

# 安卓APK文件OEM定制\_V8.0

注：该文档适用于SmartbiV8.5以下版本。

## 场景说明

通过OEM定制，客户可以在原来Smartbi Android APP基础上进行修改，修改其中的应用信息、桌面图标、启动图片、加载中的图片、主界面、登录界面等，最后打包生成一个OEM定制的Android APK文件。

在Android设备上安装该APK文件，会安装一个经过OEM定制的APP，得到在APP界面上显示客户的LOGO等显示效果。

Smartbi支持OEM定制以下内容：

### 1. 应用信息

下表是一些OEM定制中常需修改的值。其中“名称”一列在修改应用信息时会使用到，在后文中会有说明。

名称	示例	说明
app_name	Smartbi	应用显示名称
DEFAULT_SERVER_ADDR\${n}	<a href="http://demo.smartbi.com.cn/smartbi">http://demo.smartbi.com.cn/smartbi</a>	配置第n个服务器的地址
DEFAULT_USER_NAME\${n}	demo	配置第n个服务器的登录用户名
DEFAULT_PASSWORD\${n}	demo	配置第n个服务器的登录密码

### 2. 相关图片资源

OEM定制过程中，一般需要修改以下6张图片。注意，在修改时，**图片的名称和尺寸必须与下表严格一致**。

下表中所说的“路径”在后面修改图片资源时会使用到，在后文中会有说明。

图片	尺寸	路径
桌面图标	144×144	Smartbi for Android Packager\smartbi\res\drawable\logo_icon.png
LOGO	550×320	Smartbi for Android Packager\smartbi\res\drawable\logo.png
加载中图片	1024×768	Smartbi for Android Packager\smartbi\res\drawable\loading_pad.png
加载中LOGO	400×100	Smartbi for Android Packager\smartbi\res\drawable\loading_logo.png
主界面背景	1024×768	Smartbi for Android Packager\smartbi\res\drawable\mainview_background.png
桌面图标（高分辨率设备）	144×144	Smartbi for Android Packager\smartbi\res\drawable-hdpi\logo_icon.png

## OEM定制步骤

注：若是希望后期oem更新后，APP自动提醒用户更新，可先查看wiki文档，移动端oem定制更新后提醒用户下载 再进行OEM定制

为了方便客户进行OEM定制，我们提供了Smartbi for Android Packager工具，按以下步骤可以完成OEM定制：

1. 下载Smartbi for Android Packager.zip并解压。
2. 在<http://app.smartbi.com.cn/indexv8.htm> 下载Smartbi apk文件，放到解压后的Smartbi for Android Packager目录中，并重命名为smartbi.apk。
3. 修改Smartbi for Android Packager\decode.cmd中的Java路径，建议先设置好JAVA\_HOME环境变量，然后像下图那样设置。注意，JAVA路径最好包含在双引号中，否则会在路径中包含空格时报错。

```

decode.cmd
1 @echo off
2 set current=%~dp0
3 cd /d "%current%"
4 rd /s /q smartbi
5 "%JAVA_HOME%\bin\java" -jar apktool.jar decode -s "smartbi.apk" "smartbi"

```

4. 执行Smartbi for Android Packager\decode.cmd, 执行完毕后smartbi.apk会被解压生成Smartbi for Android Packager\smartbi文件夹。
5. OEM定制的APP应该避免收到各安卓商店的升级提示信息, 因为升级相当于将定制内容重置。  
用文本编辑器打开Smartbi for Android Packager\smartbi\AndroidManifest.xml, 修改<manifest>元素的android:versionCode属性的值(即下图中的值)为一个较大的数值。  
该数值是APP的版本号信息, 后续发布新的版本, 会持续提升这个值。设置一个较大的值, 例如9999, 可以避免收到各个安卓应用商店的升级提示信息。

```

AndroidManifest.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest android:versionCode="9999" android:versionName="7.0" package="cn.com.smartbi"
3 xmlns:android="http://schemas.android.com/apk/res/android">
4 <uses-permission android:name="android.permission.INTERNET" />
5 <uses-permission android:name="android.permission.READ_PHONE_STATE" />
6 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
7 <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
8 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
9 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

