

# 一、M3U8 格式标准介绍

M3U8文件是指UTF-8编码格式的M3U文件。M3U文件是记录了一个索引纯文本文件，打开它时播放软件并不是播放它，而是根据它的索引找到对应的音视频文件的网络地址进行在线播放。

M3U8是一种常见的流媒体格式，主要以文件列表的形式存在，既支持直播又支持点播，尤其在Android、iOS等平台最为常用。

下面是CCTV6直播播放地址：<http://ivi.bupt.edu.cn/hls/cctv6hd.m3u8>的M3U8的文件列表：



```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:35232
#EXT-X-TARGETDURATION:10
#EXTINF:10.000,
cctv6hd-1549272376000.ts
#EXTINF:10.000,
cctv6hd-1549272386000.ts
#EXTINF:10.000,
cctv6hd-1549272396000.ts
#EXTINF:10.000,
cctv6hd-1549272406000.ts
#EXTINF:10.000,
cctv6hd-1549272416000.ts
#EXTINF:10.000,
cctv6hd-1549272426000.ts
```



下面我们来分别说明一下相关的几个字段：

- EXTM3U：这个是M3U8文件必须包含的标签，并且必须在文件的第一行，所有的M3U8文件中必须包含这个标签。
- EXT-X-VERSION：M3U8文件的版本，常见的是3（目前最高版本应该是7）。
- EXT-X-TARGETDURATION：该标签指定了媒体文件持续时间的最大值，播放文件列表中的媒体文件在EXTINF标签中定义的持续时间必须小于或者等于该标签指定的持续时间。该标签在播放列表文件中必须出现一次。
- EXT-X-MEDIA-SEQUENCE：M3U8直播是的直播切换序列，当播放打开M3U8时，以这个标签的值作为参考，播放对应的序列号的切片。
- EXTINF：EXTINF为M3U8列表中每一个分片的duration，如上面例子输出信息中的第一片的duration为10秒。在EXTINF标签中，除了duration值，还可以包含可选的描述信息，主要为标注切片信息，使用逗号分隔开。

关于客户端播放M3U8的标准还有更多的讲究，下面我们来介绍一些：

- 分片必须是动态改变的，序列不能相同，并且序列必须是增序的。
- 当M3U8没有出现EXT-X-ENDLIST标签时，无论这个M3U8列表中有多少分片，播放分片都是从倒数第三片开始播放，如果不满三片则不应该播放。当然如果有些播放器做了特别定制了，则可以不遵照这个原则。
- 以播放当前分片的duration时间刷新M3U8列表，然后做对应的加载动作。
- 前一片分片和后一片分片有不连续的时候，播放可能会出错，那么需要X-DISCONTINUTY标签来解决这个错误。
- 如果播放列表在刷新之后与之前的列表相同，那么在播放当前分片duration一半的时间内在刷新一次。

在上面，我们提到了，一些上面例子没有出现的一些标签字段，下面我们针对一些额外的标签做一些补充说明：

- EXT-X-ENDLIST：若出现EXT-X-ENDLIST标签，则表明M3U8文件不会再产生更多的切片，可以理解为该M3U8已停止更新，并且播放分片到这个标签后结束。M3U8不仅仅是可以作为直播，也可以作为点播存在，在M3U8文件中保存所有切片信息最后使用EXT-X-ENDLIST结尾，这个M3U8即为点播M3U8。EXT-X-ENDLIST标签可能会出现在播放列表文件的任何地方，但是不能出现两次或以上。
- EXT-X-STREAM-INF：EXT-X-STREAM-INF标签出现在M3U8时，主要是出现在多级M3U8文件中时，例如M3U8中包含子M3U8列表，或者主M3U8中包含多码率M3U8时；该标签后需要跟一些属性，下面就来逐一说明一下这些属性：
  - BANDWIDTH：BANDWIDTH的值为最高码率值，当播放EXT-X-STREAM-INF下对应的M3U8时占用的最大码率（必要参数）。
  - AVERAGE-BANDWIDTH：AVERAGE-BANDWIDTH的值为平均码率值，当播放EXT-X-STREAM-INF下对应的M3U8时占用的平均码率。（可选参数）。
  - CODECS：CODECS的值用于声明EXT-X-STREAM-INF下面对应M3U8里面的音视频编码、视频编码的信息（可选参数）。
  - RESOLUTION：M3U8中视频的宽高信息描述（可选参数）。
  - FRAME-RATE：子M3U8中的视频帧率（可选参数）。

