AMERICAS APPFORUM 2019 | ELEVATING ENTERPRISE INTELLIGENCE

ZEBRA CAPTURE YOUR EDGE

Android Oreo and Pie Features for your Enterprise app

Darryn Campbell

SW Architect, Zebra Technologies
@darryncampbell
October 1st / 2nd 2019

Latest Android Oreo & Pie features for your Enterprise Application

Agenda

- Overview
- Android Oreo features
- Android Pie features

What’s new for Zebra Developers in Android Pie

Trends over time

						android
Running in the background	Job Scheduler	Doze mode	Doze “on the go”	Background restrictions	Machine learning for intelligent restrictions	Restricted access to location in the background
Notifications	Quick settings & notification shade	Long press to access options	Direct reply & bundled notifications	Notification channels & snooze	Enhanced messaging experience	Notification bubbles (native support)
One or Two other major changes affecting Enterprise	Material design	Runtime permissions	Multi-window	Changes to the Google Play Store policies	Non-SDK methods actively discouraged	External storage changes
Android Enterprise features	Android for Work, app restrictions	DO mode, lock task mode, managed configs	DPM API enhancements	DPM API enhancements	DPM API enhancements	Full transition to DO mode

Latest Android Oreo & Pie features for your Enterprise Application

Changes to the Google Play Store

In 2018 Google introduced a requirement that applications being posted to the Play Store must target a recent Android API level. These requirements have been updated in 2019 and detailed in the associated Android blog post, but the timeline is as follows:

- **August 2019**: New applications added to the Play store are required to target API level 28 (Android 9) or higher
- **November 2019**: Updates to existing applications are required to target API level 28 or higher.
- You can still assign any minimum SDK level since this restriction only affects the target SDK. Google provide additional documentation for developers who are updating their target SDK level.

Latest Android Oreo & Pie features for your Enterprise Application

Changes to the Google Play Store

- This will affect Enterprise applications *:
  - Managed Android devices will typically use the Managed Play store which is subject to the same new rules.
  - Updating applications will require increasing the target SDK level and considering any restrictions introduced in the newly supported level(s)
  - Targeting a lower SDK level to circumvent Android restrictions will no longer work for applications hosted in the Play Store. This has been a popular technique with consumer apps to avoid Marshmallow runtime permissions and Oreo background restrictions.
  - More robust workarounds are given in Zebra developer documentation.
- Existing applications that do not get updated will be allowed to stay in the Play store
- Application deployment that does not depend on the Play store may be affected (according to the official Android blog) – Zebra are working with Google on this.

* Applications distributed ONLY as private apps remain unaffected

Latest Android Oreo Features

Latest Android Oreo & Pie features for your Enterprise Application

Oreo Background Limitations

- Oreo is introducing restrictions on what an application can do in the background
- Three main types of restriction:
  - Receiving implicit broadcast intents declared in the manifest
  - Running services in the background
  - Update frequency from location APIs

  $\left.\begin{array}{l}\\ \\ \end{array}\right\}$ Google conflates these two
- Continues the trend of restricting what an app can do in the background.
  - Trend continues into P which will limit access to user input & sensor data
- Developers are advised to work with the changes where possible. **Where not possible, make us aware**.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Receiving implicit broadcast intents declared in the manifest

The limitation:

- Applications cannot receive implicit broadcast intents which they have declared in their manifest

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Receiving implicit broadcast intents declared in the manifest

- What are implicit intents?
  - An implicit intent is an intent which lacks a package or component class name

Implicit intent	Explicit intent:

`Uri uri = Uri.parse(“geo:0,0?q=London”);`
`Intent intent = new Intent(Intent.ACTION_VIEW, uri);`
	`Uri uri = Uri.parse(“geo:0,0?q=London”);`
	`Intent intent = new Intent(Intent.ACTION_VIEW, uri);`
	`intent.setPackage(“com.google.android.apps.maps”);`

- What are broadcast intents?
  - An broadcast intent is received by a broadcast receiver and sent via the sendBroadcast() API
- Can I have an explicit broadcast intent? Isn’t that a contradiction?
  - You **CAN** have an explicit broadcast intent, it will only be received by the destination component.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Receiving implicit broadcast intents declared in the manifest

- What does “declared in the manifest” mean?
  - You can register your broadcast receiver dynamically at runtime or in the manifest at build time

Manifest

```
<receiver android:name=".WifiReceiver" >
   <intent-filter>
     <action 
android:name="android.net.wifi.WIFI_STATE_CHANGED" 
/>
   </intent-filter>
</receiver>
```

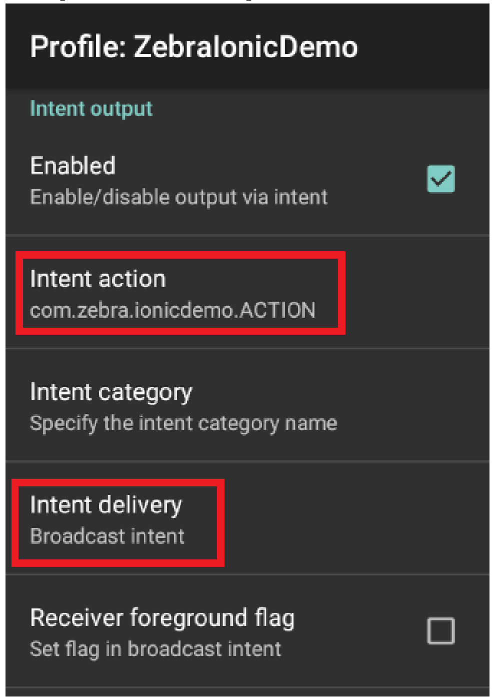
Dynamic registration:

```
BroadcastReceiver br = new MyBroadcastReceiver();
IntentFilter filter = new IntentFilter();
Filter.addAction(“android.net.wifi.WIFI_STATE_CHANG
ED”);
registerReceiver(br, filter);
//  Take care, if your application is in the 
background it is subject to being killed by the 
system
```

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Receiving implicit broadcast intents declared in the manifest

- Enterprise implications:



