**FLV结构**

##### 简述

FLV（flash video）是一种视频**封装格式，后缀名有flv、f4v。**

主要包括：FLV Header和FLV Body两部分。

Header包括文件类型，流信息(是否包含视频音频)，header长度（9字节）等信息。

Body由一个个Tag组成，每个tag前面是上一个tag的大小，PreviousTagSizeN为TagN的大小, 占4字节，包括Tag Header(一般为11字节)和Tag Data的大小之和。

Tag包含的信息主要为: **视频、音频或脚本信息。**Body中一般第一个Tag为脚本信息(meta data, 包含视频音频的编码格式，视频的宽高信息等，该类型的Tag一般有且仅有一个)，其后的Tag为Video或Audio的Tag。

二、FLV Header

FLV Header一般为9个字节：  
文件类型(3字节)：固定为16进制的0x46 0x4c 0x56,即字符F L V的ASCII值。

版本信息(1字节)：一般为0x01

流信息(1字节)：倒数第一位是1表示有视频，倒数第三位是1表示有音频，倒数第二、四位必须为0。

头长度(4字节)：一般为9。

三、FLV Body

1、FLV Tag Header 11字节：

Reserved 2BIT

Filter 1BIT

TagType 5BIT

Datasize 24BIT

Timestamp 24BIT

TimestampExtended 8BIT

StreamID 24BIT

第一个字节的后5位用于表示Tag的类型：

8 -- 音频

9 -- 视频

18 -- script data 也就是 metadata

第2字节到第4字节为DataSize，也就是Tag Data的大小(单位是字节，不包含这11字节的头)。也就是说Tag Data的最大值为2的24次方-1(0也算是一个数)。

PTS= Timestamp | TimestampExtended<<24 解码时的显示时间戳严格按照此数据执行，其它数据块里的PTS不会被FLV播放器读取。

三、script tag

flv文件头后面跟着的第一个tag一般是script tag，当TagType == 18时表示该tag是SCRIPTDATA，这个Tag Data里面在未加密的时候，是ScriptTagBody类型，里面的SCRIPTDATA编码为AMF，ScriptTagBody由Name和Value两个字段组成。类型均为SCRIPTDATAVALUE，可用的类型如下：

0 = Number

1 = Boolean

2 = String

3 = Object

4 = MovieClip (reserved, not supported)

5 = Null

6 = Undefined

7 = Reference

8 = ECMA array

9 = Object end marker

10 = Strict array

11 = Date

12 = Long string

一般来说，该Script Tag Data结构只包含两个AMF包：

第一个AMF包封装字符串类型数据，用来装入一个”onMetaData”标志。其中第1个字节表示AMF包类型，一般总是0x02，表示字符串，第2-3个字节为UI16类型值，表示字符串的长度，一般总是0x000A（“onMetaData”长度）。后面字节为字符串数据，一般总为“onMetaData”。

第二个AMF包封装一个ECMA数组类型，这个数组中包含了音视频信息项的名称和值。一般包含：FLV的音频，视频还有文件相关的基本信息，如：文件大小，时长，视频帧率码率宽高编码格式，音频码率采样率采样深度声道数码率编码格式等。其中第1个字节表示AMF包类型，一般总是0x08，表示ECMA 数组。第2-5个字节为UI32类型值，表示数组元素的个数。后面即为各数组元素的封装，数组元素为元素名称和值组成的键值对，表示方法如下：

第1-2个字节表示元素名称的长度，假设为L。后面跟着为长度为L的字符串。

第L+3个字节表示元素值的类型。SCRIPTDATAVALUE类型，后面跟着对应值，占用字节数取决于值的类型，比如0表示数值double类型，占8字节，1表示bool类型，占1个字节。

Script tag以0x000009作为结束标志。

keyframes：keyframes的信息藏在SCRIPTDATA中。

keyframes几乎是一个非官方的标准，也就是民间标准。在网上已经很难看到flv文件的metadata里面不包含 keyframes项目的视频。

也就是说keyframes中包含着2个内容filepositions and times分别指的是关键帧的文件位置和关键帧的PTS。通过keyframes可以建立起自己的Index，然后在seek的操作中，快速有效的跳转到你想要找的关键帧的位置进行处理。