## 二、HLS 与 M3U8

HLS（全称：Http Live Streaming）是由Apple公司定义的用于实时流传输的协议，HLS基于HTTP协议实现，传输内容包括两部分，一是M3U8描述文件，二是TS媒体文件。

HLS的优势为：自适应码率流播（adaptive streaming）。效果就是客户端会根据网络状况自动选择不同码率的视频流，条件允许的情况下使用高码率，网络繁忙的时候使用低码率，并且能够自动在二者之间随意切换。这对移动设备网络状况不稳定的情况下保障流畅播放非常有帮助。实现方法是服务器端提供多码率视频流，并且在列表文件中注明，播放器根据播放进度和下载速度进行自动调整。

为什么要用 TS 而不是 MP4？这是因为两个 TS 片段可以无缝拼接，播放器能连续播放，而 MP4 文件由于编码方式的原因，两段 MP4 不能无缝拼接，播放器连续播放两个 MP4 文件会出现破音和画面间断，影响用户体验。而且如果要在一段长达一小时的视频中跳转，如果使用单个 MP4 格式的视频文件，并且也是用 HTTP 协议，那么需要代理服务器支持 HTTP range request 获取大文件中的一部分。这样的话，对于代理服务器的性能来说要求较高。而 HTTP Live Streaming 则只需要根据列表文件中的时间轴找出对应的 TS 片段下载即可，不需要 range request，对代理服务器的要求小很多。所有代理服务器都支持小文件的高效缓存。

