

# Android音视频处理之MediaCodec



郑旭泽 (/u/dc225ec94e20) [+ 关注](#)

2016.12.16 15:47\* 字数 1275 阅读 12187 评论 11 喜欢 30

(/u/dc225ec94e20)

Android中可以使用MediaCodec来访问底层的媒体编解码器，可以对媒体进行编/解码。

MediaCodec的官方文档 (<https://link.jianshu.com?t=https://developer.android.com/reference/android/media/MediaCodec.html>)真的超详细，基本上所有问题都可以在文档上找到答案。

MediaCodec可以处理的数据有以下三种类型：压缩数据、原始音频数据、原始视频数据。这三种类型的数据均可以利用ByteBuffers进行处理，但是对于原始视频数据应提供一个Surface (<https://link.jianshu.com?t=https://developer.android.com/reference/android/view/Surface.html>)以提高编解码器的性能。Surface直接使用native视频数据缓存，而没有映射或复制它们到ByteBuffers，因此，这种方式会更加高效。

MediaCodec采用异步方式处理数据，并且使用了一组输入输出缓存（ByteBuffer）。通过请求一个空的输入缓存（ByteBuffer），向其中填满数据并将它传递给编解码器处理。编解码器处理完这些数据并将处理结果输出至一个空的输出缓存（ByteBuffer）中。使用完输出缓存的数据之后，将其释放回编解码器：

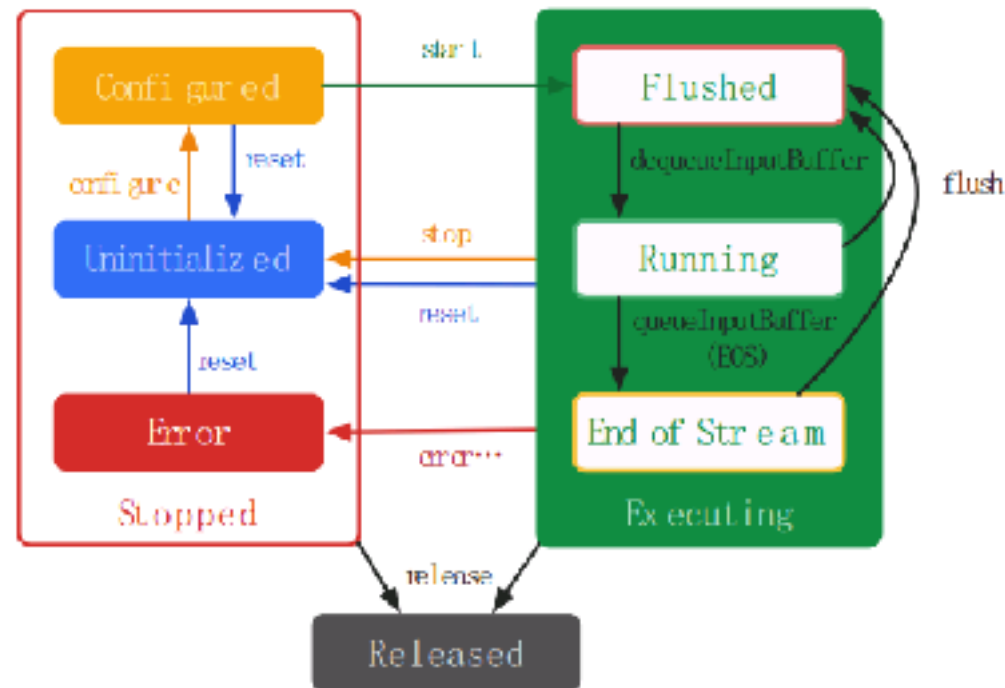




MediaCodec的工作原理

(/apps/download?  
utm\_source=sbc)

MediaCodec的生命周期有三种状态：停止态-Stopped、执行态-Executing、释放态-Released，停止状态（Stopped）包括了三种子状态：未初始化（Uninitialized）、配置（Configured）、错误（Error）。执行状态（Executing）会经历三种子状态：刷新（Flushed）、运行（Running）、流结束（End-of-Stream）：



MediaCodec的生命周期

- 当创建了一个MediaCodec对象，此时MediaCodec处于Uninitialized状态。首先，需要使用configure(...)方法对MediaCodec进行配置，这时MediaCodec转为Configured状态。然后调用start()方法使其转入Executing状态。在Executing状态可以通过上述



