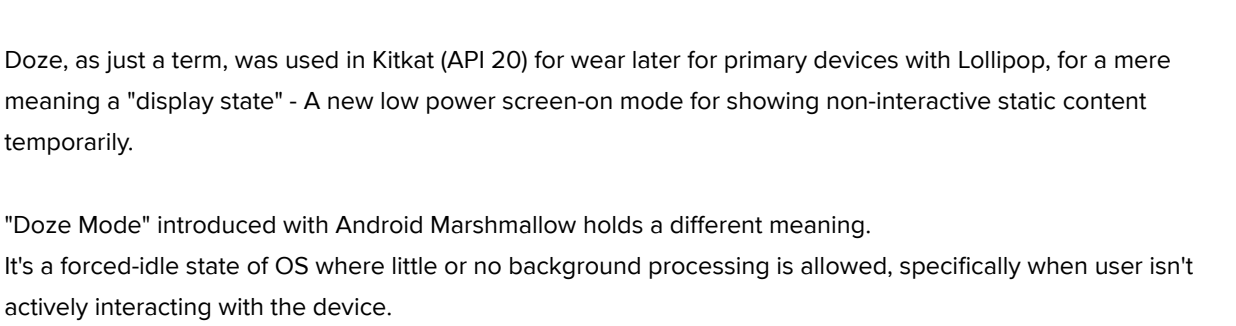
Tips and Tricks

[Tutorial] MIUI 8 (M) : Doze Explained & Answered

2016-08-06 07:21:13 9585 13

Edited by R0user at 2016-09-06 03:50 PM



Ever since the advent of MIUI over Marshmallow, there has always been Questions regarding the fate of most talked about feature of Marshmallow:
"Doze!"

- Is Doze stripped down from MIUI 8?
- Is it working down the hood?
- If not, when will it come?

with the advent of MIUI 8 (Marshmallow), It will rise only stronger.

Here we will try answering the same, for once & all, together.
Before we go & discuss about it's presence on MIUI 8 (M), let's dive into Doze First.

DOZE MODE

Doze, as just a term, was used in Kitkat (API 20) for wear later for primary devices with Lollipop, for a mere meaning a "display state" - A new low power screen-on mode for showing non-interactive static content temporarily.

"Doze Mode" introduced with Android Marshmallow holds a different meaning.
It's a forced-idle state of OS where little or no background processing is allowed, specifically when user isn't actively interacting with the device.

Q. We already have a Deep Sleep mode for our processor, how Doze differs?

- Deep Sleep is a state of Processor with very low power consumption, while Doze mode is a state, let's say, of an OS. The more time processor be in Deep sleep state while doing nothing, less power will be spent.

The system & so the apps in it, can hold the processor in the Active state to do something by the use of "wakelock". Even when the screen is off "Partial wakelock" can stop CPU going into the Deep Sleep state, so it might not even enter or for just a fraction of time; wasting energy. This is where Android 6.0 Marshmallow (API 23) came with the answer - Doze mode. It works as a guard in between the apps and cpu, by deferring Background CPU and Network Activity for apps when device is unused for longer periods of time & smartly batching the requests and processing together only at regular interval of time. All this, when it detects that you're not using the phone "actively" (by using the motion sensors)

Understanding DOZE

If a user leaves a device unplugged and stationary for a period of time, with the screen off, the device enters Doze mode. In Doze mode, the system attempts to conserve battery by restricting apps' access to network and CPU-intensive services. It also prevents apps from accessing the network and defers their jobs, syncs, and standard alarms.

// R0user
R0user
Mi Comm team

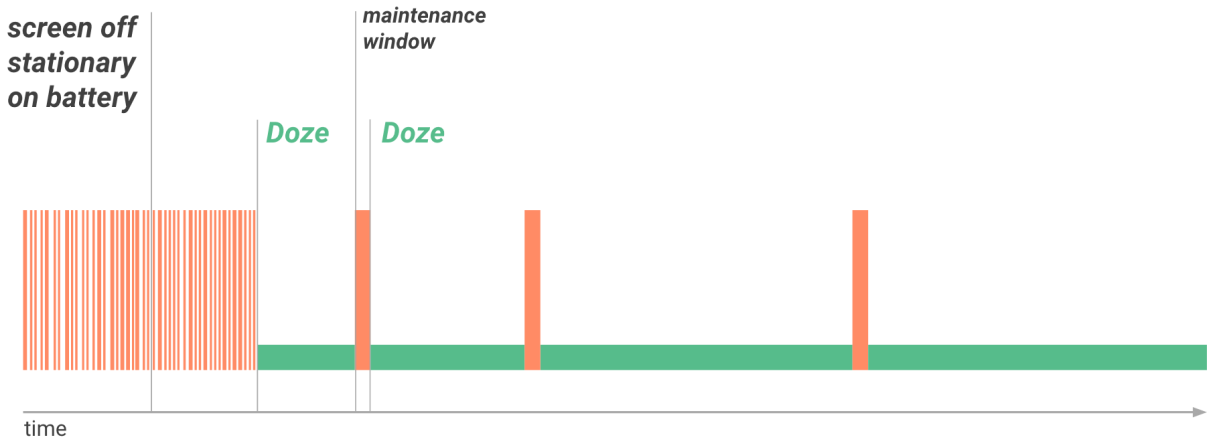
Followers	Threads	Replies	Points
2057	162	4142	

