



Android

定制开关机动画及视频开发指南

版本号: 1.0
发布日期: 2024.12.11

版本历史

| 版本号 | 日期 | 制/修订人 | 内容描述 |
|-----|------------|---------|--------|
| 1.0 | 2024.12.11 | AWA1829 | 初始版本文档 |



目 录

| | |
|----------------------------|----------|
| 1 前言 | 1 |
| 1.1 文档简介 | 1 |
| 1.2 目标读者 | 1 |
| 1.3 适用范围 | 1 |
| 2 开机动画 | 2 |
| 2.1 动画文件可配置路径 | 2 |
| 2.2 预置开机动画文件 | 2 |
| 2.3 调试开机动画 | 2 |
| 2.3.1 开机动画规格 | 2 |
| 2.3.2 配置 desc.txt | 3 |
| 2.3.3 优化 png 图片 | 4 |
| 2.3.4 配置尺寸不同图片 | 4 |
| 2.3.5 打包 bootanimation.zip | 4 |
| 2.3.6 推送开机动画文件到设备 | 4 |
| 2.3.7 播放开机动画 | 5 |
| 3 开机视频 | 6 |
| 3.1 预置开机视频 | 6 |
| 3.2 调试开机视频 | 6 |
| 3.2.1 开启播放开机视频功能 | 6 |
| 3.2.2 推送开机视频文件到设备 | 7 |
| 4 关机动画 | 8 |
| 4.1 预置关机动画文件 | 8 |
| 4.2 调试关机动画 | 8 |
| 4.2.1 开启播放关机动画的功能 | 8 |
| 4.2.2 常见问题 | 9 |

1 前言

1.1 文档简介

Android 平台开机动画、开机视频、关机动画配置，调试指南

1.2 目标读者

Android 平台开机动画、开机视频、关机动画模块维护，开发人员

1.3 适用范围

Android 平台

2 开机动画

开机动画以 zip 压缩包的形式存放，开机动画文件名称固定为 bootanimation.zip，还有暗色调的开机动画名称为 bootanimation_dark.zip

2.1 动画文件可配置路径

开机动画可放配置路径，开机动画文件播放优先级按照以下排序

```
/oem/media/bootanimation.zip  
/product/media/bootanimation.zip  
/system/media/bootanimation.zip
```

2.2 预置开机动画文件

SDK 的方案目录中完成开机动画的配置。

配置文件路径：common/media/config.mk

开机动画文件存放路径：common/media/bootanimation

以下为配置将开机动画文件放在 system/media/路径的案例：

```
BOOTANIMATION_CONFIG_PATH := $(LOCAL_MODULE_PATH)/bootanimation  
PRODUCT_COPY_FILES += \  
$(BOOTANIMATION_CONFIG_PATH)/bootanimation.zip:system/media/bootanimation.zip \  
$(BOOTANIMATION_CONFIG_PATH)/bootanimation_dark.zip:system/media/bootanimation_dark.zip \
```

在配置和编译完成后，可在 out 目录查看是否真的有把开机动画编译到方案目录中

路径：out/target/product/【方案】/system/media

2.3 调试开机动画

2.3.1 开机动画规格

文件是一个压缩包文件，其中包括了 part0、part1、audio_conf.txt、desc.txt 等四个文件。

文件结构：

```

bootanimation
|- desc.txt
|- audio_conf.txt
|- part0
  |- part000.png
  |- part001.png
  |- part0...png
  |- audio.wav(可选)
  |- trim.txt(可选)
|- part1
  |- part100.png
  |- part101.png
  |- part1...png
  |- audio.wav(可选)
  |- trim.txt(可选)
|- part...N

```

part0

--开机动画第一部分，存放的是一帧帧的动画图片

part1

--开机动画第二部分。一般情况下，此文件夹存放Logo图片文件，在part0播放完毕后，循环播放part1，直到开机完成。

audio_conf.txt

--开机动画音频的一些配置

desc.txt

--开机动画播放规格设置部分，名称不可修改。定义了开机动画的分辨率、帧率和每个部分的播放参数。

压缩开机动画时，压缩等级需要选择【仅存储】

2.3.2 配置 desc.txt

第一行：

WIDTH HEIGHT FPS

后面每行，表示部分part动画：

TYPE: 类型 (p: 播放直到开机完成, c: 播放完整动画)