的缓存队列操作处理数据。

- Executing状态包含三个子状态：Flushed、Running 以及End-of-Stream。在调用start()方法后MediaCodec立即进入Flushed子状态，此时MediaCodec会拥有所有的缓存。一旦第一个输入缓存（input buffer）被移出队列，MediaCodec就转入Running子状态，这种状态占据了MediaCodec的大部分生命周期。当你将一个带有end-of-stream marker标记的输入缓存入队时，MediaCodec将转入End-of-Stream子状态。在这种状态下，MediaCodec不再接收之后的输入缓存，但它仍然产生输出缓存直到end-of-stream标记输出。你可以在Executing状态的任何时候通过调用flush()方法返回到Flushed子状态。
- 通过调用stop()方法使MediaCodec返回到Uninitialized状态，因此这个MediaCodec可以再次重新配置。当使用完MediaCodec后，必须调用release()方法释放其资源。
- 在极少情况下MediaCodec会遇到错误并进入Error状态。这个错误可能是在队列操作时返回一个错误的值或者有时候产生了一个异常导致的。通过调用reset()方法使MediaCodec再次可用。你可以在任何状态调用reset()方法使MediaCodec返回到Uninitialized状态。否则，调用release()方法进入最终的Released状态。

MediaCodec的使用相对MediaExtractor和MediaMuxer来说要复杂一些，首先是创建一个新的MediaCodec对象然后调用configure(...)方法对MediaCodec进行配置：

```
String AUDIO_MIME = "audio/mp4a-latm";
MediaCodec mAudioEncoder = MediaCodec.createEncoderByType(AUDIO_MIME);
MediaFormat mAudioMediaFormat = new MediaFormat();
mAudioMediaFormat.setString(MediaFormat.KEY_MIME, AUDIO_MIME);
mAudioMediaFormat.setInteger(MediaFormat.KEY_BIT_RATE, 32000);
mAudioMediaFormat.setInteger(MediaFormat.KEY_CHANNEL_COUNT, 2);
mAudioMediaFormat.setInteger(MediaFormat.KEY_SAMPLE_RATE, 48000);
mAudioMediaFormat.setInteger(MediaFormat.KEY_MAX_INPUT_SIZE, BUFFER_SIZE_IN_BYTES);
mAudioMediaFormat.setInteger(MediaFormat.KEY_AAC_PROFILE,
MediaCodecInfo.CodecProfileLevel.AACObjectLC);
mAudioEncoder.configure(mAudioMediaFormat, null, null, MediaCodec.CONFIGURE_FLAG_ENCODE);
```

(/apps/download?  
utm\_source=sbc)



通过文档我们可以看到MediaCodec有几个方法可以构造MediaCodec对象，有createDecoderBy...和createEncoderBy...，分别对应解码器和编码器。此时MediaCodec处于Uninitialized状态。

调用configure(...)方法对MediaCodec进行配置，此时MediaCodec处于Configured状态。

接着调用MediaCodec的start()方法，此时MediaCodec处于Executing状态，可以通过getInputBuffers()方法和getOutputBuffers()方法获取缓存队列：

```
mAudioEncoder.start();
mAudioInputBuffers = mAudioEncoder.getInputBuffers();
mAudioOutputBuffers = mAudioEncoder.getOutputBuffers();
```

当MediaCodec处于Executing状态之后就可以对数据进行处理了。首先通过dequeueInputBuffer(long timeoutUs)请求一个输入缓存，timeoutUs代表等待时间，设置为-1代表无限等待：

```
int inputBufIndex = mAudioEncoder.dequeueInputBuffer(1000);
```

返回的整型变量为请求到的输入缓存的index，通过getInputBuffers()得到的是输入缓存数组，通过index和输入缓存数组可以得到当前请求的输入缓存，在使用之前要clear一下，避免之前的缓存数据影响当前数据：

```
mInputBuffer = mAudioInputBuffers[inputBufIndex];
mInputBuffer.clear();
```

接着就是把数据添加到输入缓存中，并调用queueInputBuffer(...)把缓存数据入队：

```
mInputBuffer.put(bytes, 0, BUFFER_SIZE_IN_BYTES);
mAudioEncoder.queueInputBuffer(inputBufIndex, 0, BUFFER_SIZE_IN_BYTES, (1000000 * mEncodedSi
```

(/apps/download?  
utm\_source=sbc)



