

背景：

在设备没有完全启动的时候 想要访问android的文件系统。

报错如下：

```
06-16 13:03:54.938 6500 6500 E AndroidRuntime: java.lang.RuntimeException: Failed to create service com.android.server.location.LocationManagerService$Lifecycle: service constructor threw an exception
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.SystemServiceManager.startService(SystemServiceManager.java:161)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.SystemServer.startOtherServices(SystemServer.java:1687)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.SystemServer.run(SystemServer.java:634)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.SystemServer.main(SystemServer.java:440)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at java.lang.reflect.Method.invoke(Native Method)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:626)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:989)
06-16 13:03:54.938 6500 6500 E AndroidRuntime: Caused by: java.lang.reflect.InvocationTargetException
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at java.lang.reflect.Constructor.newInstance(Native Method)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at java.lang.reflect.Constructor.newInstance(Constructor.java:343)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.SystemServiceManager.startService(SystemServiceManager.java:150)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     ... 6 more
06-16 13:03:54.938 6500 6500 E AndroidRuntime: Caused by: java.lang.IllegalStateException: SharedPreferences in credential encrypted storage are not available until after user is unlocked
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at android.app.ContextImpl.getSharedPreferences(ContextImpl.java:584)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.location.SettingsHelper.<init>(SettingsHelper.java:134)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.location.LocationManagerService.<init>(LocationManagerService.java:386)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.location.LocationManagerService.<init>(LocationManagerService.java:148)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     at com.android.server.location.LocationManagerService$Lifecycle.<init>(LocationManagerService$Lifecycle.java:161)
06-16 13:03:54.938 6500 6500 E AndroidRuntime:     ... 9 more
```

知识如下：

REF: <https://developer.android.com/training/articles/direct-boot>

支持“直接启动”模式

当设备已开机但用户尚未解锁设备时，Android 7.0 将在安全的“直接启动”模式下运行。为支持此模式，系统为数据提供了两个存储位置：

- 凭据加密存储，这是默认存储位置，仅在用户解锁设备后可用。
- 设备加密存储，该存储位置在“直接启动”模式下和用户解锁设备后均可使用。

默认情况下，应用不会在“直接启动”模式下运行。如果您的应用需要在“直接启动”模式下执行操作，您可以注册应在此模式下运行的应用组件。需要在“直接启动”模式下运行的一些常见应用用例包括：

- 已安排通知的应用，如闹钟应用。
- 提供重要用户通知的应用，如短信应用。
- 提供无障碍服务的应用，如 Talkback。

如果应用在“直接启动”模式下运行时需要访问数据，请使用设备加密存储。设备加密存储包含使用密钥加密的数据，该密钥只有在设备成功执行启动时验证后才可用。

对于应使用与用户凭据（如 PIN 码或密码）关联的密钥加密的数据，请使用凭据加密存储。凭据加密存储仅在用户成功解锁设备后可用，直到用户再次重启设备。如果用户在解锁设备后启用锁定屏幕，则不会锁定凭据加密存储。

请求在“直接启动”模式下运行

应用必须先向系统注册其组件，然后才能在“直接启动”模式下运行或访问设备加密存储。应用通过将组件标记为加密感知来向系统注册。如需将您的组件标记为加密感知，请在清单中将

```
android:directBootAware 属性设为 true。
```

当设备重启后，加密感知组件可以注册以接收来自系统的

`ACTION_LOCKED_BOOT_COMPLETED` 广播消息。此时，设备加密存储可用，您的组件可以执行需要在“直接启动”模式下运行的任务，例如触发已设定的闹铃。