Profile: ZebraIonicDemo

Intent output

Enabled
Enable/disable output via intent ✓

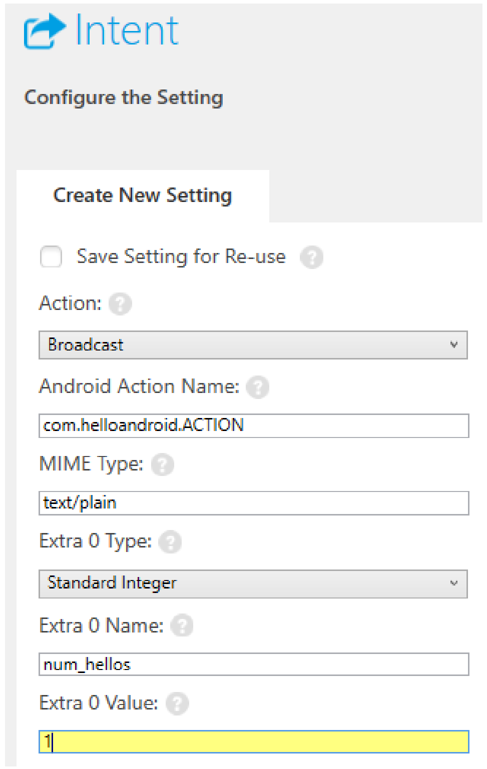
Intent action
com.zebra.ionicdemo.ACTION

Intent category
Specify the intent category name

Intent delivery
Broadcast intent

Receiver foreground flag
Set flag in broadcast intent ☐

DataWedge can only send implicit intents



Intent

Configure the Setting

Create New Setting

☐ Save Setting for Re-use

Action: Broadcast

Android Action Name: com.helloandroid.ACTION

MIME Type: text/plain

Extra 0 Type: Standard Integer

Extra 0 Name: num_hellos

Extra 0 Value: 1

Intent CSP can only send implicit intents

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Receiving implicit broadcast intents declared in the manifest

- Mitigation:
  1. Use a dynamic broadcast receiver
  2. Switch to an explicit intent (if you have control over the sender)
  3. Continue to target your application at API level 25 or lower
  4. Per Google, “Use a scheduled job to check for the condition that would have triggered the implicit broadcast”

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Running services in the background

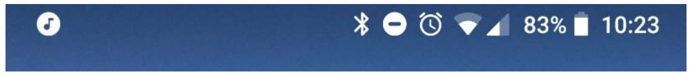
The limitation:

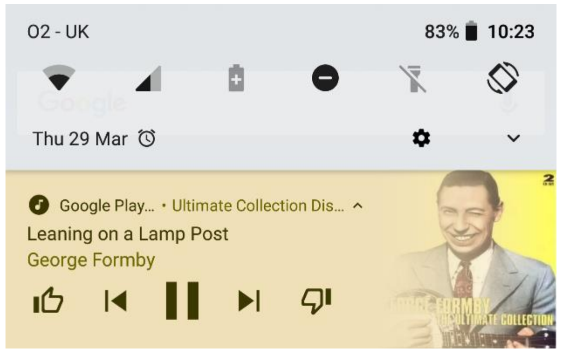
- Background applications built with API level 26 or higher and Android services associated with those background applications are subject to several limitations under Oreo to improve battery life and RAM usage. This includes IntentServices and PendingIntent services running in the background. Foreground applications are unaffected by these restrictions.
- “When the app goes into the background, it has a window of ***several minutes*** [emphasis added] in which it is still allowed to create and use services. At the end of that window, the app is considered to be idle. At this time, the system stops the app’s background services, just as if the app had called the services’ Service.stopSelf() methods.”

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Running services in the background

- What is a service?
  - From the docs: “A service is an Android application component that can perform long-running operations in the background, and it doesn’t provide a UI.” (think that definition might need updating!)
- Foreground service? Background service?
  - Most common use of a foreground service is the music player or GPS directions. A persistent notification is shown to the user, possibly with a rich UI.



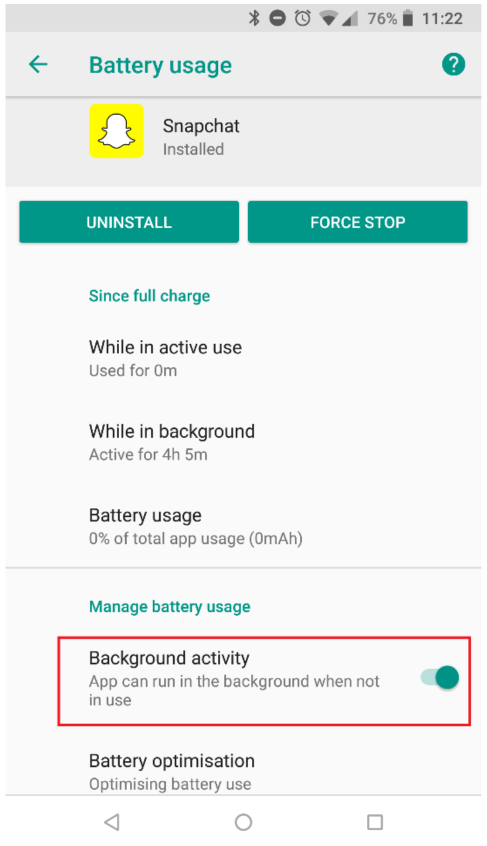


Music player notification before and after expansion

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Running services in the background

- “Built with 26 or higher”… so I can just target my application to API 25 and call it a day?
  - Not quite, apps targeting API 25 or lower and using a background service on a real device will present an option to the user for “Background activity” (*see screenshot, right*)
  - Setting is located under the battery options, under app info.
  - Default is ‘enabled’, i.e. not subject to Oreo background restrictions but the user can ‘disable’ it and the app WILL be subject to background restrictions.



