

7-8-MP4格式分析

Introduction

Overview

ftyp

moov

mdat

Moov Insider

mvhd

udta

track

tkhd

mdia

mdhd

hdlr

minf

Stbl Insider

stsd

视频的stsd

音频的stsd

stts

stss

stsc

stsz

stco

如何计算sample偏移位置

参考资料

腾讯课堂 零声学院 音视频高级开发 <https://ke.qq.com/course/468797?tuin=137bb271>

Introduction

mp4文件格式又被称为MPEG-4 Part 14，出自MPEG-4标准第14部分。它是一种多媒体格式容器，广泛用于包装视频和音频数据流、海报、字幕和元数据等。（顺便一提，目前流行的视频编码格式AVC/H264定义在MPEG-4 Part 10）。

mp4文件格式基于Apple公司的QuickTime格式，因此，[QuickTime File Format Specification](#)也可以作为我们研究mp4的重要参考。

MP4文件结构的资料 <http://www.52rd.com/Blog/wqyuwss/559/>

mp4box大杀器：<http://download.tsi.telecom-paristech.fr/gpac/mp4box.js/filereader.html>（该链接已经失效）

新地址：<http://114.215.169.66:9001/test/filereader.html>

Overview

mp4文件由box组成，每个box分为Header和Data。其中Header部分包含了box的类型和大小，Data包含了子box或者数据，box可以嵌套子box。

下图是一个典型mp4文件的基本结构：



mp4文件结构

MP4文件的基本组成单元是box，也就是说MP4文件是由各种各样的box组成的，有parent box，还有children box。因此，这些boxes之间存在一定的层次关系，总结如下表所示，表中标记出了各个box必选或可选特性，√代表Box必选。

ftyp						√	file type and compatibility
------	--	--	--	--	--	---	-----------------------------

							文件类型和兼容性
pdin							progressive download information
moov					√		container for all the metadata 所有元数据的容器
	mvhd				√		movie header, overall declarations 电影头，整体声明
	trak				√		container for an individual track or stream 单个轨或流的容器
		tkhd			√		track header, overall information about the track 轨的头部，关于该轨的概括信息，比如视频宽高
		tref					track reference container
		edts					edit list container
			elst				an edit list
		mdia			√		container for the media information in a track 轨媒体信息的容器
			mdhd		√		media header, overall information about the media 媒体头，关于媒体的总体信息
			hdlr		√		handler, declares the media (handler) type 媒体的播放过程信息
			minf		√		media information container 媒体信息容器
				vmhd			video media header, overall information (video track only)
				smhd			sound media header, overall information (sound track only)
				hmhd			hint media header, overall information (hint track only)

				nmhd			Null media header, overall information (some tracks only)
				dinf		√	data information box, container 数据信息box, 容器
					dref	√	data reference box, declares source(s) of media data in track 如何定位媒体信息
				stbl		√	sample table box, container for the time/space map 包含了track中的sample的所有时间和位置信息，以及sample的编解码等信息。利用这个表可以解析sample的时序、类型、大小以及在各自存储容器中的位置。
					stsd	√	sample descriptions (codec types, initialization etc.) 如果是视频，包含：编码类型、宽高、长度等信息； 如果是音频，包含：声道、采样率等信息
					stts	√	(decoding) time-to-sample 描述了sample时序的映射方法，我们可以通过它找到任何时间的sample。
					ctts		(composition) time to sample
					stsc	√	sample-to-chunk, partial data-offset information 用chunk组织sample可以方便优化数据获取，一个chunk包含一个或多个sample。
					stsz		sample sizes (framing) 每个sample的大小。 虽然这里没有打勾，但对于mp4还是非常必要的。
					stz2		compact sample sizes (framing)
					stco	√	chunk offset, partial data-offset information 定义了每个chunk在媒体流中的偏移位置
					co6		64-bit chunk offset

				4		
				stss		sync sample table (random access points) 用于确定media中的关键帧
				stsh		shadow sync sample table
				pad b		sample padding bits
				stdp		sample degradation priority
				sdtb		independent and disposable samples
				sbg p		sample-to-group
				sgp d		sample group description
				subs		sub-sample information
	mvex					movie extends box
		meh d				movie extends header box
		trex			√	track extends defaults
	ipmc					IPMP Control Box
moo f						movie fragment
	mfhd				√	movie fragment header
	traf					track fragment
		tfhd			√	track fragment header
		trun				track fragment run
		sdtb				independent and disposable samples
		sbgp				sample-to-group
		subs				sub-sample information
mfra						movie fragment random access

	tfra					track fragment random access
	mfro				√	movie fragment random access offset
mdat						media data container
free						free space
skip						free space
	udta					user-data
		cpri				copyright etc.
meta						metadata
	hdlr				√	handler, declares the metadata (handler) type
	dinf					data information box, container
		dref				data reference box, declares source(s) of metadata items
	ipmc					IPMP Control Box
	iloc					item location
	ipro					item protection
		sinf				protection scheme information box
			frma			original format box
			imif			IPMP Information box
			sch m			scheme type box
			sch i			scheme information box
	iinf					item information
	xml					XML container
	bxml					binary XML container
	pitm					primary item reference
	fiin					file delivery item information
		paen				partition entry
			fpar			file partition

			fecr				FEC reservoir
		segr					file delivery session group
		gitn					group id to name
		tssel					track selection
mec o							additional metadata container
	mere						metabox relation

本文使用mediainfo和mp4box进行分析

图中看到mp4文件由几个主要组成部分，下面以🔗 [2_audio_track_5s.mp4](#)文件为分析案例：

ftyp

File Type Box，一般在文件的开始位置，描述的文件版本、兼容协议等。

```

1 000000 File Type (32 bytes)
2 000000 Header (8 bytes)
3 000000 Size: 32 (0x00000020)
4 000004 Name: ftyp
5 000008 MajorBrand: isom
6 00000C MajorBrandVersion: 512 (0x00000200)
7 000010 CompatibleBrand: isom
8 000014 CompatibleBrand: iso2
9 000018 CompatibleBrand: avc1
10 00001C CompatibleBrand: mp41

```

ftyp内容

moov

Movie Box，包含本文件中所有媒体数据的宏观描述信息以及每路媒体轨道的具体信息。一般位于放在文件末尾，但如果为了支持http边下载边播放则需要将moov提前。注意，**当改变moov位置时，内部一些值需要重新计算。**