以下代码段示例说明了如何在应用清单中将

`BroadcastReceiver` 注册为加密感知并为

`ACTION_LOCKED_BOOT_COMPLETED` 添加 intent 过滤器：

在用户解锁设备后，所有组件均可访问设备加密存储和凭据加密存储。

访问设备加密存储

如需访问设备加密存储，请通过调用

`Context.createDeviceProtectedStorageContext()` 创建另一个

`Context` 实例。通过此上下文发出的所有存储 API 调用均访问设备加密存储。以下示例会访问设备加密存储并打开现有的应用数据文件：

请只将设备加密存储用于在“直接启动”模式下必须可以访问的信息。请勿将设备加密存储用作通用加密存储。对于私密用户信息，或在“直接启动”模式下不需要的加密数据，请使用凭据加密存储。

接收用户解锁通知

当用户在重启后解锁设备时，应用可以切换至访问凭据加密存储，并使用依赖用户凭据的常规系统服务。

为了在重启后用户解锁设备时收到通知，请从正在运行的组件注册

`BroadcastReceiver` 以监听解锁通知消息。在用户重启后解锁设备时：

- 如果应用具有需要立即获得通知的前台进程，请监听

`ACTION_USER_UNLOCKED` 消息。

- 如果应用仅使用可以对延迟通知执行操作的后台进程，请监听

`ACTION_BOOT_COMPLETED` 消息。

您可以通过调用

`userManager.isUserUnlocked()` 直接查询用户是否已解锁设备。

迁移现有数据

如果用户将其设备更新为使用“直接启动”模式，您可能需要将现有数据迁移到设备加密存储。使用

`Context.moveSharedPreferencesFrom()` 和

`Context.moveDatabaseFrom()` 可以在凭据加密存储与设备加密存储之间迁移偏好设置和数据库数据。

请自行判断要从凭据加密存储向设备加密存储迁移哪些数据。请勿将私密用户信息（如密码或授权令牌）迁移到设备加密存储。在某些情况下，您可能需要在这两种加密存储中管理单独的数据集。

测试加密感知应用

您可以在启用“直接启动”模式的情况下测试加密感知应用。可以通过两种方式启用“直接启动”模式：

注意：启用“直接启动”将擦除设备上的所有用户数据。

在安装 Android 7.0 的受支持设备上，您可以通过执行以下操作之一启用“直接启动”：

- 在设备上，如果您尚未启用**开发者选项**，请依次转到**设置 > 关于手机**，然后点按**版本号**七次，将其启用。开发者选项屏幕显示后，依次转到**设置 > 开发者选项**，然后选择**转换为文件加密**。
- 使用以下 adb shell 命令启用“直接启动”模式：

另外，如果您需要在测试设备上切换模式，还可以使用“直接启动”模拟模式。模拟模式只能在开发期间使用，可能会导致数据丢失。要启用“直接启动”模拟模式，请在设备上设置锁定模式，如果在设置锁定模式时系统提示使用安全启动屏幕，请选择“不用了”，然后使用以下 adb shell 命令：

要关闭“直接启动”模拟模式，请使用以下命令：

使用这些命令会导致设备重启。

检查设备政策加密状态

设备管理应用可以使用

`DevicePolicyManager.getStorageEncryptionStatus()` 检查设备目前的加密状态。如果应用面向的 API 级别低于 24.0 (Android 7.0)，且设备使用全盘加密或带“直接启动”的文件级加密，

`getStorageEncryptionStatus()` 将返回

`ENCRYPTION_STATUS_ACTIVE`。在这两种情况下，数据始终以闲时加密形式存储。如果应用面向的 API 级别为 24.0 或更高级别，且设备使用全盘加密，

`getStorageEncryptionStatus()` 将返回

`ENCRYPTION_STATUS_ACTIVE`。如果设备使用带“直接启动”的文件级加密，它将返回

`ENCRYPTION_STATUS_ACTIVE_PER_USER`。

如果面向 Android 7.0 构建设备管理应用，请务必同时检查

`ENCRYPTION_STATUS_ACTIVE` 和

`ENCRYPTION_STATUS_ACTIVE_PER_USER` 以确定设备是否已加密。

更多代码示例

[DirectBoot](#) 示例进一步演示了如何使用本页介绍的 API。