# 相册Android Q适配方案

|  |
| --- |
| 更 改 记 录 |
| 版本号 | 修改章节 | 修改页码 | 更改内容简述 |  更改日期 |
| V1.0 |  |   | 创建 | 2019.05.14 |
| V1.0.1 | 2.1 |  | 添加云相册适配方案 | 2019.05.24 |
| V1.0.2 | 2.22.32.62.8 |  | 添加相册图集数据加载、回收站、文件解析、用户数据兼容方案 | 2019.05.25 |
| V1.0.4 | 2.12.7 |  | 云相册方案调整，只针对Q的下载做处理添加 相册数据库调整 方案 | 2019.07.02 |
| V1.0.5 | 新增3.7 3.83.9 |  | 增加沙盒模式风险项 | 2020.01.11 |
|  |  |  |  |  |
|  |  |  |  |  |

相册Android Q适配方案 1

1 简介 5

1.1 背景 5

1.2 Android Q改动点 5

1.2.1沙箱视图 5

1.2.1.1沙箱模式 7

1.2.1.2兼容模式 7

1.2.1.3 权限 8

1.2.1.4 对各类型文件访问的总结 8

1.2.1.5标识特定的外部存储设备 9

1.2.2 增强了用户对定位权限的控制力 9

1.2.3 从后台启动Activity的限制 9

1.2.4 不可重置的硬件标识符 9

2 适配方案 11

2.1云相册 11

2.1.1 云相册方案概要 11

2.1.2 云相册同步范围 12

2.1.2.1 同步目录访问方式 12

2.1.2.2 多端目录对齐 13

2.1.3 云相册数据同步逻辑 15

2.1.3.1 数据上传同步流程 15

2.1.3.2 数据下载同步流程 16

2.1.3.3 私密保险箱文件判定 16

2.1.3.4 本地文件存在判定 16

2.1.3.5 同步目录适配名单 16

2.1.4 云相册文件同步 17

2.1.5风险 17

2.2 回收站 18

2.2.1 回收站方案概要 18

2.2.2 回收站方案详细介绍 18

2.2.2.1 回收站位置 18

2.2.2.2 回收站功能 19

2.2.2.3 对云相册的影响 20

2.2.2.4 回收站升级方案 20

2.2.2.5 外置t卡插拔问题 20

2.2.2.6 产品表现变化 21

2.2.2.7 风险 21

2.3 相册图集数据加载 21

2.3.1 多db数据访问兼容 21

2.3.2 \_data查询数据库改为uri 23

2.3.3 风险 24

2.4 目录名单 24

2.5 用户数据兼容 24

2.5.1 数据存储问题 24

2.5.1.1 概要 24

2.5.1.2 目录范围 24

2.5.1.3 方案 25

2.5.2 数据升级问题 26

2.5.2.1 问题概要 26

2.5.2.2 已有的数据刷新逻辑 26

2.5.2.3 方案 26

2.5.2.4 风险 27

2.6 文件访问引起的问题 27

2.6.1 共享图集 27

2.6.2 贴纸下载 27

2.6.3 Trace埋点 27

2.6.4 图片解析 27

2.6.4.1 目录范围 27

2.6.4.2 解析方式 28

2.6.5 文件Exif修改 28

2.6.6 文件modify time修改 28

2.7 相册数据库调整 29

2.7.1 建立local\_media表 29

2.7.1.1 相册数据库现状 29

2.7.2 扫描文件 30

2.7.2.1 文件解析 30

2.7.2.2申请MediaScannerJobService 30

2.7.2.1扫描时机、策略 30

2.7.3 影响范围 30

2.7.4 产品和体验影响 31

2.8 其他问题 31

2.8.1 图片地理位置信息 31

2.8.2 自定义图集和连拍问题 32

2.8.3 不可重置标识符（包含IMEI和序列号） 32

2.8.4 限制后台启动Activity的影响 32

2.8.4.1 云服务开启云相册开关无法同步问题 32

3 风险点总结 33

3.1 RelativePath 问题: 33

3.2 跨版本问题 33

3.3 升级时机问题 33

3.3.1 一次性升级完成: 33

3.3.2 分批升级: 33

3.4 代码兼容问题 34

3.5 媒体库会扫描沙箱 34

3.6 外置T卡问题 34

3.7 性能问题 34

3.8 SAF授权问题 35

3.9 名单配置问题 35

4 直接用兼容模式的风险 35

5 评审记录 35

# 1 简介

## 背景

Android Q扩展了用户对数据和应用功能的透明度和控制。隐私权功能让用户更清楚地了解并更好地控制他们的数据及为应用提供的功能。这些隐私权功能可能意味着应用所依赖的特定行为或数据可能不再可用

## Android Q改动点

### 1.2.1沙箱视图

对于targetSDKVersion是Q的应用来说，存储分为3部分：

1. 私有目录 /data/data/packagename/、/Android/data/packagename/

 文件通过File API来访问

1. 沙箱目录 /Android/data/sandbox

文件可以通过File API来访问。沙箱目录下的文件会被媒体库扫描，所以也能通过媒体库的接口来访问，可读，可修改，可删除。沙箱模式下，使用new File("/sdcard/")目录前缀写的文件会映射到沙箱目录/Android/data/sandbox下谷歌推荐app使用Context.getExternalFilesDir()存自己的文件在私有目录里

1. 共享目录 /Pictures、/Movies、/Downloads、/Documents

文件通过MediaStore API来访问.