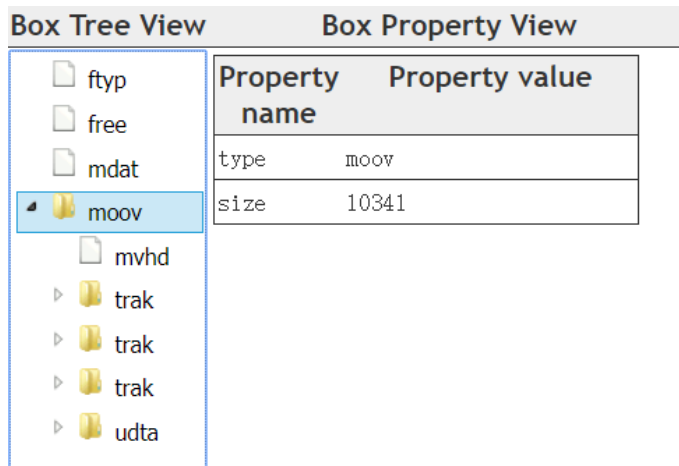
```

1 14B2CE File header (10341 bytes)
2 14B2CE Header (8 bytes)
3 14B2CE Size: 10341 (0x00002865)
4 14B2D2 Name: moov

```

moov内容

moov里面的box才是我们主要分析的box



mdat

Media Data Box，存放具体的媒体数据。

```

1 000028 Data (1356454 bytes)
2 000028 Header (8 bytes)
3 000028 Size: 1356454 (0x0014B2A6)
4 00002C Name: mdat
5 000030 Data: (1356446 bytes)
6 .....数据区域 连续存储.....
..

```

mdat内容

Moov Insider

mp4的媒体数据信息主要存放在Moov Box中，是我们需要分析的重点。moov的主要组成部分如下：

mvhd

Movie Header Box，记录整个媒体文件的描述信息，如创建时间、修改时间、时间度量标尺、可播放时长等。

下图示例中，可以获取文件信息如时长为 Duration: 5016 ms秒。

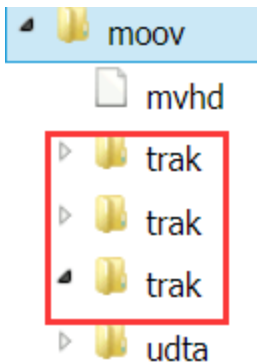
```
1 14B2D6  Movie header (108 bytes)
2 14B2D6  Header (8 bytes)
3 14B2D6  Size: 108 (0x0000006C)
4 14B2DA  Name: mvhd
5 14B2DE  Version: 0 (0x00)
6 14B2DF  Flags: 0 (0x000000)
7 14B2E2  Creation time: 0 (0x00000000) -
8 14B2E6  Modification time: 0 (0x00000000) -
9 14B2EA  Time scale: 1000 (0x000003E8)
   - 1000 Hz
10 14B2EE  Duration: 5016 (0x00001398)
   - 5016 ms
11 14B2F2  Preferred rate: 65536 (0x00010000)
   - 1.000
12 14B2F6  Preferred volume: 256 (0x0100) - 1.0
   00
13 14B2F8  Reserved: (10 bytes)
14 14B302  Matrix structure (36 bytes)
15 14B302  a (width scale): 1.000
16 14B306  b (width rotate): 0.000
17 14B30A  u (width angle): 0.000
18 14B30E  c (height rotate): 0.000
19 14B312  d (height scale): 1.000
20 14B316  v (height angle): 0.000
21 14B31A  x (position left): 0.000
22 14B31E  y (position top): 0.000
23 14B322  w (divider): 1.000
24 14B326  Preview time: 0 (0x00000000)
25 14B32A  Preview duration: 0 (0x00000000)
26 14B32E  Poster time: 0 (0x00000000)
27 14B332  Selection time: 0 (0x00000000)
28 14B336  Selection duration: 0 (0x00000000)
29 14B33A  Current time: 0 (0x00000000)
30 14B33E  Next track ID: 4 (0x00000004)
```

udta

User Data Box, 自定义数据。

track

Track Box, 记录媒体流信息, 文件中可以存在一个或多个track, 它们之间是相互独立的。



, 比如我们提供的测试文件2_audio_track_5s.mp4

每个track包含以下几个组成部分：

tkhd

Track Header Box, 包含关于媒体流的头信息。

下图示例中, 可以看到流信息如视频流宽度720, 长度1280。

```

1 14CEA6  Track Header (92 bytes)
2 14CEA6  Header (8 bytes)
3 14CEA6  Size: 92 (0x0000005C)
4 14CEAA  Name: tkhd
5 14CEAE  Version: 0 (0x00)
6 14CEAF  Flags: 3 (0x000003)
7 14CEB2  Track Enabled: Yes
8 14CEB2  Track in Movie: 2 (0x0000000000000000
002)
9 14CEB2  Track in Preview: 0 (0x0000000000000000
000)
10 14CEB2  Track in Poster: 0 (0x0000000000000000
000)
11 14CEB2  Creation time: 0 (0x00000000) -
12 14CEB6  Modification time: 0 (0x00000000) -
13 14CEBA  Track ID: 3 (0x00000003)

```

14	14CEBE	Reserved:	0 (0x00000000)
15	14CEC2	Duration:	4875 (0x0000130B)
		- 4875 (0x130B) ms	
16	14CEC6	Reserved:	0 (0x00000000)
17	14CECA	Reserved:	0 (0x00000000)
18	14CECE	Layer:	0 (0x0000)
19	14CED0	Alternate group:	2 (0x0002)
20	14CED2	Volume:	0 (0x0000) - 0.000
21	14CED4	Reserved:	0 (0x0000)
22	14CED6	Matrix structure (36 bytes)	
23	14CED6	a (width scale):	1.000
24	14CEDA	b (width rotate):	0.000
25	14CEDE	u (width angle):	0.000
26	14CEE2	c (height rotate):	0.000
27	14CEE6	d (height scale):	1.000
28	14CEEA	v (height angle):	0.000
29	14CEEE	x (position left):	0.000
30	14CEF2	y (position top):	0.000
31	14CEF6	w (divider):	1.000
32	14CEFA	Track width:	1920.000
33	14CEFE	Track height:	800.000

