

Overview and Introduction

Core Concepts

MP4 文件格式中，所有的内容存在一个称为 **movie** 的容器中。一个 **movie** 可以由多个 **tracks** 组成。每个 **track** 就是一个随时间变化的媒体序列，例如，视频帧序列。**track** 里的每个时间单位是一个 **sample**，它可以是一帧视频，或者音频。**sample** 按照时间顺序排列。注意，一帧音频可以分解成多个音频 **sample**，所以音频一般用 **sample** 作为单位，而不用帧。MP4 文件格式的定义里面，用 **sample** 这个单词表示一个时间帧或者数据单元。每个 **track** 会有一个或者多个 **sample descriptions**。**track** 里面的每个 **sample** 通过引用关联到一个 **sample description**。这个 **sample descriptions** 定义了怎样解码这个 **sample**，例如使用的压缩算法。

与其他的多媒体文件格式不同的是，MP4 文件格式经常使用几个不同的概念，理解其不同是理解这个文件格式的关键。

这个文件的物理格式没有限定媒体本身的格式。例如，许多文件格式将媒体数据分成帧，头部或者其他数据紧紧跟随每一帧视频，！！！TODO（例如 MPEG2）。而 MP4 文件格式不是如此。文件的物理格式和媒体数据的排列都不受媒体的时间顺序的限制。视频帧不需要在文件按时间顺序排列。这就意味着如果文件中真的存在这样的一些帧，那么就有一些文件结构来描述媒体的排列和对应的时间信息。

MP4 文件中所有的数据都封装在一些 **box** 中（以前叫 **atom**）。所有的 metadata(媒体描述元数据)，包括定义媒体的排列和时间信息的数据都包含在这样的一些结构 **box** 中。MP4 文件格式定义了这些 **box** 的格式。Metadata 对媒体数据（例如，视频帧）引用说明。媒体数据可以包含在同一个的一个或多个 **box** 里，也可以在其他文件中，metadata 允许使用 URLs 来引用其他的文件，而媒体数据在这些引用文件中的排列关系全部在第一个主文件中的 metadata 描述。其他的文件不一定是 MP4 文件格式，例如，可能就没有一个 **box**。

有很多种类的 **track**，其中有三个最重要，**video track** 包含了视频 **sample**；**audio track** 包含了 **audio sample**；**hint track** 稍有不同，它描述了一个流媒体服务器如何把文件中的媒体数据组成符合流媒体协议的数据包。如果文件只是本地播放，可以忽略 **hint track**，他们只与流媒体有关系。

Physical structure of the media

Box 定义了如何在 **sample table** 中找到媒体数据的排列。这包括 **data reference**(数据引用), the **sample size table**, the **sample to chunk table**, and the **chunk offset table**. 这些表就可以找到 **track** 中每个 **sample** 在文件中的位置和大小。

data reference 允许在第二个媒体文件中找到媒体的位置。这样，一部电影就可以由一个媒体数据库中的多个不同文件组成，而且不用把它们全部拷贝到另一个新文件中。例如，对视频编辑就很有帮助。

为了节约空间，这些表都很紧凑。另外，interleave 不是 sample by sample，而是把单个 track 的几个 samples 组合到一起，然后另外几个 sample 又进行新的组合，等等。一个 track 的连续几个 sample 组成的单元就被称为 **chunk**。每个 chunk 在文件中有一个偏移量，这个偏移量是从文件开头算起的，在这个 chunk 内，sample 是连续存储的。

这样，如果一个 chunk 包含两个 sample，第二个 sample 的位置就是 chunk 的偏移量加上第一个 sample 的大小。chunk offset table 说明了每个 chunk 的偏移量，sample to chunk table 说明了 sample 序号和 chunk 序号的映射关系。

注意 chunk 之间可能会有死区，没有任何媒体数据引用到这部分区域，但是 chunk 内部不会有这样的死区。这样，如果在节目编辑的时候，不需要一些媒体数据，就可以简单的留在那里，而不用引用，这样就不用删除它们了。类似的，如果媒体存放在第二个文件中，但是格式不同于 MP4 文件格式，这个陌生文件的头部或者其他文件格式都可以简单忽略掉。

Temporal structure of the media

文件中的时间可以理解为一些结构。电影以及每个 track 都有一个 **timescale**。它定义了一个时间轴来说明每秒钟有多少个 ticks。合理的选择这个数目，就可以实现准确的计时。一般来说，对于 audio track，就是 audio 的 sampling rate。对于 video track，情况稍微复杂，需要合理选择。例如，如果一个 media TimeScale 是 30000，media sample durations 是 1001，就准确的定义了 NTSC video 的时间格式（虽然不准确，但一般就是 29.97），and provide 19.9 hours of time in 32 bits.

