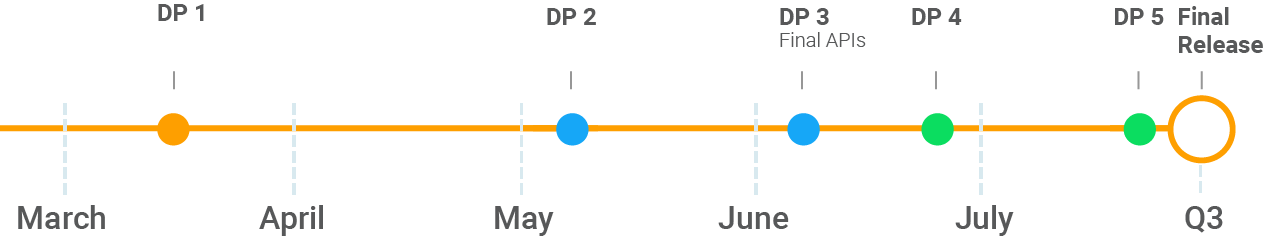
Google P本应用兼容性整改指导

# 背景说明

Android P版本中伴随很多机制和新增特性的改变，对自研以及第三方应用带来了很多兼容性问题。

本文档第一章主要介绍谷歌P版本开发环境搭建以及调试相关介绍；第2章节主要是对P版本兼容性现状一些摸底测试情况以及问题分类；第3章节主要是O版本一些特性介绍以及应用适配P版本的一些开发指导。

## Android P版本里程碑

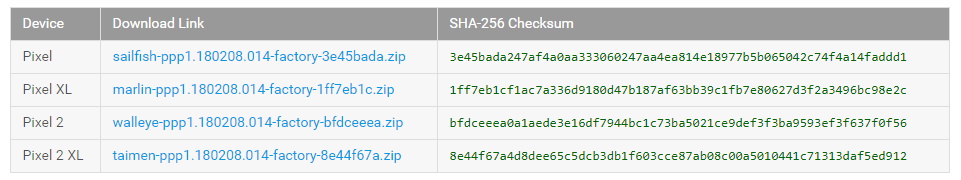


* [**Preview 1**](https://developer.android.com/preview/release-notes.html#dp1) (initial release, alpha)
* **Preview 2** (incremental update, beta)
* **Preview 3** (final APIs and official SDK, Play publishing, beta)
* **Preview 4** (release candidate for testing)
* **Preview 5** (release candidate for final testing)
* **Final release** to AOSP and ecosystem

## 刷机指导

### 版本下载

参考：<https://developer.android.com/preview/download.html>



### 刷机

刷手机的方法（**刷机前需要退出之前登录的谷歌账号**）：

1.下载对应设备的压缩包并解压；

2.手机连上USB后，执行adb reboot bootloader命令；

3.等手机出现小机器人以后，并且显示‘unlock’字样时，执行压缩包中的bat脚本。

如果是“lock”字样：

1.连续点击设置-版本号3次，打开开发人员选项；

2.在开发人员选项中打开OEM unlock；

3.连接手机到电脑，执行adb reboot bootloader命令到小机器人界面；

4.执行fastboot oem unlock，这样手机就解锁了，再使用上面的刷机方法即可。

## P版本开发环境搭建

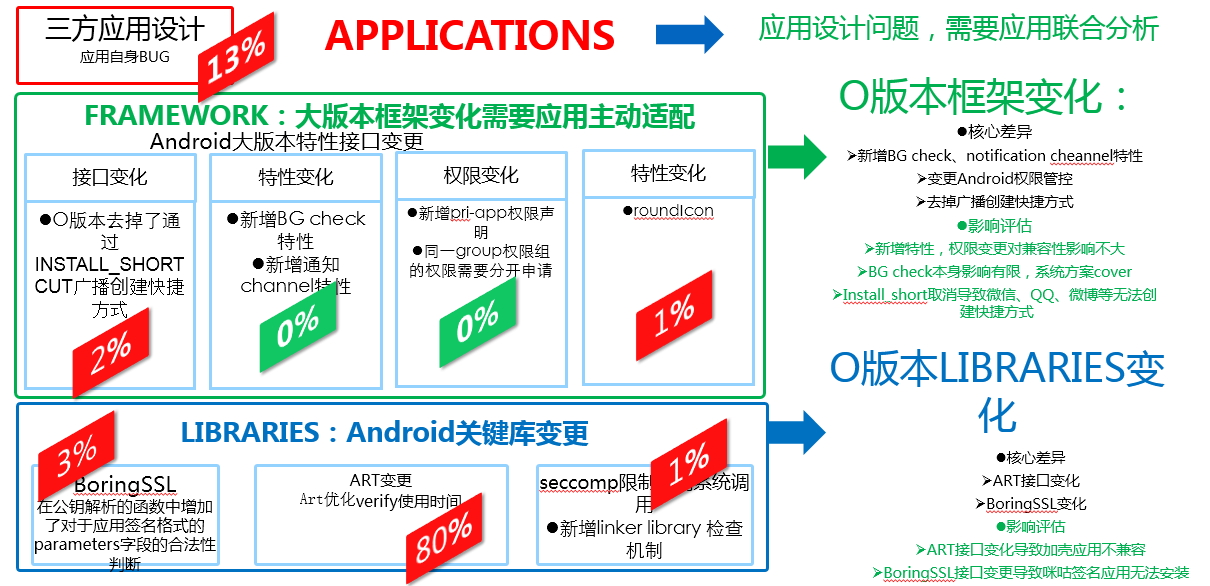
安装和配置AndroidP SDK和模拟器，参考：<https://developer.android.com/preview/setup-sdk.html>

## 华为终端开放实验室Android P 版本兼容性测试

开发者如果没有Pixcel真机并且不喜欢谷歌模拟器调试，也可以选择华为终端开放实验室提供的远程真机来进行P版本测试和调试。请参考链接：[华为终端开放实验室Android P 版本兼容性测试上线](https://mp.weixin.qq.com/s?__biz=MzI0MjgxMjU0Mg==&mid=2247485915&idx=2&sn=f6a970935f31db80bf0d3700ba383b66&chksm=e977dacade0053dc647f67a10c5904e5a7ad5966c43366c5ad1418a0db75fa0cdf23928e7043&mpshare=1&scene=1&srcid=0313yLliJWA6H7fG6l83uHeo&pass_ticket=%2FUI0xaZhCRGM7IbcxkLUzsv2InwEDumK3egORU9F3RT4FgzyM1ApMSHM41Me7qjb#rd)。