视频tkhd内容

音频的tkhd，则比如duration、volume等。

1	14B34A	Track Header (92 bytes)	
2	14B34A	Header (8 bytes)	
3	14B34A	Size:	92 (0x0000005C)
4	14B34E	Name:	tkhd
5	14B352	Version:	0 (0x00)
6	14B353	Flags:	3 (0x000003)
7	14B356	Track Enabled:	Yes
8	14B356	Track in Movie:	2 (0x0000000000000000002)
9	14B356	Track in Preview:	0 (0x0000000000000000000)
10	14B356	Track in Poster:	0 (0x0000000000000000000)

11	14B356	Creation time:	0 (0x00000000) -
12	14B35A	Modification time:	0 (0x00000000) -
13	14B35E	Track ID:	1 (0x00000001)
14	14B362	Reserved:	0 (0x00000000)
15	14B366	Duration:	5016 (0x00001398)
		- 5016 (0x1398) ms	
16	14B36A	Reserved:	0 (0x00000000)
17	14B36E	Reserved:	0 (0x00000000)
18	14B372	Layer:	0 (0x0000)
19	14B374	Alternate group:	0 (0x0000)
20	14B376	Volume:	256 (0x0100) - 1.0
		00	
21	14B378	Reserved:	0 (0x0000)
22	14B37A	Matrix structure (36 bytes)	
23	14B37A	a (width scale):	1.000
24	14B37E	b (width rotate):	0.000
25	14B382	u (width angle):	0.000
26	14B386	c (height rotate):	0.000
27	14B38A	d (height scale):	1.000
28	14B38E	v (height angle):	0.000
29	14B392	x (position left):	0.000
30	14B396	y (position top):	0.000
31	14B39A	w (divider):	1.000
32	14B39E	Track width:	0.000
33	14B3A2	Track height:	0.000

音频tkhd内容

mdia

Media Box，这是一个包含track媒体数据信息的container box。子box包括：

mdhd：Media Header Box，存放视频流创建时间，长度等信息。

hdlr：Handler Reference Box，媒体的播放过程信息。

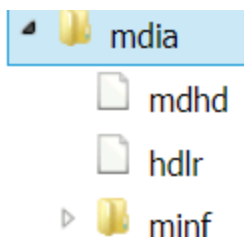
minf：Media Information Box，解释track媒体数据的handler-specific信息。minf同样是个container box，其内部需要关注的内容是stbl，这也是moov中最复杂的部分。stbl包含了媒体流每一个sample在文件中的offset，pts，duration等信息。想要播放一个mp4文件，必须根据stbl正确找到每个sample并送给解码器。

mdia展开如下图所示：

1	14CF32	Media (2975 bytes)
---	--------	--------------------

2	14CF32	Header (8 bytes)	
3	14CF32	Size:	2975 (0x00000B9F)
4	14CF36	Name:	mdia

mdia内容



mdhd

Media Header Box, 存放视频流创建时间, 长度等信息。

视频的mdhd, Time scale, Duration等信息。

1	14CF3A	Media Header (32 bytes)	
2	14CF3A	Header (8 bytes)	
3	14CF3A	Size:	32 (0x00000020)
4	14CF3E	Name:	mdhd
5	14CF42	Version:	0 (0x00)
6	14CF43	Flags:	0 (0x000000)
7	14CF46	Creation time:	0 (0x00000000) -
8	14CF4A	Modification time:	0 (0x00000000) -
9	14CF4E	Time scale:	90000 (0x00015F90)
10	14CF52	Duration:	438750 (0x0006B1DE)
) - 4875 (0x130B) ms	
11	14CF56	Language:	21956 (0x55C4) - u
		nd	
12	14CF58	Quality:	0 (0x0000)

视频mdhd

音频的mdhd, 也类似视频, 但要注意Time scale, 我们在计算时间戳的时候都要使用该Time scale, 对应我们流里面的AVStream->time_base

```

1 14B3D2      Media Header (32 bytes)
2 14B3D2      Header (8 bytes)
3 14B3D2      Size:                                32 (0x00000020)
4 14B3D6      Name:                                mdhd
5 14B3DA      Version:                            0 (0x00)
6 14B3DB      Flags:                              0 (0x000000)
7 14B3DE      Creation time:                      0 (0x00000000) -
8 14B3E2      Modification time:                  0 (0x00000000) -
9 14B3E6      Time scale:                          44100 (0x0000AC44)
10 14B3EA     Duration:                          221184 (0x00036000
    ) - 5015 (0x1397) ms
11 14B3EE     Language:                          21956 (0x55C4) - u
    nd
12 14B3F0     Quality:                            0 (0x0000)

```

音频mdhd

hdlr

Handler Reference Box，媒体的播放过程信息。

视频的hdlr，重点Component subtype: **vide**

```

1 14CF5A      Handler Reference (45 bytes)
2 14CF5A      Header (8 bytes)
3 14CF5A      Size:                                45 (0x0000002D)
4 14CF5E      Name:                                hdlr
5 14CF62      Version:                            0 (0x00)
6 14CF63      Flags:                              0 (0x000000)
7 14CF66      Component type:
8 14CF6A      Component subtype:                  vide
9 14CF6E      Component manufacturer:
10 14CF72     Component flags:                    0 (0x00000000)
11 14CF76     Component flags mask:               0 (0x00000000)
12 14CF7A     Component name:                    VideoHandler

```

音频的hdlr，Component subtype: **soun**，如果我们多个音轨的时候，Component name: 粤语

```

1 14B3F2    Handler Reference (39 bytes)
2 14B3F2    Header (8 bytes)
3 14B3F2    Size:                                39 (0x00000027)
4 14B3F6    Name:                                hdlr
5 14B3FA    Version:                            0 (0x00)
6 14B3FB    Flags:                              0 (0x000000)
7 14B3FE    Component type:
8 14B402    Component subtype:                  soun
9 14B406    Component manufacturer:
10 14B40A    Component flags:                   0 (0x00000000)
11 14B40E    Component flags mask:              0 (0x00000000)
12 14B412    Component name:                    粤语

```

我们分析的文件另一路音轨

```

1 14C0EA    Handler Reference (39 bytes)
2 14C0EA    Header (8 bytes)
3 14C0EA    Size:                                39 (0x00000027)
4 14C0EE    Name:                                hdlr
5 14C0F2    Version:                            0 (0x00)
6 14C0F3    Flags:                              0 (0x000000)
7 14C0F6    Component type:
8 14C0FA    Component subtype:                  soun
9 14C0FE    Component manufacturer:
10 14C102    Component flags:                   0 (0x00000000)
11 14C106    Component flags mask:              0 (0x00000000)
12 14C10A    Component name:                    国语

