordner ausgewählt hast, sollte in den Mobile-Support Einstellungen von LiveCode anschließend automatisch auch der Pfad zum JDK darunter erscheinen. Daran erkennst du, dass alles korrekt installiert ist und funktioniert.



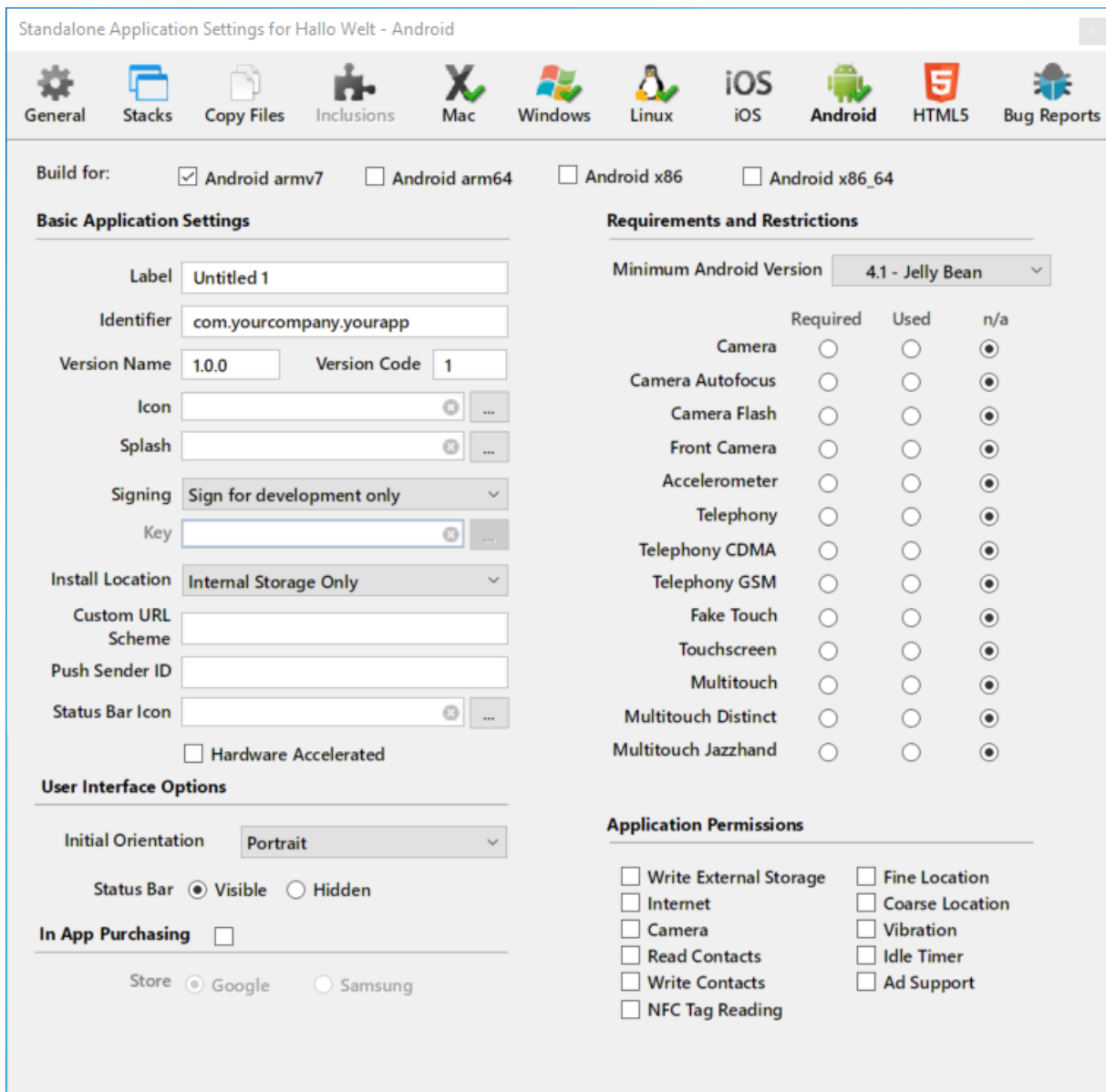
Fertig! Dein Windows-System ist jetzt für die Erstellung mobiler Android-Apps vorbereitet und dein LiveCode-System ebenfalls.