新建文件，MediaProvider insert()。可以设置display\_name,relative\_path,mimetype。MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI -- /DCIM/xxxx/ -- image/\*

删除文件，MediaProvider delete()。

修改文件，通过getContentResolver.openFileDescriptor(uri,mode)获取fd

沙箱模式下，使用new File("/sdcard/")目录前缀写的文件会映射到沙箱目录/Android/data/sandbox下

谷歌推荐app使用Context.getExternalFilesDir()存自己的文件在私有目录里.

共享目录下不能能创建隐藏文件.会报错.

**共享目录下的不同类型媒体文件，只能存放到特定的一级公共目录,这个限制是由媒体库来控制的.**

|  |  |  |  |
| --- | --- | --- | --- |
| Mine type | URI路径 | 一级目录 | 一级目录 |
| audio/\* | images/mediaimages/media/# | Environment.DIRECTORY\_ALARM Environment.DIRECTORY\_MUSICEnvironment.DIRECTORY\_NOTIFICATIONSEnvironment.DIRECTORY\_PODCASTSEnvironment.DIRECTORY\_RINGTONES  | Alarms Music Notifications Podcasts Ringtones  |
| image/\* | audio/albumartaudio/albumart/# | Environment.DIRECTORY\_MUSIC | Music  |
|  | audio/playlistsaudio/playlists/# | Environment.DIRECTORY\_MUSIC | Music  |
| video/\* | video/mediavideo/media/# | Environment.DIRECTORY\_DCIMEnvironment.DIRECTORY\_MOVIES | DCIM Movies  |
| image/\* | images/mediaimages/media/# | Environment.DIRECTORY\_DCIMEnvironment.DIRECTORY\_PICTURES | DCIM Pictures  |
| image/\* | video/thumbnailsvideo/thumbnails/# | Environment.DIRECTORY\_MOVIES | Movies |
| image/\* | images/thumbnailsimages/thumbnails/# | Environment.DIRECTORY\_PICTURES | Pictures  |
|  | downloadsdownloads/# | Environment.DIRECTORY\_DOWNLOADS | Download  |
|  | filefile/# | Environment.DIRECTORY\_DOWNLOADSEnvironment.DIRECTORY\_DOCUMENTS | Download Documents |

#### 1.2.1.1沙箱模式

 默认情况下，如果你的应用程序的目标是Android Q，它会有一个针对Android Q Beta 3版本文件的沙箱视图，而针对Android 9 (API级别28)或更低版本的应用程序默认情况下的存储方式与之前的Android版本没有变化。当您更新现有的应用程序以使用范围存储时，您可以使用一个新的manifest属性allowExternalStorageSandbox在Android Q设备上启用应用程序的新行为，即使您的应用程序的目标是Android 9或更低。

1.target为Q，默认打开沙箱模式，可以通过设置allowExternalStorageSandbox = false来退出

2.target为P及以下，默认关闭，可通过设置 allowExternalStorageSandbox = true来打开.

3.保存在沙箱中的文件,应用卸载后会被删除.

4.保存在沙箱中的文件,除了通过媒体库的接口外,不能通过其他方式访问.保护了数据隐私.

#### 1.2.1.2兼容模式

1.兼容模式的定义:

兼容模式是在Q上的应用还能通过非沙箱视图来访问文件的一种模式.beta2里只要是从P升到Q之前就安装在手机里的应用，都是兼容模式.beta3增加了allowExternalStorageSandbox这个条件一起控制，是或的关系.我们可以通过兼容模式来访问非沙箱视图下的文件，做升级。

兼容模式和之前一样，apk对sdcard，拥有完整的访问权限

兼容模式下可以通过媒体库接口访问其他应用的沙箱文件,但是不能通过file接口访问.

