

ANDROID Version History

APIs & SDK Version

Version History

- The version history of the Android mobile operating system began with the public release of the Android beta on November 5, 2007.
- The first commercial version, Android 1.0, was released on September 23, 2008.
- Android is continually developed by Google and the Open Handset Alliance, and it has seen a number of updates to its base operating system since the initial release.

Code Names

- Versions Android 1.0 and 1.1 were not released under specific code names, although Android 1.0 and 1.1 was unofficially known as '**Astro Boy**' and '**Bender**'.
- Android code names are confectionery-themed and have been in alphabetical order since 2009's Android 1.5 (Cupcake). The most recent version of Android is Android 11 (R), which was released in September, 2020.

What is API Level?

- API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform. These API levels are also known as SDK versions.
- The Android platform provides a framework API that applications can use to interact with the underlying Android system. The framework API consists of:
 - A core set of packages and classes
 - A set of XML elements and attributes for declaring a manifest file
 - A set of XML elements and attributes for declaring and accessing resources
 - A set of Intents
 - A set of permissions that applications can request, as well as permission enforcements included in the system

Unofficial names
 Astro Boy ☐ 1.0
 Bender ☐ 1.1

Name	Version number(s)	Initial stable release date	Supported (security fixes)	API level
No official codename	1.0	September 23, 2008	No	1
	1.1	February 9, 2009	No	2
Cupcake	1.5	April 27, 2009	No	3
Donut	1.6	September 15, 2009	No	4
Eclair	2.0 – 2.1	October 26, 2009	No	5 – 7
Froyo	2.2 – 2.2.3	May 20, 2010	No	8
Gingerbread	2.3 – 2.3.7	December 6, 2010	No	9 – 10
Honeycomb	3.0 – 3.2.6	February 22, 2011	No	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	No	14 – 15
Jelly Bean	4.1 – 4.3.1	July 9, 2012	No	16 – 18
KitKat	4.4 – 4.4.4	October 31, 2013	No	19 – 20
Lollipop	5.0 – 5.1.1	November 12, 2014	No	21 – 22
Marshmallow	6.0 – 6.0.1	October 5, 2015	No	23
Nougat	7.0 – 7.1.2	August 22, 2016	No	24 – 25
Oreo	8.0 – 8.1	August 21, 2017	Yes	26 – 27
Pie	9	August 6, 2018	Yes	28
Android 10	10	September 3, 2019	Yes	29
Android 11	11	September 8, 2020	Yes	30

Q -->

R -->

Application Forward Compatibility

- ❖ Each successive version of the Android platform can include updates to the Android application framework API that it delivers. These updates are also known as OTA (Over the Air) updates.
- ❖ Updates to the framework API are designed so that the new API remains compatible with earlier versions of the API.
- ❖ Most changes in the API are additive and introduce new or replacement functionality.
- ❖ As parts of the API are upgraded, the older replaced parts are deprecated but are not removed, so that existing applications can still use them.
- ❖ Hence, all other API parts from earlier revisions are carried forward without modification.
- ❖ An application running on API level 30 for instance can run on higher level APIs if an OTA update takes place.

Application Backward Compatibility

- ❖ Android applications are not necessarily backward compatible with versions of the Android platform older than the version against which they were compiled/tested.
- ❖ Each new version of the Android platform can include new framework APIs, such as those that give applications access to new platform capabilities/features.
- ❖ Earlier versions of the platform do not include the new APIs, applications that use the new APIs are unable to run on those platforms.
- ❖ An application running on API level 30 may have some features which are absent in API level 15. Thus this application can't be run on platforms with API level 15 or below.

Min, Target & Max SDK Version

- Applications can use a manifest element provided by the framework API `<uses-sdk>` to describe the minimum and maximum API Levels under which they are able to run, as well as the preferred API Level that they are designed to support. The element offers three key attributes:
 - ❑ **Minimum SDK Version** — Specifies the minimum API Level on which the application is able to run. The default value is "1".
 - ❑ **Target SDK Version** — Specifies the API Level on which the application is designed to run.
 - ❑ **Maximum SDK Version** — Specifies the maximum API Level on which the application is able to run. (Not used nowadays)

syntax:

```
<uses-sdk android:minSdkVersion="integer"  
          android:targetSdkVersion="integer"  
          android:maxSdkVersion="integer" />
```


SDK Version Relationship

- For instance, if **minSDK** = API 12 for an application then devices with less than API 12 won't be able to run the application. This minSDK is set to '1' by default so it is always necessary to declare it.
- Again if **maxSDK** = API 5 is declared for an app then devices with API > 5 won't be able to run the application. On the other hand, if a device with API 5 is running the said application but it receives an OTA (Over-the-Air) update and now is updated to API 6, then the system will remove the app from the device. Nowadays, these maxSDK are not declared since it potentially stops the applications from running.
- If **targetSDK** for an application is set to API 25 then a device with API 25 or lower can run the application. However, if the device receives an OTA update and is updated to API 26, even then this application can be run on the device. Thus targetSDK takes advantage of the forward compatibility.

Selecting SDK version/API Level

- You should declare your minSDK, targetSDK but not maxSDK.
- The minSDK can be determined through some key factors.
 - The lowest possible version of the platform that your application can support.
 - If you build an application that uses APIs or system features introduced in the latest platform version, you should set the minSDK to the API Level of the latest platform version.
- The targetSDK should be set to the highest possible version of the platform in order to get access of all the new features.
- The application should be tested/compiled against every API level from the minSDK to the targetSDK. But it is set to be compiled/tested against targetSDK by default.

minSDK (lowest possible) <= targetSDK(latest SDK) <= compileSDK(latest SDK)

THANK YOU