Wie erstelle ich die APK?

Dazu brauchst du in LiveCode natürlich eine App, die du auf dein mobiles Gerät bringen möchtest. Vielleicht ein großes Projekt, vielleicht auch nur ein Demo-Programm, das „Hallo Welt“ auf den Bildschirm schreibt.

Wenn dein Programm soweit ist, gehst du im Hauptmenü auf „File – Standalone Application Settings“.

Dort klickst du oben in der Leiste auf das Android-Symbol:



Hier kannst du wählen, dass deine App für Android kompiliert wird. In der obersten Zeile kannst du „Android armv7“, „Android arm64“ und weitere anwählen. Das sind die verschiedenen Prozessoren, die du in Android-Geräten finden kannst. Ältere Geräte haben einen armv7-Prozessor, die neueren zunehmend arm64. Wenn du nur armv7 wählst, sollte deine App auf so gut wie allen Android-Geräten laufen, da die neueren Prozessoren auch abwärtskompatibel sind. Wählst du aber zusätzlich arm64, dann wird eine für arm64-optimierte Binary zusätzlich erstellt und mit eingelinkt. Damit laufen neuere Geräte dann effizienter. Die anderen beiden Prozessoren kommen in gängigen Mobilgeräten nur selten vor.

Wichtig ist auch noch, dass du unter „Signing“ „Sign for development only“ wählst. Als „Minimum Android Version“ passt „4.1 Jelly Bean“ gut – vorausgesetzt, du hast diese Version auch im APK-Manager vorher installiert.

Vielleicht gibst du der App unter „Label“ noch einen passenden Namen. Alles Weitere kannst du erst einmal so lassen wie es ist. Später kannst du hier Finetuning betreiben, ein Icon einbinden, spezielle Rechte erteilen usw...

Nun kannst du das Fenster wieder schließen und mit „File – Save as Standalone Application“ wird deine APK erstellt, die du in deinem vorher ausgewählten Standalone-Ordner unter „Android“ findest.

Wie bekommst du die APK-Datei nun auf das Handy oder Tablet?

APK-Datei auf Mobilgerät kopieren:

Nun – die Datei auf das Mobilgerät zu bekommen sollte nicht allzu schwierig sein. Schließe das Gerät per USB an deinen PC an und kopiere die APK-Datei drauf. Ich empfehle, sie in den Ordner „Download“ zu kopieren. Statt USB geht es auch drahtlos über Bluetooth oder WLAN, zum Beispiel mit der App „myPhoneExplorer“ oder mit systeminternen Apps der Handy-Hersteller.

APK-Datei auf dem Mobilgerät installieren:

Nun musst du die App mit deinem Mobilgerät nur noch anwählen und installieren. Dazu musst du zuvor Android erlauben, Apps aus unbekanntem Quellen zu installieren. In älteren Android-Versionen kann man diese Einstellung für das gesamte System setzen. In neueren Android-Versionen kann man jeder einzelnen App das Recht geben, Apps aus unbekanntem Quellen zu installieren.

Meine Empfehlung für die Installation einer zuvor kopierten APK-Datei ist der „Total Commander“. Ihn findest du im Google PlayStore kostenlos und kannst ihn in wenigen Sekunden laden und installieren.



(Icon: Total Commander)

Dem Total Commander musst du nun in Android einmalig das Recht geben, Apps aus unbekanntem Quellen zu installieren. Suche in den Android-Einstellungen „unbekannte Apps“ – du findest es auch unter *Einstellungen zu Apps & Benachrichtigungen/Erweitert/Spezieller App-Zugriff*.

Dort musst du es dem „Total Commander“ erlauben, unbekanntem Apps zu installieren.

Danach startest du den „Total Commander“, wählst den Ordner „Download“ aus, tippst auf deine APK und wählst „Installieren“.

Wenn alles gut geht – und das hoffen wir natürlich sehr – hast du jetzt deine App auf deinem Mobilgerät und kannst sie nach Belieben testen.

Fortan ist alles viel einfacher: Änderungen in LiveCode machen, „Save as Standalone Application“, in Ordner „Download“ auf dem Mobilgerät kopieren, Total Commander starten, auswählen, installieren... weiter testen!

Viel Erfolg und viel Spaß!