

Flash Wear Management in Android Automotive

Android Automotive internal storage uses an Embedded MultiMediaCard (eMMC) with thousands of erase/write cycles; if the eMMC fails, the system can become unusable. As vehicles have long lifespans (typically 10+ years), the eMMC must be extremely reliable. This page describes eMMC behavior and how OEMs can lower the risk of a failing eMMC (and thus avoid failed Android Automotive systems).

eMMC behavior

eMMC devices use wear leveling techniques to work around erase/write limitations by arranging data and distributing writes evenly across the system (so no single block fails due to intensive writes). The estimated life of eMMC depends on:

- **Amount of writes.** On phones, the amount of data written to internal storage can be more than 10GB per day. On Automotive implementations, we don't have real world data on how much data will be written due to limited applications. However, when users are streaming high quality music and using navigation, we observe 50MB data written to eMMC every minute. In the future, we may have other types of write-intensive apps, such as dashboard camera applications that continuously record and store videos. In addition, some cars will be shared vehicles used multiple hours everyday. For these reasons and others, we expect Android Automotive implementations to have more eMMC writes than a phone.
- **Write patterns.** Writes and erasures are done in blocks. Writing data frequently in small chunks wears the eMMC faster than writing the same amount of data less frequently and in larger chunks.
- **Available size of eMMC.** Larger storage size means the wear leveling algorithm can spread the writes across larger number of blocks.
- Wear leveling techniques.
- **Environmental factors.** Examples include an operating temperature range of usually -20 ~ 85 Celsius; temperature beyond this range could further shorten the lifespan of the eMMC.

For an eMMC with 16GB usable space and 3k erase/write cycles, we estimate the following:

Daily writes	16GB	32GB
Estimated life time	10 years	5 years

However, the system would stop functioning properly long before the eMMc completely wears out as the usable storage size decreases, and the eMMC may have an even shorter lifespan depending on the leveling techniques and the write patterns used. In addition, this estimate does not consider the effects of misbehaved or malicious apps, which could attack Automotive systems by writing large blocks of junk data to eMMC without special permissions.

To detect the possible eMMC failure before it actually happens, proper storage health monitoring should be built in as part of the overall system health monitoring

Implementation

Android O supports features that enable OEMs to protect and monitor the internal storage of Android Automotive and prolong its lifespan.

Restricting third-party apps

To protect the internal storage of Android Automotive system, Android O enables OEMs to configure whether third-party apps can be installed on internal storage (apps can write only to the partition on which they were installed). To configure, set the following configuration in the resource overlay:

```
<bool name="config_allow3rdPartyAppOnInternal">false</bool>
```

Reducing flash wear

OEMs concerned about flash wear on internal storage can also add an SD card that is fast enough to be used as adopted storage. The SD card has the following behavior:

- When adopted, the SD card will be encrypted and is safe for storing app data.
- SD card slot must be in a safe location (users are not expected to remove the SD card frequently).
- SD card cannot be used for transferring data between Automotive systems and a computer.
- Ejecting the SD card will not affect a running system. However, it should not be removed unless it needs to be replaced.

To ensure second-party applications (those built by car app developers) can be installed on SD card if the car mandates, car app developers must include `android:installLocation` (<https://developer.android.com/guide/topics/manifest/manifest-element.html#install>)=`["auto" | "preferExternal"]` in the app's manifest file.

If the car does not allow third-party apps to be installed on internal storage (as described in [Restricting third-party apps](#) (`#restricting-third-party-apps`)), without this flag (or if the `installLocation` (<https://developer.android.com/guide/topics/manifest/manifest-element.html#install>)=`internalOnly` setting is configured), app installation will fail.

Getting disk metrics with `storaged`

Android O introduces *storaged*, a new system service that samples and publishes disk and eMMC metrics such as information about overall disk usage, eMMC lifetime estimation, and per application disk I/O stats. OEMs can use this information to warn users when the internal storage begins to fail or when specific applications are performing too many disk I/Os. For details, refer to [Implementing storaged](#) (<https://source.android.com/devices/tech/debug/storaged>).

Validation

This feature is tested in the `PackageManager` tests.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated August 21, 2017.