Snapchat
Installed

Since full charge

While in active use
Used for 0m

While in background
Active for 4h 5m

Battery usage
0% of total app usage (0mAh)

Manage battery usage

Background activity
App can run in the background when not in use

Battery optimisation
Optimising battery use

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Running services in the background

More about the limitation:

- A system whitelist exists whereby applications are permitted to run and start services without limitation. Applications will be temporarily added to the whitelist “for several minutes” in order to handle common background tasks, such as:
  - Handling a high-priority Firebase Cloud Messaging (FCM) message.
  - Receiving a broadcast SMS or MMS message.
  - Executing a PendingIntent from a notification.
- This list is not exhaustive, Google qualifies the list as “when [an app] handles a task that is visible to the user,” so there is room for expansion in future versions.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Running services in the background

- Whitelist? Like Doze mode right? So, we have MX APIs to turn that off?
  - No, although the terminology is the same **it is not the same whitelist as used by Doze mode** and cannot be controlled by MX.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Running services in the background

Enterprise implications:

- Many of the same enterprise apps affected by doze mode will also be affected by Oreo’s limitation on background processes:
  - Running an on-premises push messaging system (not relying on FCM)
  - Downloading or uploading large files
  - Running a continual background service to check for network traffic
- Zebra value-added applications are also affected:
  - Datawedge’s ability to call startService is curtailed since services cannot be started in the background
  - EHS’ feature to launch a list of specified services is similarly curtailed
  - Both DataWedge and EHS have added new methods to deal with the new Oreo behaviour

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Running services in the background

- Mitigation:
  1. Use the Android ~~JobScheduler~~ WorkManager API to schedule a job to perform the task
  2. Make use of the temporary whitelist
     - Especially if the app is already using FCM or running a background task after receiving a PendingIntent (e.g. in response to a notification)
  3. Use a foreground service
  4. Per Google “Defer the background work until the application is naturally in the foreground”
  5. Continue to target your application with API level 25 or below
     - Not a long term solution. You will quickly run afoul of the new Play Store rules to force applications to move to “recent API levels”.
     - Can also be circumvented by the user if they have access to the application battery settings.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Update frequency from location APIs

The limitation:

- Any background application or service making use of Android location APIs will only receive location updates “a few times each hour.”
- Unlike the previous two noted limitations, this limitation exists for any application running on an Oreo device regardless of target API level, and is not affected by the Background activity option.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Update frequency from location APIs

There is not a single location API family on Android, these are the **affected** APIs:

- Fused Location Provider Client which replaced the Fused Location Provider API. Both are affected.
- Location Manager, any available Google location API to which you can ask 'where am I' is subject to this limitation.
- Wi-Fi Manager startScan will only perform a full scan 'a few times each hour', this prevents an application using Google's Geolocation REST API to return the position based on nearby Aps
  - Note also that in 'P' this method is marked as deprecated.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Update frequency from location APIs

There is not a single location API family on Android, these are the **unaffected** APIs:

- The batched version of the fused location provider.
- Geofencing. Used to determine if a device has entered or left an area.
- ActivityRecognitionClient which replaces the earlier ActivityRecognitionApi can be used to determine if the user is performing various activities such as walking or driving.
- Any indoor-based APIs, including 3rd party APIs that rely on BLE or other hardware dependant technologies such as the magnetometer.
- APIs for Zebra technologies such as RFID or Worry Free Wi-Fi.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Update frequency from location APIs

Google's intention by implementing these restrictions is to lock down the APIs to a specific set of use cases:

- Turn by turn directions in the foreground app, bread-crumbing to determine historical location and geofencing.

Enterprise implications:

- Real time route planning for T&L use cases, updating existing routes based on the real-time outdoor position of the device.
- "Find my device" to locate a device in real time outdoors, e.g. to gather devices in preparation for the next shift or locate devices requiring battery replacement. Requires the device to maintain its location.
- Device tracking to provide historical location data. Although batched location is still available through the FLP this use case may have been previously met using the standard API, so it might require a code change in the application.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Update frequency from location APIs

Mitigation:

1. **Foreground service**: Background location limits do not apply to applications with a foreground service or are themselves in the foreground. E.g. turn-by-turn directions.
2. **Passive listener**: If a different application in the foreground is requesting location updates, a background application can “piggy-back” on the request to receive updates at a faster rate, as if they too were in the foreground.
   - E.g. real-time route planning might piggy-back on the navigation app
3. **Bread-crumbing**: If your use cases revolve solely around logging the historical location of devices, consider using the batched location functionality of the Fused Location Provider.
4. **Geofencing**: If your use cases revolve solely around detecting whether a device is inside or outside of a specified area(s), a retail store, for example, consider using a Geofence.

Latest Android Oreo & Pie features for your Enterprise Application

Oreo: Background limitations summary

	Receiving Implicit broadcast intents	Running services in the background	Update frequency from Location APIs
Description	Implicit broadcast intents can no longer be registered for from the application manifest.	An application cannot run or have running services while in the background.	Many location APIs restrict the location update frequency to 3 or 4 times per hour.
Notable affected enterprise use cases	Relies on DataWedge intents or MX Intent capability	Non-FCM push based solutions, network monitoring, long running downloads or uploads	Functionality dependent on real-time location tracking (e.g. “find my device”)
Affects APIs targeting less than 26?	No, unless user manually re-enables it from the settings UI	No, unless user manually re-enables it from the settings UI	Yes
Recommended Mitigation	Rework app to use dynamic broadcast receivers in code. Target an API level less than 26 as an interim solution.	Use a ~~Job scheduling~~ Work Manager API or a foreground service. Target an API level less than 26 as an interim solution.	Use a foreground service or a geofence / batched location (depending on use case)

Latest Android Pie Features

What’s new for Zebra Developers in Android P

Power Management changes: App standby buckets & adaptive battery

- Enhancement to the existing ‘App Standby’ feature first introduced in Marshmallow
- Behaviour of the application will change depending on how the Operating System categorizes that app. Apps are categorized into buckets
- An application may change buckets
  - No API exists to request the app be in a particular bucket
  - No API exists to determine when the app changes buckets
  - An API does exist to obtain a snapshot of which bucket the app is in
- Best advice is to work with the bucket-based restrictions rather than trying to influence them
- Official documentation for bucket based restrictions is available.