OTA升级后(P-Q):如果一个apk在P上，申请了WRITE\_EXTERNAL\_STORAGE、READ\_EXTERNAL\_STORAGE权限，并且升级到Q后没有卸载或者恢复出厂设置，那么这个apk会保持兼容模式，而不是新的沙箱模式。

#### 1.2.1.3 权限

1.保持READ\_EXTERNAL\_STORAGE 和 WRITE\_EXTERNAL\_STORAGE不变

2.没申请读权限，访问共享目录里的文件只能访问到自己应用创建的文件

3.申请了读权限，访问共享目录能访问到其他应用创建的文件

4.没申请写权限，只能写共享目录里自己应用创建的文件

5.申请了写权限，能写共享目录里其他应用创建的文件（这里的写，是指修改和删除）

这份文档里说明了权限的组别和所属

<http://opengrok.scm.adc.com:8080/source/xref/SDM710_Q_Master/android/frameworks/base/core/res/AndroidManifest.xml>

#### 1.2.1.4 对各类型文件访问的总结



图1沙盒视图文件访问摘要

#### 1.2.1.5标识特定的外部存储设备

在 Android 9（API 级别 28）及更低版本中，所有存储设备上的所有文件都显示在单个 "external" 卷名称下。Android Q 为每个外部存储设备提供唯一的卷名称。

注意：主外部存储设备始终使用卷名称 "external"。

要唯一标识外部存储设备中的特定文件，您必须同时使用卷名称和 ID。例如，主存储设备上的文件是 content://media/external/images/media/12，而名为 FA23-3E92 的辅助存储设备上的对应文件是 content://media/FA23-3E92/images/media/12。

您可以通过将此卷名称传递到特定媒体集合（例如MediaStore.Images.getContentUri()）中来访问存储在特定卷中的文件。

### 1.2.2 增强了用户对定位权限的控制力

 **AndroidQ引入了新的位置权限ACCESS\_BACKGROUND\_LOCATION，权限仅会影响应用在后台运行时对位置信息的访问权。**

**除非应用的某个活动可见或应用正在运行前台服务，否则应用将被视为在后台运行**。

### 1.2.3 从后台启动Activity的限制

Android Q增加了限制apk启动后台activity，这个改变目的是为了减少这类动作对用户的干扰。

目前测试能启动，但是会弹Toast Background activity start from package-name blocked。Google说后续版本会强制不能启动，推荐使用通知的方式，但是符合以下条件也能启动APP：

1. 有系统连接的service可以启动。

2、APP有service给其他可视的APP连接。

### 1.2.4 不可重置的硬件标识符

从Android Q开始，应用必须具有READ\_PRIVILEGED\_PHONE\_STATE特许权限才能访问设备的不可重置标识符（包含IMEI和序列号）。许多用例不需要不可重置的设备标识符。如果您的应用没有该权限，但您仍然尝试查询标识符的相关信息，则平台的响应会因目标SDK版本而异。



图 2 Android Q 隐私权限核对清单

# 适配方案

与Android P不同的是，需要兼容外置sdcard的数据库，做好数据兼容，同时针对文件系统不同的目录做访问兼容。

## 2.1云相册

### 2.1.1 云相册方案概要

 Android Q 上云相册主要影响点：

* 三方App图片、视频目录变更
* 跨Android版本多端同步目录无法对齐
* 无法获取文件绝对路径，同步逻辑受影响
* 上传下载读写文件受影响

适配原则：

* 保证云端元数据与老版本一致
* Android Q上目录做转换，

在Android Q上，将媒体库和回收站的数据进行封装，适配到与Android P一样的元数据，与P保持一样的元数据，并在元数据下载到Q的时候做路径适配。



### 2.1.2 云相册同步范围

#### 2.1.2.1 同步目录访问方式

|  |  |  |
| --- | --- | --- |
| 目录 | 目录类型 | 访问文件方式 |
| 兼容模式 | 沙箱模式 |
| DCIM/\* | 共享目录 | File类访问 | 媒体库接口 |
| Pictures/MessengerPictures/BBMPictures/LINEPictures/LINE\_MOVIE | 共享目录 | File类访问 | 媒体库接口 |
| Download/Mms | 共享目录 | File类访问 | 媒体库接口 |
| Movies/Messenger | 共享目录 | File类访问 | 媒体库接口 |
| 腾讯系：Pictures/Tencent/QQ\_Images Pictures/Tencent/QQfile\_recv Pictures/Tencent/MicroMsg/WeiXinPictures/Tencent/MicroMsg/WeChatPictures/QQ\_Screenshot其他三方app：Pictures/SnapseedPictures/sina/weibo/weiboPictures/WhatsApp/Media/WhatsApp | 共享目录 | File类访问 | 媒体库接口 |
| Bluetooth | 沙箱目录 | 媒体库接口无法插入数据 | 媒体库接口无法插入数据 |
| 相册回收站文件Android/data/com.coloros.gallery3d | 私有目录 | 直接访问文件 | File类访问 |

如表格所示，Android Q下，目录只要分为三类：共享目录、三方应用沙箱目录、相册私有目录。

* 兼容模式下，除了三方沙箱需要通过媒体库访问，其余目录可以直接通过File方式访问
* 沙箱模式下，除了私有目录可以使用File方式访问，其余的都需要使用媒体库接口访问
* 无论兼容模式还是沙箱模式，沙箱目录都没有写入文件的权限，同步目录对齐只能是相对目录对齐