Track 的时间结构受一个 **edit list** 影响，有两个用途：全部电影中的一个 track 的一部分时间片段变化（有可能是重用）；空白时间的插入，也就是空的 edits。特别注意的是如果一个 track 不是从节目开头部分开始，edit list 的第一个 edit 就一定是空的 edit。

每个 track 的全部 duration 定义在文件头部，这就是对 track 的总结，每个 sample 有一个规定的 **duration**。一个 sample 的准确描述时间，也就是他的时间戳(time-stamp)就是以前的 sample 的 duration 之和。

Interleave

文件的时间和物理结构可以是对齐的，这表明媒体数据在容器中的物理顺序就是时间顺序。另外如果多个 track 的媒体数据包含在同一个文件中，这个媒体数据可以是 interleaved。一般来说，为了方便读取一个 track 的媒体数据，同时保证每个表紧凑，以一个合适的时间间隔（例如 1 秒）做一次 interleave，而不是 sample by sample。这样就可以减少 chunk 的数据，减小 chunk offset table 的大小。

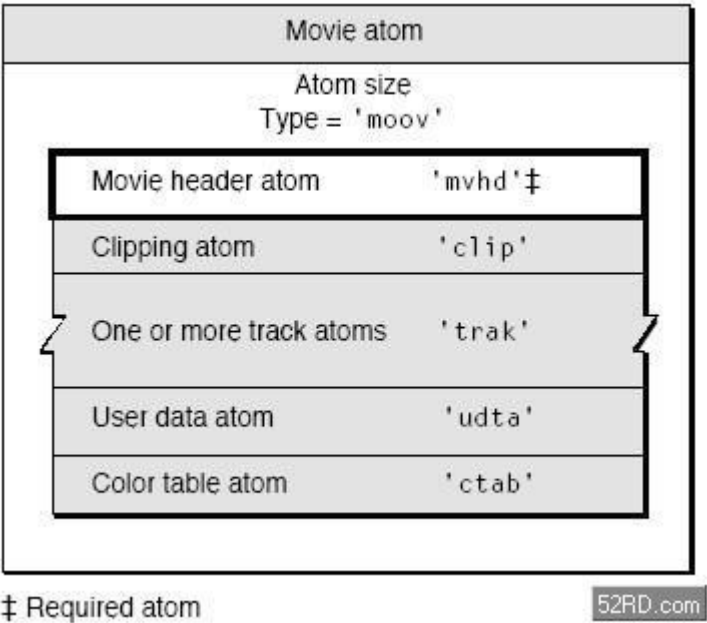
mp4 文件例子

Darwin Streaming Server 里面有一些很好的工具，可以帮助分析 mp4 文件格式。

但是如果可以自己逐字节的 parse 文件，可以更好的了解 mp4 文件格式。这里我就逐字节的分析文件结构。文件例子是 DSS 里面包含的 sample_100kbit.mp4

Movie Atom - MOOV

Movie atom 定义了一部电影的数据信息。它的类型是'moov'，是一个容器 atom，至少必须包含三种 atom 中的一种—movie header atom('mvhd'), compressed movie atom('cmov')和 reference movie atom ('rmra')。没有压缩的 movie header atom 必须至少包含 movie header atom 和 reference movie atom 中的一种。也可以包含其他的 atom，例如一个 clipping atom ('clip')，一个或几个 track atoms ('trak')，一个 color table atom ('ctab')，和一个 user data atom ('udta')。其中 movie header atom 定义了整部电影的 time scale, duration 信息以及 display characteristics。track atom 定义了电影中一个 track 的信息。Track 就是电影中可以独立操作的媒体单位，例如一个声道就是一个 track。Compressed movie atoms 和 reference movie atoms 不太使用，不在本文讨论范围内。本文主要讨论 uncompressed movie atoms。



