Doze & App Standby

Android M Battery Life Improvements

Contents

Overview

Battery life is a perennial user concern. To extend battery life, Android continually adds new features and optimizations to help the platform optimize the off-charger behavior of applications and devices.

The Android M release includes the following new improvements to battery life:

- **App Standby**. The platform can now place unused applications in App Standby mode, temporarily restricting network access and deferring syncs and jobs for those applications. Users can exempt applications using a whitelist UI in Battery Settings.
- **Doze**. The platform now enters a state of deep sleep (periodically resuming normal operations) if users have not actively used their device (screen off and stationary) for extended periods of time. As this feature requires the platform to detect the stationary state, it is available only on devices that implement the significant motion detection APIs in the Sensor HAL. Doze dramatically improves battery life; however, users can exempt applications using a whitelist UI in Battery Settings.
- **Exemptions**. System apps and cloud messaging services preloaded on a device are typically exempted by default. Users can additionally whitelist applications from App Standby and Doze, while app developers can intent their applications into this setting. For devices with GMS, Google Cloud Messaging (GCM) supports a high-priority tickle that temporarily adds the receiving app to the whitelist.

These improvements are detailed in the following sections.

Inactive applications (App Standby)

App Standby can extend battery life by deferring background network activity and jobs for applications the user is not actively using.

App Standby lifecycle

The platform detects inactive applications and places them in App Standby until the user actively engages with app.

Detection	During App Standby	Exit
The platform detects an application is inactive when the device is not charging ***and*** the user has not	The platform prevents applications from accessing the	The platform exits the app from App Standby when: • Application

launched the application directly or indirectly for a specific amount of clock time as well as a specific amount of screen-on time. (Indirect launches occur when a foreground app accesses a service in the app.)	network more than once a day, deferring application syncs and other jobs.	become active. • Device is plugged in and charging.

Active applications are unaffected by App Standby. An application is active when it has:

- A process currently in the foreground (either as an activity or foreground service, or in use by another activity or foreground service), such as notification listener, accessibility services, live wallpaper, etc.
- A notification viewed by the user, such as in the lock screen or notification tray.
- Explicitly been launched by the user.

An application is inactive if none of the above activities have occurred for a period of time.

Note: A high-priority GCM tickle temporarily adds the app to the whitelist, granting the app network access for a brief period of time. For details, see Using high-priority GCM tickles.

Testing App Standby

You can manually test App Standby using the following ADB commands:

```
$ adb shell dumpsys battery unplug
$ adb shell am set-idle packageName true
$ adb shell am set-idle packageName false
$ adb shell am get-idle packageName
```

Idle devices (Doze)

Doze can extend battery life by deferring application background CPU and network activity when a device is unused for long periods.

Idle devices in Doze periodically enter a maintenance window, during which apps can complete pending activities (syncs, jobs, etc). Doze then resumes sleep for a longer period of time, followed by another maintenance window. The platform continues the Doze

sleep/maintenance sequence, increasing the length of idle each time, until a maximum of a few hours of sleep time is reached. At all times, a device in Doze continually looks for motion and immediately leaves Doze if detected.

System services (such as telephony) and other preloaded services/apps are exempted from Doze by default. Users can also exempt specific applications from Doze in the Settings UI.

By default, Doze is **disabled in AOSP** (for details on enabling Doze, see Integrating Doze). However, GMS devices have Doze enabled by default.

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). Enables the device to know when to wake from Doze.

We strongly recommend enabling Doze and using GCM 3.0, which has many improvements to systematically improve battery life. Unlike previous versions of GCM that suffered from fragmentation, GCM 3.0 is included in GMS, which updates frequently.

Doze lifecycle

The Doze lifecycle begins when the platform detects that the device is idle and ends when the device exits Doze mode.

Detection	During Doze	Exit
The platform detects a device is idle when: • Device is stationary (using significant motion detector). • Device screen is off for some amount of time. Doze mode does not engage when plugged into a power charger.	The platform attempts to keep the system in a sleep state, periodically resuming normal operations during a maintenance window then returning the device to sleep for longer repeating periods. During the sleep state, the following restrictions are active: • Apps not allowed network access. • App wakelocks ignored. • Alarms deferred. Excludes alarm clock alarms and alarms set using	The platform exits the device from Doze when it detects: • User interaction with device. • Device movement. • Device screen turns on. Notifications do

	setAndAllowWhileIdle(). This exemption is intended for apps (such as Calendar) that must show event reminder notifications. • Wifi scans not performed. • SyncAdapter syncs and JobScheduler jobs deferred until the next maintenance window. • Apps receiving SMS and MMS messages are put on the temporary whitelist so they can complete their processing.	not exit the device from Doze.

Note: A high-priority GCM tickle temporarily adds the app to the whitelist, granting the app network access and wakelock for a brief period of time. For details, see Using high-priority GCM tickles.

Interaction with App Standby

- Time spent in Doze does not count towards App Standby.
- While the device is in Doze, idle applications are allowed to perform normal operations at least once a day.

Integrating Doze

To enable Doze for a device, perform the following tasks:

1. Confirm the device supports SENSOR_TYPE_SIGNIFICANT_MOTION. If the device does not support this sensor, it cannot support Doze.
2. Confirm the device has a cloud messaging service installed.
3. [Non-GMS devices only[1]] In the device overlay config file (overlay/frameworks/base/core/res/res/values/config.xml), set `config_enableAutoPowerModes` to **true**:

```
<bool name="config_enableAutoPowerModes">true</bool>
```

[1] In AOSP, this parameter is set to **false** (Doze disabled) by default. In GMS, this parameter is set to **true** (Doze enabled) and does not need to be changed.