#### 2.1.2.2 多端目录对齐

|  |  |  |  |
| --- | --- | --- | --- |
| **同步目录** | **Android版本** | **上传** | **下载** |
| 共享目录 | Android P | 原目录 | 原目录 |
| Android Q | **原目录** | **原目录** |
| Android R | 原目录 | 适配到新目录 |
| 回收站 | Android P | 原目录 | 原目录 |
| Android Q | **适配到原目录** | **适配到新目录** |
| 分身目录 | Android P | 原目录 | 原目录 |
| Android Q | **原目录** | **适配到手机存储目录** |
| 其他目录（非共享、非沙箱） | Android P | 原目录 | 原目录 |
| Android Q | **原目录** | **原目录** |
| Android R | 原目录 | 适配到新目录 |

1. 共享目录

例如：dcim、picture、Movie等

* Dcim、Movie、Picture等共享目录下原来已经同步的子目录，可以直接对齐，将准确的元数据直接同步到云端
* Movie、Picture共享目录下没有在同步范围的目录，即三方App产生的图片，将其适配成/Sdcard/0开头的路径，这样相同的同步目录在云端的元数据是唯一的，本地匹配也会简单很多。

例如微信的目录：Tencent/MicroMsg/WeiXin

这个与微信的开发确认过，Android Q上沙箱模式下图片会放入Pictures目录下，这样文件的真实路径会存在两个：

* /Sdcard/0/Tencent/MicroMsg/WeiXin -- Android P
* /Sdcard/0/Pictures/Tencent/MicroMsg/WeiXin -- Android Q

**方案：**

**下载：P的文件路径下载到Q的路径下；**

**上传：Q的路径直接上传，不做修改。**



1. 三方App沙箱目录（**不同步,以下方案废弃**）

例如：Tencent/MicroMsg/WeiXin、WhatsApp/Media/WhatsApp

Android P路径：/sdcard/0/目录

Android Q路径：/sdcard/0/Android/sandbox/包名/目录

这里沙箱的路径与非沙箱目录不一致，需要做适配，兼容非沙箱与沙箱、沙箱与沙箱之间的对齐问题。

Android P：直接上传、下载文件，不需要做路径转换；

Android Q：将沙箱路径与非共享路径全部转换到/Pictures目录下；

1. 相册回收站目录

Android P 回收站路径：/sdcard/0/.ColorosGalleryRecycle；

Android Q 回收站路径：/Android/data/com.coloros.gallery3d/

.ColorosGalleryRecycle；

由于Android P的路径在Android Q上无法正常访问，故将回收站的路径改动到相册私有目录下，同步目录对齐时，需要将Android Q的路径做转换，基本流程与 沙箱目录对齐 类似：

* 上传时将路径适配成 Android P的路径
* 下载时将路径适配成Android Q的路径
1. 其他遗留目录

Android Q不会存在这种路径，针对已经存在的路径，Android Q做适配，将其下载到Restored路径。

1. 云相册缩图缓存

Android P缓存目录路径：/sdcard/0/.ColorosGalleryOcloud

Android Q缓存目录路径：/Android/data/com.coloros.gallery3d/.ColorosGalleryOcloud

与相册回收站类似，这个目录在Q上无法继续使用，将其搬迁到相册私有区域。

### 2.1.3 云相册数据同步逻辑

#### 2.1.3.1 数据上传同步流程


#### 2.1.3.2 数据下载同步流程


#### 2.1.3.3 私密保险箱文件判定

云相册做好文件路径适配后，所有的目录都是唯一的路径，可以继续使用文件路径来做处理，但是需要私密保险箱一起修改，保留文件路径字段。

#### 2.1.3.4 本地文件存在判定

|  |  |  |
| --- | --- | --- |
| 兼容模式 | 私有目录 | File |
| 共享目录 | File |
| 三方App沙箱目录 | 媒体库 |
| 其他目录 | File |
| 沙箱模式 | 私有目录 | File |
| 共享目录 | 媒体库 |
| 三方App沙箱目录 | 媒体库 |
| 其他目录 | 不存在 |

#### 2.1.3.5 同步目录适配名单

|  |  |
| --- | --- |
| 云端路径 | Android Q 路径 |
| Tencent/QQ\_Images | Pictures/Tencent/QQ\_Images |
| WhatsApp/Media/WhatsApp Images | Pictures/WhatsApp/Media/WhatsApp Images |
| WhatsApp/Media/WhatsApp Video | Movies/WhatsApp/Media/WhatsApp Video |
| Tencent/MicroMsg/WeiXin下图片 | Pictures Tencent/MicroMsg/WeiXin |
| Tencent/MicroMsg/WeiXin下视频 | Movies/Tencent/MicroMsg/WeiXin |

如表格所示：

将同步目录的图片和视频全部下载到共享目录区域，按照视频和图片分别作名单。

### 2.1.4 云相册文件同步

这里主要是文件读写问题，云服务上传、下载需要读写文件，由于存在路径适配问题，故云服务元数据看到的路径有可能是虚拟的路径，云服务无法直接根据路径获取到文件，故需要使用FileProvider方式进行文件读写。

