

百度 APP 自动升级 SDK For Android 参考手册

V1.3.1

2017-11-06

时间	版本	更新说明
2015-05-25	1.0.0	创建
2015-06-05	1.2.0	增加百度助手升级接口
2016-8-23	1.2.0.1	增加对 Android 6.0 系统的支持
2017-5-8	1.3.0	增加对 Android 7.0 系统的支持； 提供的升级方式优化：默认 UI 更新、自定义 UI 更新、静默更新；
2017-11-6	1.3.1	增加对 https 下载的支持

目录

1	接入准备.....	4
1.1	环境.....	4
1.2	参数.....	4
2	API 接入说明	4
2.1	SDK 配置	4
2.2	版本检测.....	4
2.3	增量更新.....	5
2.4	其他.....	5
3	API 功能介绍	5
3.1	默认 UI 更新	5
3.2	自定义 UI 更新	6
3.2.1	版本检测.....	6
3.2.2	新版本下载.....	8
3.2.3	新版本安装.....	9
3.3	无 UI 更新	9
3.4	通过文件地址安装.....	9
	此接口被调用时会安装传入路径的文件，支持 7.0fileprovider。	9
	接口：	9
4	接入场景及接入流程.....	9
4.1	默认 UI 更新	9
4.2	自定义 UI 更新	11
4.3	静默更新.....	12

1 接入准备

1.1 环境

- 当您下载完开发包后，解压出的 autoupdatesdk-release.aar 即百度 APP 自动升级 SDK 开发包。
- 开发包支持在 Android SDK 4.0.3 及以上固件版本运行，建议使用 Android SDK 7.1.1 以上版本编译。

1.2 参数

在接入 SDK 前请先向开发者平台申请：

- AppId：应用 AppId，SDK 配置中用到，详情查看 [SDK 配置](#)。
- AppKey：应用 AppKey，SDK 配置中用到，详情查看 [SDK 配置](#)。

2 API 接入说明

2.1 SDK 配置

请将 AndroidManifest.xml 中的 application 节点下增加以下节点：

```
<meta-data android:name="BDAPPID" android:value="你的 appid"/>
<meta-data android:name="BDAPPKEY" android:value="你的 appkey"/>
```

其中 metadata 中需要填写的 AppId 和 AppKey，请向开发者平台申请。

2.2 版本检测

本 SDK 检测是否有新版本的判断依据是：如果您当前 APP 的版本号比其当前上传在服务端的最新版本号低，则说明您当前的 APP 不是最新版本，可通过本 SDK 相关 API 下载最新的 APP。本 SDK 是通过您的 APP 的 AndroidManifest.xml 的 android:versionCode 来获取 APP 的版本号。

2.3 增量更新

本 SDK 支持应用安装包增量更新。在应用进行版本更新时，SDK 会优先采用增量更新方式：即下载安装包变更部分的差分包到本地，并合成新版本的安装包，整个下载及差分包合成过程由 SDK 完成，无需开发者进行额外开发。

开发者无需单独制作增量差分包，将完整安装包上传到开发者平台即可，开发者平台自动会为安装包生成对应的差分包。

增量更新能够为用户节省下载流量，促进应用版本收敛。

2.4 其他

如果开发使用自定义 UI 更新，可以考虑将/res 目录下除 values 目录的其他目录删除来降低包的大小。

https 使用后的好处和坏处：好处是不易被劫持，坏处是下载失败率可能会微升，大约提升 1% 的失败率。

3 API 功能介绍

3.1 默认 UI 更新

建议未设计升级提示界面的应用使用，此接口被调用时会进行版本检测，若有新版本会弹出对话框提示用户。

接口：

```
static void uiUpdateAction(Context context, UICheckUpdateCallback callback, Boolean useHttps)
```

其中回调接口 UICheckUpdateCallback 需实现的方法：

方法名	描述
<code>void onCheckComplete ()</code>	当检测到无版本更新或者用户关闭版本更新提示框或者用户点击了升级下载时会触发回调该方法
<code>void onNoUpdateFound()</code>	当检测到无版本更新时会触发回调该方法

