

By: Koen Vlaswinkel  15  3


How To Build Android Apps with Jenkins

Jan 23, 2014 Java Ubuntu

In this article, Jenkins will be setup to build Android apps. Jenkins will already need to be installed, so if it hasn't been installed yet, please follow the [steps here](#) first. For this article, I assume your project uses version control like Git or Subversion. I will mostly talk about the new build system called Gradle, but this article can also be used to set up an Ant build.

Installing the software needed

To begin, the Android SDK should be installed. To find the download location, visit the [Android SDK download page](#). Then click Download for other platforms and copy the link of the Linux version (SDK Tools only).


[Developers](#)

[Design](#)

[Develop](#)

[Distribute](#)

[Training](#)
[API Guides](#)
[Reference](#)
[Tools](#)
[Google Services](#)
[Samples](#)

Developer Tools

[Download](#)

[Setting Up the ADT Bundle](#)

[Setting Up an Existing IDE](#)

[Android Studio](#)

[Exploring the SDK](#)

[Download the NDK](#)

[Workflow](#)

[Support Library](#)

[Tools Help](#)

[Revisions](#)

[Samples](#)

[ADK](#)

Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in ADT (Android Developer Tools) to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

Android Studio Early Access Preview

A new Android development environment called Android Studio, based on IntelliJ IDEA, is now available as an **early access preview**. For more information, see [Getting Started with Android Studio](#).

If you prefer to use an existing version of Eclipse or another IDE, you can instead take a more customized approach to installing the Android SDK. See the following instructions:

[USE AN EXISTING IDE](#)
[SYSTEM REQUIREMENTS](#)
[DOWNLOAD FOR OTHER PLATFORMS](#)

ADT Bundle


Platform	Package	Size	MD5 Checksum
Windows 32-bit	adt-bundle-windows-x86-20131030.zip	503599460 bytes	cd490a531ec24667354f6473e999b988
Windows 64-bit	adt-bundle-windows-x86_64-20131030.zip	503735416 bytes	ddd4bb1b9028015779d68dde01f96b14
Mac OS X 64-bit	adt-bundle-mac-x86_64-20131030.zip	470386961 bytes	3e80e7a92b549029d91bdcf2ae82657f
Linux 32-bit	adt-bundle-linux-x86-20131030.zip	496876498 bytes	d389139ad9f59a43bdd34c94bc850509
Linux 64-bit	adt-bundle-linux-x86_64-20131030.zip	497171697 bytes	99b51a4f0526434b083701a896550b72

SDK Tools Only

Platform	Package	Size	MD5 Checksum
Windows 32 & 64-bit	android-sdk_r22.3-windows.zip	108847452 bytes	9f0fe8c8884d6aee2b298fee203c62dc
	installer_r22.3-windows.exe (Recommended)	88845794 bytes	ad50c4dd9e23cee65a1ed740ff3345fa
Mac OS X 32 & 64-bit	android-sdk_r22.3-macosx.zip	74893875 bytes	ecde88ca1f05955826697848fcb4a9e7
Linux 32 & 64-bit	android-sdk_r22.3-linux.tgz	100968558 bytes	6ae581a906d6420ad67176dff25a31cc

Except as noted, this content is licensed under [Creative Commons Attribution 2.5](#). For details and restrictions, see the [Content License](#).

[About Android](#) | [Legal](#) | [Support](#)



After you have copied the link, switch over to the SSH session. CD into /opt and download the Android SDK:

```
cd /opt
```

```
wget <link you copied here>
```

At time of writing, the following command should be executed:

```
wget http://dl.google.com/android/android-sdk_r22.3-linux.tgz
```

Then unzip the file:

```
tar zxvf <filename of the just downloaded file>
```

You can now remove the file you just downloaded:

```
rm <filename of the just downloaded file>
```

Now some environment variables will need to be set. Edit `/etc/profile.d/android.sh` (`nano /etc/profile.d/android.sh`) and add the following:

```
export ANDROID_HOME="/opt/android-sdk-linux"  
export PATH="$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools:$PATH"
```

Then reload the file:

```
source /etc/profile
```

If you are going to use Git, also install Git:

```
sudo apt-get install git-core
```

Configuring the Android SDK

To build your project, the Android SDK will need a few packages. First, update the SDK itself (a few licenses will need to be accepted):

```
android update sdk --no-ui
```

This will install **all** Android versions, so it will quite a long time. Read more if you do not wish to install all packages.