字段	长度(字节)	描述
尺寸	4	这个 movie header atom 的字节数
类型	4	moov

以下是实际的 sample_100kbit.mp4 的部分字节，可以看到结果是

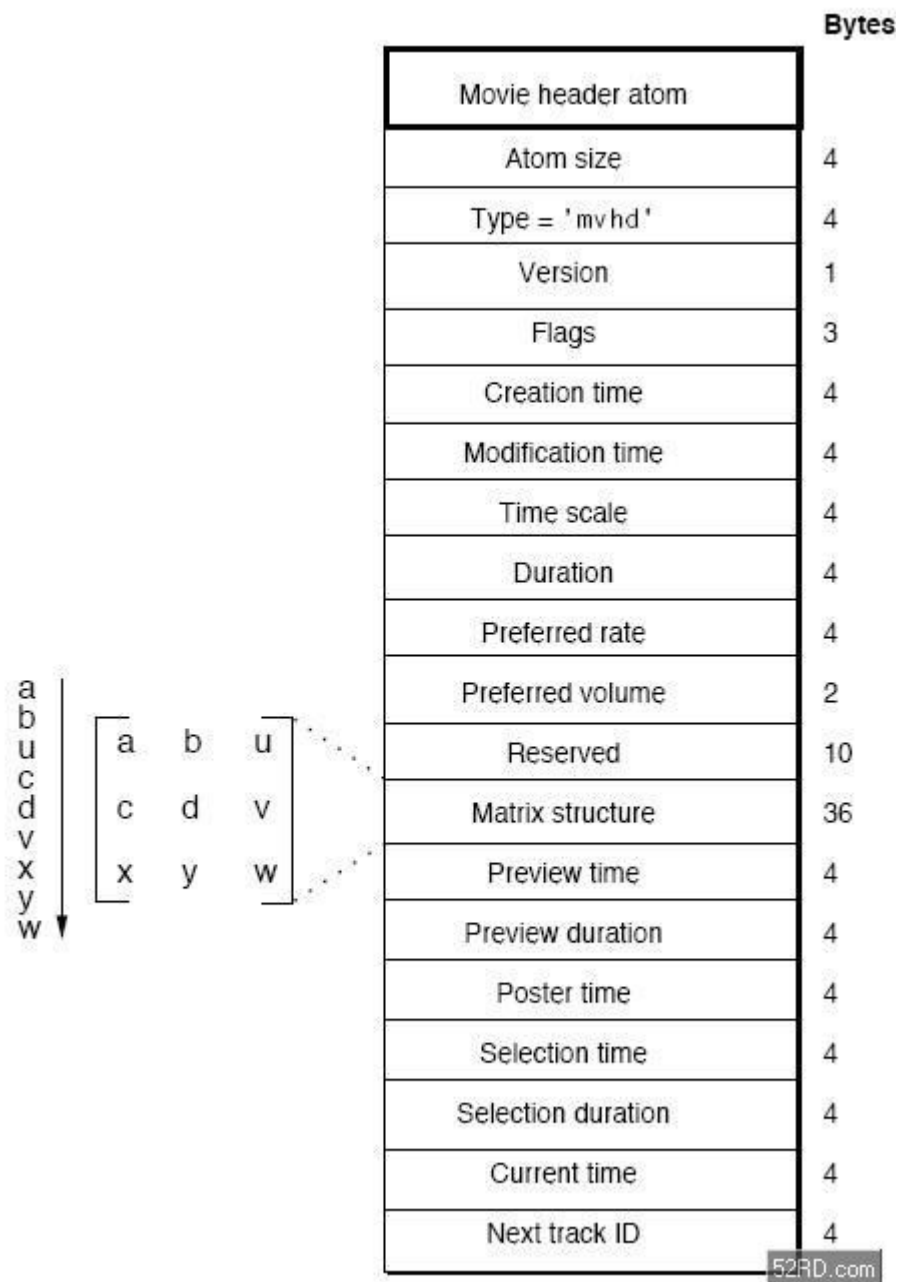
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	D 1 2 3 4 5 6 7 8 9 a b c d e f
00000	00	00	00	18	66	74	79	70	6d	70	34	32	00	00	00	01	...f t y p p i 2...
00010	5a	70	34	32	6a	70	34	31	00	00	2f	2e	6d	6e	6e	70	...mp41...7,moov
00020	00	00	00	4c	6a	76	60	64	00	00	00	00	c1	b8	4a	63	...l mvhd, ... j c
00030	c1	b8	4a	63	00	00	02	50	00	00	a9	10	00	01	00	00	j c...X, ...
00040	01	00	00	00	00	00	00	00	00	00	00	00	00	01	00	00
00050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00
00060	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	008...
00070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00080	00	00	00	00	00	00	00	00	00	00	00	03	00	10	5c	\
00090	74	72	61	4b	00	00	00	5c	74	6b	48	64	00	00	01		trak... \ tkhd...
000a0	c1	b8	4a	53	a1	b8	4a	63	00	00	00	01	00	00	00		j s j o,
000b0	00	00	a4	10	00	00	00	00	00	00	00	00	00	00	00	00

52RD.com

主要包含四个子 atom，movie header atom(mvhd), 一个 audio track atom(trak), 一个 video track atom(trak)。

Movie Header Atoms - MVHD

Movie header atom 定义了整个 movie 的特性，例如 time scale 和 duration，它的 atom 类型是'mvhd'。



字段	长度(字节)	描述
尺寸	4	这个 movie header atom 的字节数
类型	4	Mvhd
版本	1	这个 movie header atom 的版本
标志	3	扩展的 movie header 标志，这里为 0
生成时间	4	Movie atom 的起始时间。基准时间是 1904-1-1 0:00 AM
修订时间	4	Movie atom 的修订时间。基准时间是 1904-1-1 0:00 AM
Time scale	4	A time value that indicates the time scale for this movie—that is, the number of time units that pass per second in its time coordinate system. A time coordinate system that measures time in sixtieths of a second, for example, has a time scale of 60.

Duration	4	A time value that indicates the duration of the movie in time scale units. Note that this property is derived from the movie's tracks. The value of this field corresponds to the duration of the longest track in the movie.
播放速度	4	播放此 movie 的速度。1.0 为正常播放速度
播放音量	2	播放此 movie 的音量。1.0 为最大音量
保留	10	这里为 0
矩阵结构	36	该矩阵定义了此 movie 中两个坐标空间的映射关系
预览时间	4	开始预览此 movie 的时间
预览 duration	4	以 movie 的 time scale 为单位，预览的 duration
Poster time	4	The time value of the time of the movie poster.
Selection time	4	The time value for the start time of the current selection.
Selection duration	4	The duration of the current selection in movie time scale units.
当前时间	4	当前时间
下一个 track ID	4	下一个待添加 track 的 ID 值。0 不是一个有效的 ID 值。

00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 0123456789abcdef	
00000 00 00 00 18 66 74 79 70 6d 70 34 32 00 00 00 01ftypmp42....
00010 6d 70 34 32 6d 70 34 31 00 00 2f 2c 6d 6f 6f 76	mp42mp41.../moov
00020 00 00 00 6c 6d 76 68 64 00 00 00 00 c1 b8 4a 63	...lmvhd....Jc
00030 c1 b8 4a 63 00 00 02 58 00 00 a4 10 00 01 00 00	Jc...X.....
00040 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00050 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00060 00 00 00 00 00 00 00 00 00 00 00 00 40 00 000....
00070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00080 00 00 00 00 00 00 00 00 00 00 00 03 00 00 10 5c[...]
00090 74 72 61 6b 00 00 00 5c 74 6b 68 64 00 00 00 01	trak...tkhd....
000a0 c1 b8 4a 63 c1 b8 4a 63 00 00 00 01 00 00 00 00	Jc Jc.....
000b0 00 00 a4 10 00 00 00 00 00 00 00 00 00 00 00
000c0 01 00 00 00 00 01 00 00 00 00 00 00 00 00 00
000d0 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00

