

3.1 File Type Box (ftyp)

通常放在MP4文件的开头，告诉解码器基本的解码版本和兼容格式。

- 基本格式如下:



```
aligned(8) class FileTypeBox
extends Box('ftyp') {
    unsigned int(32) major_brand;
    unsigned int(32) minor_version;
    unsigned int(32) compatible_brands[];
}
```

- 字段说明:

字段	长度	说明	默认值
major_brand	4	推荐兼容性的版本	iso6
minor_version	4	最低兼容性的版本	1
compatible_brands	列表值	所有的兼容性的版本	'iso6' 'isom' 'dash'

- Nginx模块实现



```
ngx_int_t
ngx_rtmp_mp4_write_ftyp(ngx_buf_t *b)
{
    u_char *pos;

    pos = ngx_rtmp_mp4_start_box(b, "ftyp");

    /* major brand */
    ngx_rtmp_mp4_box(b, "iso6");

    /* minor version */
    ngx_rtmp_mp4_field_32(b, 1);

    /* compatible brands */
    ngx_rtmp_mp4_box(b, "isom");
    ngx_rtmp_mp4_box(b, "iso6");
    ngx_rtmp_mp4_box(b, "dash");

    ngx_rtmp_mp4_update_box_size(b, pos);

    return NGX_OK;
}
```

3.2 Movie Box (moov)

作为容器盒子，存放相关的trak及meta信息。

- 基本格式如下:



```
aligned(8) class MovieExtendsBox extends Box('mvex'){ }
```

3.2.1 Movie Header Box (mvhd)

mvhd 是 moov 下的第一个 box，用来描述 media 的相关信息:

- 基本格式如下:



```
aligned(8) class MovieHeaderBox extends FullBox('mvhd', version, 0) {
    if (version==1) {
        unsigned int(64) creation_time;
        unsigned int(64) modification_time;
        unsigned int(32) timescale;
        unsigned int(64) duration;
    } else { // version==0
        unsigned int(32) creation_time;
        unsigned int(32) modification_time;
        unsigned int(32) timescale;
        unsigned int(32) duration;
    }

    template int(32) rate = 0x00010000; // typically 1.0
    template int(16) volume = 0x0100; // typically, full volume
    const bit(16) reserved = 0;
    const unsigned int(32)[2] reserved = 0;
    template int(32)[9] matrix =
    { 0x00010000,0,0,0,0x00010000,0,0,0,0x40000000 };
    // Unity matrix
    bit(32)[6] pre_defined = 0;
    unsigned int(32) next_track_ID;
}
```

- 字段说明:

字段	长度	说明	默认值
version	4	版本	0 or 1s
creation_time	4	创建的UTC时间。从1904年开始算起。用秒来表示	无
modification_time	4	最后一次修改时间	无
timescale	4	文件媒体在1秒时间内的刻度值，可以理解为1秒长度的时间单元数	无
duration	4	该track的时间长度，用duration和time scale值可以计算track时长	实际时间为：duration/timescale = xx 秒
rate	4	推荐播放速率	0x00010000
volume	2	音量大小	0x0100 为最大值
reserved	10	保留字段	0
matrixs	4*9	视频变换矩阵	{0x00010000,0,0,0,0x0001s0000,0,0,0,0x40000000}
next_track_ID	4	下一个track使用的id号	

- Nginx模块实现



```
static ngx_int_t
ngx_rtmp_mp4_write_mvhd(ngx_buf_t *b)
{
    u_char *pos;

    pos = ngx_rtmp_mp4_start_box(b, "mvhd");

    /* version */
    ngx_rtmp_mp4_field_32(b, 0);

    /* creation time */
    ngx_rtmp_mp4_field_32(b, 0);

    /* modification time */
    ngx_rtmp_mp4_field_32(b, 0);

    /* timescale */
    ngx_rtmp_mp4_field_32(b, 1000);

    /* duration */
    ngx_rtmp_mp4_field_32(b, 0);

    /* reserved */
}
```

```
ngx_rtmp_mp4_field_32(b, 0x00010000);
ngx_rtmp_mp4_field_16(b, 0x0100);
ngx_rtmp_mp4_field_16(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);

ngx_rtmp_mp4_write_matrix(b, 1, 0, 0, 1, 0, 0);

/* reserved */
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);
ngx_rtmp_mp4_field_32(b, 0);

/* next track id */
ngx_rtmp_mp4_field_32(b, 1);

ngx_rtmp_mp4_update_box_size(b, pos);

return NGX_OK;
}
```

