

顶部 shui 已于 2022-07-21 16:10:44 修改 9806 收藏 24

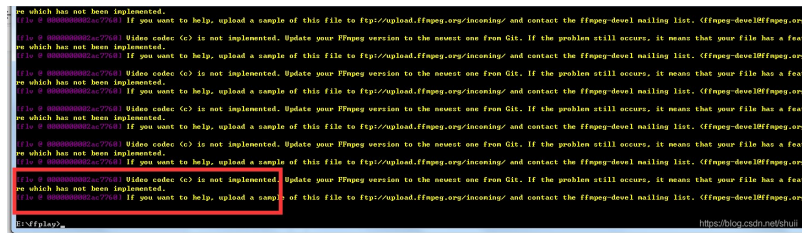
分类专栏: 音视频 ffmpeg 文章标签: ffmpeg hevc flv flv-rtmp rtmp

立即加入开发者社区 文章已被社区收录 加入社区

音视频 同时被 2 个专栏收录 1 订阅 2 篇文章 订阅专栏

很多初学者不知道rtmp或者ffmpeg是不支持flv封装的rtmp/http-flv流的，其原因是flv不支持H265编码payload的，因为当时制定flv封装协议的时候，H265还没出来，现在Adobe又要淘汰flashie，自然不想支持了。但是我们当前大部分直播平台还是使用rtmp来推流，又想使用H265/Hevc来编码视频数据，怎么办呢，只能修改ffmpeg的flv解复用的相关部分代码。

先来看看使用原版的rtplay来拉H265的rtmp流，是一个什么样的效果：



我们可以看到，上图中提示的错误信息，首先红色的部分，提示的是flv @开通的错误，说明是flv解复用层的错误，然后看见是黄色的Video codec 没有实现，那就是flv解复用的问题了。

修改方法：

1. 新增一个video codec id:

先看看flv官方文档，视频包 Video Tags的解析字段：

### E.4.3 Video Tags

Video tags are similar to the VideoFrame tag in the SWF file format, and their payload data is identical.

See also the SWF File Format Specification at [http://www.adobe.com/go/swf\\_file\\_format](http://www.adobe.com/go/swf_file_format)

#### E.4.3.1 VIDEODATA

The VideoTagHeader contains video-specific metadata.

##### VideoTagHeader

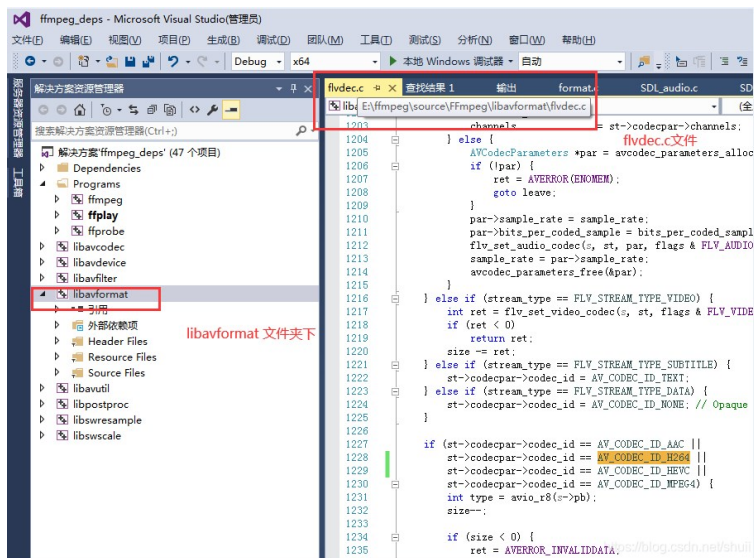
Field	Type	Comment
Frame Type	UB [4]	Type of video frame. The following values are defined: 1 = key frame (for AVC, a seekable frame) 2 = inter frame (for AVC, a non-seekable frame) 3 = disposable inter frame (H.263 only) 4 = generated key frame (reserved for server use only) 5 = video info/command frame
CodedID	UB [4]	Codec Identifier. The following values are defined: 2 = Sorenson H.263 3 = Screen video 4 = On2 VP6 5 = On2 VP6 with alpha channel 6 = Screen video version 2 7 = AVC
AVCPacketType	IF CodedID == 7 UI8	The following values are defined: 0 = AVC sequence header 1 = AVC NALU 2 = AVC end of sequence (lower level NALU sequence ender is not required or supported)
CompositionTime	IF CodedID == 7 SI24	IF AVCPacketType == 1 Composition time offset ELSE 0 See ISO 14496-12, 8.15.3 for an explanation of composition times. The offset in an FLV file is always in milliseconds.