52RD.com

Track Atoms - TRAK

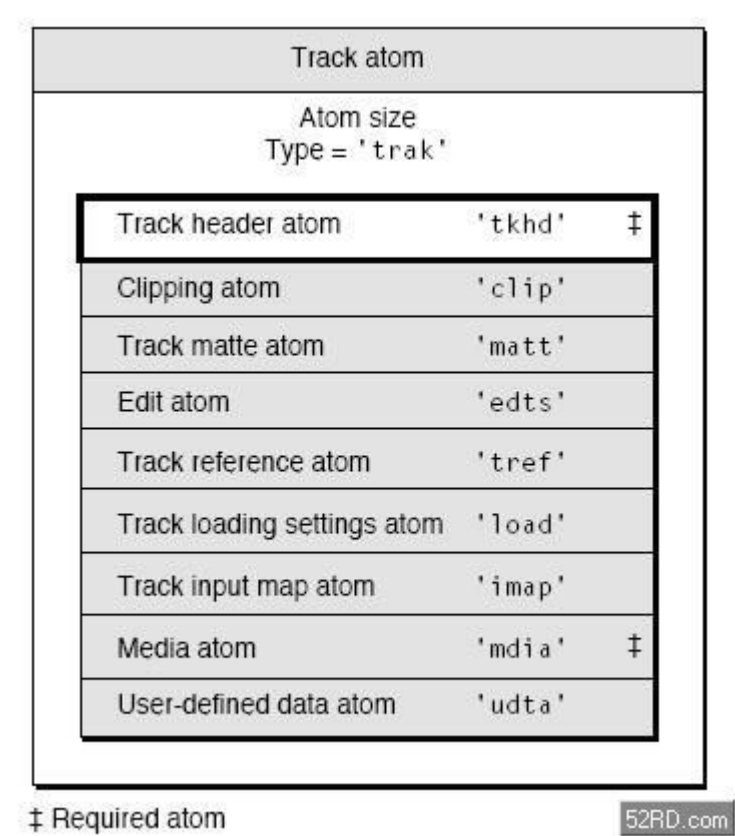
一个 Track atom 定义了 movie 中的一个 track。一部 movie 可以包含一个或多个 tracks，它们之间相互独立，各自有各自的时间和空间信息。每个 track atom 都有与之关联的 media atom。

Track 主要用于以下目的：

- 包含媒体数据引用和描述(media tracks)
- 包含 modifier tracks (tweens 等)
- 对于流媒体协议的打包信息(hint tracks)。Hint tracks 可以引用或者复制对应的媒体 sample data。

Hint tracks 和 modifier tracks 必须保证完整性，同时和至少一个 media track 一起存在。换句话说，即使 hint tracks 复制了对应的媒体 sample data，media tracks 也不能从一部 hinted movie 中删除。Track atoms 的 atom 类型是'trak'. Track atom 要求必须有一个 track header atom ('tkhd') 和一个 media atom ('mdia')。其他的 track clipping atom ('clip'), track matte atom ('matt'), edit atom ('edts'), track reference atom ('tref'), track load settings atom ('load'), a track input map atom ('imap')以及 user data atom ('udta')都是可选的。

Track atoms 是一个容器 atom，本身没有特别的字段，需要子 atom 来进一步说明有效的内容。



字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	Edts

Audio track 的值

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
000000	00	00	00	18	66	74	79	7d	6d	7d	34	32	00	00	00	01	...ftypmp42...
000100	6d	70	34	32	6d	70	34	31	00	00	22	2c	6d	62	62	76	mp42mp41...moov
000200	00	00	00	6c	6d	76	68	64	00	00	00	00	e1	b8	4a	63	...lmvhd....Jc
000300	e1	b8	4a	63	00	00	03	58	00	00	a4	10	00	01	00	00	Jc...%.....
000400	01	00	00	00	00	00	00	00	00	00	00	00	00	01	00	00
000500	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00
000600	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	000...
000700	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000800	00	00	00	00	00	00	00	00	00	00	00	03	00	00	10	5c\...
000900	74	72	61	6b	00	00	00	5c	74	6b	68	64	00	00	00	01	trak...\tkhd....
000a00	e1	b8	4a	53	e1	b8	4a	63	00	00	00	01	00	00	00	00	JS Jc.....
000b00	00	00	a4	10	00	00	00	00	00	00	00	00	00	00	00	00
000c00	01	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
000d00	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00

52RD.com

Video track 的值

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
010d00	00	0c	bc	f0	00	0c	bf	83	00	0c	d1	ac	00	0c	d4	23
010e00	00	0c	df	45	00	0c	e1	f6	00	00	1e	5c	74	72	61	6b	.. E.. ...\trak
010f00	00	00	00	5c	74	6b	68	64	00	00	00	01	e1	b8	4a	54	...\tkhd....JT
011000	e1	b8	4a	63	00	00	00	02	00	00	00	00	00	00	a4	10	Jc.....
011100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
011200	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00
011300	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00
011400	40	00	00	00	00	c0	00	00	00	f2	00	00	00	00	00	44	0.... ..D
011500	74	61	70	74	00	00	00	14	63	6c	65	66	00	00	00	00	tapt....clef....
011600	00	c0	00	00	00	f2	00	00	00	00	00	14	70	72	6f	66prof
011700	00	00	00	00	00	c0	00	00	00	f2	00	00	00	00	00	14
011800	65	6e	6f	66	00	00	00	00	c0	00	00	00	f2	00	00	00	eof.....
011900	00	00	00	24	65	64	74	73	00	00	00	1e	65	6b	73	74	...3edtr....ext

52RD.com

每个 trak 都包含了一个 track header atom

Track Header Atoms - TKHD

每个 trak 都包含了一个 track header atom. The track header atom 定义了一个 track 的特性, 例如时间, 空间和音量信息, 它的类型是('tkhd').

