# Setting up the Android development environment

## Overview

**Prerequisite:** If you intend to use IBM MobileFirst Platform Studio or CLI to develop your application, first read the Setting up the MobileFirst development environment (../../setting-up-your-development-environment/setting-up-the-mobilefirst-development-environment/) tutorial.

Before you can develop, deploy, and test your Android applications, you must follow several steps to set up the Android development environment. These steps include installing the Android SDK and ADT plug-in (if you are using Eclipse) or installing the Android Studio.

This tutorial covers the following topics:

- Android SDK
- Android Studio
- ADT Eclipse plug-in
- SDK platform and virtual devices
- Devices and developments

## Android SDK

The Android SDK provides the tools and APIs that you need to develop applications on the Android platform by using the Java programming language.

To install the SDK, visit the Android Developer Tools page (http://developer.android.com/sdk/index.html) and scroll to **Other Download Options → SDK Tools Only** to download the SDK installer.

**Note:** To use the Android SDK, use Oracle Java Runtime Environment (JRE).

## Android Studio

You can download Android Studio from https://developer.android.com/sdk/index.html (https://developer.android.com/sdk/index.html)
If you choose to work with Android Studio, you do not need to install the ADT Eclipse plug-in.

After installing Android Studio, you must configure MobileFirst Studio to use it.
In Eclipse, select **File > Preferences > MobileFirst** and provide the root location for the Android Studio folder.

**Note:** Gradle is currently not supported in hybrid applications that are generated either in MobileFirst Studio or via the command-line interface (CLI).

## ADT Eclipse plug-in

The Android Development Tools (ADT) plug-in for Eclipse is an integrated environment in which you can build rich Android apps. If you are using Android Studio, you can skip this step.

To install ADT, in Eclipse:

1. Click **Help > Install New Software**.
2. Click **Add** and enter the name and location of the ADT plug-in.
    - The plug-in location is: `https://dl-ssl.google.com/android/eclipse/`

3. Select **Developer Tools** and click **Next** to proceed with the plug-in installation.

# SDK platform and virtual devices (Eclipse users only)

## Adding the SDK platform

1. After the ADT plug-in installation is complete and Eclipse restarted, click **Window > Android SDK Manager**.
2. In the SDK Manager, Under Packages, select the required API Levels, then click **Install**.

If the Android SDK Manager / AVD Manager is not visible on the Eclipse menu bar:

1. From the Window menu bar item, click on **Window > Customize Perspective > Command Groups Availability**.
2. Select the Android SDK and AVD Manager check boxes.

MobileFirst applications support API Levels 10 (Android 2.3) and above.
In a fresh installation of the ADT plug-in, it is bundled only with the latest available API Level. You might want to use the SDK Manager and add additional API Levels, such as 17, 18, and 19.

## Adding a virtual device

To create a virtual device (emulator), click **Window > AVD Manager** and click **New** to create an Android Virtual Device (AVD) according to your needs.

For a detailed explanation of all available options, see the Android Developer documentation at Managing AVDs with AVD Manager (http://developer.android.com/tools/devices/managing-avds.html).

You can now set up a device and develop applications.

# Devices and developments

## Setting up a device

To set up an Android device for testing and debugging, see the Android Developer documentation at Using Hardware Devices (http://developer.android.com/tools/device.html).

## Developing applications

Building a MobileFirst application generates a separate Android project that is automatically displayed in Eclipse. For more information, see the *Previewing your application on Android* (file:////home/travis/build/MFPSamples/DevCenter/_site/tutorials/en/foundation/7.0/hello-world/previewing-application-android/) tutorial.

You can then run the Android application on the Android Emulator or device: Right-click the Android project and select **Run As > Android Application**.