* 相册提供GalleryFileProvider，允许云服务可以根据元数据中的文件路径获取文件FD，完成文件上传；
* 云服务提供CloudFileProvider，公开下载的文件cache，允许相册读取文件，完成文件下载。


### 2.1.5风险

无

## 2.2 回收站

### 2.2.1 回收站方案概要



Android Q上回收站的主要影响点：

* 回收站db位置改变
* 已回收文件的位置改变
* 相册清数据回收站数据会被清除
* 沙箱目录的文件无法恢复到原路径
* 导致云相册多一个同步目录
* T卡在多台机器上插拔后回收站的表现受影响

### 2.2.2 回收站方案详细介绍

2.2.2.1 回收站位置



原因：

1. 沙箱模式下原目录不可用，也不能放在公共和沙箱目录（回收站文件不能被其他应用访问）。只剩下/Android/data和/data/data目录。
2. DB放在外置T卡上会有升级和降级问题（从高版本相册的机器到低版本相册的机器），现在将外置T卡的DB移动到/data/data/下。

2.2.2.2 回收站功能

删除和恢复功能的目录和访问方式受到影响。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 原始路径 | 恢复路径 | 访问方式 |
| 现状 | sdcard | 原位 | File |
| 新方案 | 兼容模式 | sdcard下应用的自建目录，公共目录，自己的沙箱目录 | 原位 | File |
| 其他应用沙箱目录 | /DCIM/Recovery | 媒体库+File |
| 沙箱模式 | 公共目录 | 原位 | 媒体库 |
| 自己沙箱目录 | 原位 | Flie |
| 其他应用沙箱目录 | /DCIM/Recovery | 媒体库+File |

备注：过期删除和异常数据处理功能不改动。

2.2.2.3 对云相册的影响

云服务同步回收站里文件的范围也是根据同步名单来的。

其他应用的沙箱目录删到回收站的文件以及相册清数据之后找不到原始路径的文件会放到/DCIM/Recovery，这个目录也会进行同步。

2.2.2.4 回收站升级方案

1.升级操作

移动db：在/data/data/com.coloros.gallery3d/databases/下创建db，把原路径下的db内容填充过来。

移动文件：把原来手机存储和外置t卡的回收站文件从/.ColorOSGalleryRecycler下移动到对应的/Andorid/data/com.coloros.gallery3d/files/recycler/下。

1. 升级时机

相册在Q的第一个版本就升级。

在插入t卡之后进行检测，如果t卡里的回收站是旧版的就进行升级。

2.2.2.5 外置t卡插拔问题



2.2.2.6 产品表现变化

1.清除相册数据，回收站的数据就被清空了
2.t卡的回收站表现变化

2.2.2.7 风险

Android P上的外置t卡不能写的问题，王彬正在看怎么修改。

## 2.3 相册图集数据加载

### 2.3.1 多db数据访问兼容

1、原因：

媒体库内外置存储数据会存储到不同的db。当插入外置T卡时,会存在卷名问题,使得数据来源发生了变化，数据库目录下会多一个external-卷名。db 例如（external-413c-12f7.db）这样就存在两个db，因此需要做兼容。因为涉及到的地方很多，这里就以LocalImageVideoAlbum为例。

2、解决方案：

* 新建BaseMergeAlbum该类继承MediaSet。MediaSet[]作为它的一个构造参数。
* 新建LocalImageVideoMergeAlbum 继承 BaseMergeAlbum。
* 在LocalImageVideoAlbum 的构造函数中增加一个是否有外置T卡的参数通过该参数和内外sdcard卡卷名（volumeName）设置对应的Uri。
* 在用到LocalImageVideoAlbum 的地方都替换成LocalImageVideoMergeAlbum。





3、风险及影响点：

* 通过查询2个db，然后将数据进行合并，会有一定的性能影响。
* 涉及到的改动点非常多，工作量比较大，后期工作量比较大，需要测试的地方很多。



### 2.3.2 \_data查询数据库改为uri

1、原因：

因为在Q的沙箱模式下，目录结构发生了变化，filePath变的不可用。

1. 解决方案：

之前通过filePath方式查询数据库的方式要改成通过uri查询。

1. 风险点及影响：

影响的地方较多，需要一个一个去排查，后期工作量比较大，需要测试的地方很多。

### 2.3.3 风险

Android P上只有回收站是使用多个DB的，相册每次会查询两次数据库，然后合并排序。

 Android Q上所有的图集都需要兼容多DB查询的方案，速度和IO都会比之前差一点，但是从目前的评估看，多查询一次数据库，已经合并排序，没有性能隐患。

## 2.4 目录名单

时间轴图片目录的名单（云同步目录）、黑名单、不常用图集白名单、图集名称映射等。

由于相册配置的是相对目录，如果只是路径前缀发生变化，比如从根目录切换到Pictures下，那对于相册来说不影响，可以继续使用相对目录查询数据；如果相对路径也发生变化，则需要考虑增加名单。