获取输出缓存和获取输入缓存类似，首先通过`dequeueOutputBuffer(BufferInfo info, long timeoutUs)`来请求一个输出缓存，这里需要传入一个`BufferInfo`对象，用于存储`ByteBuffer`的信息：

```
int outputBufIndex = mAudioEncoder.dequeueOutputBuffer(mOutBufferInfo, 1000);
```

然后通过返回的`index`得到输出缓存，并通过`BufferInfo`获取`ByteBuffer`的信息，通过`BufferInfo`我们可以得到当前数据是否`Codec-specific Data`：

```
mOutBuffer = mAudioOutputBuffers[outputBufIndex];
if ((mOutBufferInfo.flags & MediaCodec.BUFFER_FLAG_CODEC_CONFIG) != 0) {
    // Codec-specific Data, 这里可以从ByteBuffer中获取csd参数
    // audioFormat.setByteBuffer("csd-0", mOutBuffer);
} else {
    // 处理数据
}
mAudioEncoder.releaseOutputBuffer(outputBufIndex, false);
```

注意一定要调用`releaseOutputBuffer`方法。

使用完`MediaCodec`后释放资源：

```
mAudioEncoder.stop();
mAudioEncoder.release();
```

下面例子展示了把PCM数据编码为AAC数据：

(/apps/download?  
utm\_source=sbc)



```

private void encodePCMTtoAAC(byte[] bytes) throws IOException {
    int inputBufIndex = mAudioEncoder.dequeueInputBuffer(1000);
    if (inputBufIndex >= 0) {
        mInputBuffer = mAudioInputBuffers[inputBufIndex];
        mInputBuffer.clear();
        mInputBuffer.put(bytes, 0, BUFFER_SIZE_IN_BYTES);
        mAudioEncoder.queueInputBuffer(inputBufIndex, 0, BUFFER_SIZE_IN_BYTES, (1000000 * mEncoderTimeUs) / 1000, 0);
        mEncodedSize += BUFFER_SIZE_IN_BYTES;
    }

    int outputBufIndex = mAudioEncoder.dequeueOutputBuffer(mOutBufferInfo, 1000);
    if (outputBufIndex >= 0) {
        mOutBuffer = mAudioOutputBuffers[outputBufIndex];
        if ((mOutBufferInfo.flags & MediaCodec.BUFFER_FLAG_CODEC_CONFIG) != 0) {
            // 这里可以从ByteBuffer中获取csd参数
            mAudioEncoder.releaseOutputBuffer(outputBufIndex, false);
            return;
        }
        if (mOutBufferInfo.size != 0 && mLastAudioPresentationTimeUs < mOutBufferInfo.presentationTimeUs) {
            mOutBuffer.position(mOutBufferInfo.offset);
            mOutBuffer.limit(mOutBufferInfo.offset + mOutBufferInfo.size);

            addADTSHeader(mADTSHeader, mOutBufferInfo.size + 7);
            mMixedFileOutputStream.write(mADTSHeader);

            mOutBuffer.get(mEncodedBytes, 0, mOutBufferInfo.size);
            mOutBuffer.clear();
            mMixedFileOutputStream.write(mEncodedBytes, 0, mOutBufferInfo.size);

            mLastAudioPresentationTimeUs = mOutBufferInfo.presentationTimeUs;
        }

        mAudioEncoder.releaseOutputBuffer(outputBufIndex, false);
    } else if (outputBufIndex == MediaCodec.INFO_OUTPUT_BUFFERS_CHANGED) {
        mAudioOutputBuffers = mAudioEncoder.getOutputBuffers();
    }
}

```

(/apps/download?  
utm\_source=sbc)

小礼物走一走，来简书关注我

赞赏支持





郑旭泽 (/u/dc225ec94e20)

写了 3865 字, 被 88 人关注, 获得了 79 个喜欢  
(/u/dc225ec94e20)

+ 关注

(/apps/download?  
utm\_source=sbc)

♡ 喜欢 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-like-button) | 30



更多分享

(http://cwb.assets.jianshu.io/notes/images/7705868/weibo/image\_7



(/apps/download?utm\_source=nbc)



登录 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-comment-form)

11条评论

只看作者

按喜欢排序 按时间正序 按时间倒序



JianShu\_bigT (/u/79413de72649)

