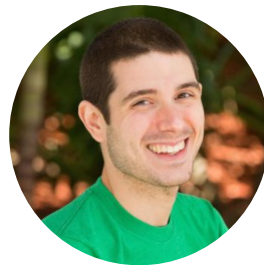


[Products ▾](#) [Customers](#) [Pricing](#) [Developers](#) [Support ▾](#) [Resources ▾](#)[Sign In](#)[XAMARIN BLOG](#) [All posts](#) [Developers](#) [Enterprise](#)

# Understanding Android's Doze Functionality

By James Montemagno • October 12, 2016 • Android, Xamarin Platform



The Android operating system enables developers to create complex applications that take advantage of background services, alarm scheduling, and data synchronization. With users downloading more apps than ever, it's important that these mass amounts of apps don't slow

down and drain the users batteries, which is why Google introduced **Doze** in Android Marshmallow and expanded it's reach in Android Nougat.

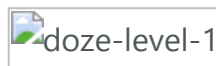
## When Devices Doze

Just like the name implies, Doze kicks in when a user's device is unplugged from a power supply and the handset isn't being used. Previously, Doze would only occur when the device was stationary for a set amount of time with the screen off, but with the release of Android 7.0, Doze has several states that can occur even if the device is not stationary.



### Doze Level 1

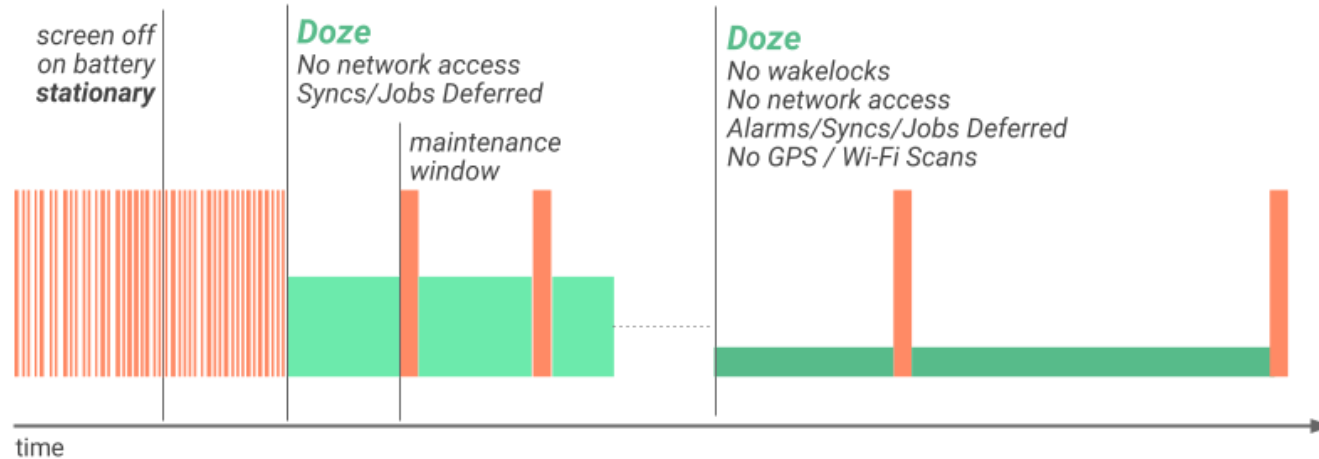
In Android 7.0, Doze can occur any time the device is on battery and the screen is off. After a certain amount of time, applications will cease to have network access and all syncs and background jobs will be deferred to a maintenance window that Android determines. Normal operation resumes as soon as the user plugs in their phone to charge or turns on the screen. Here's an illustration of Android 7.0's Doze Level 1:



### Doze Level 2

If the device remains stationary or is already in Level 1 Doze and becomes stationary, Level 2 will begin and start additional restrictions. Level 2 is the standard Doze from Android

Marshmallow, which ignores all wakelocks, defers all alarms, syncs, and jobs, and restricts network access, GPS, and Wi-Fi scans.