Bytes	
Track header atom	
Atom size	4
Type = 'tkhd'	4
Version	1
Flags	3
Creation time	4
Modification time	4
Track ID	4
Reserved	4
Duration	4
Reserved	8
Layer	2
Alternate group	2
Volume	2
Reserved	2
Matrix structure	36
Track width	4
Track height	4

52RD.com

字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	tkhd
版本	1	这个 atom 的版本
标志	3	有效的标志是 <input type="checkbox"/> 0x0001 - the track is enabled <input type="checkbox"/> 0x0002 - the track is used in the movie <input type="checkbox"/> 0x0004 - the track is used in the movie's preview <input type="checkbox"/> 0x0008 - the track is used in the movie's poster

生成时间	4	Movie atom 的起始时间。基准时间是 1904-1-1 0:00 AM
修订时间	4	Movie atom 的修订时间。基准时间是 1904-1-1 0:00 AM
Track ID	4	唯一标志该 track 的一个非零值。
保留	4	这里为 0
Duration	4	The duration of this track (in the movie's time coordinate system). Note that this property is derived from the track's edits. The value of this field is equal to the sum of the durations of all of the track's edits. If there is no edit list, then the duration is the sum of the sample durations, converted into the movie timescale.
保留	8	这里为 0
Layer	2	The track's spatial priority in its movie. The QuickTime Movie Toolbox uses this value to determine how tracks overlay one another. Tracks with lower layer values are displayed in front of tracks with higher layer values.
Alternate group	2	A collection of movie tracks that contain alternate data for one another. QuickTime chooses one track from the group to be used when the movie is played. The choice may be based on such considerations as playback quality, language, or the capabilities of the computer.
音量	2	播放此 track 的音量。1.0 为正常音量
保留	2	这里为 0
矩阵结构	36	该矩阵定义了此 track 中两个坐标空间的映射关系
宽度	4	如果该 track 是 video track，此值为图像的宽度
高度	4	如果该 track 是 video track，此值为图像的高度

Audio track 的值

00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
00000 00 00 00 18 66 74 79 7d 6d 70 34 32 00 00 00 01ftypmp42....
00010 6d 70 34 32 6d 70 34 31 00 00 2e 2e 6d 6e 6e 76	mp42mp41../moov
00020 00 00 00 6c 6d 76 68 64 00 00 00 00 c1 b8 4a 63	...lmvhd.... Jc
00030 c1 b8 4a 63 00 00 02 58 00 00 a4 10 00 01 00 00	Jc...X... ..
00040 01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00050 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00060 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 000...
00070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00080 00 00 00 00 00 00 00 00 00 00 00 03 00 00 10 5c\
00090 74 72 61 6b 00 00 00 5c 74 6b 68 64 00 00 00 01	trak...tkhd....
000a0 c1 b8 4a 63 c1 b8 4a 63 00 00 00 01 00 00 00 00	Jc Jc.....
000b0 00 00 a4 10 00 00 00 00 00 00 00 00 00 00 00 00
000c0 01 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00
000d0 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00
000e0 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 000.....
000f0 00 00 00 24 65 64 74 73 00 00 00 1c 65 6c 73 74	...iedts....elist
00100 00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
00110 00 01 00 00 00 00 0f 64 6d 64 69 61 00 00 00 20mdia...

Video track 的值

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcedef	
0100d	00	0e	be	45	00	0e	bf	03	00	0e	d1	ae	00	0e	69	25 %	
0100e	00	0e	4c	15	00	0c	e1	16	00	00	1e	50	74	72	61	6b	.. E.. ..\trak	
0100f	00	00	00	5e	74	6b	68	64	00	00	00	01	e1	29	4a	54	...tkhd... JT	
01100	e1	b8	4a	63	00	00	00	02	00	00	00	00	00	00	00	a9	10	Jo.....
01110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01120	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01130	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
01140	40	00	00	00	00	e0	00	00	00	12	00	00	00	00	00	00	44	8.....\3...D
01150	74	61	70	74	00	00	00	14	63	6e	65	66	00	00	00	00	00	tapt....cief....
01160	00	e0	00	00	00	22	00	00	00	00	00	14	70	72	68	66 prof	
01170	00	00	00	00	00	e0	00	00	00	12	00	00	00	00	00	00	14
01180	65	6e	6c	66	00	00	00	00	00	00	00	00	00	12	00	00	00	enof.....
01190	00	00	00	24	65	64	74	73	00	00	00	1e	65	6e	73	74 sedes....cist	
011a0	00	00	00	00	00	00	00	01	00	00	00	a4	10	00	00	00	00
011b0	00	01	00	00	00	00	1d	90	6d	64	69	61	00	00	00	00	20 mdia...
011c0	6d	64	68	64	00	00	00	01	b8	4a	63	e1	b8	4a	63	00	00	mdhd.... Jo Jo
011d0	00	00	02	58	00	00	a4	10	15	e7	00	00	00	00	00	00	35	..X.. .. . 5
011e0	68	64	6e	73	00	00	00	00	00	00	00	26	69	64	65	00	00	hdtr.....vide