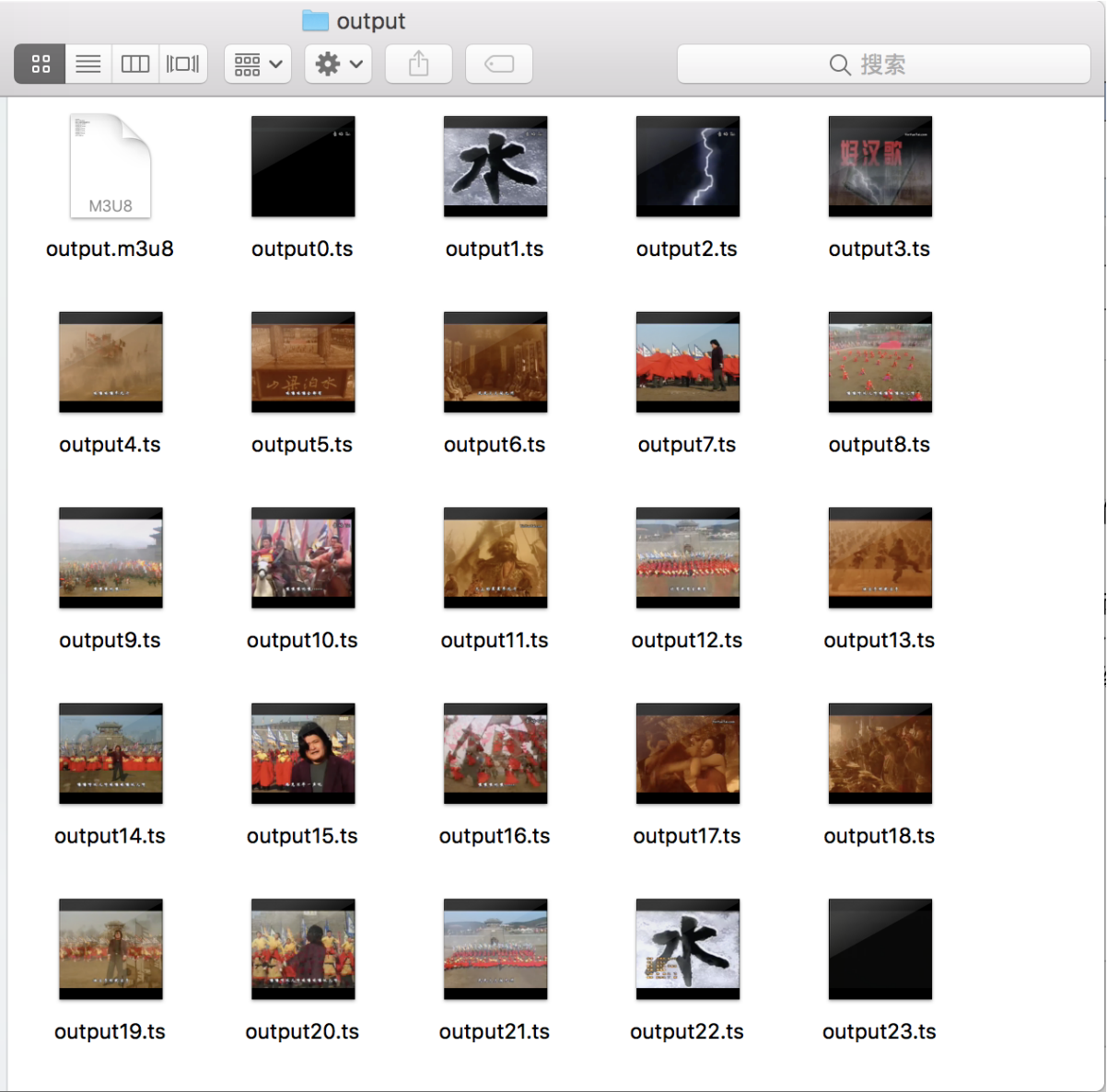
## 三、FFmpeg转HLS文件(M3U8)实战

### 1. FFmpeg转MP4为HLS(M3U8)文件

将MP4文件转换成HLS(M3U8)命令行：

ffmpeg -re -i 好汉歌.mp4 -c copy -f hls -bsf:v h264\_mp4toannexb output.m3u8

可以看到生成的M3U8及相应的ts文件：



查看一下生成的M3U8文件：

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:19
#EXTINF:10.000000,
output19.ts
#EXTINF:10.000000,
output20.ts
#EXTINF:9.280000,
output21.ts
#EXTINF:4.120000,
output22.ts
#EXTINF:2.440000,
output23.ts
#EXT-X-ENDLIST
```

细心的人可能发现一个问题，就是生成的m3u8文件里只有最后的五个片段的信息。这是因为ffmpeg 默认的list size 为5，所以只获得最后的5个片段。为了解决这个问题，需要指定参数-hls\_list\_size 0，这样就能包含所有的片段。

下面是优化后的命令行：

```
ffmpeg -re -i 好汉歌.mp4 -c copy -f hls -hls_list_size 0 -bsf:v h264_mp4toannexb output.m3u8
```

这时，我们可以看到从output0.ts到output23.ts的文件列表了。

可能有人会发现，无论是优化之前的命令行，还是优化后的命令行都有一个参数-bsf:v h264\_mp4toannexb，这个参数的作用是将MP4中的H.264数据转换成为H.264 AnnexB标准的编码，AnnexB标准的编码常见于实时传输流中。如果源文件为FLV、TS等可以作为直播传输流的视频，则不需要这个参数。

下面我们逐一介绍下使用FFmpeg生成HLS时还可以配置的其他参数。

## 四、FFmpeg 转 HLS (M3U8) 文件命令参数

### 1. start\_number 参数

start\_number 参数用于设置M3U8列表中的第一片的序列数。

下面的例子中，我们使用start\_number参数设置M3U8中的第一片序列书为100，命令行如下：