```

minf

minf: Media Information Box, 解释track媒体数据的handler-specific信息。minf同样是个container box, 其内部需要关注的内容是stbl, 这也是moov中最复杂的部分。stbl包含了媒体流每一个sample在文件中的offset, pts, duration等信息。想要播放一个mp4文件, 必须根据stbl正确找到每个sample并送给解码器。

而且需要注意的是, minf里面的子容器, 音频和视频轨是有区别的, 比如视频轨: **vmhd**, 音频轨则为: **smhd**

vmhd

```
1 14CF8F      Video Media Header (20 bytes)
2 14CF8F      Header (8 bytes)
3 14CF8F      Size:                                20 (0x00000014)
4 14CF93      Name:                                vmhd
5 14CF97      Version:                             0 (0x00)
6 14CF98      Flags:                               1 (0x00000001)
7 14CF9B      Graphic mode:                        0 (0x0000)
8 14CF9D      Graphic mode color R:                 0 (0x0000)
9 14CF9F      Graphic mode color G:                 0 (0x0000)
10 14CFA1     Graphic mode color B:                 0 (0x0000)
```

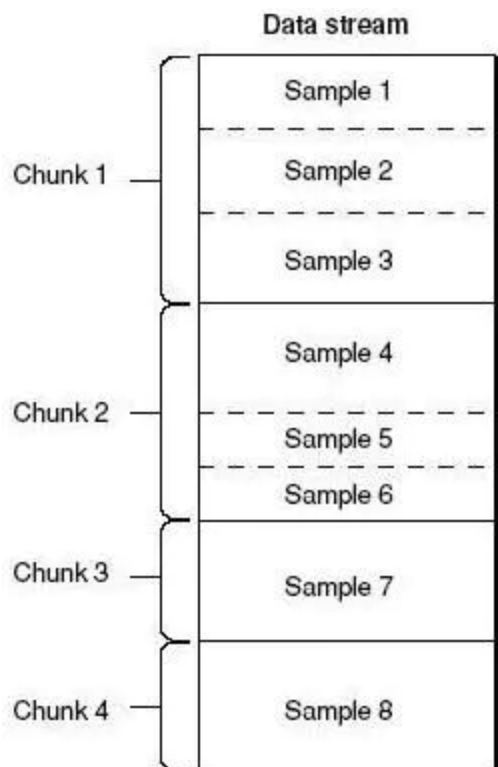
smhd

```
1 14B421      Sound Media Header (16 bytes)
2 14B421      Header (8 bytes)
3 14B421      Size:                                16 (0x00000010)
4 14B425      Name:                                smhd
5 14B429      Version:                             0 (0x00)
6 14B42A      Flags:                               0 (0x00000000)
7 14B42D      Audio balance:                       0 (0x0000)
8 14B42F      Reserved:                           0 (0x0000)
```

Stbl Insider

Sample Table Box, 上文提到mdia中最主要的部分是存放文件中每个sample信息的stbl。在解析stbl前，我们需要区分chunk和sample这两个概念。

在mp4文件中，sample是一个媒体流的基本单元，例如视频流的一个sample代表实际的nal数据。chunk是数据存储的基本单位，它是一系列sample数据的集合，一个chunk中可以包含一个或多的sample。



一个chunk包含一个或多个sample

stbl用来描述每个sample的信息，包含以下几个主要的子box：

stsd

Sample Description Box，存放解码必须的描述信息。

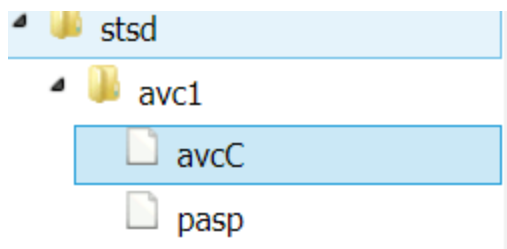
下图示例中，对于h264的视频流，其具体类型为 `avc1`，extensions中其中存放有sps，pps等解码必要信息。

视频的stsd

1	14CFCF	Sample Description (174 bytes)	
2	14CFCF	Header (8 bytes)	
3	14CFCF	Size:	174 (0x000000AE)
4	14CFD3	Name:	stsd
5	14CFD7	Version:	0 (0x00)
6	14CFD8	Flags:	0 (0x000000)
7	14CFDB	Count:	1 (0x00000001)

视频stsd内容

里面包含了avc1，avc1里面又包含了avcC和pasp



avc1: 包含了视频Width、Height

avcC: 包含了视频编码器相关的信息，包括sps、pps等信息

1	14CFDF	Video (158 bytes)	
2	14CFDF	Header (8 bytes)	
3	14CFDF	Size:	158 (0x0000009E)
4	14CFE3	Name:	avc1
5	14CFE7	Reserved:	0 (0x000000000000)
6	14CFED	Data reference index:	1 (0x0001)
7	14CFEF	Version:	0 (0x0000)
8	14CFF1	Revision level:	0 (0x0000)
9	14CFF3	Vendor:	
10	14CFF7	Temporal quality:	0 (0x00000000)
11	14CFFB	Spatial quality:	0 (0x00000000)
12	14CFFF	Width:	1920 (0x0780)
13	14D001	Height:	800 (0x0320)
14	14D003	Horizontal resolution:	4718592 (0x00480000)
15	14D007	Vertical resolution:	4718592 (0x00480000)
16	14D00B	Data size:	0 (0x00000000)
17	14D00F	Frame count:	1 (0x0001)
18	14D011	Compressor name size:	0 (0x00)
19	14D012	Padding:	(31 bytes)
20	14D031	Depth:	24 (0x0018)
21	14D033	Color table ID:	65535 (0xFFFF)
22	14D035	AVC decode (56 bytes)	
23	14D035	Header (8 bytes)	
24	14D035	Size:	56 (0x00000038)
25	14D039	Name:	avcC
26	14D03D	Version:	1 (0x01)
27	14D03E	Specific (47 bytes)	