## 2.5 用户数据兼容

### 2.5.1 数据存储问题

2.5.1.1 概要

相册在回忆，收藏，人物，标签，ocr等数据库里都使用了filepath字段，用来标识文件和访问文件。

Android Q上媒体库返回的\_data不再是真实的绝对路径，但是我们可以用媒体库的其他字段组合得到filepath。

2.5.1.2 目录范围

相册在Android Q上显示的媒体文件的范围如下：

|  |  |  |
| --- | --- | --- |
| 目录类型 | 目录 | 访问方式 |
| 兼容模式 | 沙箱模式 |
| 共享目录 | DCIM/\*Pictures/Tencent/QQ\_ImagesMovie/\* | File类访问 | 媒体库接口 |
| 沙箱目录 | Android/sandbox/packagename/Android/sandbox/packagename/Pictures | 媒体库接口 | 媒体库接口 |
| 三方目录 | Sina/weibo/ | File类访问 | 无法访问 |

结论：共享目录和沙箱目录需要使用媒体库接口访问，不能直接拿到\_data，和以往方式不同，需要拼接得出filepath。

2.5.1.3 方案

1.使用媒体库字段拼接FilePath

例如：

共享目录：手机存储 /storage/emulated/0/Pictures/tencent/xxxx.jpg

 外置t卡 /storage/A0AF-14FB/Picture/tencent/xxxx.jpg

沙箱目录：手机存储 /storage/emulated/0/Android/sandbox/package/Pictures/tencent/xxxx.jpg

 外置t卡 /storage/A0AF-14FB/Android/sandbox/package/Pictures/tencent/xxxx.jpg



2. 通过filepath查媒体库记录

显示一个回忆图集的内容，通过filepath查到媒体库记录，然后构造相册的Path。



沙箱目录和共享目录都能通过这种方式查询到记录。

### 2.5.2 数据升级问题

2.5.2.1 问题概要

三方应用会逐步把文件从sdcard下的自建目录转移到共享目录，相册的数据库里存储的记录失效。

2.5.2.2 已有的数据刷新逻辑

如回忆模块，有同步的机制，在Gallery.onCreate会调用syncMeta,会根据当前的MediaMeta的filepath去媒体库里查对应的记录，如果没有的话，会删除MediaMeta的数据库记录。

2.5.2.3 方案

1. 人脸和ocr功能的不需要处理，一方面是它们本来就会做数据的同步处理，另一方面是它们能够重新扫描。
2. 回忆和收藏功能需要处理。通过relativepath+displayname的方式在媒体库里查询对应的记录，如果存在，判断size和dataTaken，如果相同，就确定是同个文件，更新媒体库里的对应字段。

2.5.2.4 风险

1. 回忆和收藏功能对于被三方应用移动的文件在媒体库里记录的判断，如果相对路径、size和dataTaken都相同，但是文件却和移动前不相同，就会出现问题，如收藏了一张没收藏过的图片。
2. 如果三方应用移动文件后把相对路径也改了，那相册就会判定数据库里的对应记录失效并删除，出现的问题是回忆或者收藏里少了部分图片。

## 2.6 文件访问引起的问题

### 2.6.1 共享图集

共享图集的缩图保存在.ColorOS/OCloud/.Cache/ShareAlbum 目录。通过云服务sdk获取共享图集的数据和封面。通过FilePath的方式访问图片。因目录不可用，因此会有影响。

方案：云服务添加FileProvider，相册用来获取文件。

### 2.6.2 贴纸下载

贴纸下载使用/sdcard/ColorOS/.Sticker这个目录作为中转，这个目录在Q上也不可用，因此会有影响。

 方案：放到相册私有目录下。

### 2.6.3 Trace埋点

在/sdcard/ColorOS/Gallery/log/下有日志文件，这个目录在Q上也不可用。因此会有影响。

 方案：跟随系统log方案即可，如果没有权限，需要更手软同事商议新的目录，影响不大。

### 2.6.4 图片解析

2.6.4.1 目录范围

|  |  |  |
| --- | --- | --- |
| 目录类型 | 目录 | 访问方式 |
| 兼容模式 | 沙箱模式 |
| 共享目录 | DCIM/\*Pictures/Tencent/QQ\_ImagesMovie/\* | File类访问 | 媒体库接口 |
| 沙箱目录 | Android/sandbox/packagename/Android/sandbox/packagename/Pictures | 媒体库接口 | 媒体库接口 |
| 三方目录 | Sina/weibo/ | File类访问 | 无法访问 |
| 私有目录 | Android/data/com.coloros.gallery3d/files/recycler | File类访问 | File类访问 |

2.6.4.2 解析方式

图片和视频缩图的解析都是通过BitmapFactory来做的。

传入BitmapFactory的参数有这几种：

* String filepath
* FileDescriptor
* FileInputStream
* Byte[]

但其实最终都是由filepath来构造的。

对于只能用媒体库接口访问的文件，用filepath来构造FileDescriptor等对象时会直接报错。

不再使用FilePath的接口，改为使用FD接口实现，相册需要做兼容，同时提供FilePath和FD的解析接口，兼容不同的版本。

### 2.6.5 文件Exif修改

 