Follow Send PM

Reply Author

New Thread

Featured



At the conclusion of each maintenance window, the system again enters Doze, suspending network access and deferring jobs, syncs, and alarms. Over time, the system schedules maintenance windows less and less frequently, helping to reduce battery consumption in cases of longer-term inactivity when the device is not connected to a charger.

As soon as the user wakes the device by moving it, turning on the screen, or connecting a charger, the system exits Doze and all apps return to normal activity.

Doze Restrictions

The following restrictions apply to apps while in Doze:

- Network access is suspended.
- The system ignores wake locks.
- Standard AlarmManager alarms are deferred to the next maintenance window (special alarms fucntions for doze are there)
- Alarms set with setAlarmClock() continue to fire normally - the system exits Doze shortly before those alarms fire.
- The system does not perform Wi-Fi scans.
- The system does not allow sync adapters to run.
- The system does not allow JobScheduler to run.

Doze Requirements

Doze support requires the following:

- Device implements the significant motion detector (SMD) APIs in the Sensor HAL. Devices that do not implement these APIs cannot support Doze.
- Device has a cloud messaging service, such as Google Cloud Messaging (GCM). This enables the device to know when to wake from Doze.

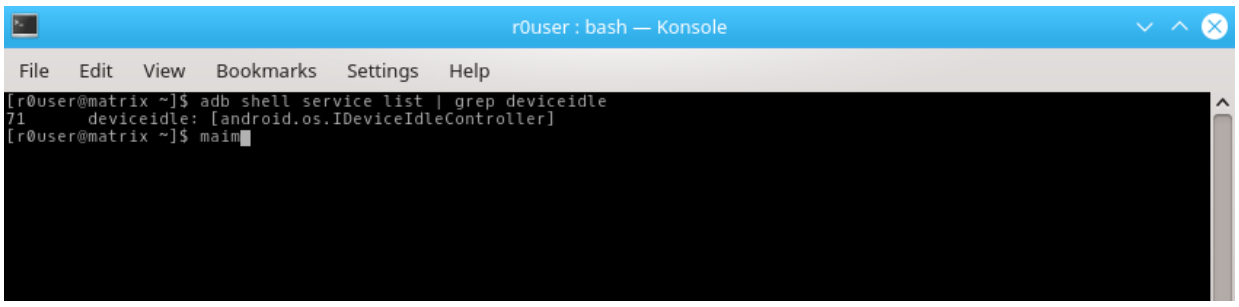
For MI3, testing shows this

```
  mEnabled=true
  mForceIdle=false
  mSigMotionSensor={Sensor name="Significant Motion Detector", vendor="Qualcomm", version=1, type=17, maxRange=1.0, re
solution=1.0, power=0.011, minDelay=-1}
  mCurDisplay=Display id 0: DisplayInfo{"Built-in Screen", uniqueId "local:0", app 1080 x 1920, real 1080 x 1920, larg
est app 1920 x 1860, smallest app 1080 x 1020, mode 1, defaultMode 1, modes [{id=1, width=1080, height=1920, fps=60.0}
], colorTransformId 1, defaultColorTransformId 1, supportedColorTransforms [{id=1, colorTransform=0}], rotation 0, den
sity 480 (449.704 x 447.412) dpi, layerStack 0, appVsyncOff 0, presDeadline 17666667, type BUILT_IN, state OFF, FLAG_S
ECURE, FLAG_SUPPORTS_PROTECTED_BUFFERS}, DisplayMetrics{density=3.0, width=1080, height=1920, scaledDensity=3.0, xdpi=
449.704, ydpi=447.412}, isValid=true
  mScreenOn=false
  mCharging=true
  mSigMotionActive=false
  mSensing=false mNotMoving=false
  mLocating=false mHaveGps=true mLocated=false
  mState=ACTIVE
  mInactiveTimeout=+30m0s0ms
```

"Deviceidlecontroller" - The Warrior Behind

DeviceIdleController is the primary driver of Doze - Let's refer it to **device idle mode** instead of doze mode since that better fits the code. Going through the official documentation gives the commands to manually enter into doze mode.