# 第三方应用兼容性测试情况

## O版本兼容性问题回顾



## P版本DP1兼容性摸底测试情况

基于谷歌P版本的DP1版本进行了国内Top1000兼容性摸底测试，测试对象：华为应用市场中各领域中热门的应用。详细情况可以参考链接：[P版本国内首份千款主流应用Android P版本兼容性测试报告发布](https://mp.weixin.qq.com/s?__biz=MzI0MjgxMjU0Mg==&mid=2247485977&idx=1&sn=20273db9218f0e88814aaaf1f8b5a90a&chksm=e977d908de00501efc835e6fc07e7c9c5f5aef80bd021d49a063e6570f5857efd997acfd617b&mpshare=1&scene=1&srcid=0316ZVHGpmGrv4mTtvk75Mom&pass_ticket=%2FUI0xaZhCRGM7IbcxkLUzsv2InwEDumK3egORU9F3RT4FgzyM1ApMSHM41Me7qjb#rd)。

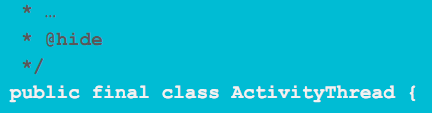
# 兼容性分析

## Restrictions on non-SDK interfaces

### 背景介绍

non-SDK interfaces的定义

以@hide标记的类/方法/属性



应用滥用non-SDK interfaces的危害

这些non-SDK接口在大版本之间的变化可能很频繁，带来兼容性问题。

影响范围：

三方应用都会受到该特性影响。

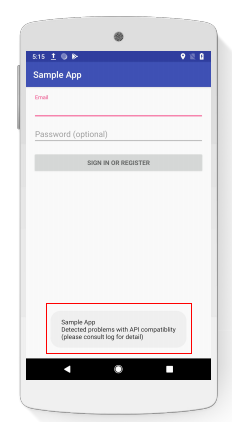
解决办法

* 三方整改；
* 必须要调用的non-SDK接口可以向谷歌申诉把接口加入到灰名单：https://[b.corp.google.com/hotlists/825416](https://b.corp.google.com/hotlists/825416)；（申诉结果是否通过完全由谷歌审核决定）

名单分类

* Light grey list: targetSDK>=P时，警告；
* Dark grey list: targetSDK<P时，警告；>=p时，不允许调用；
* Black list:三方应用不允许调用；

注：在发货版本上不会有警告。



non-SDK interfaces名单列表

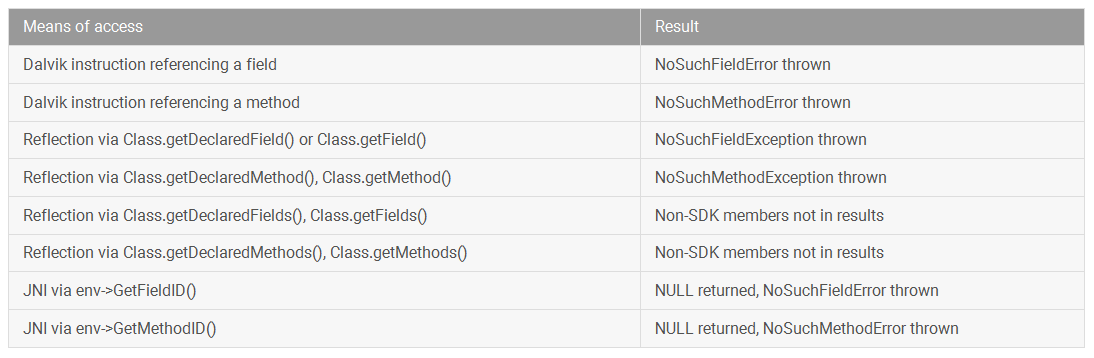
触发警告的non-SDK接口会被上报至谷歌，以便进一步分析判断，调整接口名单的内容。4月份前都有调整的机会，6月份定稿。谷歌将于近日给出初始的接口名单，且有申诉渠道，有可能由此将特定non-SDK接口加入gray list，DP1的时候，谷歌发布black list，OEM进行评估，同时谷歌测试top1000；

API列表查看：<https://android.googlesource.com/platform/frameworks/base/+/master/config/>



### 兼容性影响

三方应用的热补丁、加壳方案、调用non-SDK接口的所有三方应用可能会受到影响；在Developer Preview的后续版本中，访问non-SDK接口的各种方法都会产生错误或其他不良结果。下表提供了有关访问方式及其各自结果的详细信息。



需要三方应用排查non-SDK接口的调用，并提前整改，否则可能现在或者是将来会出现兼容性问题。

### 适配指导

1. 通过日志，找出应用调用的所有non-SDK接口名单，关键日志：

Accessing hidden field Landroid/os/Message;->flags:I (light greylist, JNI)

Accessing hidden method Landroid/app/ActivityThread;->currentActivityThread()Landroid/app/ActivityThread; (dark greylist, reflection)

1. 对于已经被禁止调用的接口需要整改，寻找替换该non-SDK接口的方案；
2. 如果必须要调用的non-SDK接口可以向谷歌申诉把接口加入到灰名单：https://[b.corp.google.com/hotlists/825416](https://b.corp.google.com/hotlists/825416)；

参考谷歌指导：<https://developer.android.com/preview/restrictions-non-sdk-interfaces.html>

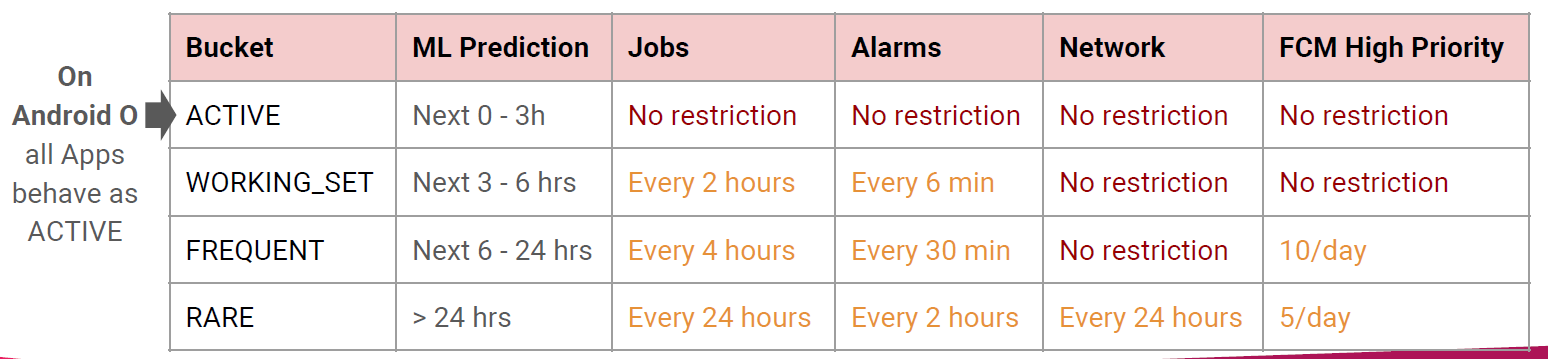
## Battery Improvements

### 背景介绍

谷歌在P版本之前没有一个完整的功耗解决方案，OEM厂商分别开发各自的功耗方案，管控手段都包括了清理应用，功耗得到优化，但是同时也影响了三方应用的一些功能正常使用，谷歌为了解决这个问题在P版本提出了自己的功耗解决方案，该方案主要包含：