3.2 自定义 UI 更新

建议已有升级提示界面的应用使用，开发者可以通过自定义 UI 更新相关数据接口自己实现更新的相关提示界面。

3.2.1 版本检测

此接口用于查询当前服务端是否有新版本，有的话取回新版本信息。

接口：

```
static void cpUpdateCheck(Context context, CPCheckUpdateCallback callback, Boolean useHttps)
```

其中回调接口 `CPCheckUpdateCallback` 需实现的方法：

```
void onCheckUpdateCallback(AppUpdateInfo info, AppUpdateInfoForInstall infoForInstall)
```

- `AppUpdateInfo`：服务端新版本信息。
- `AppUpdateInfoForInstall`：本地已经下载完成可安装的最新版本信息。

接口参数回调场景：

场景	参数	是否有值	当前 APP 是否是最新版本	场景处理
场景一	<code>AppUpdateInfo</code>	不为空	否	开发者可直接通过 <code>AppUpdateInfoForInstall</code> 里的内容进行新版本提示及安装
	<code>AppUpdateInfoForInstall</code>	不为空		
场景二	<code>AppUpdateInfo</code>	不为空	否	开发者可直接通过 <code>AppUpdateInfo</code> 里的内容进行新版本提示，并将 <code>AppUpdateInfo</code> 传入 新版本下载接口 进行新版本下载
	<code>AppUpdateInfoForInstall</code>	为空		
场景三	<code>AppUpdateInfo</code>	为空	否	开发者可直接通过 <code>AppUpdateInfoForInstall</code> 里
	<code>AppUpdateInfoForInstall</code>	不为空		

	Install			的内容进行新版本提示及安装
场景四	AppUpdateInfo	为空	是	当前已经是最新版本，无需处理
	AppUpdateInfoFor	为空		
	Install			

AppUpdateInfo:

字段	类型	描述
appName	String	应用名称
appVersionName	String	应用版本名称
appPackage	String	应用包标识符
appVersionCode	int	应用版本号
appUrl	String	应用全量更新包下载地址（开发者无需关注）
appSize	long	应用全量更新包大小（开发者无需关注）
appPath	String	应用增量更新包下载地址，当 <code>appPath</code> 和 <code>appPathSize</code> 不为空时， <code>appUrl</code> 和 <code>appSize</code> 也不为空
appPathSize	long	应用增量更新包大小，当 <code>appPath</code> 和 <code>appPathSize</code> 不为空时， <code>appUrl</code> 和 <code>appSize</code> 也不为空
appIconUrl	String	目前没有用到，属于保留字段
appChangeLog	String	新版本更新信息
appMd5	String	目前没有用到，属于保留字段
forceUpdate	int	1 为强制更新，不可关闭

AppUpdateInfoForInstall:

字段	类型	描述
----	----	----

<code>appSName</code>	String	应用名称
<code>appVersionName</code>	String	应用版本名称
<code>appChangeLog</code>	String	新版本更新信息
<code>installPath</code>	String	新版本 APK 包的本地路径

3.2.2 新版本下载

此接口用于下载当前服务端最新版本，并回调下载进度和状态。接口中的 `AppUpdateInfo` 参数从[版本检测](#)接口中获取。

接口：

```
static void cpUpdateDownload(Context context, AppUpdateInfo info,
    CPUUpdateDownloadCallback callback)
```

其中回调接口 `CPUUpdateDownloadCallback` 需实现的方法：

方法名	参数	描述	是否必调
<code>void onStart()</code>	无	当 <code>cpUpdateDownload</code> 被调用时会触发回调该方法	是
<code>void onPercent(int percent, long rcvLen, long fileSize)</code>	Percent : 进度百分比 rcvLen : 已下载文件大小 fileSize : 文件总大小	下载进度通过该方法通知应用	否
<code>void onDownloadComplete(String apkPath)</code>	apkPath : 下载完成后的 apk 包路径	下载完成后本地的 APK 包路径回调接口	否
<code>void onFail(Throwable error, String content)</code>	Error : 异常信息 Content : 异常提示	下载失败或者发送错误时回调此接口	否

content)			
void onStop()	无	下载流程结束后统一调此接口	是

3.2.3 新版本安装

接口：

```
static void cpUpdateInstall(Context context, String apkPath)
```

其中 apkPath 为本地安装包路径。

3.3 无 UI 更新

此接口被调用时会先检查本地是否有已下载完成的新版本，如果有则会弹出提示框提示用户安装，否则，APP 会在后台进行无感知版本检测并下载。**此接口只支持在 WIFI 环境。**