The system will wake and allow applications to perform normal operations at specific intervals in all levels of Doze. If your application is using any of these restricted APIs, it's important to realize that your app may not perform actions when you thought it would and that Doze should be tested for. The one important thing to remember with Doze is that there is one loophole when it comes to push notifications and Google Cloud Messaging—any push notifications marked "High Priority" will still come through when Doze is occurring.

## Testing for Doze

You can easily test for Doze mode to make sure your app is acting correctly when testing on a device or emulator running Android 6.0 (API Level 23) or higher. Simply run your application, leave it active, and shut off the device screen (ensuring the app remains active). You don't need

to wait for Doze to kick in; you can force Doze via the ADB command prompt with two simple commands:

```
$ adb shell dumpsys battery unplug  
$ adb shell dumpsys deviceidle step
```

You can run the second command multiple times to continue cycling through Doze modes to fully ensure your app is responding correctly when in and out of Doze mode.

## Critical Alarms

Many apps may be using the `AlarmManager` to schedule events such as calendar notifications. To help manage this, there have been two new methods added to the `AlarmManager` in API 23: [SetAndAllowWhileIdle](#) and [SetExactAndAllowWhileIdle](#). They are very similar, but the `SetExact` will ensure the alarm goes off at the exact time requested instead of a window. These alarms should be used sparingly and are still restricted to firing just once every 9 minutes per app.

## Take Advantage of Job Scheduler

The new Job Scheduler API introduced in Android 5.0, now part of Google Play services as `GcmNetworkManager`, enables developers to schedule background tasks and allow Android to correctly run them when it can batch multiple tasks together for efficiency and battery savings, making it great for one-off or periodic tasks. You can learn more about the Job Scheduler in [our samples](#) and in the [GcmNetworkManager blog](#) on data refresh.

## Learn more

Google has provided [great documentation](#) on how to fully optimize apps for Doze, including many of the tips I've provided here plus several more. Additionally, be sure to read about all of the API changes in Android N, including the [updates to Doze](#) and how it can effect apps.

Written on October 12, 2016 by  
James Montemagno



## Keep reading

[All Xamarin posts ▶](#)

### Xamarin University Webinar Recording | Building Your First Android App with Xamarin for Visual Studio



Courtney Witmer  
March 17, 2017

## Building Android Apps with Entity Framework



Jon Douglas

February 9, 2017

## Round Launcher Icons in Android 7.1



James Montemagno

January 30, 2017

Stay updated on the latest Xamarin announcements.

your email address

Subscribe

☐ I agree to the Terms of Use and Privacy Statement

## Products

[Xamarin for Visual Studio](#)[Xamarin Test Cloud](#)[Xamarin Insights](#)[Xamarin University](#)[Xamarin Studio](#)[Visual Studio](#)[Xamarin.Forms](#)[Pre-built apps](#)[Mono Licensing](#)

## Company

[About Us](#)[Customers](#)[Partners](#)[Blog](#)[Jobs](#)[Press](#)[Pricing](#)

## Developer Center

[Get Started](#)[Guides](#)[Recipes](#)[Xamarin.iOS APIs](#)[Xamarin.Android APIs](#)[Xamarin.Mac APIs](#)[Forums](#)[Components](#)[Videos](#)

## Connect

[Developer Support](#)[Resources](#)[Consulting](#)[Services](#)[Contact Sales](#)[Twitter](#)[GitHub](#)[Stack Overflow](#)[Facebook](#)[Google+](#)[YouTube](#)[+1 \(855\) 926-2746](#)[Terms of Use](#)

© 2017 Xamarin Inc.

Xamarin Inc., as a wholly-owned Microsoft subsidiary acting as a separate legal entity, adheres to the Microsoft Privacy Statement: [Privacy & cookies](#)