1. AAB（Auto Awesome Battery)：

* 通过ML算法将应用进行分类，不同类型的应用功耗管控策略不一样
* Firebase Cloud Messaging (FCM):管控三方消息接收的频率
* 谷歌提供了统一的应用的管控方法：Forced App Standby (FAS)，谷歌不会通过清理应用来优化功耗



1. Extreme Battery Saver（EBS）谷歌超级省电模式；
2. Smart screen brightness：屏幕亮度调节优化算法。

### 兼容性影响

谷歌功耗方案对三方应用各种管控，存在导致应用后台功能无法正常使用的可能，特别是：IM、邮箱、闹钟、音乐（直播）、地图导航、运动健康、下载、日历等应用影响比较大。目前通过谷歌提供的调试命令验证：所有的应用都有可能会被分到管控的类型，对三方的后台功能是有影响的。

### 适配指导

如何判断是不是谷歌AAB方案导致的问题？谷歌提供了调试命令，可以获取应用所处的管控分类，并且可以让某个应用进入指定的分类进行管控，测试验证对应管控策略下的应用行为。

1. Unplug (or adb shell dumpsys battery unplug )

2. adb shell am get-standby-bucket <package name> ; //获取某个指定应用分类数据（5-50）

5 EXEMPT/WHITELISTED

10 ACTIVE

20 WORKING\_SET

30 FREQUENT

40 RARE

50 NEVER

3. adb shell am set-standby-bucket <package name> <bucket>//强制把某个应用设置到指定的类别

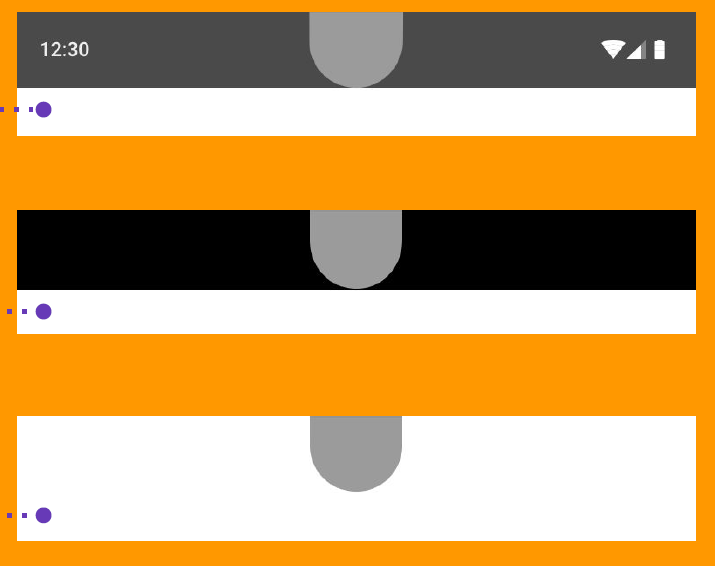
4. Programmatic: UsageStatsManager::getAppStandbyBucket()

## 挖孔屏适配

### 背景介绍

谷歌P版本提供了统一的挖孔屏方案和三方适配挖孔屏方案：

* 对于有状态栏的页面，不会受到挖孔屏特性的影响；
* 全屏显示的页面，系统挖孔屏方案会对应用界面做下移避开挖孔区显示；
* 已经适配的P的应用的全屏页面可以通过谷歌提供的适配方案使用挖孔区，真正做到全屏显示。



P版本三方适配挖孔屏方案：

* 新增挖孔屏挖孔尺寸和位置接口

class WindowInsets {

DisplayCutout getDisplayCutout();

}

class DisplayCutout {

int getSafeInsetLeft();

int getSafeInsetTop();

int getSafeInsetRight();

int getSafeInsetBottom();

Region getBounds();

}

* 新窗口布局模式，允许应用程序请求是否在挖孔区域布局：

class WindowManager.LayoutParams {

int layoutInDisplayCutoutMode;

final int LAYOUT\_IN\_DISPLAY\_CUTOUT\_MODE\_DEFAULT;

final int LAYOUT\_IN\_DISPLAY\_CUTOUT\_MODE\_ALWAYS;

final int LAYOUT\_IN\_DISPLAY\_CUTOUT\_MODE\_NEVER;