28	14D03E	Profile:	100 (0x64)
29	14D03F	Compatible profile:	0 (0x00)
30	14D040	Level:	40 (0x28)
31	14D041	Reserved:	63 (0x3F) – (6 bits)
32	14D041	Size of NALU length minus 1:	3 (0x3) – (2 bits)
33	14D042	Reserved:	7 (0x7) – (3 bits)
34	14D042	seq_parameter_set count:	1 (0x01) – (5 bits)
35	14D043	seq_parameter_set (30 bytes)	
36	14D043	Size:	28 (0x001C)
37	14D045	nal_ref_idc:	3 (0x3) – (2 bits)
38	14D045	nal_unit_type:	7 (0x7) – (5 bits)
39	14D046	profile_idc:	100 (0x64)
40	14D047	constraints (1 bytes)	
41	14D047	constraint_set0_flag:	No
42	14D047	constraint_set1_flag:	No
43	14D047	constraint_set2_flag:	No
44	14D047	constraint_set3_flag:	No
45	14D047	constraint_set4_flag:	No
46	14D047	constraint_set5_flag:	No
47	14D047	reserved_zero_2bits:	0 (0x0)
48	14D048	level_idc:	40 (0x28) – (8 bits)
49	14D049	seq_parameter_set_id:	0 (0x0)
50	14D049	high profile specific (1 bytes)	
51	14D049	chroma_format_idc:	1 (0x1) – 4:2:0
52	14D049	bit_depth_luma_minus8:	0 (0x0)
53	14D049	bit_depth_chroma_minus8:	0 (0x0)
54	14D049	qpprime_y_zero_transform_bypass_flag:	No
55	14D049	seq_scaling_matrix_present_flag:	No
56	14D04A	log2_max_frame_num_minus4:	0 (0x0)
57	14D04A	pic_order_cnt_type:	0 (0x0)
58	14D04A	log2_max_pic_order_cnt_lsb_minus4:	2 (0x2)
59	14D04A	max_num_ref_frames:	3 (0x3)
60	14D04B	gaps_in_frame_num_value_allowed_flag:	No

```

61 14D04B      pic_width_in_mbs_minus1:      119 (0x077)
62 14D04D      pic_height_in_map_units_minus1: 49 (0x031)
63 14D04E      frame_mbs_only_flag:          Yes
64 14D04E      direct_8x8_inference_flag:     Yes
65 14D04E      frame_cropping_flag:          No
66 14D04E      vui_parameters_present_flag (17 bytes)
67 14D04E      vui_parameters_present_flag: Yes
68 14D04E      aspect_ratio_info_present_flag (2 bytes)
69 14D04E      aspect_ratio_info_present_flag: Yes
70 14D04F      aspect_ratio_idc:              1 (0x01) - (8 bit
s) - 1.000
71 14D050      overscan_info_present_flag: No
72 14D050      video_signal_type_present_flag (3 bytes)
73 14D050      video_signal_type_present_flag: Yes
74 14D050      video_format:                  5 (0x5) - (3 bits
) -
75 14D050      video_full_range_flag:         0 (0x0) - (1 bits
) - Limited
76 14D050      colour_description_present_flag (3 bytes)
77 14D050      colour_description_present_flag: Yes
78 14D050      colour_primaries:              1 (0x01) - (8 bit
s) - BT.709
79 14D051      transfer_characteristics:      1 (0x01) - (8 bit
s) - BT.709
80 14D052      matrix_coefficients:           1 (0x01) - (8 bit
s) - BT.709
81 14D053      chroma_loc_info_present_flag: No
82 14D054      timing_info_present_flag (8 bytes)
83 14D054      timing_info_present_flag:      Yes
84 14D054      num_units_in_tick:             1 (0x00000001) -
(32 bits)
85 14D058      time_scale:                    48 (0x00000030) -
(32 bits)
86 14D05C      fixed_frame_rate_flag:         Yes
87 14D05C      nal_hrd_parameters_present_flag: No
88 14D05C      vcl_hrd_parameters_present_flag: No
89 14D05C      pic_struct_present_flag:       No
90 14D05C      bitstream_restriction_flag (3 bytes)
91 14D05C      bitstream_restriction_flag: Yes
92 14D05C      motion_vectors_over_pic_boundaries_flag: Yes

```

```

93 14D05D          max_bytes_per_pic_denom:      0 (0x0)
94 14D05D          max_bits_per_mb_denom:        0 (0x0)
95 14D05D          log2_max_mv_length_horizontal: 11 (0x0B)
96 14D05E          log2_max_mv_length_vertical:  11 (0x0B)
97 14D05F          max_num_reorder_frames:       2 (0x2)
98 14D05F          max_dec_frame_buffering:      4 (0x4)
99 14D061          pic_parameter_set count:      1 (0x01)
100 14D062          pic_parameter_set (6 bytes)
101 14D062          Size:                        5 (0x0005)
102 14D064          nal_ref_idc:                 3 (0x3) - (2 bits
)
103 14D064          nal_unit_type:               8 (0x8) - (5 bits
)
104 14D065          pic_parameter_set_id:        0 (0x0)
105 14D065          seq_parameter_set_id:       0 (0x0)
106 14D065          entropy_coding_mode_flag:    Yes
107 14D065          bottom_field_pic_order_in_frame_present_flag: N
0
108 14D065          num_slice_groups_minus1:     0 (0x0)
109 14D065          num_ref_idx_l0_default_active_minus1: 3 (0x3)
110 14D066          num_ref_idx_l1_default_active_minus1: 0 (0x0)
111 14D066          weighted_pred_flag:         No
112 14D066          weighted_bipred_idc:        2 (0x2) - (2 bits
)
113 14D066          pic_init_qp_minus26:        0 (0x0)
114 14D067          pic_init_qs_minus26:        0 (0x0)
115 14D067          chroma_qp_index_offset:      0 (0x0)
116 14D067          deblocking_filter_control_present_flag: Yes
117 14D067          constrained_intra_pred_flag: No
118 14D067          redundant_pic_cnt_present_flag: No
119 14D067          transform_8x8_mode_flag:     Yes
120 14D067          pic_scaling_matrix_present_flag: No
121 14D067          second_chroma_qp_index_offset: 0 (0x0)
122 14D068          -----
123 14D068          ---   AVC, accepted   ---
124 14D068          -----
125 14D069          Padding?:                    (4 bytes)
126 14D06D          Pixel Aspect Ratio (16 bytes)
127 14D06D          Header (8 bytes)
128 14D06D          Size:                      16 (0x00000010)