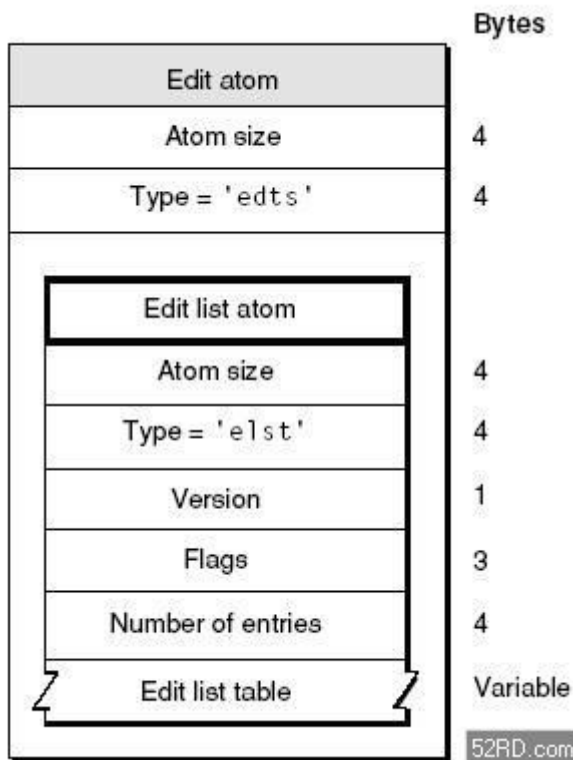
52RD.com

Edit Atoms - EDTS

Edit atoms 定义了创建 movie 中一个 track 的一部分媒体。所有的 edit 都在一个表里面，包括每一部分的时间偏移量和长度。Edit atoms 的类型是'edts'。如果没有该表，则此 track 会被立即播放。一个空的 edit 用来偏移 track 的起始时间。

如果没有 edit atom 或 edit list atom，则此 track 使用全部媒体。

Edit atoms 是一个容器 atom，本身没有特别的字段，需要子 atom 来进一步说明有效的内容。



字段	长度(字节)	描述
----	--------	----

尺寸	4	这个 atom 的字节数
类型	4	Edts

Audio track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
00000	00 00 00 18 66 74 79 70 6d 70 34 32 00 00 00 01ftypmp42....
00010	6d 70 34 32 6d 70 34 31 00 00 2f 2c 6d 6f 6f 76	mp42mp41..f,moov
00020	00 00 00 6c 6d 76 68 64 00 00 00 00 c1 b8 4a 63	...lmvhd.... Jc
00030	c1 b8 4a 63 00 00 02 58 00 00 a4 10 00 01 00 00	Jc...X..
00040	01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00050	00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00060	00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00@...
00070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00080	00 00 00 00 00 00 00 00 00 00 00 00 03 00 10 5c\
00090	74 72 61 6b 00 00 00 5c 74 6b 68 64 00 00 00 01	trak...\tkhd....
000a0	c1 b8 4a 53 c1 b8 4a 63 00 00 00 01 00 00 00 00	JS Jc.....
000b0	00 00 a4 10 00 00 00 00 00 00 00 00 00 00 00 00
000c0	01 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00
000d0	00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00
000e0	00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00@.....
000f0	00 00 00 24 65 64 74 73 00 00 00 1c 65 6c 73 74	...edts...elst
00100	00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
00110	00 01 00 00 00 00 01 a4 6d 64 69 61 00 00 00 20mdia...
00120	6d 64 68 64 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd.... Je Je

Video track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
010e0	00 0c 8f 45 00 0c e1 f6 00 00 1e 5c 74 72 61 6b	.. E.. ... \trak
010f0	00 00 00 5c 74 6b 68 64 00 00 00 01 c1 b8 4a 54	... \tkhd.... JT
01100	c1 b8 4a 63 00 00 00 02 00 00 00 00 00 00 a4 10	Jc.....
01110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01120	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01130	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01140	40 00 00 00 00 c0 00 00 00 f2 00 00 00 00 00 44	@.....D
01150	74 61 70 74 00 00 00 14 63 6c 65 66 00 00 00 00	tapt....clef....
01160	00 c0 00 00 00 f2 00 00 00 00 00 14 70 72 6f 66prof
01170	00 00 00 00 00 c0 00 00 00 f2 00 00 00 00 00 14
01180	65 6c 6f 68 00 00 00 00 c0 00 00 00 f2 00 00	eof.....
01190	00 00 00 24 65 64 74 73 00 00 00 1c 65 6c 73 74	...edts...elst
011a0	00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
011b0	00 01 00 00 00 00 1d 50 6d 64 69 61 00 00 00 20mdia...
011c0	6d 64 68 64 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd.... Je Je
011d0	00 00 02 58 00 00 a4 10 15 c7 00 00 00 00 00 35	...X.. .. .S

Edit List Atoms - ELST

1. Edit list atom 用来映射 movie 的时间到此 track media 的时间。所有信息在一个 edit list 表中，见下图。Edit list atoms 的类型是'elst'.

字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	elst
版本	1	这个 atom 的版本
标志	3	这里为 0
条目数目	4	后面的 edit list 表中的条目数目
edit list	可变	每一个条目包含 3 项，见下图和下表

表		
---	--	--

Track duration	Media time	Media rate	Field
4	4	4	Bytes

字段	长度(字节)	描述
Track duration	4	duration of this edit segment in units of the movie's time scale.
时间	4	starting time within the media of this edit segment (in media timescale units). 值为-1 表示是空 edit。Track 中的最后一个 edit 永远不能为空。Any difference between the movie's duration and the track's duration is expressed as an implicit empty edit.
速度	4	relative rate at which to play the media corresponding to this edit segment. 不能是 0 或负数。