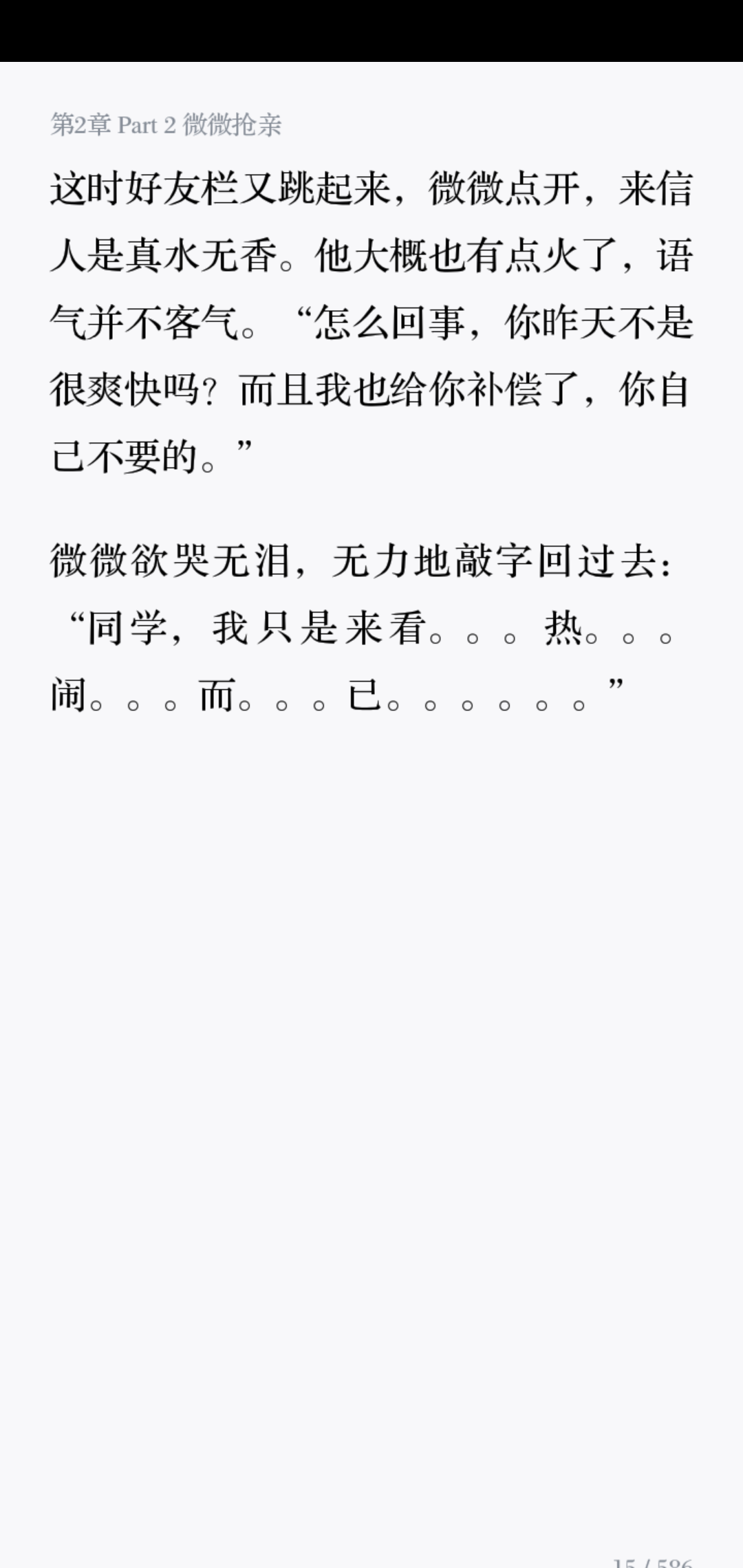
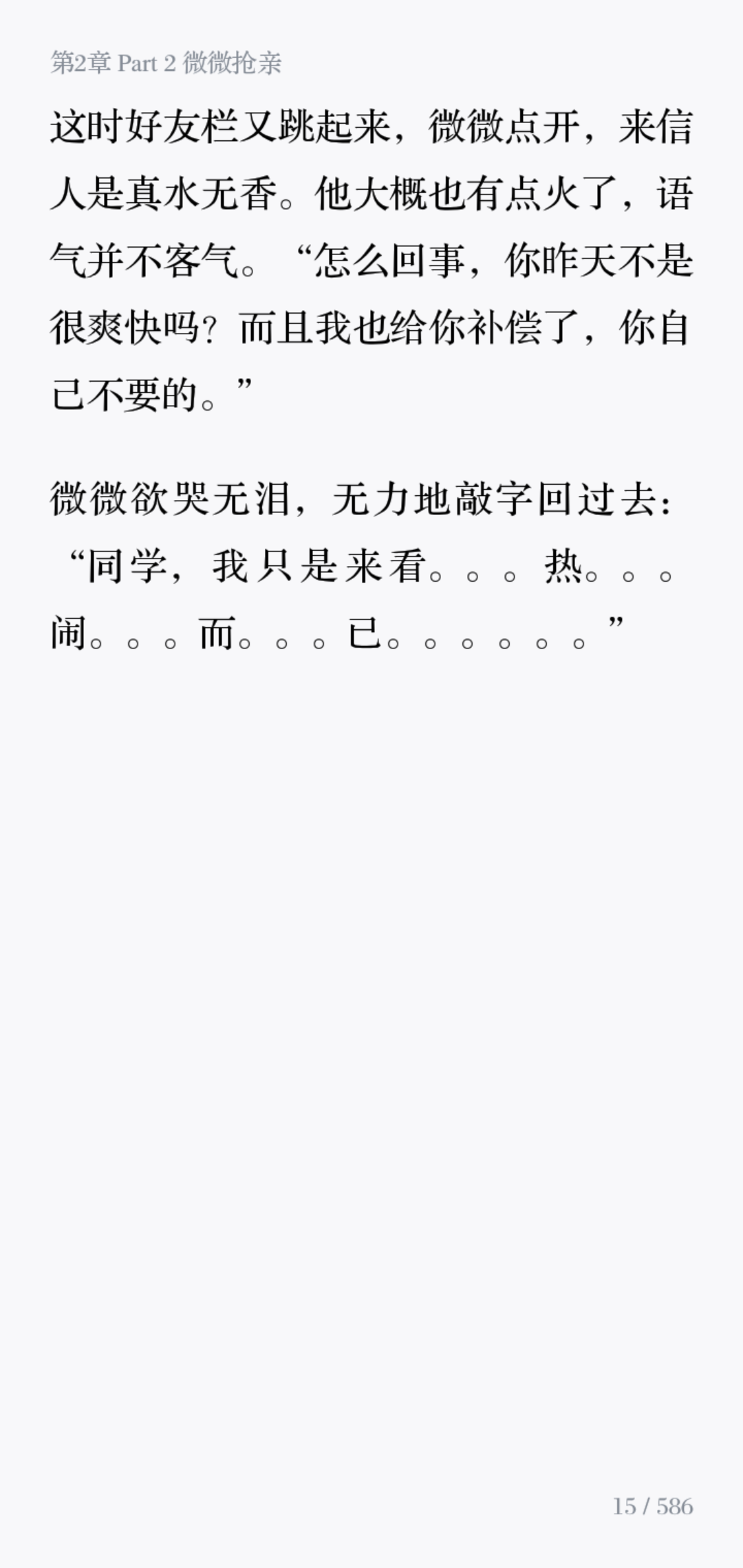
}

layoutInDisplayCutoutMode值说明：

1. LAYOUT\_IN\_DISPLAY\_CUTOUT\_MODE\_DEFAULT：默认情况下，全屏窗口不会使用到挖孔区域，非全屏窗口可正常使用挖孔区域。
2. LAYOUT\_IN\_DISPLAY\_CUTOUT\_MODE\_ALWAYS：窗口声明使用挖孔区域
3. LAYOUT\_IN\_DISPLAY\_CUTOUT\_MODE\_NEVER：窗口声明不使用挖孔区域