What’s new for Zebra Developers in Android P

Power Management changes: App standby buckets & adaptive battery

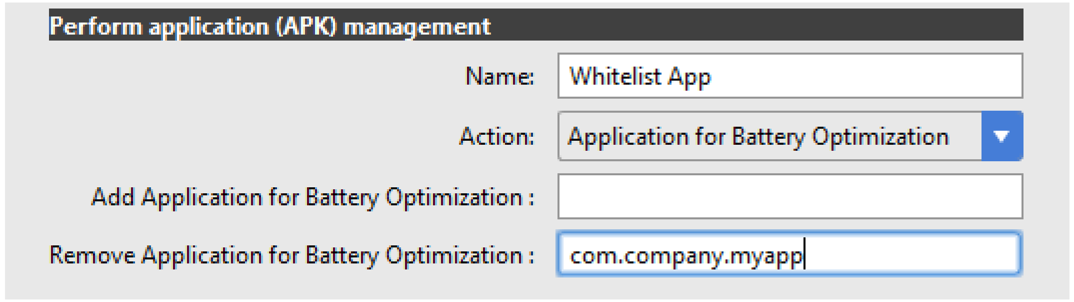
Bucket	Description
Active	User is currently engaged with the application and Android considers it to be in the foreground **Restrictions:** none
Working set	Application is not currently active but runs often. **Restrictions:** Jobs and Alarms will be deferred. No restrictions on network access or Firebase Cloud Messaging.
Frequent	Application is used regularly but not necessarily every day. **Restrictions:** Jobs and Alarms will be deferred for longer than applications in the working set. No restrictions on network access but Firebase Cloud Messaging is limited to 10 high priority messages a day.
Rare	Application is not often used. **Restrictions:** Jobs and Alarms will be deferred for longer than applications in the frequent set. Network access is deferred and Firebase Cloud Messaging is limited to 5 high priority messages a day.
Never	Application has been installed but never run. “The system imposes **severe restrictions** on these apps.”

What's new for Zebra Developers in Android P

Power Management changes: App standby buckets & adaptive battery

Applications which are on the doze whitelist are exempted from bucket-based restrictions

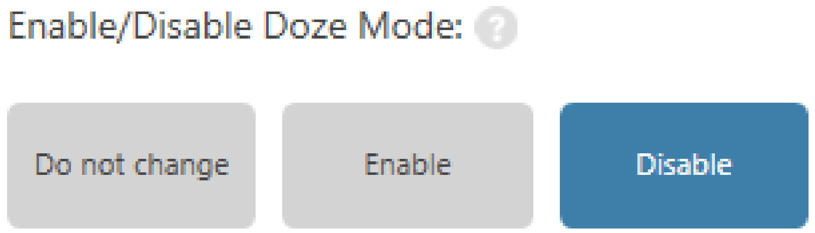
- Zebra have a couple of administrator features to whitelist an application:
  - Whitelist a particular app with the App Manager



Perform application (APK) management

Name:	Whitelist App
Action:	Application for Battery Optimization
Add Application for Battery Optimization :	
Remove Application for Battery Optimization :	com.company.myapp

  - Disable Doze Mode entirely on the device with the Power Manager



Enable/Disable Doze Mode: Do not change Enable [✓ Disable]

Remember: Whitelisting an app ***may*** *increase your device battery consumption*

What's new for Zebra Developers in Android P

Power Management changes: App standby buckets & adaptive battery

- Testing: ADB commands exist for controlling and checking an app bucket

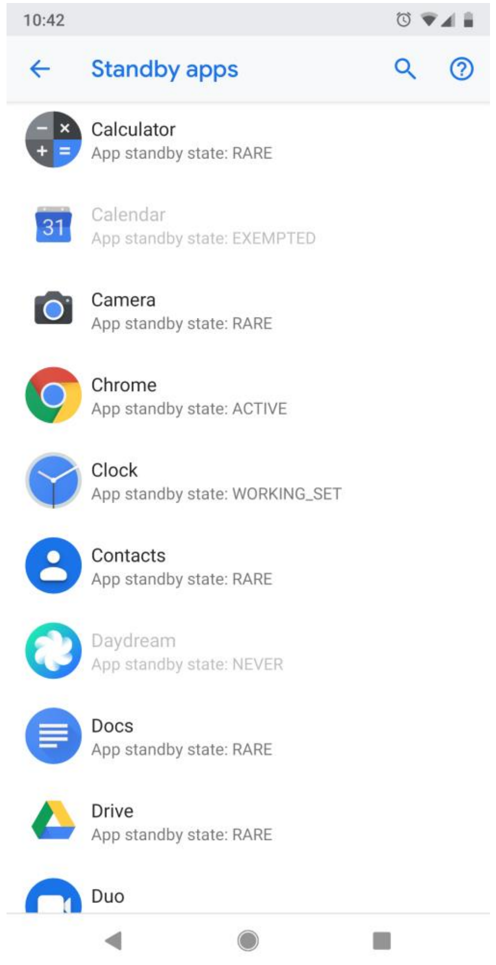
Manually assign an app to a bucket

```
$ adb shell am set-standby-bucket packagename active |
working_set | frequent | rare
```

Check what bucket an app is in

```
$ adb shell am get-standby-bucket packagename
```

- You can see the bucket for every app from 'Standby apps' configuration under 'Developer Options' (Right)



Calculator
App standby state: RARE
Calendar
App standby state: EXEMPTED
Camera
App standby state: RARE
Chrome
App standby state: ACTIVE
Clock
App standby state: WORKING_SET
Contacts
App standby state: RARE
Daydream
App standby state: NEVER
Docs
App standby state: RARE
Drive
App standby state: RARE
Duo

