

# Installing the Android SDK

Archive note: all links and images have been disabled as their destinations and/or source locations no longer exist.

Created by Brian Immel, last modified on Aug 8, 2019

- Overview
- Compatibility and Download
  - Required Android Packages
  - Android SDK / Target Android platform
    - Supported versions
    - Unsupported versions
  - Android Native Add-on Module Development on macOS
- Installation
  - macOS
  - Installing Android SDK Tools on macOS
- Windows
  - Installing Android SDK Tools on Windows
- Installing Android Packages with Android SDK Manager
  - To install Android SDK on macOS
  - To install Android SDK on Windows
- Installing x86 Emulator Packages
- Configuring Studio
- Configuring the CLI

## Overview

This guide describes where to obtain and how to install the Google Android SDK and associated packages.

## Compatibility and Download

Titanium requires the Android SDK Tools to be installed in order to allow you to develop Android applications.

Note: Oracle JDK is a prerequisite for Android development and should be installed first.

Note that macOS systems need to have the Java 6 runtime installed. See Installing Oracle JDK for instructions.

The *Android SDK Manager* installer may be obtained from the Android Studio and SDK tools (<https://developer.android.com/studio>) site.

## Required Android Packages

The following packages, and the respective versions shown, must be installed to be able to develop for Android:

Package	Minimum Version
Android SDK Tools	Rev 25
Android SDK Platform-tools/Build-tools	Rev 23
Android SDK	You need to have at least one Android SDK installed within the supported minimum and maximum target versions. For Android modules, you need to install Android SDK 6.0.x (API 23) and SDK 6.0.0+.

Titanium SDK 8.0.0+ requires Android SDK Tools v28 or newer as a minimum requirement.

Titanium SDK 7.0.0+ requires Android SDK Tools v25 or newer as a minimum requirement.

Take caution before upgrading these packages, as changes to the way they work has broken the Titanium toolchain a number of times in the past. Although these problems are often beyond our control, we always do our utmost to fix them as soon as we are made aware of them.

With this in mind, it's important to only upgrade these packages *between* major projects, so that you have time to fix any problems that may result. Always consult the Android Tools Release Notes (<http://developer.android.com/sdk/tools-notes.html>) and Known Issues (<http://tools.android.com/knownissues>) first, and refer to our Installation Troubleshooting guide to check whether there are any further actions that need to be taken to make the Titanium and the Android SDK compatible.

# Android SDK / Target Android platform

Each Titanium SDK supports building against a specific range of Android versions, as shown in the following table, and requires at least one of these versions to be installed. If you specify a `android:targetSdkVersion` in the `tiapp.xml` file of your project, you must specify one within the target min and max values. The minimum Android/SDK version column indicates the minimum version of Android that a device can run, which a Titanium application supports.

## Supported versions

Titanium SDK Version	Min Target Android/SDK Version ( <code>android:targetSdkVersion</code> )	Max Target Android/SDK Version ( <code>android:targetSdkVersion</code> )	Minimum Android/SDK Version ( <code>android:minSdkVersion</code> )
8.0.0 - latest	6.0.x (API 23)	9.0.x (API 28)	4.4.x (API 19)
7.5.0 - 7.5.1	6.0.x (API 23)	9.0.x (API 28)	4.1.x (API 16)
7.3.0 - 7.4.1	6.0.x (API 23)	8.1.x (API 27)	4.1.x (API 16)
7.0.0 - 7.2.0	6.0.x (API 23)	7.0.x (API 25)	4.1.x (API 16)

## Unsupported versions

Titanium SDK Version	Min Target Android/SDK Version ( <code>android:targetSdkVersion</code> )	Max Target Android/SDK Version ( <code>android:targetSdkVersion</code> )	Minimum Android/SDK Version ( <code>android:minSdkVersion</code> )
6.2.0 - 6.3.0	7.1.x (API 25)	6.0.x (API 23)	4.1.x (API 16)
6.0.0 - 6.1.x	6.0.x (API 23)	6.0.x (API 23)	4.1.x (API 16)
5.1.0 - 5.5.x	6.0.x (API 23)	6.0.x (API 23)	4.0.x (API 14)
5.0.0 - 5.0.x	5.0.x (API 21)	6.0.x (API 23)	4.0.x (API 14)
4.0.0 - 4.1.x	5.0.x (API 21)	5.1.x (API 22)	4.0.x (API 14)
3.4.1 - 3.5.1	4.0.x (API 14)*	5.0.x (API 21)**	2.3.x (API 10)
3.3.0 - 3.4.0	4.0.x (API 14)*	4.4.x (API 19)	2.3.x (API 10)
3.2.0 - 3.2.3	2.3.x (API 10)*	4.4.x (API 19)	2.3.x (API 10)
3.1.2 - 3.1.3	2.3.x (API 10)	4.3.x (API 18)	2.3.x (API 10)
3.1.1	2.3.x (API 10)	4.2.x (API 17)	2.3.x (API 10)
3.1.0	2.2 (API 8)	4.2.x (API 17)	2.2 (API 8)
2.1.2 - 3.0.2	2.2 (API 8)	4.1.x (API 16)	2.2 (API 8)
2.0 - 2.1.1	2.2 (API 8)	4.0.x (API 15)	2.2 (API 8)
1.8.x	2.2 (API 8)	3.x.x (API 11)	2.2 (API 8)
1.7.x	2.1 (API 7)	3.x.x (API 11)	2.1 (API 7)