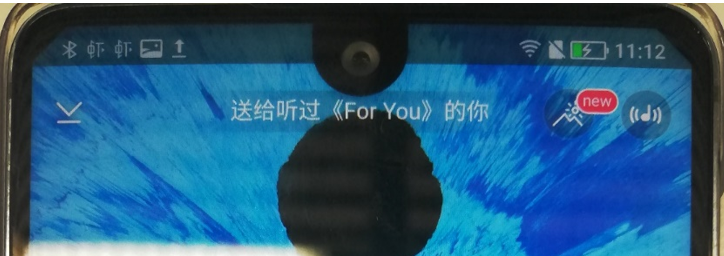
### 兼容性影响

系统下移方案导致布局问题（截断、错乱，按钮热区错位）；

小说页码被截断问题

状态栏高度写死问题；



状态栏背景高度写死问题

沉浸式布局遮挡问题；



搜索框被遮挡问题

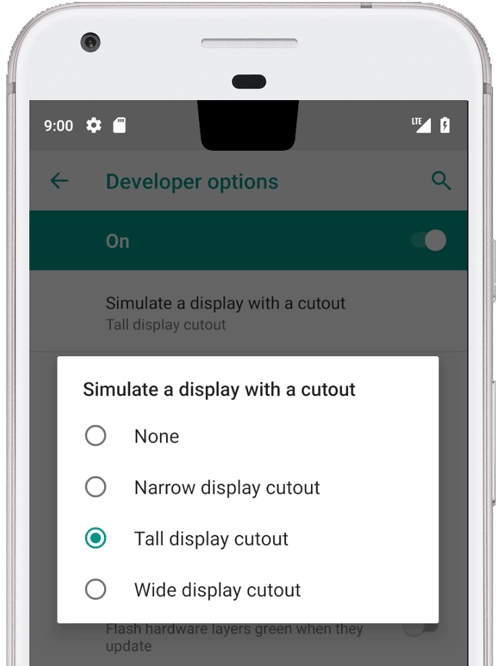
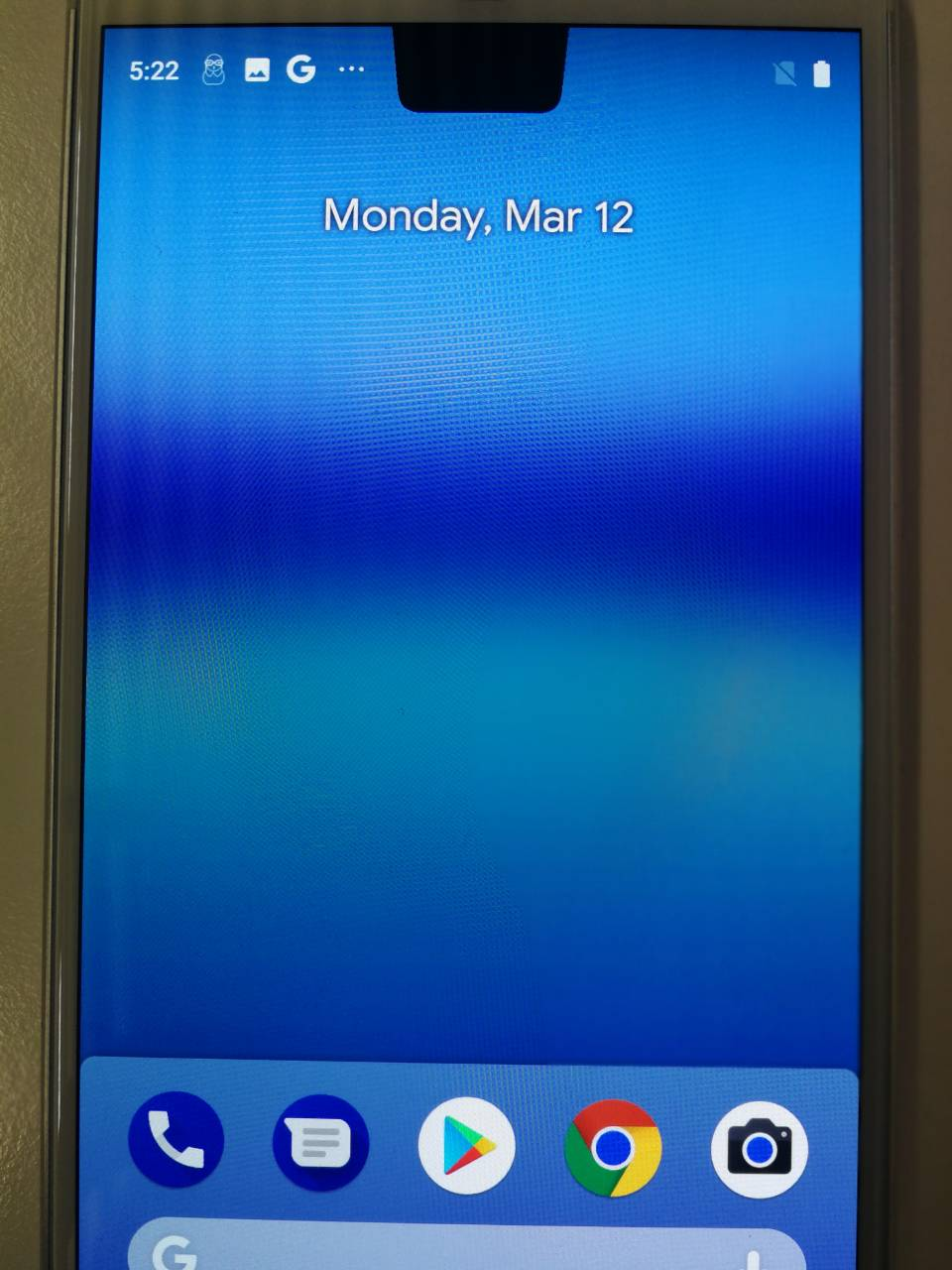


视频内容被遮挡问题

### 适配指导

在非挖孔屏P版本手机可以开启模拟挖孔屏调试的功能：

* 在开发人员选项屏幕中，向下滚动到绘图部分，然后点击“模拟具有凹口的显示屏”设置项；
* 选择挖孔尺寸信息；

在挖孔屏调试打开之后，浏览应用的所有页面，测试所有遮挡问题，或者是下移导致的问题，对有问题的页面进行布局适配，适配方案：

可以通过谷歌提供的适配方案，使用挖孔区全屏显示解决：

WindowManager.LayoutParams lp = getWindow().getAttributes();  
lp.**layoutInDisplayCutoutMode** = WindowManager.LayoutParams.***LAYOUT\_IN\_DISPLAY\_CUTOUT\_MODE\_ALWAYS***;  
getWindow().setAttributes(lp);

布局调整，建议布局调整策略：

应用页面背景可以充满整个屏幕显示，控件和文字等关键信息布局在状态栏以外的区域显示，保证关键信息不会出现遮挡（谷歌要求：状态栏高度和挖孔高度要保持一致），需要用到的接口：

1） 获取挖孔尺寸信息接口，具体可以参考：3.3.1章节；

2） 获取系统状态栏高度接口：

public static int getStatusBarHeight(Context context) {

int result = 0;

int resourceId = context.getResources().getIdentifier("status\_bar\_height", "dimen", "android");

if (resourceId > 0) {

result = context.getResources().getDimensionPixelSize(resourceId);

}

return result;

}

P版本适配工作：谷歌提供的适配方案要求应用必须要适配到P版本才能使用，所以如果应用需要适配挖孔屏，还需要验证应用适配到P版本的其他兼容性问题并解决，可参考：<https://developer.android.com/preview/migration.html>。

## 蓝牙后台扫描管控

### 背景介绍

特性介绍

减少功耗，减少后台占用CPU，提供更加智能的扫描策略来降低功耗，对后台应用、灭屏场景下的扫描作限制，提供更加低功耗的扫描间隔和窗口。

影响APP范围：没有设置过滤条件的所有应用

private void startScan(List<ScanFilter> filters, ScanSettings settings,  
            final ScanCallback callback, List<List<ResultStorageDescriptor>> resultStorages)

第一个参数(ScanFilter)：筛选条件,可以通过设置过滤器的mDeviceName、mDeviceAddress、mServiceUuid等作为过滤条件进行过滤。也就是如果第一个参数传入的是null，该应用会受该特性影响。