相册目前已经有Exif的工具类，google也给出了使用文件流去编辑Exif的接口，相册需要结合google的建议，升级当前工具类。

### 2.6.6 文件modify time修改

 目前系统会删除个类，后续会继续寻找新的方案实现此功能，应用可以继续使用，但从解耦和系统维护来看，相册最好解耦此接口的使用。

 场景：

1、旋转图片，有Eixf的文件，写入后方向后modify time改变，影响云服务同步

 2、云相册下载文件后，反写文件modify time，保证时间排序与原手机一致

 应对方案：

1. 云相册继续使用modify time作对比，对于modify time变化的，继续对比datetaken、文件大小。文件md5、缩图gid等，这里对性能影响不大。
2. 相册显示云相册范围内的文件时，重新获取云相册数据库的modify time，做二次排序。云相册做好cache，相册调整数据结构，理论上也不会有性能瓶颈。

## 2.7 相册数据库调整

媒体库针对File表的字段访问做了修改：

* 经纬度字段被删除
* 不同的uri（File、Image、Video），只允许访问特定的字段，视频resoulution、Album、Artist等字段无法直接从File表中获取到，需要二次获取，影响性能

针对此问题，相册需要自己实现解析经纬度和resoulution，存储到相册数据库。

### 建立local\_media表

2.7.1.1 相册数据库现状

|  |  |  |  |
| --- | --- | --- | --- |
| DB | Table | 用途 | 改进方案 |
| album | locked\_pictures | 相册app加密后拍照图集 | 保持现状 |
| specified\_album\_name | 图集显示名称 | 保持现状 |
| timershaftextension | 云相册同步目录 | 保持现状 |
| whitelist\_dirs | 白名单 | 保持现状 |
| hide\_dirs，albums，albums\_map，comment，comments，already\_upload |  | 历史遗留，评估删除table |
| **local\_media** | **本地媒体文件数据库** | **新增，用于存储经纬度、resoulution等文件属性，后期可以扩展为相册本地媒体文件数据库** |
| attrs | faceclusterattrs | 人物二期同步状态 | 建议删除数据库，改为preference存储，并封装Provider给云服务或者云服务通过sdk获取。 |
| geo | geo\_mappinggeo\_route | 存储经纬度对应的地址 | 保持现状 |
| memories |  | 回忆 | 保持现状 |
| ocloud | label | 存储云端识别的标签数据（LabelOcrProvider） | 保持现状 |
| ocr | 存储云端ocr识别的文字信息（LabelCloudHelper） | 保持现状 |
| ocr |  | ocr源数据 | 保持现状 |
| sticker |  | 贴纸 | 保持现状 |
| suggestions |  | 最近搜索 | 建议放到albums中去，不需要单独数据库 |
| tags | facescluster | 人脸信息 | 保持现状 |
| labels | 场景、标签信息 | 保持现状 |
| virtualset |  | 虚拟图集，已经废弃 | 历史遗留，评估删除table |
| movebackup |  | 虚拟转实体，历史遗留 | 保持现状 |

### 扫描文件

* + - 1. 文件解析

参照2.8.1 图片地理位置信息

2.7.2.2申请MediaScannerJobService

ComponentName componentName = **new** ComponentName(context, ***MediaScannerService***.**class**);
JobScheduler js = (JobScheduler) context.getSystemService(Context.***JOB\_SCHEDULER\_SERVICE***);
JobInfo job = **new** JobInfo.Builder(GalleryUtils.JobServiceIdPool.M***ediaScannerService***.getJobId(), componentName).setRequiresBattIdle(**true**, 0).build();
js.schedule(job);

2.7.2.1扫描时机、策略

* 进程启动，触发job，并根据job控制是否停止扫描
* 根据电量和是否充电决定是否继续执行扫描
* 确定任务为高优先级，在其他扫描之前执行
* 扫描过程中，如果一切正常，就可以考虑添加进程保护

### 影响范围

|  |  |
| --- | --- |
| 页面 | 功能点 |
| 地点图集 | 按照地点分类显示图片 |
| 时间轴 | 地图相册时间轴地点显示 |
| 大图页 | 左上角地址详情地址 |
| 搜索 | 搜索地点按照地点分类显示图片 |
| 旅游回忆 | 使用经纬度获取国家、城市，生成旅游回忆 |
| 云相册 | 作为元数据上传云端，云端根据经纬度做地点图集 |

### 产品和体验影响

* 相关图集数据显示不及时
* 多级页面，存在数据刷新不一致问题

## 2.8 其他问题

### 2.8.1 图片地理位置信息

1.将新的 ACCESS\_MEDIA\_LOCATION 权限添加到您应用的清单中。

2.在 MediaStore 对象中，调用 setRequireOriginal() 并传入照片的 URI。

3.Exif 数据在兼容 模式和沙箱模式都可以进行读写.沙箱模式下需要用fd来修改



### 2.8.2 自定义图集和连拍问题

可以通过MediaStore.Images.Media.RELATIVE\_PATH的方式在DCIM目录下创建多级目录(Google在开发文档中也有这样的代码片段)