```

6. 应用信息修改: 用文本编辑器打开Smartbi for Android Packager\smartbi\res\values\strings.xml, 就可以对目标信息进行修改。  
应用信息的修改可以参考该表, 表中的“名称”列对应的就是strings.xml中<string>的name属性。  
除了上表提到的信息外, 还需修改<string name="updateCheckType">这一元素的值为**private**, 如下图所示, 这样可以避免访问app.smartbi.com.cn去获取新版本。  
其中, app\_name是app的显示名称, version\_n / webaddr\_v / support\_v / copyright\_v等在设置界面显示, DEFAULT\_SERVER\_ADDR1 / DEFAULT\_USER\_NAME1 / DEFAULT\_PASSWORD1等设置了默认的服务器地址和默认访问用户/密码

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Smartbi</string>
  <string name="app_name_gddw">广电信息门户</string>
  <string name="username">用户名: </string>
  <string name="password">密码: </string>
  <string name="login">登录</string>
  <string name="serveraddr">服务器地址</string>
  <string name="save">保存</string>
  <string name="ok">确定</string>
  <string name="updateCheckType">global</string>
  <string name="globalUpdateKey">androidAppVersion</string>
  <string name="version_n">Smartbi for Android</string>
  <string name="version_1">V7.2</string>
  <string name="version_v">V7.2(Build 2017/09/27 16:00:00)</string>
  <string name="version_d">2017-09-27 16:00:00</string>
  <string name="webaddr_n">网址:</string>
  <string name="webaddr_v">www.smartbi.com.cn</string>
  <string name="support_n">技术支持:</string>
  <string name="support_v">support@smartbi.com.cn</string>
  <string name="copyright_n">版权所有</string>
  <string name="copyright_v">广州思迈特软件有限公司</string>
  <string name="loading">加载中</string>
  <string name="menu_settings">设置</string>
  <string name="menu_about">关于</string>
  <string name="menu_exit">退出</string>
  <string name="menu_handwrite">手写</string>
  <string name="menu_send">发送</string>
  <string name="menu_favorite">收藏</string>
  <string name="menu_screenshot">缩略图</string>

```

```
<string name="menu_handwrite">手写</string>
<string name="menu_send">发送</string>
<string name="menu_favorite">收藏</string>
<string name="menu_screenshot">缩略图</string>
<string name="menu_online">在线</string>
<string name="menu_offline">离线</string>
<string name="delete">删除</string>
<string name="catalogtitle">资源目录</string>
<string name="firstload">正在为首次运行初始化, 需要几分钟.....</string>
<string name="downloadOffline">正在处理离线包.....</string>
<string name="DEFAULT_SERVER_ADDR1">http://demo.smartbi.com.cn/smartbi</string>
<string name="DEFAULT_USER_NAME1">demo</string>
<string name="DEFAULT_PASSWORD1">demo</string>
<string name="DEFAULT_SERVER_ADDR2" />
<string name="DEFAULT_USER_NAME2" />
<string name="DEFAULT_PASSWORD2" />
<string name="DEFAULT_SERVER_ADDR3" />
<string name="DEFAULT_USER_NAME3" />
<string name="DEFAULT_PASSWORD3" />
<string name="SYSTEM_CONFIG_TEXT">系统配置</string>
<string name="SERVER_TEXT">服务器</string>
```

- 7. 相关图片资源修改:  
相关图片资源都在Smartbi for Android Packager\smartbi\res\drawable-xxx (各目录对应不同分辨率的设备)下。OEM定制中主要涉及到6张图片, 如该表所示。表中“路径”一列表示的就是对应图片文件在解压目录下的位置及名称。要修改这些资源, 只需替换对应的图片文件即可。注意, 必须保证替换图片的文件名和图片尺寸与原图片文件完全一致。
- 8. 修改Smartbi for Android Packager\encode.cmd中的Java路径, 然后执行encode.cmd。

执行完成后, Smartbi for Android Packager\smartbi目录下的内容将会被编译成Smartbi for Android Packager\Smartbi\_new.apk文件。