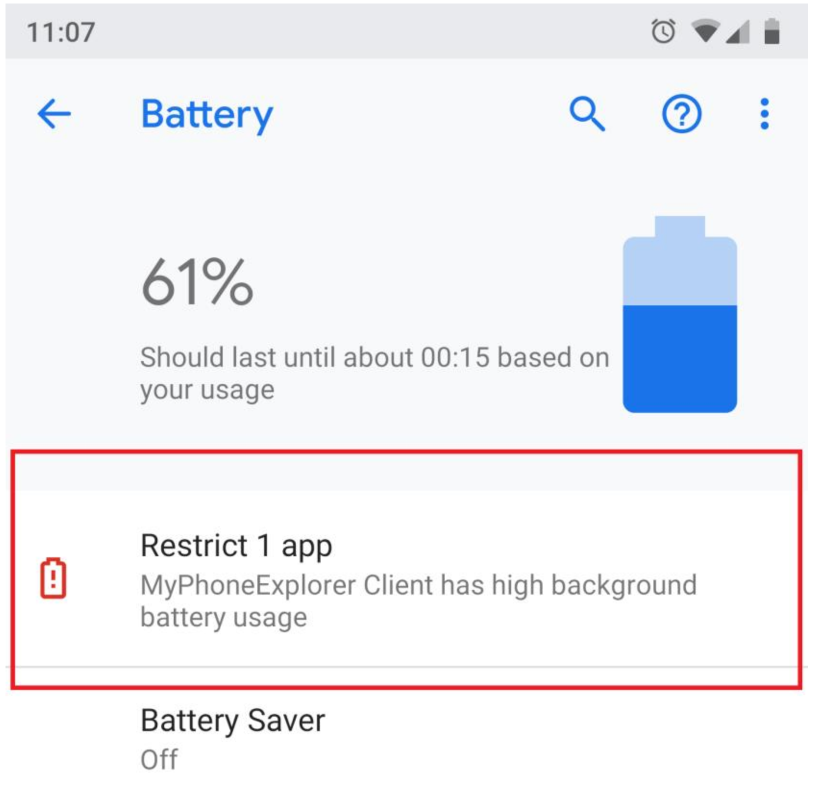
What’s new for Zebra Developers in Android P

Power Management changes: App restrictions

- The user can put an application into the ‘restricted’ state
- Until the app comes to the foreground a restricted app will have the following restrictions:
  - Jobs or Alarms will not fire
  - The app will have no access to the network or device location
  - The app will not be able to receive Firebase Cloud Messages regardless of their priority
- This is an evolution of Oreo’s background limits, but stricter
  - E.g. on Oreo, background apps can still receive Firebase Cloud Messages

What’s new for Zebra Developers in Android P

Power Management changes: App restrictions


11:07

Battery

61%

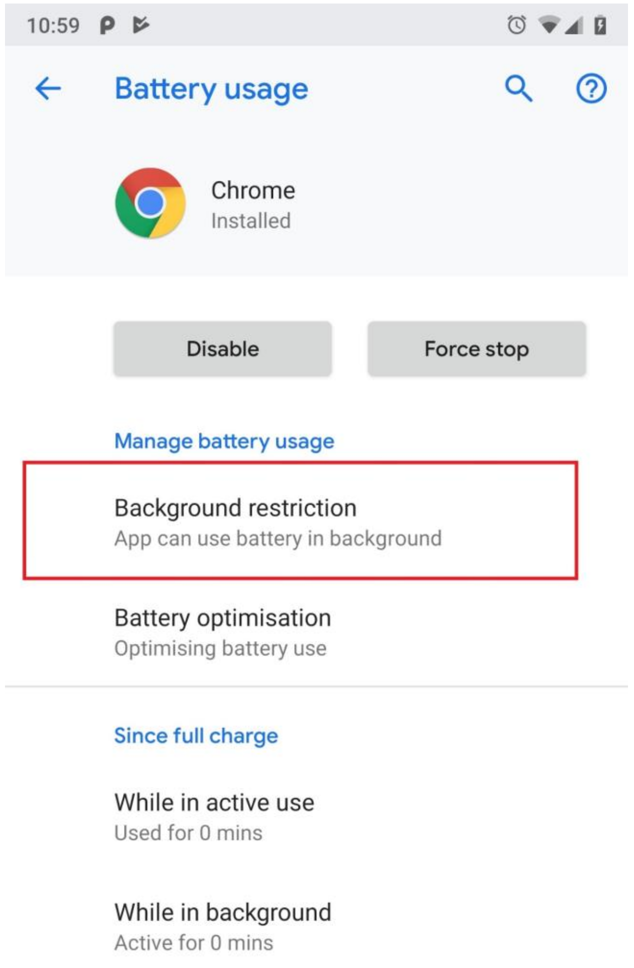
Should last until about 00:15 based on your usage

Restrict 1 app
MyPhoneExplorer Client has high background battery usage

Battery Saver
Off

Apps are restricted **by the user**, either by:

- From the **automatically generated** list on the “Battery” Settings UI (Settings → Battery), Left.
  - Criteria for this determination is subject to change.
- From the “Battery usage” Settings UI (Apps & notifications → App → Advanced → Battery → Background restriction), Right.


10:59

Battery usage

Chrome
Installed

Manage battery usage

Background restriction
App can use battery in background

Battery optimisation
Optimising battery use

Since full charge

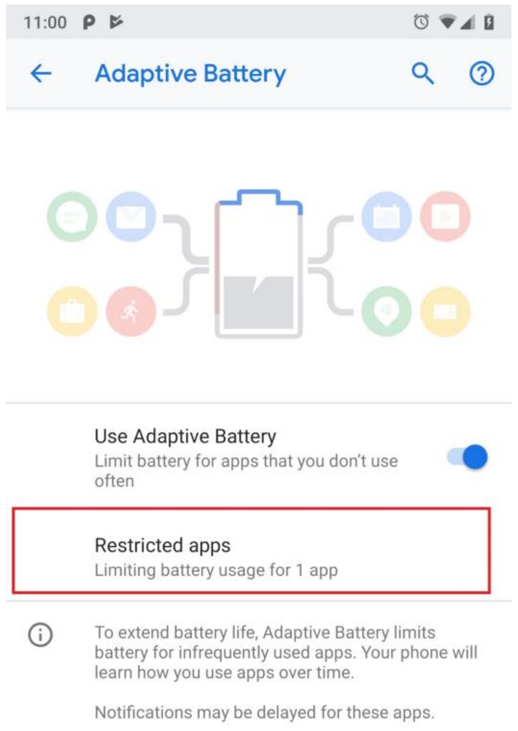
While in active use
Used for 0 mins

While in background
Active for 0 mins

What’s new for Zebra Developers in Android P

Power Management changes: App restrictions

- You can view which apps are restricted from the Adaptive Battery Screen:
  - Settings → Battery → Adaptive Battery → Restricted apps
- There is no API to determine if your app is restricted or to control whether or not it should be restricted
  - Again, Google advises to work with the restriction rather than counteract it.



Use Adaptive Battery
Limit battery for apps that you don't use often

Restricted apps
Limiting battery usage for 1 app

To extend battery life, Adaptive Battery limits battery for infrequently used apps. Your phone will learn how you use apps over time.