```
ffmpeg -re -i huijia.mp4 -c copy -f hls -start_number 100 -hls_list_size 0 -bsf:v h264_mp4toannexb output.m3u8
```

输出的M3U8内容如下：

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:3
```

```
#EXT-X-MEDIA-SEQUENCE:100
#EXTINF:3.000000,
output100.ts
#EXTINF:3.000000,
output101.ts
#EXTINF:3.000000,
output102.ts
#EXTINF:3.000000,
output103.ts
#EXTINF:3.000000,
output104.ts
#EXTINF:3.000000,
output105.ts
#EXTINF:3.000000,
output106.ts
#EXTINF:1.000000,
output107.ts
#EXT-X-ENDLIST
```



从输出可以看出，切片的第一片编号是100，上面的命令行参数的-start\_number参数已生效。

## 2. hls\_time 参数

hls\_time参数用于设置M3U8列表中切片的duration。

下面的例子中，我们使用hls\_time参数设置M3U8的TS文件的每一片时长为9秒左右。命令行如下：

```
ffmpeg -re -i huijia.mp4 -c copy -f hls -hls_time 9 -hls_list_size 0 -bsf:v h264_mp4toannexb output.m3u8
```

然后查看输出的M3U8内容如下：



```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:9
#EXT-X-MEDIA-SEQUENCE:0
#EXTINF:9.000000,
output0.ts
#EXTINF:9.000000,
output1.ts
#EXTINF:4.000000,
output2.ts
#EXT-X-ENDLIST
```



可以看到TS的文件每一片的时常都是9秒左右，hls\_time参数生效。

(注意：hls\_time设置后效果不一定准确，会受到关键帧大小及其他因素影响。)

如果需要相对非常准确的切片，可以添加hls\_flags的子参数split\_by\_time来保证生成的切片能够与hls\_time设置的切片时长差不多。

(注意：split\_by\_time参数必须与hls\_time配合使用，并且使用split\_by\_time参数有可能会影响首画面体验，例如花屏或者首画面显示慢的问题，因为视频的第一帧不一定是关键帧。)

## 3. hls\_list\_size 参数

hls\_list\_size参数用于为M3U8列表中的TS切片的个数。其中设置为0的时候，将包含所有。

这个命令，我们在第3节优化MP4转HLS文件的命令行时使用到了。

下面的例子中，我们使用hls\_list\_size参数设置只保留2片TS切片。命令行如下：

```
ffmpeg -re -i huijia.mp4 -c copy -f hls -hls_list_size 2 -bsf:v h264_mp4toannexb output.m3u8
```

查看输出的M3U8内容如下：



```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:3
#EXT-X-MEDIA-SEQUENCE:6
#EXTINF:3.000000,
output6.ts
#EXTINF:1.000000,
output7.ts
#EXT-X-ENDLIST
```



从输出的M3U8内容可以看出，在M3U8文件中只保留了2片TS的文件信息，可以看出hls\_list\_size设置生效了。

## 4. hls\_base\_url 参数

hls\_base\_url 参数用于为M3U8列表的文件路径设置前置基本路径参数，因为在FFmpeg中生成M3U8时写入的TS切片路径默认为M3U8生成的路径相同，但是实际上TS所存储的路径既可以为本地绝对路径，也可以为相对路径，还可以为网络路径，因此使用hls\_base\_url参数可以达到该效果，命令行如下：

```
ffmpeg -re -i huijia.mp4 -c copy -f hls -hls_base_url /Users/renhui/Desktop/test/ -bsf:v h264_mp4toannexb output.m3u8
```

查看输出的M3U8内容如下：



```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:3
#EXT-X-MEDIA-SEQUENCE:3
#EXTINF:3.000000,
/Users/renhui/Desktop/test/output3.ts
#EXTINF:3.000000,
/Users/renhui/Desktop/test/output4.ts
#EXTINF:3.000000,
/Users/renhui/Desktop/test/output5.ts
#EXTINF:3.000000,
/Users/renhui/Desktop/test/output6.ts
#EXTINF:1.000000,
/Users/renhui/Desktop/test/output7.ts
#EXT-X-ENDLIST
```



可以看到，TS的路径变为绝对路径了，使用ffplay output.m3u8播放，看到播放是能够正常播放的。这样就可以说明hls\_base\_url生效了。