```
encode.cmd
1 @echo off
2 set current=%~dp0
3 cd /d "%current%"
4 "%JAVA_HOME%\bin\java" -jar apktool.jar build "smartbi" "Smartbi_new.apk"
```

## APK文件签名

上文中生成的APK文件其实已经可以安装到一部分Android设备中, 但对于部分设备会安装失败, 失败原因是安装包解析出错, 因此为了兼容所有安卓设备, 且为了安全性起见, 一般都要对APK文件进行签名。

对APK文件进行签名的主要目的, 是防止第三方通过在设备上安装包名完全一致的APP来替换掉自己的APP, 而且没有签名的APK文件有可能会被某些设备认为不安全而拒绝安装。

如果对这方面没有需求, 理论上可以跳过APK文件签名, 直接安装到Android设备上, 但不建议这么做。

进行APK文件签名, 可以按以下步骤来完成:

- 1. 生成keystore证书, 这是Android应用签名必须的。
  - a. 如果已经有自己的证书, 直接将自己的证书文件重命名为Smartbi.keystore然后放在Smartbi for Android Packager目录下, 然后跳到第2步;
  - b. 如果上次已经生成过证书, 且证书已经在Smartbi for Android Packager目录下, 则直接跳到第2步;
  - c. 如果这是第一次操作, 且没有证书, 请按以下步骤生成证书文件:
    - i. 修改Smartbi for Android Packager\makekey.cmd中的Java路径、keypass和storepass。  
其中keypass和storepass是keystore中需要设置的两个密码, **两个密码需要保持一致。**

```
makekey.cmd
1 @echo off
2 set current=%~dp0
3 cd /d "%current%"
4 if exist "%current%\Smartbi.keystore" ren "Smartbi.keystore" "Smartbi-%DATE:~0,4%-&DATE:~0,2%-&DATE:~8,2%.keystore.bak"
5 "%JAVA_HOME%\bin\keytool" -genkey -alias Smartbi -keyalg RSA -validity 40000 -dname "CN=localhost, OU=localhost, O=localhost, L=SH, ST=SH, C=CN" -keystore
Smartbi.keystore [keypass password] [storepass password]
设置keypass和storepass, 两个值需要保持一致
```

- ii. 执行Smartbi for Android Packager\makekey.cmd, 执行完成后, 将生成Smartbi for Android Packager\Smartbi.keystore证书文件。

aapt.exe	2013/5/13 12:47	应用程序	835 KB
apktool.jar	2013/2/3 2:37	Executable Jar File	2,594 KB
decode.cmd	2015/9/7 14:26	Windows 命令脚本	1 KB
encode.cmd	2015/9/7 14:26	Windows 命令脚本	1 KB
makekey.cmd	2017/5/4 10:51	Windows 命令脚本	1 KB
resign.cmd	2015/9/7 14:26	Windows 命令脚本	1 KB
Smartbi.keystore	2017/5/4 10:51	KEYSTORE 文件	3 KB
zipalign.exe	2013/5/13 18:49	应用程序	163 KB

2. 修改Smartbi for Android Packager\resign.cmd中的Java路径、keypass和storepass, 其中keypass和storepass需要与第1步中设置的值一样。