管控措施：

后台：降低扫描的空占比；

灭屏：禁止蓝牙扫描。

### 兼容性影响

影响应用后台蓝牙扫描功能。

### 适配指导

发起蓝牙扫描的时候添加过滤条件。

## Camera、sensor和麦克风后台管控

### 背景介绍

增加Camera、sensor和麦克风的background限制，对所有SDK都生效；限制sensor event向后台应用或服务的传输，对前台应用无影响；主要是考虑隐私原因，功耗不是主要动机；对应用来说是很大的变化。

### 兼容性影响

影响范围：

应用处于idle和gone的状态就会被管控，切换到后台一分钟应用就会进入idle状态，但是有前台服务的不会进入idle状态；

影响三方功能：后台录音，拍照、摄像、定位和计步。

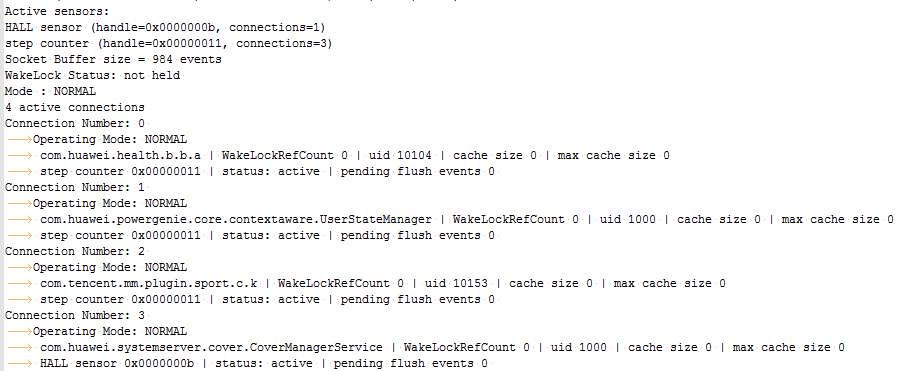
### 适配指导

问题定位，确认是不是该特性管控导致的：

* 查看应用状态：adb shell dumpsys activity p com.sina.weibo ，然后搜索UID states查看UID的状态



* 查看应用有没有注册监听sensor情况：adb shell dumpsys sensorservice



适配方案：需要后台访问麦克风、sensor和camera的时候增加前台服务。

如果满足以下任意条件，应用将被视为处于前台：

* 具有可见 Activity（不管该 Activity 已启动还是已暂停）。
* 具有前台服务。
* 另一个前台应用已关联到该应用（不管是通过绑定到其中一个服务，还是通过使用其中一个内容提供程序）。 例如，如果另一个应用绑定到该应用的服务，那么该应用处于前台：
* IME
* 壁纸服务
* 通知侦听器
* 语音或文本服务

适配指导：<https://developer.android.com/about/versions/oreo/background.html?hl=zh-cn>

前台服务：<https://developer.android.com/guide/components/services.html#Foreground>

## 更严格的Seccomp filter

### 背景介绍

大量的系统调用暴露在用户空间，但其实在程序的整个生命周期内并没有使用，增加了系统安全的攻击面。而随着系统调用的改变和成熟，可以产生出一组尽量少的系统调用暴露在用户空间。Seccomp-bpf就是在应用程序的使用中，限制只能调用有限指定的系统调用。Seccomp过滤器提给了一种手段，为一个进程调用系统调用时指定了过滤器，而这个过滤器则是BPF。

### 兼容性影响

谷哥对zygote进程设置了seccomp filter，使的app进程仅能够通过bionic库调用系统调用。其他系统调用将会被系统拦截，可能会导致一些应用闪退兼容性问题，如果应用调用了不允许的系统调用，会接收到SIGKILL信号；P版本比O版本限制更严格，在P版本同样存在兼容性问题的风险。

### 适配指导

app进程通过bionic库调用systemcalls：

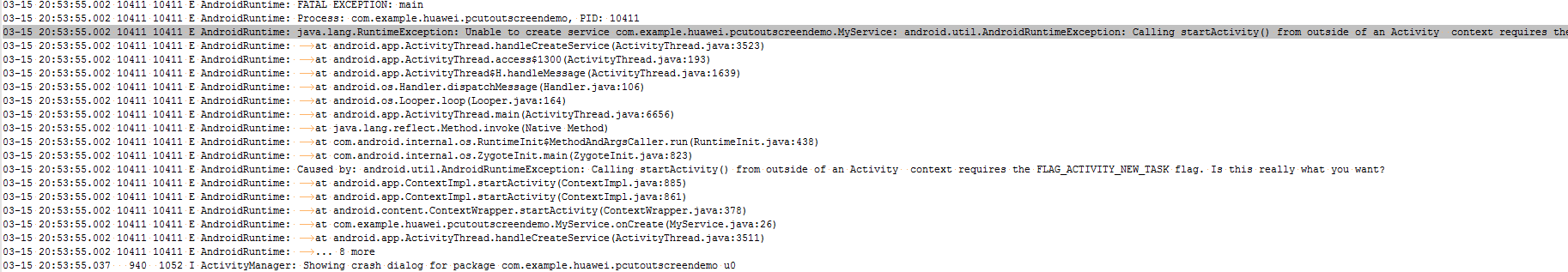
<https://android.googlesource.com/platform/bionic/>

## FLAG\_ACTIVITY\_NEW\_TASK被强制要求

在P版本，如果不在Intent添加FLAG\_ACTIVITY\_NEW\_TASK，将无法通过非Activity的Context启动一个Activity，并且会抛异常。

比如在Service中启动Activity，如果Intent不添加FLAG\_ACTIVITY\_NEW\_TASK，就会抛异常：

@Override  
 **public void** onCreate() {  
 Log.*v*(***TAG***, **"ServiceDemo onCreate"**);  
 **super**.onCreate();  
 Intent intent = **new** Intent(**this**, Main2Activity.**class**);  
*// intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);* startActivity(intent);  
 }



## 应用禁止访问xt\_qtaguid目录的文件

对应用程序不再允许直接读取/ proc / net / xt\_qtaguid文件夹中的文件。原因是为了确保和运行Android P的设备保持一致，这些设备根本没有这些文件。

依赖这些文件的公共API，TrafficStats和NetworkStatsManager继续按预期工作。但是，不支持的cutils函数（如qtaguid\_tagSocket()）可能无法按预期方式工作，或者根本无法在不同的设备上工作。

## Crypto provider在P版本去掉了

从P版本开始，Crypto JCA provider被去掉了，调用SecureRandom.getInstance(“SHA1PRNG”, “Crypto”) 将会报NoSuchProviderException。

## 前台服务权限

应用如果想使用前台服务需要申请FOREGROUND\_SERVICE权限，这个权限是普通权限，如果不申请权限就直接启动前台服务会抛SecurityException。