COUNT: 循环次数, 0表示无限循环直到开机结束

PAUSE: part结束后暂停帧数

PATH: 文件加路径 (如: part0)

RGBHEX: (可选) 背景颜色: #RRGGBB

CLOCK: (可选) 画当前时间的y坐标(for watches)

例如：

1080 1920 25

p 1 0 part0

p 0 0 part1 #FF0000

2.3.3 优化 png 图片

```
编译zopflipng工具
$ cd external/zopfli
$ make zopflipng
开始优化（一张图约几分钟）
for fn in *.png; do zopflipng -m ${fn} ${fn}.new && mv -f ${fn}.new ${fn} & done
注意：加入 '&' 符号表示后台多线程执行，命令后会直接返回，但是需要等待全部执行完毕才能进行打包。
```

2.3.4 配置尺寸不同图片

```
参考frameworks/base/cmds/bootanimation/FORMAT.md
如：trim.txt每幅图片尺寸不同时可做配置
```

2.3.5 打包 bootanimation.zip

```
$ zip -0qry -i \*.txt \*.png \*.wav @ bootanimation.zip *.txt part*
如果不需要开机音乐
$ zip -0qry -i \*.txt \*.png @ bootanimation.zip *.txt part*
```

2.3.6 推送开机动画文件到设备

设备固件为 userdebug 版本，并且需要先 oem 解锁，之后 remount 设备

```
# oem 解锁
adb reboot bootloader

fastboot oem unlock

fastboot reboot

# remount设备
adb root

adb remount

adb reboot

adb root

adb remount

# 推开机动画到设备中
adb push bootanimation.zip /system/media
```

2.3.7 播放开机动画

仅用于调试使用，可用于播放当前配置的开机动画

```
setprop service.bootanim.exit 0  
bootanimation
```



3 开机视频

开机视频的文件推荐使用 mp4 格式，文件名称固定为 bootanimation.mp4，mp4 格式。

3.1 预置开机视频

SDK 的方案目录中完成开机视频的配置。

配置文件路径：common/media/config.mk

开机动画文件存放路径：common/media/bootanimation

推荐配置：开机视频时长可控制在 15 秒以内；mp4 格式

以下为配置将开机视频文件放在 system/media/路径的案例：

```
BOOTANIMATION_CONFIG_PATH := $(LOCAL_MODULE_PATH)/bootanimation
PRODUCT_COPY_FILES += \
$(BOOTANIMATION_CONFIG_PATH)/bootanimation.mp4:system/media/bootanimation.mp4
```

在配置和编译完成后，可在 out 目录查看是否真的有把开机机动画编译到方案目录中

路径：out/target/product/【方案】/system/media

开机视频和开机动画同时存在的情况下，优先会播放开机视频。开机视频文件播放优先级按照以下排序。

```
/oem/media/bootanimation.mp4
/product/media/bootanimation.mp4
/system/media/bootanimation.mp4
```

默认是不开启播放开机视频的功能

3.2 调试开机视频

3.2.1 开启播放开机视频功能

由于不是所有的方案都需要开启开机视频，所以提供两种方式来开启播放开机视频的功能。

一直开启播放开机视频的方式

在 sdk 修改，可参考方式：

```
frameworks/base/cmds/bootanimation/BootAnimation.cpp  
  
@@ -876,7 +880,7 @@ void BootAnimation::findBootAnimationFile() {  
    SYSTEM_VIDEO_FILE, DATA_VIDEO_FILE, PRODUCT_VIDEO_FILE,  
};  
  
- const bool playVideo = android::base::GetIntProperty(USER_BOOTVIDEO_ENABLE, 0) == 1;  
+ const bool playVideo = android::base::GetIntProperty(USER_BOOTVIDEO_ENABLE, 0) == 0;  
  
}
```