接口：

```
static void silenceUpdateAction(Context context, Boolean useHttps)
```

3.4 通过文件地址安装

此接口被调用时会安装传入路径的文件，支持 7.0fileprovider。

接口：

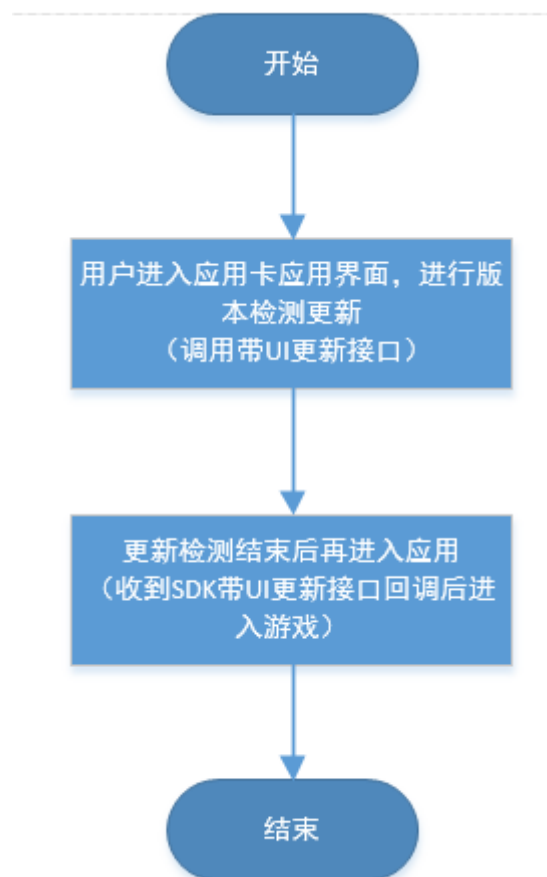
```
static void cpUpdateInstall(Context context)
```