## 去除通过Build.serial获取设备识别码

应用需要验证手机序列号必须要申请READ\_PHONE\_STATE权限，然后通过P版本新增的接口Build.getSerial() 来获取：

通过Build.SERIAL获取不到真实数据：Build.serial:unknown

通过Build.getSerial()才能获取真实的数据，需要用户授权READ\_PHONE\_STATE权限

该特性只影响已经适配P的应用，也就是targetSDK Version>=P。

## 不允许共享WebView数据目录

应用程序不能再跨进程共享单个WebView数据目录。如果您的应用有多个使用WebView，CookieManager或android.webkit包中的其他API的进程，则当第二个进程调用WebView方法时，您的应用将崩溃。

该特性只影响已经适配P的应用，也就是targetSDK Version>=P。

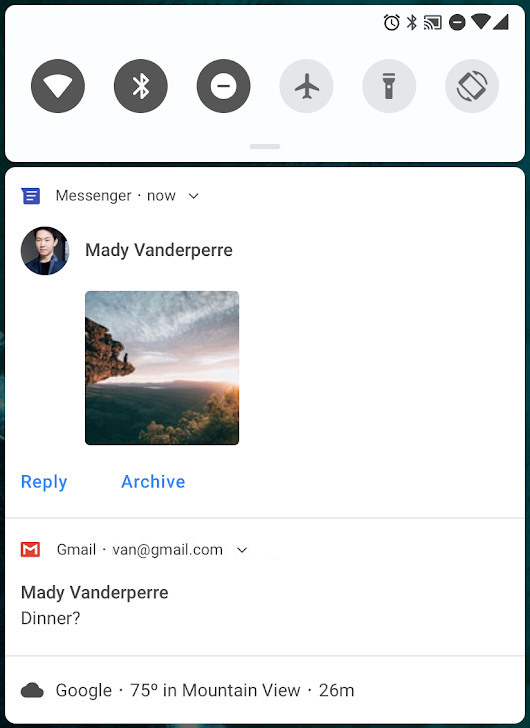
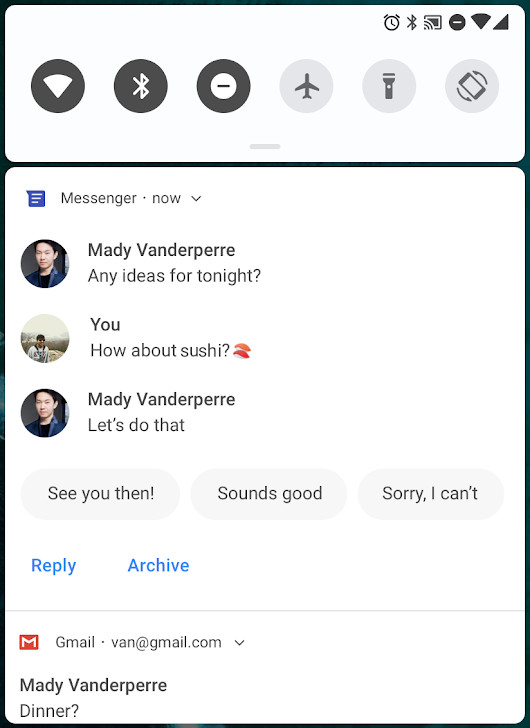
## Notifications

### 背景介绍

增强的消息体验：

从Android 7.0开始，你可以增加一个action以对消息或从通知直接进入文字的行为作出反应。Android P有如下增强:

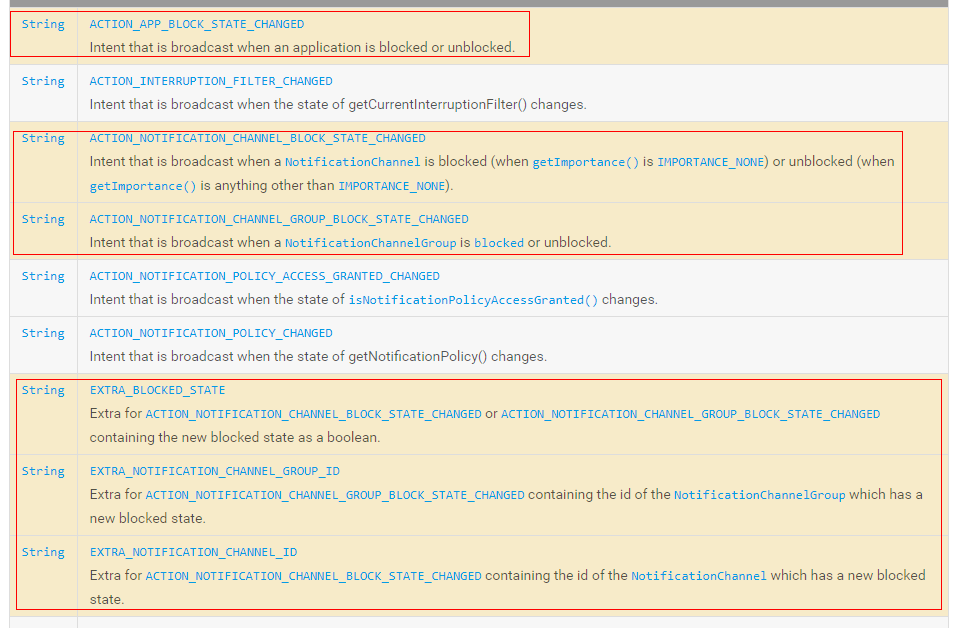
* 支持图像: Android P现在可在消息中展示图像，通过在消息中setData()即可实现此功能;
* 为了便捷性而做出的简化支持: 新的Notification.Person类用于识别对话中的人物，包含他们的头像和URI. 很多其他的API,例如addMessage()，现在利用Person类而不是CharSequence;
* 保存回复为草稿:当用户不小心关闭消息通知时，你的app可以获取由系统发出的EXTRAREMOTEINPUT\_DRAFT。你可以使用这个extra值来在app中进行预填充，从而使用户可以更快地完成回复
* 识别是否对话为群聊:你可以使用setGroupConversation()来识别一个对话是否为群聊
* 为Intent设置语义动作:setSemanticAction()方法可以让开发者给一个行为增加语义，如标记为读，删除，回复等等
* 智能回复:Android P支持相同的建议性回复。使用RemoteIntent.setChoices()来提供一组标准回复给用户

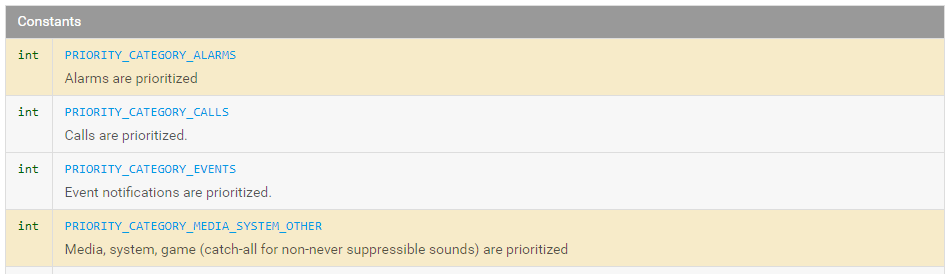
Channel设置，广播和免打扰

Android O引入了通知Channel的概念，从而 可以让开发者为每类通知创建自定义的channel. Android P通过如下变化简化了通知channel设置：