## Notes

\* If you are building an Android module, you need to have Android SDK 6.0.x (API 23) installed if using Release 6.0.0 and greater.

\*\* The Titanium SDK does not support the Android 4.4W SDK (<http://developer.android.com/wear/index.html>) (API 20), also known as the Android Wear SDK.

Most mobile device manufacturers have been licensed to use Google's enhanced API, which provides support for Maps and other functionality. If this is the case for your target devices, you will need to install the relevant Google packages, listed as *Google APIs by Google Inc., Android API x...* by the **Android SDK Manager** tool. In Studio, choose the SDKs with the naming format "Google APIs x.x" to use the enhanced APIs, or those without the "Google APIs" prefix otherwise.

Android SDK packages can be installed using the **Android SDK Manager** tool. See Installing the Android SDK for detailed instructions.

The default Android SDK can be configured using Studio's *Preferences*, and then selected per-project using the *Run Configurations*.

For Android API levels 14 and above, the emulator system images are installed separately from the SDK platform packages. To run in the emulator for Android API 14+, you must install at least one of the emulator system images.

## Android Native Add-on Module Development on macOS

The Xcode command line tools are required to develop native Android add-on modules on macOS.

See Installing the iOS SDK: Installing Command Line Tools for detailed instructions.

## Installation

### macOS

Note that the typical filesystem location of this software can be found in the macOS Software Locations section of these guides.

### Installing Android SDK Tools on macOS

1. Point a browser towards Android Studio and SDK tools (<https://developer.android.com/studio?pkg=studio>).
2. Click **Download Android Studio**.
3. Once the .dmg file has downloaded, the **Android Studio installer** should pop up. **Drag the Android Studio icon** to the **Applications folder**.

## Windows

Note that the typical filesystem location of this software can be found in the Windows Software Locations section of these guides.

### Installing Android SDK Tools on Windows

1. Point a browser towards Android Studio and SDK tools (<https://developer.android.com/studio?pkg=studio>).
2. Click **Download Android Studio**.
3. Once the .exe file has downloaded, **double-click on the executable file** to start the install process.
4. Click the **Next >** button in the Android Studio Setup window.
5. **Select the components** you wish to install. By default, you should install both **Android Studio** and **Android Virtual Device**. Click the **Next >** button.
6. Choose the install location and click **Next >** button.
7. Click the **Install** button.
8. Once installation is complete, click **Next >** button.
9. Click the **Finish** button.
10. Launch Android Studio so it can download any necessary components for your version of Windows.
11. Click the **Finish** button once Android Studio has finished downloading components.

Refer to the Windows Software Locations section, and add the path of the **Android SDK Tools** directory to your system's PATH.

Then proceed to the Installing Android Packages with Android SDK Manager section.

# Installing Android Packages with Android SDK Manager

The *Android SDK Tools* package that has been installed contains the *Android SDK Manager* that is used to install the rest of the Android SDK packages.

Launching the *Android SDK Manager* differs depending on the platform.

- On macOS, go to Launchpad and click on Android Studio.
- On Windows, double-click on the Android Studio icon. Note: You may get prompted to run as an administrator. If so, from the Windows Start Menu and select *Run as administrator*. Enter the Administrator password.

In the Android SDK Manager, you will see different versions of the following items listed:

- **Android SDK Tools** (mandatory) - includes the *Android SDK Manager* and *Android Virtual Device Manager* (android executable)
- **Android SDK Platform Tools** (mandatory) - includes Android Debug Bridge, (adb executable)
- **SDK Platform** - provides the libraries for building for a specific version of the Android OS that uses the standard Google APIs.
- **Google APIs by Google Inc** - provides the libraries for building for a specific version of Android that uses the *enhanced* Google APIs (includes maps support).
- **System Image** - For Android API versions 14 and above, the emulator system images are packaged separately from the SDK Platform package. Three versions of the system images are provided - ARM, Intel x86, and MIPS. The ARM system image is supplied by default in earlier SDK Platform packages, and corresponds to the hardware in most phones.