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

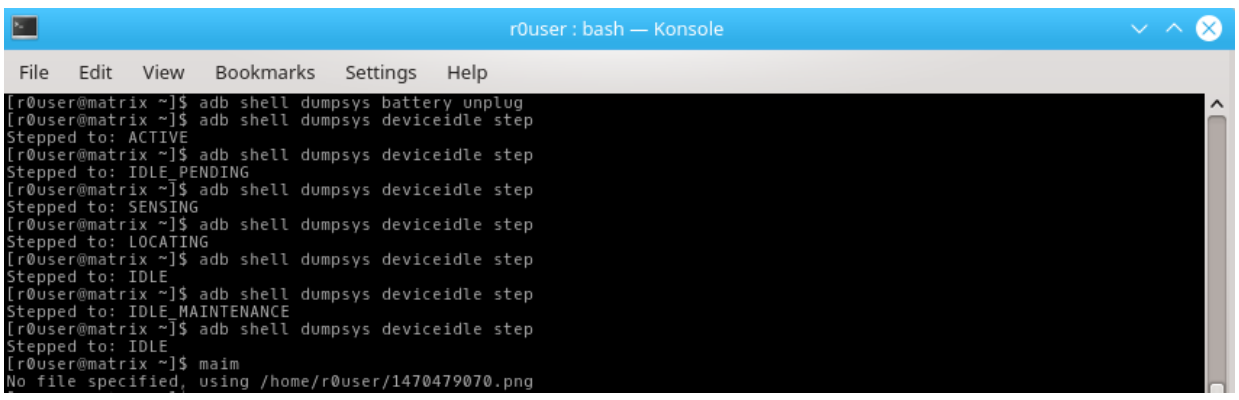


As you can see, the service actually exists.
(I'm assuming you already know how to be in debugging via ADB)

This new service is registered at all times to listen for the following system events, which can trigger it into (and out of) this new idle mode:

- Screen on/off
- Charging status
- Significant motion detect

DeviceIdleController maintains a state machine that contains different states. we can manually drive into each state like below & check the behaviour of system.

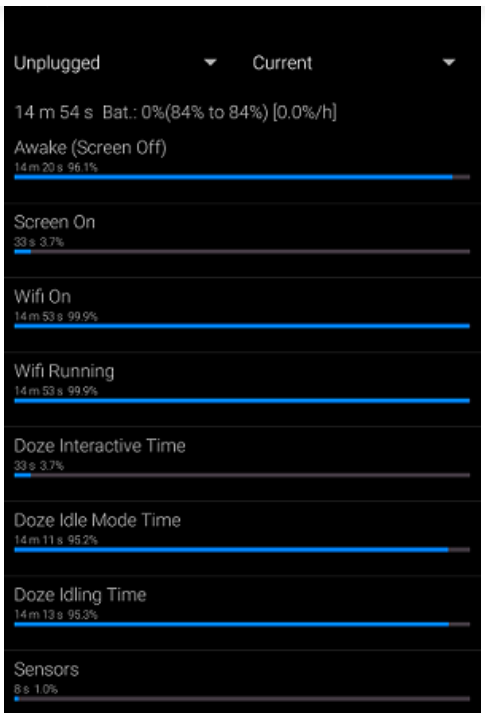


after entering into the **IDLE** , it wll only switch back & forth between **IDLE** & **IDLE_MAINTENANCE**
Press power button once & you will be stepped to **ACTIVE** state, whenever you want.

Note:
- Remember, we are manually driving into the doze(idle) mode.
- The device needs to be locked via power button to switch to different states, otherwise only "Active" you will see.
- After execution of "battery unplug" , "adb shell dumpsys battery reset" is necessary at the end of test or you will not see USB charging.
- You can also use "adb shell dumpsys deviceidle force-idle" too skipping both the commands directly.

Manual Testing

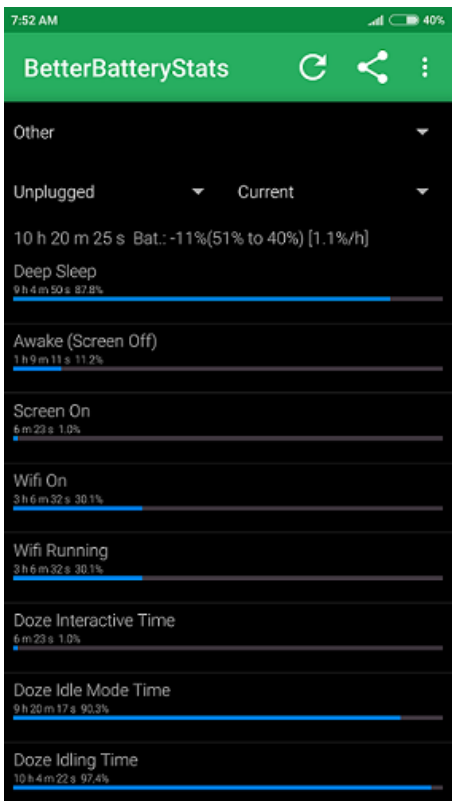
Let's leave it to the IDLE state for about 15 Minutes and check for BBS graphs.