* 限制channel组：用户现在可以通过对一个app的通知设置来限制整个channel的通知组。你可以使用isBlocked()方法来判断一个通知组是否被限制，从而，不再发送通知到那个组的channel中。此外，你可通过getNotificationChannelGroup()方法获取当前channel组的设置
* 新的广播Intent类型:当通知channel以及channel组的阻塞状态发生变化时，Android系统会发送广播。被阻塞通知channel的app可以监听这些intent并作出相应的反应。

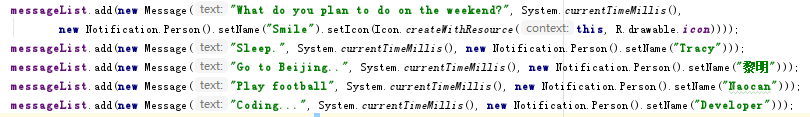


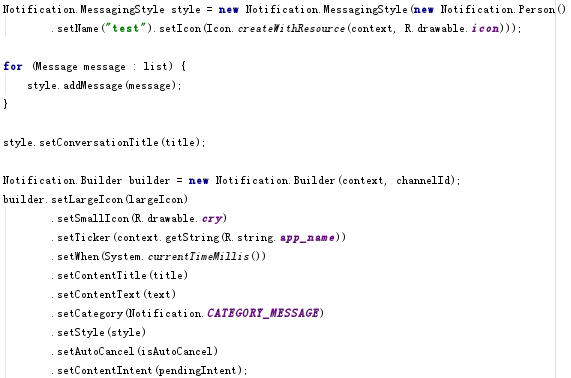
* 新的不打扰类别: NotificationManager.Policy有两个策略常量: PRIORITY\_CATEGORYALARMS(闹铃优先)和PRIORITY\_CATEGORY\_MEDIA\_SYSTEM\_OTHER(媒体，系统和游戏声音优先)



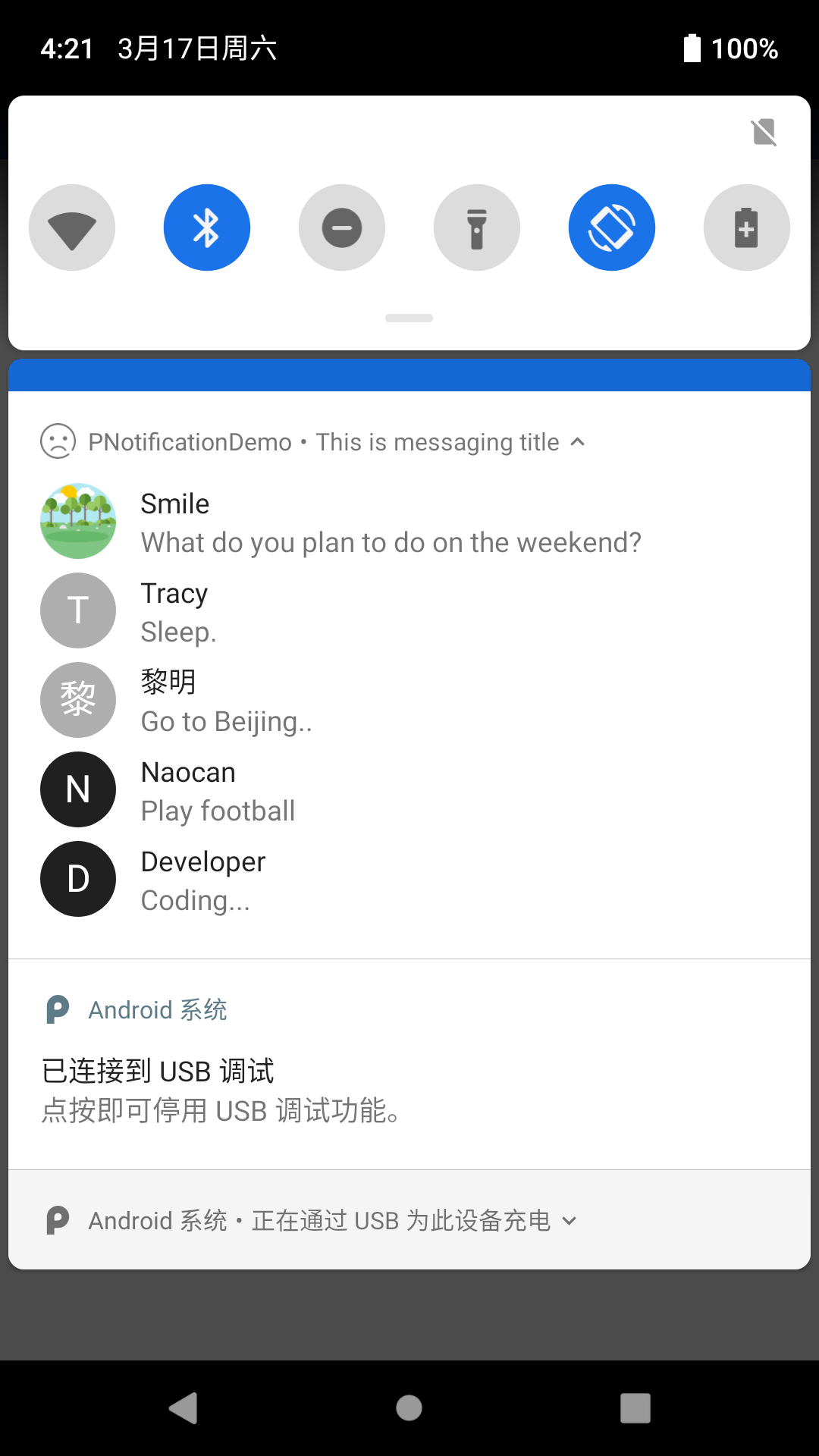
### 适配指导

使用Person（P版本新增）显示发消息用户头像：





显示效果：



### 兼容性影响

新特性，对三方兼容性无影响，可以增强IM应用的通知体验。

# 参考文献

1. Google P版本的P-preview SDK已经发布, P-preview的SDK下载地址: https://developer.android.com/preview/overview.html

2. Google P版本的P-preview source code地址： https://android.googlesource.com/platform/frameworks/base/+/android-p-preview-1

3. Google P版本的P-preview source code下载方法： https://source.android.com/source/downloading.html

4. 目前Pixel、Pixel XL、Pixel 2和Pixcel 2 XL可以升级P-preview版本，下载地址如下： https://developer.android.com/preview/download.html

5. Google P版本版本计划：https://developer.android.com/preview/migration.html

6. Google P版本新特性和主要行为变更：https://developer.android.com/preview/features.html

https://developer.android.com/preview/behavior-changes.html