```

129	14D071	Name:	pas
130	14D075	hSpacing:	1 (0x00000001)
131	14D079	vSpacing:	1 (0x00000001)

音频的stsd

用 **Hexinator** 分析，包含了音频相关的信息，比如采样率，通道数量等。

✓	0x14B45D	0	103	0 stsd	
	0x14B45D	0	4	0 Size	103
	0x14B461	+4	4	1 Type	stsd: 0x73747364
	0x14B465	+8	1	2 Version	0
	0x14B466	+9	3	3 Flags	0
	0x14B469	+12	4	4 Number of entries	1
✓	0x14B46D	+16	87	5 Sample Descriptions	
✓	0x14B46D	0	87	0 Sound Sample Description (V0)	
	0x14B46D	0	4	0 Size	87
	0x14B471	+4	4	1 Type	mp4a: 0x6D703461
	0x14B475	+8	6	2 Reserved	00 00 00 00 00 00
	0x14B47B	+14	2	3 Data reference index	1
	0x14B47D	+16	2	4 Version	Version: 0
	0x14B47F	+18	2	5 Revision level	0
	0x14B481	+20	4	6 Vendor	0x0
	0x14B485	+24	2	7 Number of channels	2
	0x14B487	+26	2	8 Sample size (bits)	16
	0x14B489	+28	2	9 Compression ID	0
	0x14B48B	+30	2	10 Packet size	0
	0x14B48D	+32	2	11 Sample rate M	44100
	0x14B48F	+34	2	12 Sample rate Q	0
	0x14B491	+36	4	13 SizeOfStruct	51
	0x14B495	+40	8	14 <Binary Fill Bytes>	65 73 64 73 00 00 00 00
	0x14B49D	+48	4	15 NumAudioChannels	58753152
	0x14B4A1	+52	4	16 Reserved-1	22 00 01 00
	0x14B4A5	+56	4	17 ConstantBitsPerChannel	75530368
	0x14B4A9	+60	4	18 FormatSpecificFlags	339744000
	0x14B4AD	+64	4	19 ConstantBytesPerAudioPacket	2
	0x14B4B1	+68	4	20 ConstantFramesPerAudioPacket	3515744258
	0x14B4B5	+72	15	21 15 padding bytes	

stts

Time-to-Sample Box，定义每个sample时长。Time-To-Sample的table entry布局如下：

Sample count	Sample duration	Field
4	4	Bytes

stts table entry布局

- sample count：sample个数
- sample duration：sample持续时间

持续时间相同的连续sample可以放到一个entry里达到节省空间的目的。

这里先给出来的是视频的stts，Number of entries，**这个参数需要注意并不是sample的个数，sample的实际数量需要将每个entry的sample count进行累加才是真正的sample个数。**

下图示例中，第1个sample时间为3720，单位用mdhd的time scale进行换算，比如视频的是90000，此时换算成秒为3720/90000 = 0.0413333333333333秒。

```
1 14D07D      Time to Sample (664 bytes)
2 14D07D      Header (8 bytes)
3 14D07D      Size:                               664 (0x00000298)
4 14D081      Name:                               stts
5 14D085      Version:                            0 (0x00)
6 14D086      Flags:                              0 (0x000000)
7 14D089      Number of entries:                  81 (0x00000051)
8 14D08D      Sample Count:                       1 (0x00000001)
9 14D091      Sample Duration:                    3720 (0x00000E88)
10 14D095     Sample Count:                       1 (0x00000001)
11 14D099     Sample Duration:                    3780 (0x00000EC4)
12 14D09D     Sample Count:                       1 (0x00000001)
13 14D0A1     Sample Duration:                    3690 (0x00000E6A)
14 14D0A5     Sample Count:                       2 (0x00000002)
15 14D0A9     Sample Duration:                    3780 (0x00000EC4)
16 14D0AD     Sample Count:                       1 (0x00000001)
17 14D0B1     Sample Duration:                    3690 (0x00000E6A)
18 14D0B5     Sample Count:                       2 (0x00000002)
19 14D0B9     Sample Duration:                    3780 (0x00000EC4)
20 14D0BD     Sample Count:                       1 (0x00000001)
21 14D0C1     Sample Duration:                    3690 (0x00000E6A)
22 14D0C5     Sample Count:                       2 (0x00000002)
```

23	14D0C9	Sample Duration:	3780 (0x00000EC4)
24	14D0CD	Sample Count:	1 (0x00000001)
25	14D0D1	Sample Duration:	3690 (0x00000E6A)
26	14D0D5	Sample Count:	2 (0x00000002)
27	14D0D9	Sample Duration:	3780 (0x00000EC4)
28	14D0DD	Sample Count:	1 (0x00000001)
29	14D0E1	Sample Duration:	3690 (0x00000E6A)
30		
31	14D305	Sample Count:	2 (0x00000002)
32	14D309	Sample Duration:	3780 (0x00000EC4)
33	14D30D	Sample Count:	1 (0x00000001)
34	14D311	Sample Duration:	3750 (0x00000EA6)
35	14D315	结束位置	

视频stts内容

再给出个音频的stts，只是mdhd的时间 scale的差别，之前我们看到音频为44100，则计算第一个sample的时间

$1024/44100=0.0232199546485261$ 秒。

1	14B4C4	Time to Sample (1048 bytes)	
2	14B4C4	Header (8 bytes)	
3	14B4C4	Size:	1048 (0x00000418)
4	14B4C8	Name:	stts
5	14B4CC	Version:	0 (0x00)
6	14B4CD	Flags:	0 (0x000000)
7	14B4D0	Number of entries:	129 (0x00000081)
8	14B4D4	Sample Count:	1 (0x00000001)
9	14B4D8	Sample Duration:	1024 (0x00000400)
10	14B4DC	Sample Count:	1 (0x00000001)
11	14B4E0	Sample Duration:	1025 (0x00000401)
12	14B4E4	Sample Count:	2 (0x00000002)
13	14B4E8	Sample Duration:	1024 (0x00000400)
14	14B4EC	Sample Count:	1 (0x00000001)
15	14B4F0	Sample Duration:	1023 (0x000003FF)