If you only want to install specific parts, more information on updating the SDK is available in [this StackOverflow question](#).

The packages you will most likely need are installed by the following command (replacing the "19" in `android-19` with the most recent Android SDK version:

```
android update sdk -u --filter platform-tools,android-19
```

However, this won't install the build tools needed for Gradle. For that, execute the following command:

```
android list sdk --all
```

In this list, find the first item called Android SDK Build-tools, version xx.xx.xx. Remember the number that is listed before the item and execute the following:

```
android update sdk -u --all --filter <number>
```

Also do this for Android Support Repository and Android Support Library.

If you know your project will be using a specific version of the build tools, look for that item in the list. It can be seen in the build.gradle file as the `buildToolsVersion`:

```
android {  
    buildToolsVersion "18.1.1"  
    compileSdkVersion 18  
}
```

Also install the appropriate platform version that can be seen in the build.gradle as `compileSdkVersion`.

Now, for the Android SDK to be accessible by Jenkins, execute the following:

```
sudo chmod -R 755 /opt/android-sdk-linux
```

Also, if you are on a 64-bit OS, install the 32-bit libraries:

```
sudo apt-get install ia32-libs
```

Then reboot your Droplet:

```
sudo shutdown -r now
```

Configuring your Android project

If your project is still built by Eclipse, a build script will need to be created. The easiest option is to do this right from Eclipse, by going to File -> Export, then selecting Android -> Generate Gradle build files. Then select the project and click Finish. This should create a Gradle project, which can be built by Jenkins.

If you don't want to use Gradle, Apache Ant can still be used. You can skip this step if you don't know what this is or when the Gradle project export fails. The only step needed is to execute the following command on a PC with the Android SDK installed:

```
android update project -p .
```

Configuring Jenkins

Jenkins also needs to be configured. First, a JDK needs to be installed. To do this, visit Manage Jenkins -> Configure System. Then find JDK and click Add JDK. You need to have an Oracle account. To enter your credentials, click on Please enter your username/password in the JDK section. If you don't have one, follow the link in the next screen. When you've entered those, go back to the JDK configuration. Enter a name – I recommend naming it after the Java version. Then check Install automatically and select the most recent Java SE Development Kit version. This should install the JDK later on.

If you are using Ant, execute the same steps in the Ant section.

Now visit Manage Jenkins -> Manage Plugins -> Available. Check the following plugins and click Install without restart:

- Gradle Plugin (not needed if using Ant)
- Git Plugin (if using Git)
- Android Emulator Plugin (if you want to use an emulator)

This will automatically install a few other plugins. If you are going to use Gradle, execute the same steps as for the JDK to install Gradle automatically. At the time of writing, you will need Gradle 0.9. This is not needed if you are using the Gradle wrapper, i.e. you have a gradlew file in your project that is checked in to source control.

Setting up the Job

Now, the job can be setup. Click New Job, give it a name and select Build a free-style software project. Then click OK. In the section Source Code Management, select the version control system your project is using and give the information. Then in the Build section, add Invoke Gradle script if using Gradle and Invoke Ant if using Ant. Fill in all parameters. For Gradle this will be the most likely parameters:

The screenshot shows the 'Build' configuration section of a Jenkins job. The 'Invoke Gradle script' option is selected. The 'Gradle Version' is set to '0.9'. The 'Build step description' is empty. The 'Switches' field is empty. The 'Tasks' field is set to 'clean build'. The 'Root Build script' field is empty. The 'Build File' field is empty. A note below the 'Build File' field states: 'Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)'. The checkbox 'Force GRADLE_USER_HOME to use workspace' is checked. There are 'Add build step' and 'Delete' buttons at the bottom.