### 2.8.3 不可重置标识符（包含IMEI和序列号）

Android Q 开始在 Android 中,应用必须有的READ\_PRIVILEGED\_PHONE\_STATE为了允许访问特权的装置的标识符可包括 IMEI 、序列号。

### 2.8.4 限制后台启动Activity的影响

#### 2.8.4.1 云服务开启云相册开关无法同步问题

 问题：开关解耦后会移动到云服务进程，此时相册没有申请权限，无法读取数据库，访问文件。

 方案：相册做处理，如果没有申请权限，就拒绝同步流程，不需要弹出权限框，规避此问题。

# 3 风险点总结

# 3.1 RelativePath 问题:

beta2版本用来控制写文件到共享目录里的位置的PRIMARY\_DIRECTORY和SECONDARY\_DIRECTORY到beta3里没有了，然后我们发现使用MediaStore.MediaColumns.RELATIVE\_PATH也可以控制写文件到共享目录的位置，甚至可以创建多级目录,不清楚google 后续会不会改动。

# 3.2 跨版本问题

问题：用户有可能一直不升级Q，等到Q的下一个版本出来,再直接跨过Q升级。

方案：跟系统同事邹小龙确认过，理论上不会出现这种情况。

# 3.3 升级时机问题

###  3.3.1 一次性升级完成:

在兼容模式下,第一次启动相册,弹出一个loading框,一次性将数据需要升级的数据升级完成,比如:将回收站用移动到data/data 目录下,升级全部完成后才能使用相册.

### 3.3.2 分批升级:

1. 回收站:在Q上的兼容模式,第一次使用相册的时候开始做升级,设置一个标志位,在onResume中使用RecycleDataManager.needAdjust时通过该标志位来判断是否调用该方法.
2. 云服务缩图/共享相册 第一次启动云服务的时候,下载图片到/data/data/packName/目录之后通过FileProvider传给相册,然后相册将该图片保存到沙箱中.
3. Bitmap缩图缓存: 在Q上的兼容模式,第一次使用相册的时候开始做升级

## 3.4 代码兼容问题

因为设置是否沙箱模式使用manifest文件里的属性,无法用代码动态设置,本来需要拉两个分支,但是

1.Q的新机器,默认是沙箱模式,使用适配Q的代码，各个接口进行升级，例如数据库的结构升级.

2.从P升级到Q,相册默认是兼容模式,兼容模式下能够访问所有的目录,才能够将老数据(文件)升级.

相册target是Q,就能使用Q新行为的接口,不被兼容模式影响.但是由于兼容模式的原因,比如相册在查询媒体库时,返回的\_data是真实的绝对路径,而不是沙箱模式下的类似content uri的东西,要考虑有没有这些场景需要适配.

## 3.5 媒体库会扫描沙箱

在沙箱模式下:媒体库可以扫描所有三方应用的沙箱,可以对里面的文件访问,修改,删除,但是不能往第三方应用的沙箱立面插入文件.

在兼容模式下:有可能存在第三方应用是沙箱模式,

## 3.6 外置T卡问题

兼容模式下需要把t卡/.ColorOSGalleryRecycler的文件移动到t卡的/Android/data/com.coloros.gallery3d/，所以需要media\_rw权限。这个需要系统组的人将这个权限移植到Q上,这个还没有得到系统组同事的明确答复.

## 3.7 性能问题

因为沙箱模式文件访问发生了变化，file接口不能使用，需要通过MediaStore（媒体库）和SAF（存储访问框架），但是MediaStore 接口和SAF接口比File接口慢很多。



## 3.8 SAF授权问题

1.通过SAF访问文件需要用户授权，但是用户可以自由选择授权的目录，我们只能操作用户选择的目录下的文件，访问其他文件需要重新申请权限。设计到很多流程的问题修改，比如批量操作文件的时候，有些文件有权限，有些文件没有权限.

2.外置卡插拔，权限有可能会丢失，丢失后又要重新授权。

3.SAF的授权页面（Document UI）和ColorOS 的设计风格不一样，UI很丑，但是google源代码不开放，

系统那边无法做修改，产品体验不好。

## 3.9 名单配置问题

相册的时间轴和图集都是通过名单控制，目前尚不清楚，微信，QQ，微博等这些三方应用保存媒体文件在沙盒模式下的目录，如果名单配置错误，则会引发图片在时间轴不显示，图集出现在其他中。容易引发用户投诉。

# 4 直接用兼容模式的风险

1.外置设备有卷名问题,数据来源

2.外置设备写权限的移植问题.

3.兼容模式下有可能三方应用是沙箱模式

4.Q的下一个版本兼容模式不可用

# 5 评审记录