音频stts内容节选

stss

Sync Sample Box，同步sample表，存放关键帧列表，关键帧是为了支持随机访问。
stss的table entry布局如下：

Number	Sample 1
Number	Sample 2
Number	Sample 3
Number	Sample 4
Number	Sample 5

stss table entry布局

下图示例中，该视频track有3个关键帧：

```
1 14D315      Sync Sample (28 bytes)
2 14D315      Header (8 bytes)
3 14D315      Size:                               28 (0x0000001C)
4 14D319      Name:                               stss
5 14D31D      Version:                             0 (0x00)
6 14D31E      Flags:                               0 (0x000000)
7 14D321      entry-count:                         3 (0x00000003)
8 14D325      number:
  1 darren补充 (mediainfo没有解析出来)
9 14D329      number:
  54 darren补充
10 14D32D     number:
  103 darren补充
```

stss内容

Box Tree View

- dref
 - url
- stbl
- stsd
 - avc1
 - avcC
 - pasP
 - stts
 - stss**

Box Property

Property name	Property value
type	stss
size	28
flags	0
version	0
sample_numbers	1, 54, 103

stsc

Sample-To-Chunk Box，sample-chunk映射表。上文提到mp4通常把sample封装到chunk中，一个chunk可能会包含一个或者几个sample。Sample-To-Chunk Atom的table entry布局如下图所示：

First chunk	Samples per chunk	Sample description ID	Fields
4	4	4	Bytes

stsc table entry布局

- First chunk：使用该表项的第一个chunk序号
- Samples per chunk：使用该表项的chunk中包含有几个sample
- Sample description ID：使用该表项的chunk参考的stsd表项序号

下图示例中，可以看到该视频track一共有1个stsc表项，chunk序列1-x，每个chunk包含一个sample。这里则说明每个chunk里面只有一个sample（一个chunk是可以有多个sample）。

Box Tree View	Box Property Vie																
<ul style="list-style-type: none"> dref <ul style="list-style-type: none"> url stbl <ul style="list-style-type: none"> stsd <ul style="list-style-type: none"> avc1 <ul style="list-style-type: none"> avcC pasP stts stss ctts stsc 	<table> <thead> <tr> <th>Property name</th><th>Property value</th></tr> </thead> <tbody> <tr> <td>type</td><td>stsc</td></tr> <tr> <td>size</td><td>28</td></tr> <tr> <td>flags</td><td>0</td></tr> <tr> <td>version</td><td>0</td></tr> <tr> <td>first_chunk</td><td>1</td></tr> <tr> <td>samples_per_chunk</td><td>1</td></tr> <tr> <td>sample_description_index</td><td>1</td></tr> </tbody> </table>	Property name	Property value	type	stsc	size	28	flags	0	version	0	first_chunk	1	samples_per_chunk	1	sample_description_index	1
Property name	Property value																
type	stsc																
size	28																
flags	0																
version	0																
first_chunk	1																
samples_per_chunk	1																
sample_description_index	1																

stsc内容

stsz

Sample Size Box，指定了每个sample的size。Sample Size Atom包含两sample总数和一张包含了每个sample size的表。

sample size 表的entry布局如下图：

Size	Sample 1
Size	Sample 2
Size	Sample 3
Size	Sample 4
Size	Sample 5

stsz table entry布局

下图示例中，该视频流一共有110个sample，第1个sample大小为42072字节，第2个sample大小为7354个字节。

```

1 14D705      Sample Size (488 bytes)
2 14D705      Header (8 bytes)

```

3	14D705	Size:	488 (0x000001E8)
4	14D709	Name:	stsz
5	14D70D	Version:	0 (0x00)
6	14D70E	Flags:	0 (0x000000)
7	14D711	Sample Size:	0 (0x00000000)
8	14D715	Number of entries:	117 (0x00000075)

Box Tree View

- dref
 - url
- stbl
 - stsd
 - avc1
 - avcC
 - pasp
 - stts
 - stss
 - ctts
 - stsc
 - stsz

Property name

type	stsz
size	488
flags	0
version	0
sample_sizes	172818, 20829, 722, 567, 25207, 1946, 822, 674, 23828, 2141, 824, 974, 22426, 2794, 899, 936, 24037, 1729,
sample_size	0
sample_count	117

这里只是节选了部分

stsz内容

stco

Chunk Offset Box，指定了每个chunk在文件中的位置，这个表是确定每个sample在文件中位置的关键。该表包含了chunk个数和一个包含每个chunk在文件中偏移位置的表。每个表项的内存布局如下：

Offset	Chunk 1
Offset	Chunk 2
Offset	Chunk 3
Offset	Chunk 4
Offset	Chunk 5

stco table entry布局

需要注意，这里stco只是指定的每个chunk在文件中的偏移位置，并没有给出每个sample在文件中的偏移。想要获得每个sample的偏移位置，需要结合 Sample Size box(stsz)和Sample-To-Chunk(stsc) 计算后取得。

下图示例中，该视频流第1个chunk在文件中的偏移为4750，而这里是每个chunk只有一个sample，此时第一个sample的起始位置就为4750->0x1D78，数据大小则参照stsz，第一个sample size为172818。

Box Tree View

- dref
 - url
- stbl
 - std
 - avc1
 - avcC
 - pasp
 - stts
 - stss
 - ctts
 - stsc
 - stsz
 - stco

Property name	
type	stco
size	484
flags	0
version	0
chunk_offsets	7544, 182562, 204381, 206907, 209520, 236820, 240924, 242781, 245574, ...

