

## 原 视音频编解码学习工程：FLV封装格式分析器

2014年01月12日 00:01:19 阅读数：38365

=====

视音频编解码学习工程系列文章列表：

[视音频编解码学习工程：H.264分析器](#)

[视音频编解码学习工程：AAC格式分析器](#)

[视音频编解码学习工程：FLV封装格式分析器](#)

[视音频编解码学习工程：TS封装格式分析器](#)

[视音频编解码学习工程：JPEG分析器](#)

=====

本文介绍一个自己的开源小项目：FLV封装格式分析器。FLV全称是Flash Video，是互联网上使用极为广泛的视频封装格式。像Youtube，优酷这类视频网站，都使用FLV封装视频。我这个项目规模不大，主要可以用来学习FLV封装格式结构。此外它还支持分离FLV中的视频流和音频流。使用VC 2010的MFC开发完成。在对FLV进行视音频分离的过程中，用到了一个Github开源小工程：flvparse。在此插一句：我发现Github上优秀的东西真的还是挺多的，许多零散的小工程，效果都很不错。这个flvparse做的就不错。

软件的exe以及源代码已经上传到了SourceForge上。和之前的H.264码流分析器一样，增加了一个英文界面，紧跟国际潮流~

项目地址：<https://sourceforge.net/projects/flvformatanalysis/>

CSDN下载地址（程序+源代码）：<http://download.csdn.net/detail/leixiaohua1020/6838805>



更新记录=====

1.1版（2014.7.8）

- \* 更换了界面
- \* 原工程支持Unicode编码
- \* 支持中英文切换

CSDN源代码：<http://download.csdn.net/detail/leixiaohua1020/7767613>

PUDN源代码：<http://www.pudn.com/downloads644/sourcecode/multimedia/detail2605189.html>

新版（2016.1.1）

- \* 精简了代码，使之更通俗易懂
- \* 修改了少量界面UI
- \* 修正了少量解析错误
- \* 添加了对TagData首字节的解析

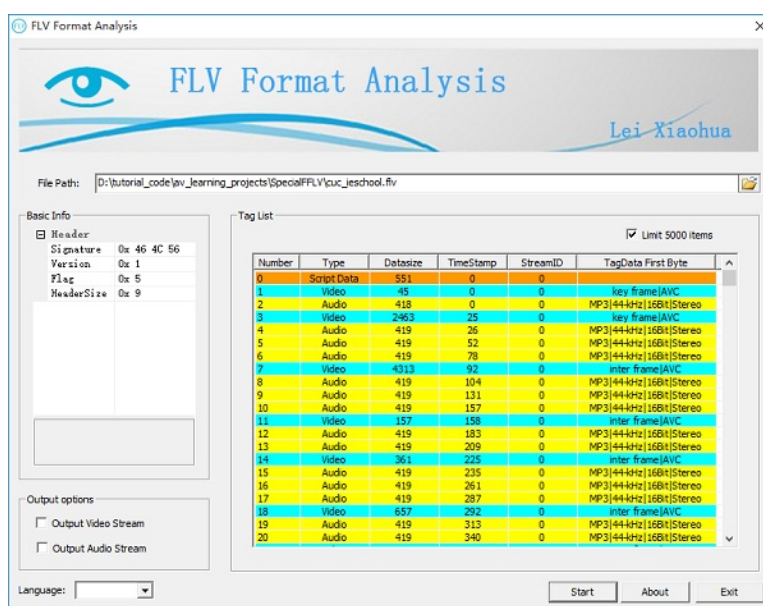
已经更新至SourceForge上

### 软件使用介绍

软件的使用相当简单。软件运行后，首先打开一个FLV文件。单击“开始”，可以解析出一系列Tag，列表显示在软件右侧。不同种类的Tag被标记成了不同的颜色。列表中包含了每个Tag的类型、大小、时间戳、StreamID、TagData首字节。软件的左侧，显示了FLV文件头信息。



此外软件做了一个英文界面，如下所示。



注：如果勾选上“输出视频”，“输出音频”的话，可以输出分离后的视频流和音频流。在这里要注意的是音频支持MP3格式，AAC格式貌似有点问题。

## 软件源代码简析

源代码方面和普通的MFC程序差不多，懂得MFC的人应该很快就能看懂。唯一比较特殊的地方，就在于对开源项目flvparse进行了一些改动，在此就不细说了。注释方面还是很充分的。

## FLV封装原理

FLV格式的封装原理，贴上来辅助学习之用。

FLV (Flash Video) 是Adobe公司设计开发的一种流行的流媒体格式，由于其视频文件体积小、封装简单等特点，使其很适合在互联网上进行应用。此外，FLV可以使用Flash Player进行播放，而Flash Player插件已经安装在全世界绝大部分浏览器上，这使得通过网页播放FLV视频十分容易。目前主流的视频网站如优酷网，土豆网，乐视网等网站无一例外地使用了FLV格式。FLV封装格式的文件后缀通常为“.flv”。