7 = AVC，即H264  
那么我们需要自顶一个类型：  
12 = HEVC

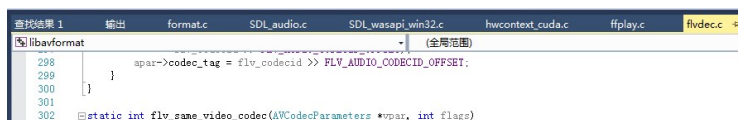
Codedid 占4 bits，范围是[0, 15] 总共能表示16种解码ID，例如 Codedid = 7,表示 AVC编码，也就是H264，那么8-15都是没有使用的，这个区间我们可以自顶一个类型，作为H265的ID，国内大厂可能已经达成共识，12表示HEVC，也就说12是H265的。

在代码中新增一个枚举值：

代码位置为：libavformat文件夹下的flvdec.c文件，如下图：



先看看FLV\_CODEC\_ID\_H264在哪定义的，选中他->右键->转到定义：



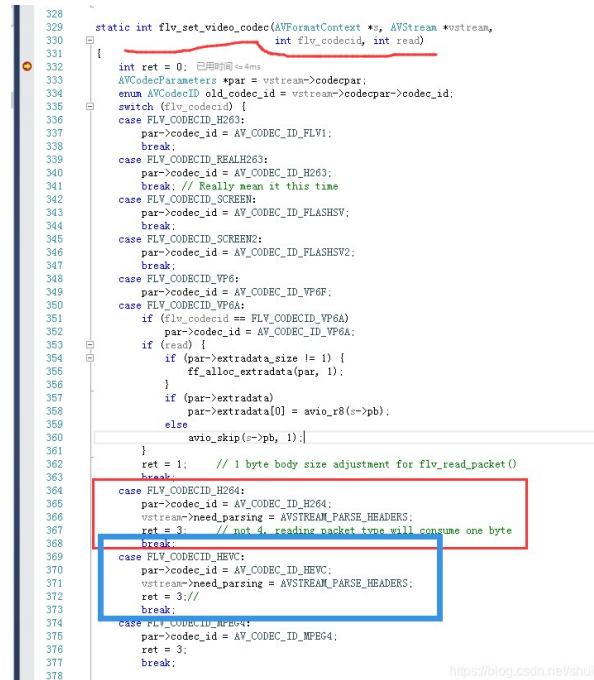


然在FLV\_CODECID\_H264后, 新增一个FLV\_CODECID\_HEVC:

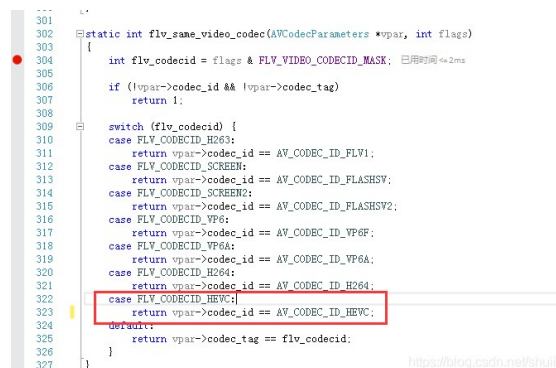


如何将FLV\_CODECID\_HEVC和FLV的Video Tag中的 CodecID关联起来:

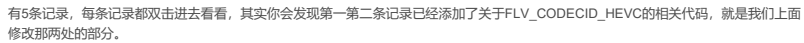
flv解复用的时候, 发现Video Tag会先将 codecid 存起来, 通过下面这个函数来实现, 红色框起来那个是H264的, 蓝色框那个case语句是我们加进去的, FLV\_CODECID\_HEVC就是刚才我们定义的12, 这里其实只需要模仿H264就可以了, codecid, need\_parsing还有ret = 3, 这几个值就好了。



接下来, 还会重新校验一次, avformat\_find\_stream\_info的时候, 内部会调用这个flv\_same\_video\_codec, 校验一次flv\_codecid, 这里也是需要添加一个FLV\_CODECID\_HEVC的case分支来判断的:



需要修改的地方不多，我们可以直接通过仿照H264的方法来修改，先在当前文档flvdec.c下查找AV\_CODEC\_ID\_H264，



```

1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261

```

```

if (st->codecpar->codec_id == AV_CODEC_ID_AAC ||
    st->codecpar->codec_id == AV_CODEC_ID_H264 ||
    st->codecpar->codec_id == AV_CODEC_ID_HEVC ||
    st->codecpar->codec_id == AV_CODEC_ID_MPEG4) {
    int type = avio_r8(s->pb);
    size--;

    if (size < 0) {
        ret = AVERORR_INVALIDDATA;
        goto leave;
    }

    if (st->codecpar->codec_id == AV_CODEC_ID_H264 || st->codecpar->codec_id == AV_CODEC_ID_HEVC
        || st->codecpar->codec_id == AV_CODEC_ID_MPEG4) {
        // sign extension
        int32_t cts = (avio_rb24(s->pb) & 0xffff800000) | 0xffff800000;
        pts = cts + cts;
        if (cts < 0) { // dts might be wrong
            if (!flv->wrong_dts)
                av_log(s, AV_LOG_WARNING,
                    "Negative cts, previous timestamps might be wrong.\n");
            flv->wrong_dts = 1;
        } else if (FFABS(cts - pts) > 1000*60*15) {
            av_log(s, AV_LOG_WARNING,
                "Invalid timestamps %\"PRIi64\" %\"PRIi64\"\\n", dts, pts);
            dts = pts = AV_NOPTS_VALUE;
        }
    }
}

if (type == 0 && (!st->codecpar->extradata || st->codecpar->codec_id == AV_CODEC_ID_AAC ||
    st->codecpar->codec_id == AV_CODEC_ID_H264 ||
    st->codecpar->codec_id == AV_CODEC_ID_HEVC)) {
    av_freep(&st->extradata);
}

if (st->codecpar->extradata) {
    if ((ret = flv_queue_extradata(flv, s->pb, stream_type, size)) < 0)

```

效果如下：

[illegible]