Audio track 的值

00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 0123456789abcdef	
00000 00 00 00 18 66 74 79 70 6d 70 34 32 00 00 00 01 ...rctypsp42...	
00010 6d 70 34 32 6d 70 34 31 00 00 2f 2e 6d 6f 6e 76 mp42mp41.../mov	
00020 00 00 00 6c 6d 76 68 64 00 00 00 00 c1 b8 4a 63 ...lmvhd... Jc	
00030 c1 b8 4a 63 00 00 02 58 00 00 a4 10 00 01 00 00 Jc...X... ..	
00040 01 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00	
00050 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00	
00060 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00	
00070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00080 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 10 5c	
00090 74 72 61 6b 00 00 00 5c 74 6b 69 64 00 00 00 01 track...tchhd...	
000a0 c1 b8 4a 53 c1 b8 4a 63 00 00 00 01 00 00 00 00 Jc Jc.....	
000b0 00 00 a4 10 00 00 00 00 00 00 00 00 00 00 00 00	
000c0 01 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00	
000d0 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00	
000e0 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 ...0.....	
000f0 00 00 00 24 65 64 74 73 00 00 00 1c 65 6c 73 74 ...\$edts...e1st	
00100 00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00	
00110 00 01 00 00 00 00 0f d4 6d 64 69 61 00 00 00 20 media...	
00120 6d 64 68 64 00 00 00 00 c1 b8 4a 63 c1 b8 4a 63 mdhd... Jc Jc	
00130 00 00 1f 40 00 08 8c 00 15 e7 00 00 00 00 00 355	
00140 68 64 6c 72 00 00 00 00 00 00 00 73 6f 75 6e hdir.....son	

Video track 的值

00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 0123456789abcdef	
010e0 00 0c 8f 45 00 0c e1 f6 00 00 1e 5c 74 72 61 6b .. E.. ...trak	
010f0 00 00 00 3c 74 6b 68 64 00 00 00 01 c1 b8 4a 54 ...tchhd... JT	
01100 c1 b8 4a 63 00 00 00 02 00 00 00 00 00 00 a4 10 Jc.....	
01110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01120 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01130 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01140 40 00 00 00 00 c0 00 00 00 12 00 00 00 00 44 0	
01150 74 61 70 74 00 00 14 63 6c 65 66 00 00 00 00 00 capt...clef...	
01160 00 c0 00 00 00 12 00 00 00 00 14 70 72 5f 66	
01170 00 00 00 00 00 c0 00 00 00 12 00 00 00 00 14	
01180 65 6e 6f 66 00 00 00 00 c0 00 00 00 12 00 00 enof.....	
01190 00 00 00 24 65 64 74 73 00 00 00 1c 65 6c 73 74 ...\$edts...e1st	
011a0 00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00	
011b0 00 01 00 00 00 00 1d 90 6d 64 69 61 00 00 00 20 media...	
011c0 6d 64 68 64 00 00 00 c1 b8 4a 63 c1 b8 4a 63 mdhd... Jc Jc	
011d0 00 00 02 58 00 00 a4 10 15 e7 00 00 00 00 00 35 ...X... ..5	

Track Reference Atoms - TREF

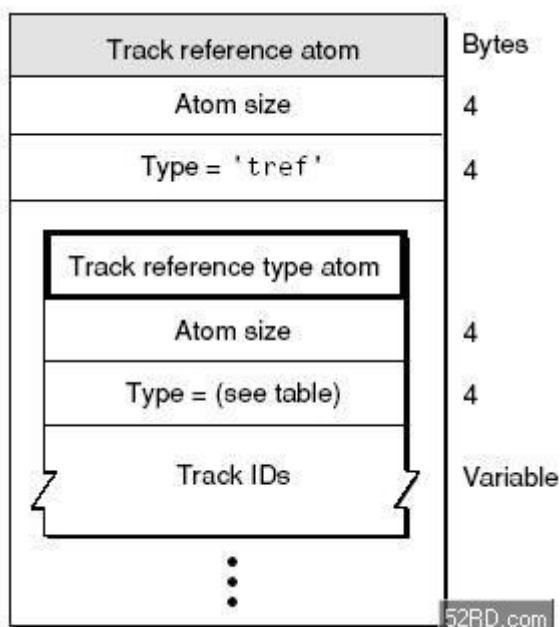
Track reference atoms define relationships between tracks. Track reference atoms allow one track to specify how it is related to other tracks. For example, if a movie has three video tracks and three sound tracks, track references allow you to identify the related sound and video tracks. Track reference atoms have an atom type value of 'tref'.

Track references are uni-directional and point from the recipient track to the source track. For example, a video track may reference a time code track to indicate where its time code is stored, but the time code track would not reference the video track. The time code track is the source of time information for the video track.

A single track may reference multiple tracks. For example, a video track could reference a sound track to indicate that the two are synchronized and a time code track to indicate where its time code is stored.

A single track may also be referenced by multiple tracks. For example, both a sound and video track could reference the same time code track if they share the same timing information.

If this atom is not present, the track is not referencing any other track in any way. Note that the array of track reference type atoms is sized to fill the track reference atom. Track references with a reference index of 0 are permitted. This indicates no reference.



Each track reference atom defines relationships with tracks of a specific type. The reference type implies a track type. Following table shows the track reference types and their descriptions.