总体上看，FLV包括文件头（File Header）和文件体（File Body）两部分，其中文件体由一系列的Tag组成。因此一个FLV文件是如图1结构。



图1.文件结构（简图）

其中,每个Tag前面还包含了Previous Tag Size字段,表示前面一个Tag的大小。Tag的类型可以是视频、音频和Script,每个Tag只能包含以上三种类型的数据中的一种。图2展示了FLV文件的详细结构。

|            |   |            |  |
|------------|---|------------|--|
| Flv Header | Signature (3 字节) 为文件标识, 总为"FLV", (0x46, 0x4c, 0x66)                             |            |  |
|            | Version (1 字节) 为版本, 目前为 0x01  |            |  |
|            | Flags (1 字节) 前 5 位保留, 必须为 0。第 6 位表示是否存在音频 Tag。第 7 位保留, 必须为 0。第 8 位表示是否存在视频 Tag。 |            |  |
|            | Headersize (4 字节) 为从 File Header 开始到 File Body 开始的字节数, 版本 1 中总为 9。              |            |  |
| Flv Body   | Previous Tag Size #0 (4 字节) 表示前一个 Tag 的长度                                       |            |  |
|            | Tag #1  | Tag Header | Type (1 字节) 表示 Tag 类型, 包括音频 (0x08), 视频 (0x09) 和 script data (0x12), 其他类型值被保留 |
|            |   |            | Datasize (3 字节) 表示该 Tag Ddata 部分的大小  |
|            |   |            | Timestamp (3 字节) 表示该 Tag 的时间戳  |
|            |   |            | Timestamp_ex (1 字节) 表示时间戳的扩展字节, 当 24 位数值不够时, 该字节最为最高位将时间戳扩展为 32 位数值          |
|            |   |            | StreamID (3 字节) 表示 stream id 总是 0  |
|            |   | Tag Data   | 不同类型 Tag 的 data 部分结构各不相同, 当 header 的结构是相同的                                   |
|            | Previous Tag size #1 即 Tag #1 的大小 (11 + Datasize)                               |            |  |
|            | Tag #2  |            |  |
|            | Previous Tag size #2  |            |  |
|            | ... ..  |            |  |
|            | Tag #N  |            |  |
|            | Previous Tag size #N  |            |  |

图2.FLV文件结构（详图）

下面详细介绍一下三种Tag的Tag Data部分的结构。

(a)Audio Tag Data结构（音频Tag）

音频Tag开始的第1个字节包含了音频数据的参数信息，从第2个字节开始为音频流数据。结构如图3所示。

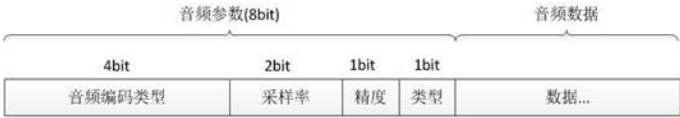


图3. Audio Tag Data结构

第1个字节的前4位的数值表示了音频编码类型。如表1所示。

表1. 音频编码类型

| 值  | 含义                           |
|----|------------------------------|
| 0  | Linear PCM, platform endian  |
| 1  | ADPCM                        |
| 2  | MP3                          |
| 3  | Linear PCM, little endian    |
| 4  | Nellymoser 16-kHz mono       |
| 5  | Nellymoser 8-kHz mono        |
| 6  | Nellymoser                   |
| 7  | G.711 A-law logarithmic PCM  |
| 8  | G.711 mu-law logarithmic PCM |
| 9  | reserved                     |
| 10 | AAC                          |
| 14 | MP3 8-Khz                    |
| 15 | Device-specific sound        |

第1个字节的第5-6位的数值表示音频采样率。如表2所示。

表2.音频采样率

| 值 | 含义     |
|---|--------|
| 0 | 5.5kHz |
| 1 | 11KHz  |
| 2 | 22 kHz |
| 3 | 44 kHz |

PS：从上表可以发现，FLV封装格式并不支持48KHz的采样率。

第1个字节的第7位表示音频采样精度。如表3所示。

表3. 音频采样精度

| 值 | 含义     |
|---|--------|
| 0 | 8bits  |
| 1 | 16bits |

第1个字节的第8位表示音频类型。

表4. 音频类型

| 值 | 含义        |
|---|-----------|
| 0 | sndMono   |
| 1 | sndStereo |

(b)Video Tag Data结构（视频Tag）

视频Tag也用开始的第1个字节包含视频数据的参数信息，从第2个字节为视频流数据。结构如图4所示。



图4.Video Tag Data结构

第1个字节的前4位的数值表示帧类型。如表5所示。

表5. 帧类型

| 值 | 含义   |
|---|--|
| 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） |
| 5 | video info/command frame                     |

第1个字节的后4位的数值表示视频编码类型。如表6所示。

表6.视频编码类型

| 值 | 含义                      |
|---|-------------------------|
| 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                        |

(c)Script Tag Data结构（控制帧）

该类型Tag又通常被称为Metadata Tag,会放一些关于FLV视频和音频的元数据信息如：duration、width、height等。通常该类型Tag会跟在File Header后面作为第一个Tag出现，而且只有一个。结构如图5所示。



图5.Script Tag Data结构

第一个AMF包：

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，标识字符串的长度，一般总是0x000A（“onMetaData”长度）。后面字节为具体的字符串，一般总为“onMetaData”（6F,6E,4D,65,74,61,44,61,74,61）。

第二个AMF包：

第1个字节表示AMF包类型，一般总是0x08，表示数组。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对。常见的数组元素如表7所示。

表7.常见MetaData

| 值               | 含义     |
|-----------------|--------|
| duration        | 时长     |
| width           | 视频宽度   |
| height          | 视频高度   |
| videodatarate   | 视频码率   |
| framerate       | 视频帧率   |
| videocodecid    | 视频编码方式 |
| audiosamplerate | 音频采样率  |
| audiosamplesize | 音频采样精度 |
| stereo          | 是否为立体声 |
| audiocodecid    | 音频编码方式 |
| filesize        | 文件大小   |

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/17934487>

文章标签：[flv](#) [封装格式](#) [解析](#) [开源项目](#)

个人分类：[我的开源项目](#)

此PDF由[spygg](#)生成,请尊重原作者版权!!!  
我的邮箱:liushidc@163.com