Notifications may be delayed for these apps.

What's new for Zebra Developers in Android P

Power Management changes: App restrictions

Most enterprises will want their line of business applications to **never be restricted**:

- Best practice (even before app restrictions) is to **limit the settings available to the end user**, e.g.:
  - Zebra's own Enterprise Home Screen application can reduce the available settings
  - Restricting access to device settings has been a feature of Zebra's MX Access Manager for a long time now
  - If you are using an EMM, they will almost certainly offer the ability to lock down the device settings from the end user.
  - You could use the kiosk features which are part of Android Enterprise for task specific devices

What’s new for Zebra Developers in Android P

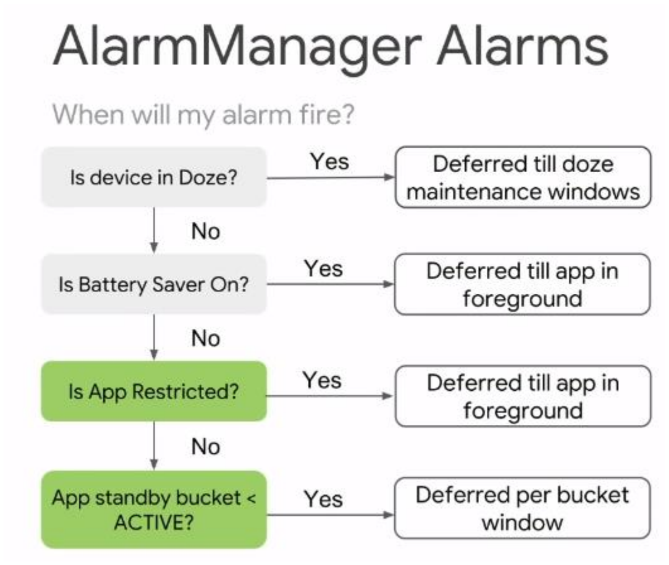
Power Management changes: Summary

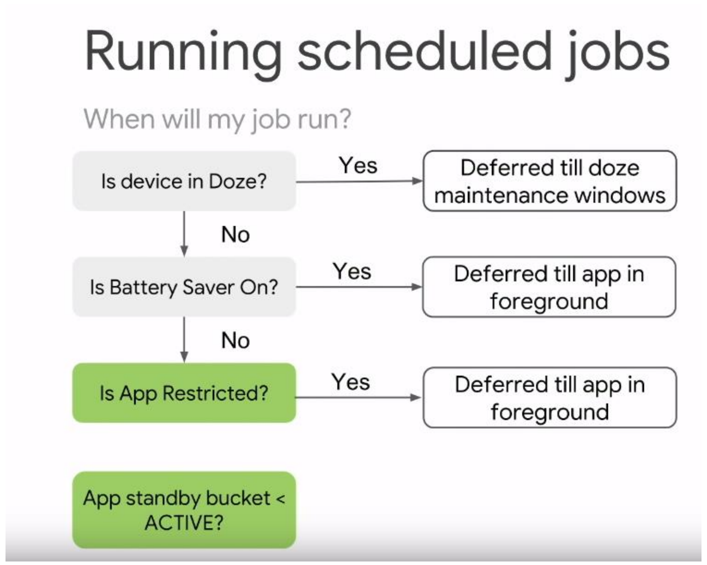
- Google have given more **user control** over how applications use battery
- In general, Enterprises want to limit what the user can do rather than give more control.
- Most customers already use some form of lock-down over the device settings
- Zebra offer full configurability over the device doze mode (which affects App Standby Buckets)
- Actual impact to your deployment may be minimal but worth understanding what changes have been made.

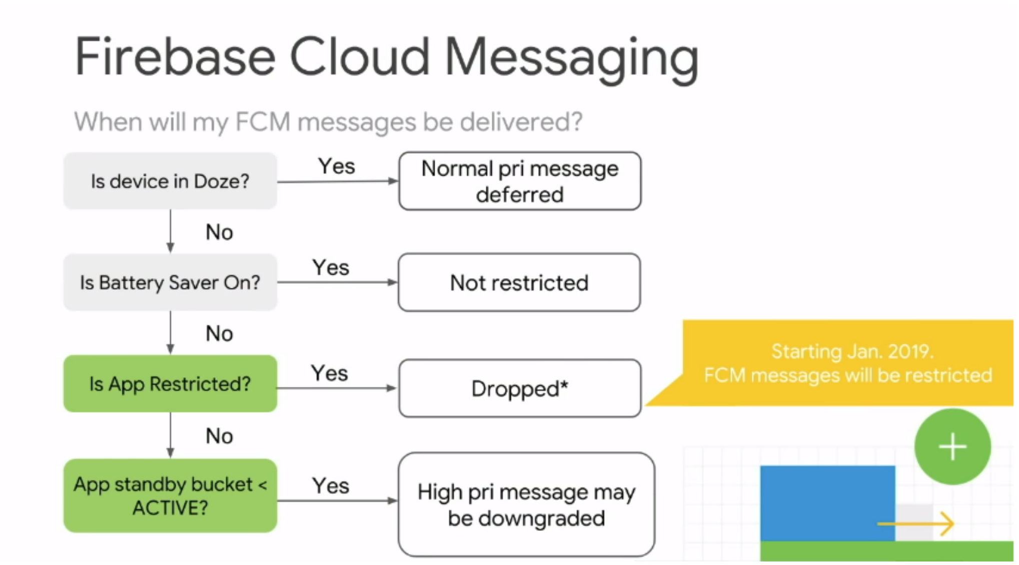
What’s new for Zebra Developers in Android P

Power Management changes: Summary

- For an alternative summary, I would recommend Google’s Android DevSummit presentation from November 2018: https://www.youtube.com/watch?v=-7eZL3XRqas
- Gives a good overview of Alarms, Jobs and FCM and how they are affected by the Power changes in Pie
  - Screenshots from that talk below:


AlarmManager Alarms
When will my alarm fire?
Is device in Doze?
Yes
Deferred till doze maintenance windows
No
Is Battery Saver On?
Yes
Deferred till app in foreground
No
Is App Restricted?
Yes
Deferred till app in foreground
No
App standby bucket < ACTIVE?
Yes
Deferred per bucket window


Running scheduled jobs
When will my job run?
Is device in Doze?
Yes
Deferred till doze maintenance windows
No
Is Battery Saver On?
Yes
Deferred till app in foreground
No
Is App Restricted?
Yes
Deferred till app in foreground
App standby bucket < ACTIVE?