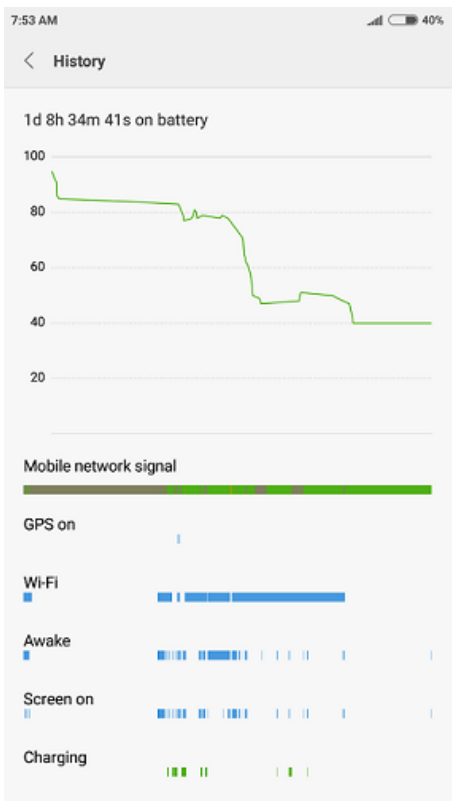


Yes, surely It works manually.

MIUI 8 (M) & Doze In Real World Scenarios

In Real world scenarios, Doze doesn't fire that fast, it fires only after predefined time (after above mentioned 3 conditions are met)
Leaving the phone on baterry, unplugged & untouched - Here are the results.





Conclusion

MIUI 8 (M) users, Yes DOZE will work in the background (as long as your device meets the requirements)
For MI3W, yes Doze is definitely working in the background of MIUI 8.

Signing off,
R0user

Rate

Number of participants 6 **Experience +42** Reason Pack

View Rating Log

2016-08-06 07:21:13 Reply

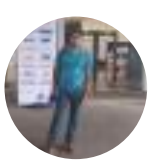
MIUI beta tester

lavnidhi | #1

Wow finally we can see the most talked feature of Android M working on MIUI. Thanks for the share bro.

2016-08-06 07:33:11 Reply Report Rate

Proud Mi Fan



Advanced Bunny

Dr.Chinmay | #2

Very good thread, thanks for explaining in detail about doze feature, am sure many of us have got clear idea about doze feature of miui8 now

2016-08-06 07:44:01 Reply Report Rate

MIUI beta tester

luvxiaomi | #3

One of the best written thread. You explained everything in detail. Kudos for writing such original thread.

2016-08-06 07:54:48 Reply Report Rate

Rookie Bunny

322988076 | from mobile #4

For those wondering. Root or not (some work for non root but very easy) . Simply download ForceDoze on play store and your battery will thank you.

2016-08-06 08:15:58 Reply Report Rate

MIUI beta tester

Er. Deep | #5

Wow.. very good work..
No one goes this deep.. Its been a while I see a post without copy pasting ss & details from Internet. You explained with actual examples.
Keep up ur good work R0user

2016-08-07 03:07:24 Reply Report Rate

Harish Mohanty | #6

MIUI beta tester

Really very useful for educational purposes also it helps to know more about Android system... Thanks for sharing this awesome article

2016-08-07 03:45:21 Reply Report Rate

MI FC BHUBANESWAR PRESIDENT

Apurv Gupta | #7

Advanced Bunny

Thanks for this advanced info. Great indeed !

2016-08-27 04:25:52 Reply Report Rate

Samuel Thomas | from Redmi Note 3 #8

Rookie Bunny

is it compatible wid Redmi note 3 running Android 6.0 on top of MIUI 8

2016-12-19 02:48:55 Reply Report Rate

R0user Author | #9

Mi Comm team

Samuel Thomas replied at 2016-12-19 01:18 PM
is it compatible wid Redmi note 3 running Android 6.0 on top of MIUI 8

Yes, have checked it. It is there

2016-12-19 05:17:41 Reply Report Rate

New to MIUI, Start Here | Rookie Bunny, Start Here

Rookie
Bunny

R0user

Yes, have checked it. It is there

But the thing is, in nighttime upto 10% battery juice lost....

2016-12-19 06:21:51 Reply Report Rate

Next >

1 2 Go to page 1