```
resign.cmd
1 | echo off
2 | set current=%~dp0
3 | cd /d "%current%"
4 | "C:\JAVA\jdk8\bin\jarsigner" -keypass password -storepass password -verbose -sigalg SHA1withRSA -digestalg SHA1 -sigfile CERT -keystore Smartbi.keystore
   "Smartbi_new.apk" Smartbi
5 | del "Smartbi_new_Unsigned.apk"
6 | ren "Smartbi_new.apk" "Smartbi_new_Unsigned.apk"
7 | zipalign.exe -v 4 "Smartbi_new_Unsigned.apk" "Smartbi_new.apk"
8 |
```

3. 执行Smartbi for Android Packager\resign.cmd。

执行完成后, Smartbi for Android Packager\Smartbi\_new.apk文件已经被签名成功。

可以注意到还生成了一个Smartbi for Android Packager\Smartbi\_new\_Unsigned.apk文件, 这个文件是原来没有签名的APK文件的备份。

## 移动端首页资源修改

移动端首页等资源的修改没有上面的工具中提供, 因为它们是服务器提供、APP加载的。

要修改移动端首页, 可以下载MobilePortalChange.ext扩展包, 用压缩工具打开扩展包, 然后替换其中相应的图片资源。注意, 替换的图片资源名称和尺寸必须与对应图片文件一致。

移动端首页对应的图片资源在这扩展包中这两个目录下: 【MobilePortalChange.ext\vision\android\login\img】, 【MobilePortalChange.ext\vision\android\portraitweb\img】

下面两个对应LOGO的文件:

MobilePortal.ext\vision\android\login\img\login\_logo.png

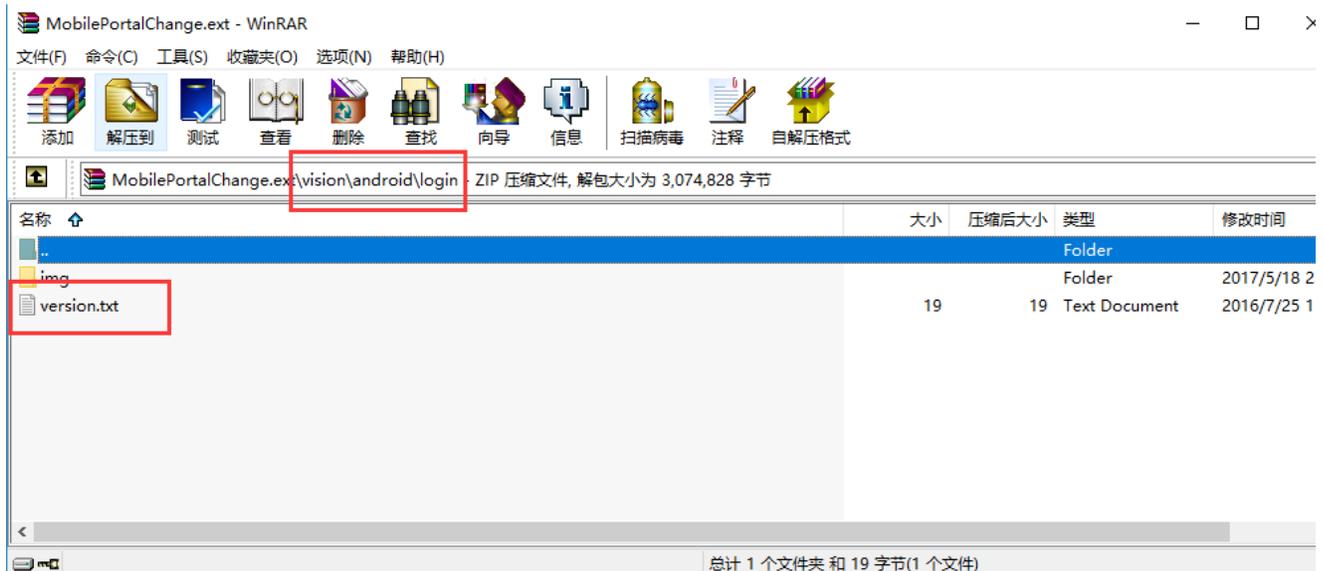
MobilePortal.ext\vision\android\portraitweb\img\homepage\_banner.png

(280×58)

(544×72)



这一步必做的：为了防止修改的logo后出现不生效的问题，修改了MobilePortalChange.ext 中哪个目录下的图片，就需要保证这个目录下的version.txt上的时间和产品war包中mobileproatal.ext扩展包中的相同路径下的version.txt的时间不能相同。安卓版本的version.txt路径为：MobilePortalChange.ext\vision\android\login（如下图），将里边的时间信息重新修改下，保持和war包版本中的mobileproatal.ext扩展包（mobileproatal.ext在war包中扩展包路径为：C:\Smarbti\_Insight\Tomcat\webapps\smartbi\WEB-INF\extensions）中的version时间不一样即可：





最后，将扩展包部署到服务器上，详见[扩展包部署](#)。

## “分享到微信”功能

由于“分享到微信”功能已经绑定了Smartbi的数字签名，因此按以上步骤生成的apk将无法使用“分享到微信”的功能。如果还需要此功能，有两种方式处理：

1. 将“OEM定制步骤”中第8步生成的smartbi\_new.apk发到思迈特技术支持团队进行签名。这种方式的缺点是，此apk在微信中显示的小图标还是Smartbi的图标。
2. 用户自行在微信开发平台上申请新的应用，并将申请的app包名发到思迈特技术支持团队进行定制化开发。