四、video tag

当TagType == 09时表示该tag是video tag。

tag data中的video header：长度1字节：

前4位表示帧信息：值为1: keyframe (for AVC, a seekable frame)；2: inter frame (for AVC, a nonseekable frame)；3: disposable inter frame (H.263 only)；4: generated keyframe (reserved for server use only)；5: video info/command frame。后4位标识视频编码。1: JPEG (currently unused)；2: Sorenson H.263；3: Screen video；4: On2 VP6； 5: On2 VP6 with alpha channel；6: Screen video version 2；7: AVC

如果codecID是avc，则后面4个字节是：第一个字节：AVCPacketType。2 3 4字节是CompositionTime。

AVCPacketType：

0 = AVC sequence header AVC的序列参数信息

1 = AVC NALU avc一般的数据负载单元NAL

2 = AVC end of sequence (lower level NALU sequence ender is not required or supported)

如果是avc编码格式，则第一个tag一定是AVC sequence header即AVCDecoderConfigurationRecord。该tag包含着H.264解码相关比较重要的sps和[pps](https://www.baidu.com/s?wd=pps&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)信息，再给AVC解码器送数据流之前一定要把sps和pps信息送出，否则的话解码器不能正常解码。而且在解码器stop之后再次start之前，如seek、快进快退状态切换等，都需要重新送一遍sps和pps的信息。在FLV文件中一般情况也是出现1次，也就是第一个video tag。

AVCDecoderConfigurationRecord包含h264的profile信息、sps个数长度及nalu、pps个数长度及nalu，一般h264编码的sps nalu以0x67开始，pps的nalu以0x68开始。

第1字节：版本号，2-4字节sps[i]，5字节保留，6字节后五位是sps数组个数。7-8字节为第一个sps长度，后面是第一个sps值。sps结束后第1字节是pps数组个数，2-3字节是第一个pps长度，后跟第一个pps值。

如果是nalu单元，则以0x000001或0x00000001开始，起始码。

五、audio tag

如果TAG包中的TagType==8时，就表示这个TAG是audio。StreamID之后的数据就表示是AudioTagHeader，AudioTagHeader结构如下：

SoundFormat：4bit，标识音频数据的格式，如：0x2表示的是MP3数据，当前合法的数值为0，1，2，3，4，5，6，7，8，9，10，11，14，15（7，8，14，15保留为内部使用），10表示AAC。

SoundRate：2bit，第5，6位bit表示采样率（AAC，总是44khz）

SoundSize：1bit，采样精度，0为8bits，1为16bits

SoundType：1bit，音频类型，mono=0，stereo=1

如果是AAC则多出1字节：AACPacketType：0 = AAC sequence header，1 = AAC raw。

AAC sequence header也就是包含了AudioSpecificConfig，AudioSpecificConfig包含着一些更加详细音频的信息，具体只有2字节：

audioObjectType 5bits 编码类型，AAC-LC为2

samplingFreguencyIndex 4bits 音频采样率索引值

channelConfiguration 4bits 音频声道数

GASpecificConfig

frameLengthFlag 1bits 标志位，用于表明IMDCT窗口长度，为0

dependsOnCoreCoder 1bits 标志位，用于表明是否依赖core coder，为0

extensionFlag 1bits 扩展标志位，选择了AAC-LC，这里必须为0

解码需要AudioSpecificConfig中的采样率信息和采样通道信息。在FLV的文件中，一般情况下AAC sequence header这种包只出现1次，而且是第一个audio tag，如果是AAC的音频，需要在每帧AAC ES流前边添加7个字节ADTS头，就是AAC的纯ES流要打包成ADTS格式的AAC文件才能被解码器正确解码。

AAC raw这种包含的就是音频ES流，不带adts头。