2楼 · 2017.05.18 17:19

(/u/79413de72649)

int outputBufIndex = mAudioEncoder.dequeueOutputBuffer(mOutBufferInfo, 1000); 在  
执行这句话的时候 返回值为-2 也就是说走到了格式改变这一部分 这种原因可能有什么  
我该如何去做 (不用mux类 直接编码成h.264 然后打包推流) 望指点一二...😓

👍 赞    💬 回复





哎疯 (/u/899e2d74bdeb)

3楼 · 2017.09.21 19:12

(/u/899e2d74bdeb)

请问 用mediacodec编码h264后 播放出来的视频速度很快, 这是与pts有关的, 请问如何设置pts,网上都方法都没有效果。

👍 赞    💬 回复

this\_is\_for\_u (/u/96fea18cbfe3): 你现在这个问题解决了吗, 我也遇到这个问题了, 在性能比较高的机器上没什么问题, 到性能差的机器上这个问题就比较明显。

2017.11.06 22:53    💬 回复

哎疯 (/u/899e2d74bdeb): @this\_is\_for\_u (/users/96fea18cbfe3) 是的, 你说的没错, 之所以产生这个原因是我们在代码中做了两个比较耗时的操作第一个是旋转90度, 第二个是转换颜色格式, 在性能好的机子上没问题, 在性能不好的机子上有问题, 你可以把耗时的这两个操作放到jni层去做, 问题就可以解决了, 应该能帮到你。

2017.11.07 10:28    💬 回复

this\_is\_for\_u (/u/96fea18cbfe3): @哎疯 (/users/899e2d74bdeb) 好的, 谢谢, 我把视频用axure改成了mp4格式, 之后每帧加上时间戳就好啦

2017.11.17 21:04    💬 回复

✍️ 添加新评论    |    还有1条评论, 展开查看



asla (/u/1c861d2f4b22)

4楼 · 2017.10.13 14:42

(/u/1c861d2f4b22)

如果在连接相机的时候晃动手机, 刚开始看到画面的时候会有马赛克的现象, 这是怎么回事? 跟解码有关吗? 还是说是因为摄像头采集的数据有问题?

👍 赞    💬 回复

郑旭泽 (/u/dc225ec94e20): @asla (/users/1c861d2f4b22) 马赛克或花屏一般是丢帧, 特别关注帧

2017.10.13 18:11    💬 回复

asla (/u/1c861d2f4b22): 跟解码没有关系吗?

(/apps/download?  
utm\_source=src)





2017.10.13 18:25 回复

郑旭泽 (/u/dc225ec94e20): @asla (/users/1c861d2f4b22) 解码器一般不会出问题, 花屏比较常见的是丢帧导致

2017.10.13 18:27 回复

添加新评论

(/apps/download?  
utm\_source=sbc)



我和你\_164c (/u/de8ef940e01c)

5楼 · 2017.10.31 18:15

(/u/de8ef940e01c)

java.nio.BufferOverflowException

java.nio.Buffer.checkPutBoundsajava.nio.DirectByteBuffer.put

java.nio.ByteBuffer.put

请问有遇到过这个问题没? 非常感谢

赞 回复

被以下专题收入, 发现更多相似内容



Android知识 (/c/3fde3b545a35?utm\_source=desktop&utm\_medium=notes-included-collection)



android杂记 (/c/51f7a870dcab?utm\_source=desktop&utm\_medium=notes-included-collection)



音视频开发经验之路 (/c/404039678493?  
utm\_source=desktop&utm\_medium=notes-included-collection)





移动流媒体 (/c/a4254ed19b65?utm\_source=desktop&utm\_medium=notes-included-collection)



android (/c/293ada6864f3?utm\_source=desktop&utm\_medium=notes-included-collection)



 Android... (/c/723f9851c1b2?utm\_source=desktop&utm\_medium=notes-included-collection)

 音视频处理 (/c/207472d1f8b5?utm\_source=desktop&utm\_medium=notes-included-collection)

(/apps/download?utm\_source=src)


(/p/35857aff28bc?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

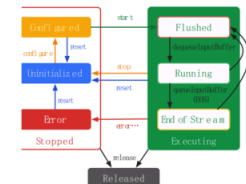
**MediaCodec (/p/35857aff28bc?utm\_campaign=maleskine&utm\_conten...**

原文: <https://developer.android.com/reference/android/media/MediaCodec.html> MediaCodec class can be used to access low-level media codecs, i...

 thebestofrocky (/u/f90292cbab13?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/8e6c87b105d0?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

**Android MediaCodec (/p/8e6c87b105d0?utm\_campaign=maleskine&ut...**

MediaCodec的官方文档 一、Android MediaCodec简单介绍 Android中可以使用MediaCodec来访问底层的媒体编解码器，可以对媒体进行编/解码。MediaCodec可以处理的数据有以下三种类型：压缩数据、原始音...