## To install Android SDK on macOS

1. Open **Android Studio**.
2. Go to **Tools > SDK Manager**.
3. Under **Appearance & Behavior > System Settings > Android SDK**, you will see a list of SDK Platforms to choose from. **Select the SDK(s)** you wish to use and click **OK** button.
4. Android Studio will confirm your selection. Click **OK** button.
5. Once the requested components have been installed, click **Finish** button.

## To install Android SDK on Windows

1. Open **Android Studio**.
2. In the Welcome to Android Studio window, click on **Configure > SDK Manager**.
3. Under **Appearance & Behavior > System Settings > Android SDK**, you will see a list of SDK Platforms to choose from. **Select the SDK(s)** you wish to use and click **OK** button.
4. Android Studio will confirm your selection. Click **OK** button.
5. Click the **Accept** radio button and click the **Next** button.
6. Once the requested components have been installed, click **Finish** button.

The following steps are the normal process to install an Android SDK with Axway Appcelerator Studio (Appc Studio) but due to the bug noted in TISTUD-9035, Studio cannot install Android SDKs at this time.

In **Appc Studio**, confirm that your Android SDK(s) have been installed:

1. Launch **Axway Appcelerator Studio**.
2. **Login** using your credentials (if you haven't already).
3. Go to **Axway Appcelerator Studio > Preferences** in the menu bar.
4. In the Preferences window, go to **Studio > Platforms > Android**.
5. Click the **Install SDKs...** button in the **General** section.
6. Select the **Android SDK(s)** you wish to use.
7. Click the **Install** button.
8. Click the **Accept** button for the **Review License** window.
9. In the **Progress** tab, you should see the status of the Android SDK(s) being downloaded and installed.
10. Review the installed SDKs in the General section of the Preferences window. You should now see the newly installed Android SDK(s).

Even though Appc Studio not display newly installed Android SDK(s), they are installed and available for use. You can confirm this by creating a new AVD (Android Virtual Device) and see the options to use the newly installed Android SDK(s).

There are two ways to create an AVD to use in Appc Studio: Create the AVD in Appc Studio or create it in Android Studio. Appc Studio should pick up the new AVD automatically if you created a new one via Android Studio.

To create an AVD via Appc Studio:

1. Open **Appc Studio**.
2. In the menu, go to **Axway Appcelerator Studio > Preferences**.
3. In the Preferences window, go to **Studio > Platforms > Android**.
4. In the Android Preferences section, click on the **Create AVDs...** button.
5. Give the AVD a name in the **AVD Name** field using a-z A-Z 0-9 . \_ - characters only (no spaces or other special characters).
6. Select the desired device in the **Device** field.
7. Select the **Android SDK** from the **System Image** field.
8. After making your selections, click **OK** button to build the AVD.

To create an AVD via Android Studio:

1. Open **Android Studio**.
2. In the menu, go to **Tools > AVD Manager**.
3. Click the **Create Virtual Device...** button.
4. Select the **category** of virtual device you want to use.
5. Select the **device** and click the **Next** button.
6. Select the **System Image** you want to use in the virtual device and click the **Next** button.
7. Android Studio will provide a name for you automatically based on the virtual device and system image you selected. Confirm you selections and click the **Finish** button.

Refer to the Titanium Compatibility Matrix for the latest list of required packages.

## Installing x86 Emulator Packages

Some versions of the Android API are available with Intel x86 emulators. Look for the "Intel x86 Atom System Image" in the *Android SDK Manager*.

The x86 emulator may run faster than the standard ARM emulator, but may show minor differences with the ARM emulator. Extra software is required to take advantage of the x86 emulator - see *Configuring Virtual Machine Acceleration* (<http://developer.android.com/tools/devices/emulator.html#accel-vm>) in the Android Tools documentation.

## Configuring Studio

To use the Android SDK with Studio, the path to the SDK must be set in the **Preferences** dialog.

1. Open the Preferences dialog:
  1. On macOS, from the menu bar, select **Appcelerator Studio > Preferences**.
  2. On Windows, from the menu bar, select **Window > Preferences**.
2. In the **Preferences** dialog, navigate to **Studio > Platforms > Android SDK**.
3. Set your Android SDK path. Enter the path to your Android SDK in the **Android SDK Home** textbox or click the **Configure** button to navigate to the SDK directory.
4. Select your default Android SDK. Select a **Google APIs** item from the **Default Android SDK** drop-down menu.

## Configuring the CLI

To use the Android SDK with the Titanium CLI, the path to the SDK must be set in the Titanium CLI configuration settings. Run the following command and replace `/Users/appc/sdk/android-sdk/` with the path to your Android SDK:

```
titanium config android.sdk /Users/appc/sdk/android-sdk/
```