Firebase Cloud Messaging
When will my FCM messages be delivered?
Is device in Doze?
Yes
Normal pri message deferred
No
Is Battery Saver On?
Yes
Not restricted
No
Is App Restricted?
Yes
Dropped*
Starting Jan. 2019. FCM messages will be restricted
No
App standby bucket < ACTIVE?
Yes
High pri message may be downgraded

What’s new for Zebra Developers in Android P

Lock down enhancements

- Lock Task Mode is Android Enterprise’s Kiosk Mode
- Callable from a DO (or PO), so aimed primarily at EMMs. Application developers need to be aware that they could be running in Lock Task Mode
- In Marshmallow, Nougat and Oreo:
  - The ***application developer*** had to request that lock task mode be set
- In Pie and newer:
  - The DO (EMM) can put any application into lock task mode
- Impact to developers:
  - You may find yourself in lock task mode unexpectedly
  - You may need to communicate your dependencies to the administrator
- New Flags have been introduced in P to control what ‘Lock Task Mode’ looks like

What’s new for Zebra Developers in Android P

Restrictions on non-SDK interfaces

- Google have produced an entire documentation subpage describing the restrictions being imposed on applications calling non-SDK methods via reflection or via JNI. Please consult that documentation for information on how to determine which APIs your application is using, details of the errors thrown and additional technical information about the different SDK lists.

Whitelist • Interface is part of the Android SDK • No restrictions on use	Blacklist • Interface can only be called by Platform apps • Will throw an error when invoked
Light Greylist • No public alternative exists for this API • Warning will be generated if target SDK >= P • API will continue to function	Dark Greylist • Warnings will be generated if the target SDK < P • Will throw an error when invoked if the target SDK >= P

What’s new for Zebra Developers in Android P

Restrictions on non-SDK interfaces

- Although OEMs can *add* interfaces to these lists, Zebra have no current plans to do so.
  - OEMs cannot *remove* interfaces from these lists (& still be CTS compliant!)
- Action for Enterprise developers targeting Pie devices:
  - Consult the Google documentation for ‘testing for non-SDK interfaces’
    - Run your application on a Pie device or emulator (where possible) and observe any log messages or runtime errors because of blacklisted or dark greylisted applications
    - Run the veridex static analysis tool on your apk to detect calls to non-SDK methods via reflection.
  - Modify your application so it is not making any calls to the Blacklisted or dark greylisted APIs
  - Monitor use of APIs on the light greylist and transition when possible.

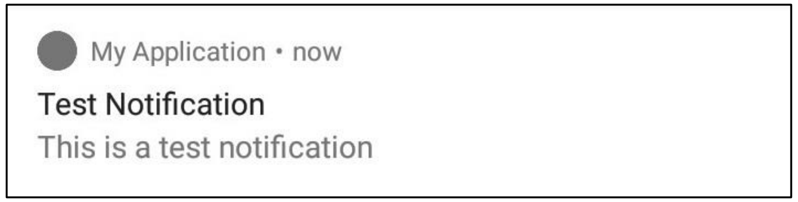
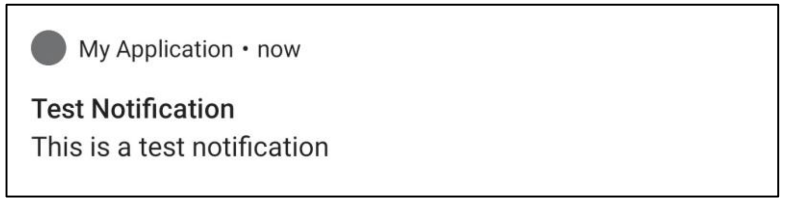
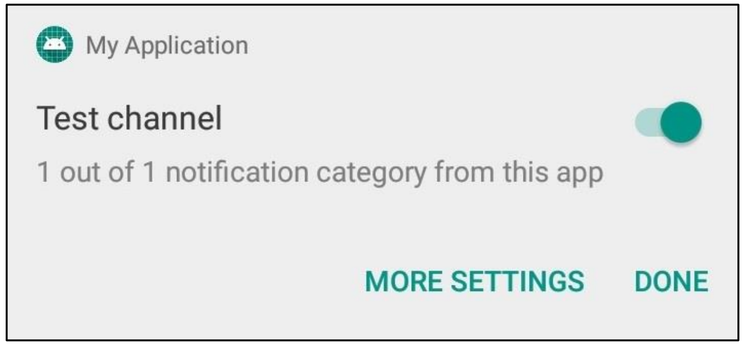
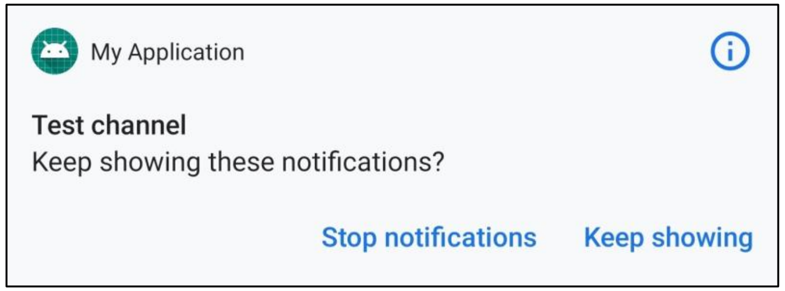
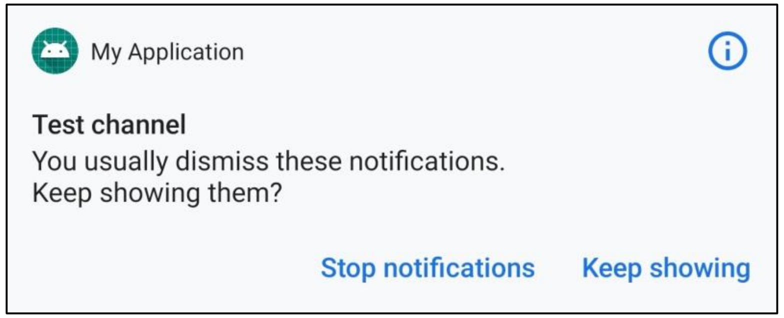
What’s new for Zebra Developers in Android P