 黄海佳 (/u/05402282c2d9?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/14da1baaf08f?

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)



## android中MediaCodec类解析 (/p/14da1baaf08f?utm...

简介 MediaCodec类可以获取底层媒体编码/解码库，是Android底层多媒体支持库的一部分（一般和MediaExtractor、MediaSync、MediaMuxer、...



昵称真难选 (/u/d6cb6f6d87b3?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation) (/apps/download?utm\_source=sbc)



(/p/2721548cbae4?

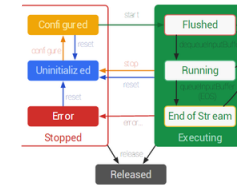
utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## MediaCodec中文API (/p/2721548cbae4?utm\_campa...

MediaCodec类可用于访问低级媒体编解码器，即编码器/解码器组件。它是Android低级多媒体支持架构的一部分（通常与MediaExtractor，MediaSync...

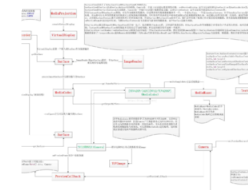


Karolinezhang (/u/ea61839c2b91?)



utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/392497c60993?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 转：Android中直播视频技术探究之---基础知识大纲介绍 (/p/392497c60993...

原文详见：<http://www.2cto.com/kf/201607/525000.html> 一、前言 最近各种视频直播app到处都是，各种霸屏，当然我们也是需要体验的，关于视频直播的软件这里就不介绍了，在不是技术的人来看，直播是一种...



玉圣 (/u/65e19270f991?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)



(/p/a9a9fb10751e?



(/apps/download?  
utm\_source=sbc)

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 不舍与银杏叶 (/p/a9a9fb10751e?utm\_campaign=maleskine&utm\_conte...

连洗完澡关上储物柜的一瞬间都觉得不舍 我捡了一朵银杏叶



三里三 (/u/33965e2e0c13?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/eebcb239d28c?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 听张宇的沧桑 (/p/eebcb239d28c?utm\_campaign=maleskine&utm\_cont...

很多年前， 浪荡的“大女人”跳跃在大街小巷里， 从节奏欢快的音乐声中走过的时候， 内心总能突然奔放起来。

对于渴望解放内心矜持的女人来说， 恰巧犹如获得自由般的快活。 幻想着， 如果有个男人， 能接纳你所有...



榴莲的柔软 (/u/ab13895136ce?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/4700923fd945?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

## 传承民俗文化，从娃娃抓起 (/p/4700923fd945?utm\_campaign=maleskine...

2016年12月8日，在埃塞俄比亚首都的斯亚贝巴，中国传统民俗文化之一——二十四节气申遗成功，被正式

列入联合国教科文组织人类非物质文化遗产代表作名录。这一成功，既是现代人对古代人智慧的一种肯定...



清浅光阴 (/u/63bf7222b79?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)



---

(/p/f12d2c5fc365?



(/apps/download?  
utm\_source=sbc)

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

### 贵州小七孔即景 (/p/f12d2c5fc365?utm\_campaign=maleskine&utm\_cont...

贵州小七孔是我走过的自我感觉最似仙境的地方，赋诗二首，以作纪念。其一 苍寂桂黔陌，玲珑碧水间。  
长河宕飞雪，乳霭浣重山。瑶鼓彩云动，苗歌翠嶂传。无须青鸟使，瑶水洗尘颜。其二 仙女一凝思， ...



霞彩w (/u/ed36a00fe2ef?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

---

(/p/2f7282c8277e?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

### 嘴上人生自证预言 (/p/2f7282c8277e?utm\_campaign=maleskine&utm\_c...

有没有发现，人们总在过着跟自己嘴上说的情况相似的人生？比如，整天哭穷的人好像真的也过得很穷，喜欢抱怨的人，好像生活中也总是遇到不顺心的事情。这听上去有点像玄学？其实不然，这中间也是有科学...



魏德强Mike (/u/fd5635803294?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