stco内容

比如偏移位置，7544->0x1D78

	00	01	02	03	04	05	06	07	
0x001D78	00	00	00	04	68	E9	2B	CB	. . . h . . .
0x001D80	00	00	00	02	09	F0	00	00
0x001D88	00	1B	67	64	00	28	AC	D9	. . g d . (. .
0x001D90	00	78	06	5A	6A	02	02	02	. x . Z j . . .
0x001D98	80	00	00	03	00	80	00	00
0x001DA0	18	47	8C	18	CB	00	00	00	. G
0x001DA8	04	68	E9	2B	CB	00	02	A2	. h
0x001DB0	D9	65	88	82	00	17	FF	DA	. e
0x001DB8	B8	E4	29	17	5A	79	53	92	. .) . Z y S .
0x001DC0	76	2A	F4	FA	EF	49	B2	C8	v * . . . I . .
0x001DC8	B3	1C	32	74	37	B4	9A	3C	. . 2 t 7 . . <
0x001DD0	3F	D2	91	AA	3A	2E	5C	CC	? . . . : . . .
0x001DD8	8F	EF	30	E5	84	8D	38	A2	. . 0 . . . 8 .
0x001DE0	C0	4C	08	97	14	A1	AD	0D	. L
0x001DE8	3C	7C	DB	F5	79	D6	23	C5	< . . y . # .
0x001DF0	80	83	95	76	4B	56	3A	69	. . . v K V : i
0x001DF8	E7	67	AD	35	83	7E	FD	91	. g . 5
0x001E00	14	6D	B6	95	6B	10	67	69	. m . . k . g i
0x001E08	87	AB	92	E5	1E	A5	67	FB g .
0x001E10	BB	7C	B8	81	3C	12	AF	1D	. . . < . . .
0x001E18	C5	55	C2	9B	D3	E9	EB	35	. U 5
0x001E20	6F	88	F7	57	BA	7F	41	22	o . . W . . A "
0x001E28	76	A5	6E	B0	CB	2E	3C	11	v . n . . . < .
0x001E30	7F	3B	51	70	0B	41	56	0F	. ; Q p . A V .
0x001E38	7D	62	61	D3	FC	48	18	D8	} b a . . H . .
0x001E40	0C	D9	9B	F9	8D	E5	4D	54 M T
0x001E48	31	CB	1C	90	15	3E	C4	7F	1 > . .
0x001E50	63	FB	59	80	A3	C3	0B	8F	c . Y
0x001E58	EC	2A	91	08	0E	D2	F7	61	. * a
0x001E60	ED	CD	D6	5A	8F	77	EA	B6	. . . Z . w . .
0x001E68	F3	9C	BC	3F	5B	5F	BD	1B	. . . ? [. .

如何计算sample偏移位置

上文提到通过stco并不能直接获取某个sample的偏移位置，下面举例说明如何获取某一个pts对应的sample在文件中的位置。

通

大体需要以下步骤：

- 1.将pts转换到媒体对应的时间坐标系
- 2.根据stts(**decoding time-to-sample**)计算某个pts对应的sample序号
- 3.根据stsc(**sample-to-chunk**)计算sample序号存放在哪个chunk中
- 4.根据stco(**chunk offset**)获取对应chunk在文件中的偏移位置
- 5.根据stsz获取sample在chunk内的偏移位置并加上第4步获取的偏移，计算出sample在文件中的偏移

例如，想要获取3.64秒视频sample数据在文件中的位置（使用我们上课用的2_audio_track_5s.mp4）：

- 1.根据time scale参数，将3.64秒转换为视频时间轴对应的3640000（假如时间刻度不为毫秒）
 - 视频轨：time scale为90000，转成对应的时间戳为3.64秒*90000
- 2.遍历累加下表所示stts所有项目，计算得到3640000位于第110个sample = 327600
- 3.计算出多个sample_deltas叠加才到了327600，我们这里姑且按3780作为平均值计算，**实际是**
3720*1+3780*1+3690*1+3780*2 这样一直叠加进行。327600/3780 =
86.66666666666667，取整为86

```
1 type      stts
2 size      664
3 flags     0
4 version   0
5 sample_counts 1,1,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,
  2,1,2,1,2,1,2,1,2,1,1,1,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,
  2,1,2,1,2,1,2,1,2,1,2,1,1,1,1,2,1,2,1,2,1,2,1,2,1
6 sample_deltas 3720,3780,3690,3780,3690,3780,3690,3780,3690,3780,
  3690,3780,3690,3780,3690,3780,3690,3780,3690,3780,3690,3
  780,3690,3780,3690,3780,3690,3780,3690,3780,3690,3780,37
  50,3720,3780,3690,3780,3690,3780,3690,3780,3690,3780,369
  0,3780,3690,3780,3690,3780,3690,3780,3690,3780,3690,3780
  ,3690,3780,3690,3780,3690,3780,3750,3720,3780,3690,3780,3690,3780,
  3690,3780,3690,3780,3750
```

- 4.查询下表所示stsc所有项目，计算得到第86个sample位于第86个chunk，并且在该chunk中位于第1个sample（因为我们的码流是每个chunk对应了一个sample）

```
1 Property name  Property value
2 type          stsc
3 size          28
4 flags         0
5 version       0
```

```

6 first_chunk 1
7 samples_per_chunk 1
8 sample_description_index 1

```

- 5.查询下表所示stco所有项目，得到第86个chunk在文件中偏移位置为1004678。使用hexinator

```

1 Property name    Property value
2 type            stco
3 size            484
4 flags           0
5 version         0
6 chunk_offsets    7544,182562,204381,206907,209520,236820,240924,242
7                  781,.....省略

```

- 6.查询下表所示stsz所有项目，得到第86个sample的size为20934。计算得到3.64秒视频sample数据在文件中
offset: 1004678+0 = 1004678
size: **20934**

```

1 Property name    Property value
2 type            stsz
3 size            488
4 flags           0
5 version         0
6 sample_sizes     172818,20829,722,567,25207,1946,822,674,23828,2141
7                  ,824,974,22426,2794..省略
7 sample_count     117

```

- 验证：用编辑器打开mp4文件，定位到文件偏移1004678位置，。

09分隔符，这里占用了6个字节， 再看真正的数据区域，前4字节也为 NALU的长度0x000051bc=20924

总共占用的字节计算 4+2+4+20924 = **20934**

	00	01	02	03	04	05	06	07	
0x0F5480	00	0E	8E	17	6A	70	00	00	.
0x0F5488	00	02	09	F0	00	00	51	BC	.
0x0F5490	41	9A	23	56	A8	95	CB	FF	A
0x0F5498	AD	60	DE	93	C7	C9	C7	FA	.
0x0F54A0	F9	0D	33	C5	59	A0	CB	47	.
0x0F54A8	CC	A0	37	5E	60	06	88	D8	.
0x0F54B0	17	5A	91	6D	EC	E2	8E	45	.
0x0F54B8	6A	4C	A2	24	52	9E	49	6E	j
0x0F54C0	89	B7	CE	9C	F6	F7	E7	47	.

参考资料

扩展：《整理mp4协议重点，将协议读薄》<https://www.cnblogs.com/shakin/p/8543719.html>

<https://www.cnblogs.com/ranson7zop/p/7889272.html>

[在线mp4解析工具](#)

[QuickTime File Format Specification](#)

一个chunk含有多个sample的情景参考下面链接进行分析：

mp4文件格式重点解析 <https://www.jianshu.com/p/44c9567d8fcb>