4 接入场景及接入流程

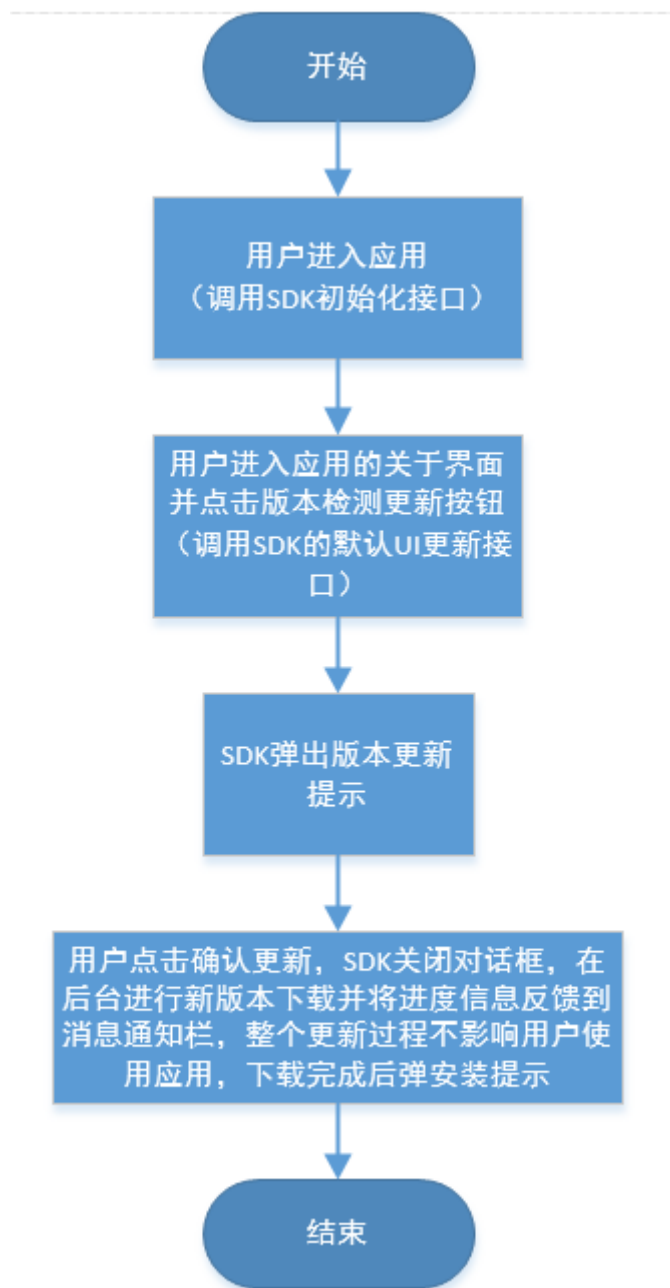
4.1 默认 UI 更新

场景一：开发者希望 APP 在应用初始化界面时先卡界面进行版本检测更新，无更新时进入应用，有更新时提示用户应用有新版本更新，用户选择更新后进入应用，后台进行版本更新，并将新版本 APK 下载进度反馈到消息通知栏。

接入流程：



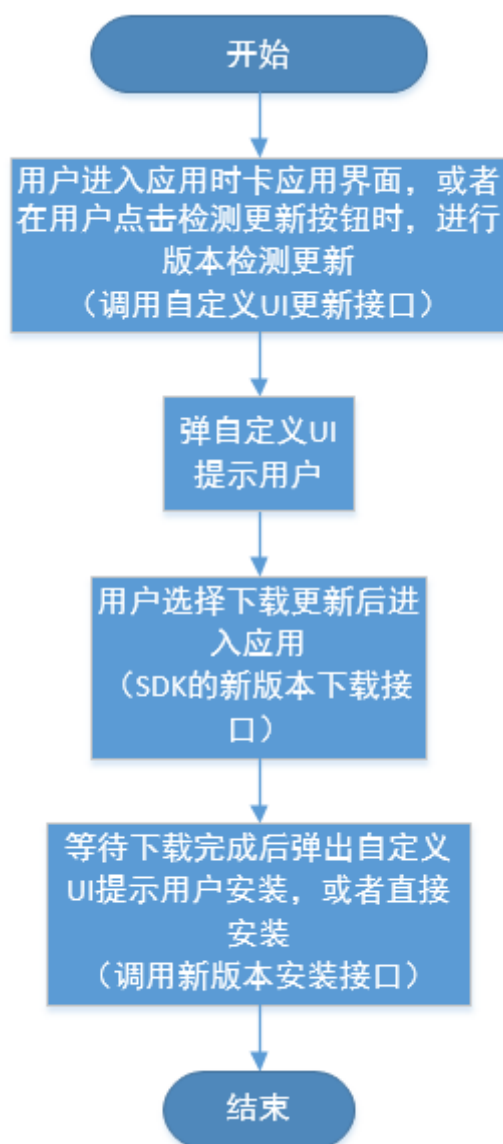
场景二：开发者希望将版本检查更新的主动权交给用户，将版本检查更新功能放在应用的关于模块里面，由用户自己进入关于模块，手动触发版本检查更新功能进行升级。



4.2 自定义 UI 更新

场景：开发者希望能够自己定义 APP 版本检测更新的 UI 及场景逻辑，此时 SDK 提供版本检测接口、APK 下载接口以及 APK 安装接口，开发者可通过这三个接口定义各种更新场景。

接入流程：



4.3 静默更新

只支持 WiFi 环境。

场景：开发者希望在用户无感知情况下对 APP 进行更新，进入应用时，后台进行新版本检测并下载。从版本检测，新版本下载，下载完成，下载进度，用户均无感知。下载完成后用户再次进入应用时提示用户进行安装。

接入流程：