4. Confirm that preloaded apps and services:
    - Use the new power-saving optimization guidelines. For details, see Testing and optimizing applications.

      OR

    - Are exempted from Doze and App Standby. For details, see Exempting applications.
5. [GMS devices only] Confirm the necessary services are exempted from Doze. For details, see Confirming system-privileged/exempted services.

Testing and optimizing applications

Test all applications (especially preloaded applications) in Doze mode. For details, refer to Testing Doze and App Standby.

Note: MMS/SMS/Telephony services function independently of Doze and will always wake client apps even while the device remains in Doze mode.

For apps and services that must be woken during Doze mode, consider using Google Cloud Messaging (GCM) high-priority messages. For details, see Using high-priority GCM tickles.

Confirming system-privileged/exempted services

For GMS devices, confirm that Play Services (GmsCore.apk) is included in application exemptions. Play Services is a system-privileged app that must wake up other 3rd-party apps when a high-priority message arrives for device in Doze mode. For correct Doze operation, confirm the following:

- The GmsCore.apk file is located in `/system/priv-app`.
- The `device system config` includes the following lines:

```
<!-- GmsCore must always have network access for GCM and
other things. -->

<allow-in-power-save package="com.google.android.gms" />
```

  These lines should be present in the config file if you followed the GMS integration guide and added the gms.mk line in the device makefile.

Testing Doze

GTS verifies that GMS devices with a Significant Motion Detector (SMD) have Doze enabled. You can manually test Doze by turning off the device screen and running the following commands:

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

To ensure consistent behavior, set up two (2) test scenarios for each preloaded app, using one scenario with Doze enabled and the other with Doze disabled. If you encounter issues with existing apps after enabling Doze, work with your Google Technical Account Manager (TAM) to determine the root cause—do not immediately request to whitelist the app.

Exempting applications

You can whitelist an application to exempt it from being subject to Doze or App Standby.

Warning: Do not whitelist apps to avoid testing and optimizing. Whitelisting undermines the benefits of Doze and App Standby and can compromise the user experience, so we strongly suggest minimizing whitelisting as it allows applications to defeat beneficial controls the platform has over power use.

Apps exempted by default are listed in a single whitelist, which is used for exempting the app from both Doze and App Standby modes. To provide transparency to the user, the settings UI **MUST** show all exempted applications.

Users can manually exempt apps (add to the whitelist) using the settings UI. However, users cannot unexempt (remove from the whitelist) any application or service that is whitelisted by default in the system image. If a user becomes unhappy about the power consumption of these apps, it can lead to frustration, bad user experiences (and negative user reviews for the app), and customer support questions. For this reason, we strongly recommend that you do not whitelist 3rd-party applications and instead whitelist only cloud messaging services or apps with similar function.

Alternatives to whitelisting

Before whitelisting an application, consider the following options:

- (Option 1) Migrate applications to use GCM on GMS devices. This reduces the number of persistent connections kept alive during Doze and thus optimizes power consumption. GCM 3.0 has many improvements to systematically improve battery life; for details, see the Google I/O 2015 GCM 3.0 talk and Cloud Messaging on developer.android.com.
- (Option 2) Migrate the messaging platform to tunnel through GCM. This hybrid approach minimizes migration effort across a large number of apps and maintains current custom solution infrastructure.
- (Option 3) Use an alternative Cloud Messaging system that adopts the battery saving concepts of Doze and App Standby, (i.e. wake devices only for high priority notifications and avoid battery-draining syncs).

GMS approval

Google may block GMS approval if a 3rd-party app available for download from Google Play Store or any 3rd-party application distribution channel is exempted by default. Such exemptions increase risk to the user experience due to shorter battery standby time and undermine new Android features.

However, Android is an open platform and we do not intend to block GMS approval because of other cloud messaging services or app distribution channels being exempted from Doze and App Standby. Instead, we will work with you to mitigate battery consumption impact by having more apps/services whitelisted.

Using high-priority GCM tickles

GCM 3.0 supports a high-priority tickle that temporarily adds the receiving application to the whitelist and enables intents to trigger an immediate user action (e.g. an incoming IP video/voice call or instant message). Use the high-priority tickle only for important user notifications (those that must be seen right away); do not use for notifications that can wait until the device wake (e.g. Status updates, Network Configurations).

To display a notification as a result of a tickle, include the contents of the notification in the payload of a GCM tickle such that subsequent network access is not required to generate the notification.

Example:

```
https://gcm-http.googleapis.com/gcm/send
  Content-Type:application/json
  Authorization:key=AIzaSyZ-1u...0GBYzPu7Udno5aA
{
    "to": "/topics/foo-bar",
    "data": {
      "message": "This is a GCM Topic Message!",
     },
    "priority":"10"
            }
```

For more parameters, refer to the GCM Connection Server Reference.

Note: GCM server side support is present now, but platform client side support is not yet complete (coming soon).

Using push notifications

If an app maintains a direct channel to a 3rd party app server, that channel is disabled when the device enters Doze or App Standby.

To prevent this, you can wake the app using GCM while the app maintains a direct channel to the 3rd party app server.

When using GCM, the app is disabled when the device enters Doze or App Standby mode. However, it can send notifications through GCM to wake the app client and re-establish the direct channel.