userdebug 固件开启开机视频功能方式

为了便于 userdebug 固件调试，通过配置 persist.sys.bootanim.video_enable 来开启或者关闭播放开机视频的功能。

开启播放开机视频的功能

```
setprop persist.sys.bootanim.video_enable 1
```

关闭播放开机视频功能

```
setprop persist.sys.bootanim.video_enable 0
```

3.2.2 推送开机视频文件到设备

推送开机动画文件到设备可参考开机动画章节。

4 关机动画

关机动画是设备关机时播放的动画。关机动画文件名固件为 shutdownanimation.zip，文件配置和开机动画相同。

存放路径，关机动画文件播放优先级按照以下排序：

```
/oem/media/shutdownanimation.zip  
/product/media/shutdownanimation.zip  
/system/media/shutdownanimation.zip
```

SDK 默认是不开启播放关机动画的功能

4.1 预置关机动画文件

SDK 的方案目录中完成关机动画的配置。

配置文件路径：common/media/config.mk

关机动画文件存放路径：common/media/shutdownanimation

以下为配置将关机动画文件放在 system/media/路径的案例：

```
BOOTANIMATION_CONFIG_PATH := $(LOCAL_MODULE_PATH)/shutdownanimation  
PRODUCT_COPY_FILES += \  
$(BOOTANIMATION_CONFIG_PATH)/shutdownanimation.zip:system/media/shutdownanimation.zip
```

在配置和编译完成后，可在 out 目录查看是否真的有把开机动画编译到方案目录中

路径：out/target/product/【方案】/system/media

4.2 调试关机动画

4.2.1 开启播放关机动画的功能

一直开启播放关机动画的功能

参考方式，对 SDK 对应的两个代码路径完成以下修改：

```
1- frameworks/base/services/core/java/com/android/server/power/ShutdownThread.java
```

```
index db231cf66e09..87356b755c16 100644
--- a/services/core/java/com/android/server/power/ShutdownThread.java
+++ b/services/core/java/com/android/server/power/ShutdownThread.java
@@ -437,7 +437,7 @@ public final class ShutdownThread extends Thread {
    /// AW CODE:[feat]:shut down screen directly
    final boolean shutdownForceScreenOff = context.getResources().getBoolean(
        com.android.internal.R.bool.config_shutdownForceScreenOff);
-   final boolean showShutdownAnimation = SystemProperties.getBoolean("ro.init.shutdown_animation", false);
+   final boolean showShutdownAnimation = SystemProperties.getBoolean("ro.init.shutdown_animation", true);
    ...
}
```

2、system/core/init/reboot.cpp

```
@@ -666,7 +666,7 @@ static void DoReboot(unsigned int cmd, const std::string& reason, const std::str
    abort();
}

- bool do_shutdown_animation = GetBoolProperty("ro.init.shutdown_animation", false);
+ bool do_shutdown_animation = GetBoolProperty("ro.init.shutdown_animation", true);
// watchdog is a vendor specific component but should be alive to complete shutdown safely.
const std::set<std::string> to_starts{"watchdog"};
std::set<std::string> stop_first;
```

调试播放关机动画

调试可先推关机动画到设备对应，详情可见推送开机动画文件到设备。

设置属性：

```
adb shell
su
setprop ro.init.shutdown_animation true
```

关机时便看到关机动画。

4.2.2 常见问题

1、如果 Android15 的代码编译的固件无法播放关机动画，那么可进行如下修改

```
// 代码路径:
// system/core/init/reboot.cpp
static void DoReboot{
.....
-- } else if (do_shutdown_animation && s->classnames().count("animation") > 0) {
++ } else if (do_shutdown_animation) {
.....
}
```

2、要是不想要播放关机动画前有转圈圈的图标，可做如下修改

```
// 代码路径:
// frameworks/base/services/core/java/com/android/server/power/ShutdownThread.java
private static void beginShutdownSequence(Context context) {
```

```
.....  
-- sInstance.mProgressDialog = showShutdownDialog(context);  
++ //sInstance.mProgressDialog = showShutdownDialog(context);  
.....  
}
```






著作权声明

版权所有 ©2025 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

、、**全志科技**、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。