And for Ant these:

The screenshot shows the 'Build' configuration section of a Jenkins job. The 'Invoke Ant' option is selected. The 'Ant Version' is set to 'Default'. The 'Targets' field is set to 'clean build'. There are 'Advanced...' and 'Delete' buttons at the bottom.

This will build the project, but the APK files will not be saved yet. To configure this, add the Post-build action called Archive the artifacts. For files to archive type `**/*.apk`.

Click Save and the building can start! Click Build now. The JDK will be installed and after a while the build will have finished! On the page of the project, there is a heading called Last Successful Artifacts. Here all the APK files generated will be showed:

Project TestProject



The screenshot shows the Jenkins interface for a project named 'TestProject'. It features three main sections: 'Workspace' with a folder icon, 'Last Successful Artifacts' with a box icon, and 'Recent Changes' with a notepad icon. The 'Last Successful Artifacts' section lists four APK files with their sizes and 'view' links.

Artifact Name	Size	View Link
TestProject-beta-unsigned.apk	801.05 KB	view
TestProject-debug-unaligned.apk	819.91 KB	view
TestProject-debug.apk	820.25 KB	view
TestProject-release-unsigned.apk	801.05 KB	view

What more?

The first Android project has been built on your Jenkins instance. Congratulations!

The Jenkins instance can hold more than 1 job, so create more and develop more Android apps!

Gradle can also sign the apps automatically, but this sensitive information should be stored in the build.gradle file, which is not recommended. This is an example of a complete build.gradle that can be used in combination with Jenkins:

```
buildscript {  
    repositories {  
        mavenCentral()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:0.7.+'  
    }  
}
```

```
    }  
  }  
  apply plugin: 'android'  
  
  repositories {  
    mavenCentral()  
    flatDir {  
      dirs 'libs'  
    }  
  }  
}  
  
android {  
  compileSdkVersion 19  
  buildToolsVersion '19.0.1'  
  
  signingConfigs {  
    release {  
      storeFile file("keystores/release.keystore")  
      storePassword "your-keystore-password"  
      keyAlias "your-alias"  
      keyPassword "your-alias-password"  
    }  
  }  
  
  buildTypes {  
    release {  
      zipAlign true  
      signingConfig signingConfigs.release  
    }  
    debug {  
      zipAlign true  
    }  
  }  
}  
  
dependencies {  
  compile 'com.android.support:appcompat-v7:19.0.0'  
  compile 'com.android.support:support-v4:19.0.0'  
}
```

Submitted by: Koen Vlaswinkel



Heart

15



Share

Subscribe



Author:

Koen Vlaswinkel

Spin up an SSD cloud server in under a minute.

Simple setup. Full root access.
Straightforward pricing.

DEPLOY SERVER

Related Tutorials

How To Install Java with Apt-Get on Ubuntu 16.04

How To Optimize Your Tomcat Installation on Ubuntu 14.04

How To Install Apache Kafka on Ubuntu 14.04

How To Install Solr 5.2.1 on Ubuntu 14.04

How To Install Apache Tomcat 8 on CentOS 7

3 Comments

Leave a comment...

Logged in as:



Notify me of replies
to my comment

Comment

nullcognitiondev *June 10, 2014*

Thanks, after trying a lot of tuts this one was the thorough-est.
Prior to any bash "android update sdk" commands, I had to change my permission with
sudo chmod -R 777 /opt/android-sdk-linux
Then do the update then change to sudo chmod -R 755 /opt/android-sdk-linux



spamme *July 11, 2014*

How do you install the Android Support Library without the UI?



marc769294 *August 14, 2014*

Great article, thanks!



This work is licensed under a Creative
Commons Attribution-NonCommercial-
ShareAlike 4.0 International License.



Copyright © 2016 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [RSS](#) 

[Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Get Paid to Write](#)