3.2.2 Movie Extends Box (mvex)(fMP4专有)

mvex 是 fMP4 的标准盒子。它的作用是告诉解码器这是一个 fMP4 的文件，具体的 samples 信息内容不再放到 trak 里面，而是在每一个 moof 中。基本格式为：

```
aligned(8) class MovieExtendsHeaderBox extends FullBox('mehd', version, 0) { if (version==1) {
    unsigned int(64)  fragment_duration;
} else { // version==0
    unsigned int(32)  fragment_duration;
}
}
```

3.2.2.1 Track Extends Box (trex)(fMP4专有)

trex 是 mvex 的子一级 box 用来给 fMP4 的 sample 设置默认值。基本内容为

```
aligned(8) class TrackExtendsBox extends FullBox('trex', 0, 0){
    unsigned int(32) track_ID;
    unsigned int(32) default_sample_description_index;
    unsigned int(32) default_sample_duration;
    unsigned int(32) default_sample_size;
    unsigned int(32) default_sample_flags
}
```

3.2.3 Track Box (trak)

trak box 就是主要存放相关 media stream 的内容。

3.2.3.1 Track Header Box (tkhd)

tkhd 是 trak box 的子一级 box 的内容。主要是用来描述该特定 trak 的相关内容信息。其主要内容为：

- 基本格式如下:

```
aligned(8) class TrackHeaderBox
extends FullBox('tkhd', version, flags){
    if (version==1) {
        unsigned int(64)  creation_time;
        unsigned int(64)  modification_time;
        unsigned int(32)  track_ID;
        const unsigned int(32) reserved = 0;
        unsigned int(64)  duration;
    } else { // version==0
        unsigned int(32)  creation_time;
        unsigned int(32)  modification_time;
        unsigned int(32)  track_ID;
        const unsigned int(32) reserved = 0;
        unsigned int(32)  duration;
    }

    const unsigned int(32)[2] reserved = 0;
    template int(16) layer = 0;
    template int(16) alternate_group = 0;
    template int(16) volume = (if track_is_audio 0x0100 else 0); const unsigned int(16) reserved = 0;
    template int(32)[9] matrix=
{ 0x00010000,0,0,0,0,0x00010000,0,0,0,0x40000000 };
    // unity matrix
    unsigned int(32) width;
    unsigned int(32) height;
}
```

- 字段说明:

字段	长度	说明	默认值
version	4	版本	
creation_time	4	创建时间，非必须	0
modification_time	4	修改时间，非必须	0
track_ID	4	指明当前描述的 track ID	1
reserved	4	保留	0
duration	4	当前 track 内容持续的时间。通常结合 timescale 进行相关计算	0
reserved	12	保留字段	0
reserved	2	保留字段	0
alternate_group	2	保留字段	0
volume	2	保留字段	if track_is_audio 0x0100 else 0
reserved	2	保留字段	0
matrix	9 * 4	matrix	b, 1, 0, 0, 1, 0, 0, width, height

3.2.3.2 Media Box (media)

mdia 主要用来包裹相关的 media 信息。

(1) Media Header Box (mdhd)

- 基本格式如下:

```
aligned(8) class MediaHeaderBox extends FullBox('mdhd', version, 0) { if (version==1) {
    unsigned int(64)  creation_time;
    unsigned int(64)  modification_time;
    unsigned int(32)  timescale;
    unsigned int(64)  duration;
} else { // version==0
    unsigned int(32)  creation_time;
    unsigned int(32)  modification_time;
    unsigned int(32)  timescale;
    unsigned int(32)  duration;
}

bit(1) pad = 0;
unsigned int(5)[3] language; // ISO-639-2/T language code unsigned int(16) pre_defined = 0;
}
```

- 字段说明:

字段	长度	说明	默认值
version	4	版本	
creation_time	4	创建时间, 非必须	0
modification_time	4	修改时间, 非必须	0
timescale	4	文件媒体在1秒时间内的刻度值, 可以理解为1秒长度的时间单元数	无
duration	4	当前 track 内容持续的时间。通常结合 timescale 进行相关计算	0
lanuage	4s	表明当前 trak 的语言。因为该字段总长为 15bit, 通常是和 pad 组合成为 2B 的长度。	-

(2) Handler Reference Box(hdlr)

- 基本格式如下:



```
aligned(8) class HandlerBox extends FullBox('hdlr', version = 0, 0) {
    unsigned int(32) pre_defined = 0;
    unsigned int(32) handler_type;
    const unsigned int(32)[3] reserved = 0;
    string name;
}
```

- 字段说明:

字段	长度	说明	默认值
version	4	版本	
pre_defined	4	版本	0
handler_type	4	是代指具体 trak 的处理类型	0
reserved	4 * 3	reserved	0
data	string	reserved	"VideoHandler" or "SoundHandler"

- handler_type 类型如下:



```
vide : Video track
soun : Audio track
hint : Hint track
meta : Timed Metadata track
auxv : Auxilliary Video track
```

3.2.3.3 Media Information Box (minf)

minf 是子属内容中, 重要的容器 box, 用来存放当前 track 的基本描述信息。

(1) Video Media Header Box (vmhd)

- 基本格式如下:



```
aligned(8) class VideoMediaHeaderBox
    extends FullBox('vmhd', version = 0, 1) {
    template unsigned int(16) graphicsmode = 0; // copy, see below
    template unsigned int(16)[3] opcolor = {0, 0, 0};
}
```

(2) Sound Media Header Box (smhd)

- 基本格式如下:



```
aligned(8) class SoundMediaHeaderBox
    extends FullBox('smhd', version = 0, 0) {
    template int(16) balance = 0;
    const unsigned int(16) reserved = 0;
}
```

(3) Data Information Box (dinf)

dinf 是用来说明在 trak 中, media 描述信息的位置。其实本身就是一个容器, 没啥内容:

- 基本格式如下:



```
aligned(8) class SoundMediaHeaderBox
    extends FullBox('smhd', version = 0, 0) {
    template int(16) balance = 0;
    const unsigned int(16) reserved = 0;
}
```

(4) Data Reference Box (dref)

dref 是用来设置当前Box描述信息的 data_entry。

- 基本格式如下:



```
aligned(8) class DataReferenceBox
    extends FullBox('dref', version = 0, 0) {
    unsigned int(32) entry_count;

    for (i=1; i <= entry_count; i++) {
        DataEntryBox(entry_version, entry_flags) data_entry;
    }
}
```

- 字段说明:

字段	长度	说明	默认值
version	4	版本	0
entry_count	4	入口数	1
entry_version	4	入口数	0
entry_flags	3	入口数	0

3.3 Moof Box

3.4 Sidxs Box

作者: lo酷风

链接: <https://www.jianshu.com/p/dee55d54fd54>

来源: 简书

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。

分类: 音视频文件, 音视频, 流媒体



DoubleLi
关注 - 29
粉丝 - 2052

+加关注

« 上一篇: Ffmpeg 接口使用 - 基础和转封装

» 下一篇: Web视频播放一杆到底

posted on 2021-02-11 12:02 DoubleLi 阅读(814) 评论(0) 编辑 收藏 举报

0

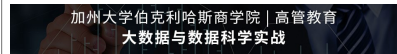
推荐

0

反对

编辑推荐：

- 我给冯圣尔克写了一个720°看鞋展厅
- 带团队后的日常（二）
- 你为什么不想向上汇报？
- 传统.NET 4.x应用容器化体验（4）
- CSS 世界中的方位与顺序



据新闻：

- 10月28日上市！微软又晒《帝国时代4》：玩法、画质大提升
- 小米降噪耳机Pro全球首发骁龙聆听技术有何用？官方科普
- Intel警告重回世界第一！第二天 台积电2nm正式获批
- 《仙剑奇侠传七》上架腾讯WeGame：最高直降30元
- 拿下高通、亚马逊后 Intel CEO放话：还有100多家公司等着我们代工
- » 更多新闻_