Ephemeral Users

- Extension to Android’s multi-user capability and are implemented as secondary users
- Deleted when the user logs out (stops) or switches away
- Ephemeral users are one piece in an overall solution
  - Primary / secondary user creation is controlled by the DO (or PO), i.e. the EMM
- Use cases are around shared devices where a worker will take a device out of a rack and that device is ready to go.
  - Does not specifically require user log-in / log-out but that could be accomplished via the EMM
- Want feedback. Does this meet your shared device use cases?
- Zebra Pie devices are / will be compatible with Ephemeral users, where supported by the EMM

What’s new for Zebra Developers in Android P

Notification enhancements

Notification	Oreo	Pie
Standard notification	 My Application • now Test Notification This is a test notification	 My Application • now **Test Notification** This is a test notification
Long press the notification	 My Application Test channel 1 out of 1 notification category from this app MORE SETTINGS DONE	 My Application **Test channel** Keep showing these notifications? Stop notifications Keep showing
After repeatedly dismissing the notification	No change in behaviour	 My Application **Test channel** You usually dismiss these notifications. Keep showing them? Stop notifications Keep showing

What's new for Zebra Developers in Android P

Notification enhancements

- Recommendations for Enterprise apps
  - Continue to enable the Settings Manager's App Notification Control option to prevent the user from accessing the application settings via a long press
  - Look out for additional MX features to ensure notifications can be fully locked down under Pie
  - Optionally: make your notifications persistent to avoid them being dismissed by the user

Latest Android Oreo & Pie Features: In Summary

Latest Android Oreo & Pie features for your Enterprise Application

Conclusions

- Android Oreo and Pie have both introduced many new features for application developers to take advantage of
- Android Enterprise continues to gain functionality and traction in the industry
- Oreo and Pie continue Android's trend of giving more control to users whilst limiting what application developers can do in the background
  - This can be challenging to understand every nuance of the change but:
    - Google provide great classes like **WorkManager** which "just work" in the background (with caveats)
    - Zebra offer enterprise configuration to re-enable a number of use cases we are hearing from our customers
  - The end result will be extended battery life – ***essential*** for any Enterprise deployment.

Latest Android Oreo & Pie features for your Enterprise Application

Resources

- What's New for Android 'O' and the impact on Zebra Developers:
  - [Developer portal post | DevTALK]
- What's New for Android 'P' and the impact on Zebra Developers:
  - https://developer.zebra.com/blog/what%E2%80%99s-new-android-pie-and-impact-zebra-developers
- Google published documentation for each new release (samples, behaviour changes, API changes)
  - Lollipop, Marshmallow, Nougat, Oreo, Pie, Android 10
- Google published documentation for new Android Enterprise features (primarily EMM focused)
  - Nougat, Oreo, Pie, 10
- Recommended Google resources for specific Pie features:
  - Android Enterprise talk on Power changes
  - Background execution advice (blog)

Questions?

ZEBRA DEVELOPER PORTAL
http://developer.zebra.com

Sign up for news
Join the ISV program

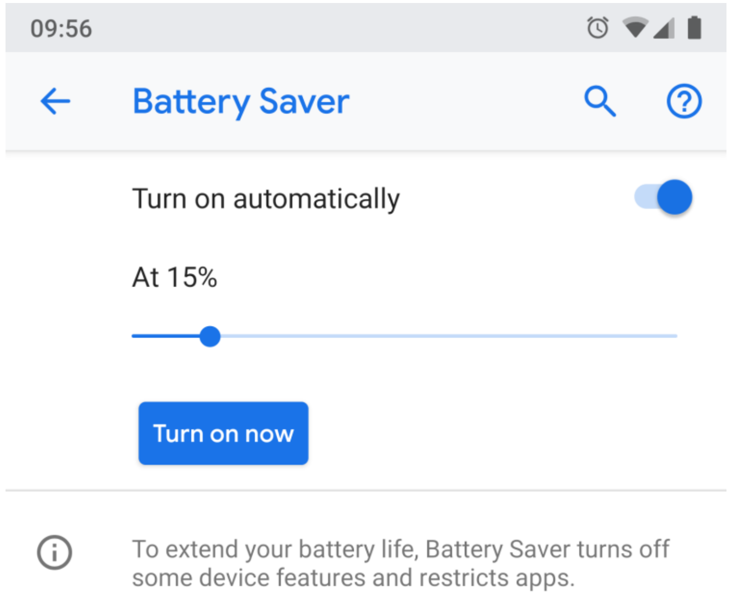
Thank You

Backup

What’s new for Zebra Developers in Android P

Power Management changes: Battery saver improvements

- Battery saver is similar to application restrictions but applies restrictions to all applications on the device.
- When battery saver is turned on the device will take several steps to reduce battery consumption for example stopping all applications from performing background work (official documentation).
- Battery saver is not a new feature in Pie but it has been modified:
- Although disabled out of the box, battery saver can either be turned on manually by the user or be configured to turn on automatically at some specified battery percentage, by default 15%.
- Restrict access to the Settings screen to prevent your users enabling Battery Saver mode



To extend your battery life, Battery Saver turns off some device features and restricts apps.

What’s new for Zebra Developers in Android P

Power Management changes: Battery saver improvements

Battery saver	Oreo	Pie
Out of box state	Off and battery saver will not automatically enable	Off and battery saver will not automatically enable
How to manually enable	Either from the Battery Saver menu (Settings → Battery → Battery Saver) or from the Quick Settings icon	Either from the Battery Saver menu (Settings → Battery → Battery Saver) or from the Quick Settings icon
Supported ‘turn on automatically at X%’	Supports either 10% battery or 18% battery	Configurable anywhere between 5% and **75%**
How to turn off battery saver	Either manually from the Battery Saver menu / Quick Settings icon or by providing power to the device	Manually from the Battery saver menu / Quick Settings icon. In contrast with Oreo, providing power to the device will only disable battery saver ***whilst*** that power is applied.