Table: Track reference types

Reference type	Description
tmcd	Time code. Usually references a time code track.
chap	Chapter or scene list. Usually references a text track.
sync	Synchronization. Usually between a video and sound track. Indicates that the two tracks are synchronized. The reference can be from either track to the other, or there may be two references.
scpt	Transcript. Usually references a text track.
ssrc	Nonprimary source. Indicates that the referenced track should send its data to this track, rather than presenting it. The referencing track will use the data to modify how it presents its data. See “Track Input Map Atoms” (page 51) for more information.
hint	The referenced tracks contain the original media for this hint track.

Each track reference type atom contains the following data elements.

Size

A 32-bit integer that specifies the number of bytes in this track reference type atom.

Type

A 32-bit integer that identifies the atom type; this field must be set to one of the values shown in above table.

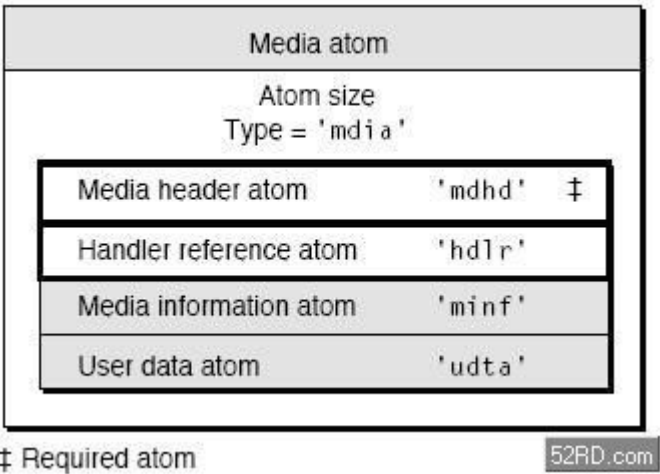
Track IDs

A list of track ID values (32-bit integers) specifying the related tracks. Note that this is one case where track ID values can be set to 0. Unused entries in the atom may have a track ID value of 0. Setting the track ID to 0 may be more convenient than deleting the reference. You can determine the number of track references stored in a track reference type atom by subtracting its header size from its overall size and then dividing by the size, in bytes, of a track ID.

Media Atoms - MDIA

Media atoms 定义了 track 的媒体类型和 sample 数据，例如音频或视频，描述 sample 数据的 media handler component，media timescale and track duration 以及 media-and-track-specific 信息，例如音量和图形模式。它也可以包含一个引用，指明媒体数据存储在另一个文件中。也可以包含一个 sample table atoms，指明 sample description, duration, and byte offset from the data reference for each media sample.

Media atom 的类型是'mdia'。它是一个容器 atom，必须包含一个 media header atom ('mdhd')，一个 handler reference ('hdlr')，一个媒体信息引用('minf')和用户数据 atom('udta')。



字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	Edts

Audio track 的值

000000	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
000000	00 00 00 18 66 74 79 70 6d 70 34 32 00 00 00 01ftypmp42....
000010	6d 70 34 32 6d 70 34 31 00 00 1f 2c 6d 6f 6f 76	mp42mp41../moov
000020	00 00 00 6c 6d 76 68 64 00 00 00 00 c1 b8 4a 63	...lmvhd.... Jc
000030	c1 b8 4a 63 00 00 02 58 00 00 a4 10 00 01 00 00	Jc...X... ..
000040	01 00 00 00 00 00 00 00 00 00 00 00 01 00 00
000050	00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
000060	00 00 00 00 00 00 00 00 00 00 00 00 40 00 00B...
000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000080	00 00 00 00 00 00 00 00 00 00 00 00 03 00 10\
000090	74 72 61 6b 00 00 00 5c 74 6b 68 64 00 00 01	trak...\tkhd....
0000a0	c1 b8 4a 63 e1 b8 4a 63 00 00 01 00 00 00 00	JJ Jc.....
0000b0	00 00 a4 10 00 00 00 00 00 00 00 00 00 00 00
0000c0	01 00 00 00 01 00 00 00 00 00 00 00 00 00 00
0000d0	00 00 00 00 01 00 00 00 00 00 00 00 00 00 00
0000e0	00 00 00 40 00 00 00 00 00 00 00 00 00 00 00B.....
0000f0	00 00 00 24 65 64 74 73 00 00 1c 65 6c 73 74	...fedts....elst
000100	00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
000110	00 01 00 00 00 00 cf d4 6d 64 69 61 00 00 20mdia...
000120	6d 64 69 64 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd.... Jc Jc
000130	00 00 1f 40 00 08 8c 00 15 c7 00 00 00 00 35	...@... ..5

Video track 的值

011800	65 6e 62 66 00 00 00 00 c0 00 00 00 f2 00 00	eno2.....
011810	00 00 00 24 65 64 74 73 00 00 1c 65 6c 73 74	...fedts....elst
011820	00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
011830	00 01 00 00 00 00 1d 90 6d 64 69 61 00 00 20mdia...
011840	6d 64 69 64 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd.... Jc Jc
011850	00 00 02 58 00 00 a4 10 15 c7 00 00 00 00 35	...X... ..5
011860	69 64 6c 72 00 00 00 00 00 00 00 00 76 69 64	hdlr.....vide
011870	00 00 00 00 00 00 00 00 00 00 00 00 c6 bb b9 fb
011880	ca d3 c6 b5 c3 bd cc e5 b4 ae c0 ed b3 cc d0 f2
011890	00 00 00 1d 33 6d 69 6e 6d 00 00 14 76 6d 683minf....vmh
0118a0	64 00 00 01 00 00 00 00 00 00 00 00 00 00 00	d.....
0118b0	24 64 69 6e 66 00 00 1c 64 72 65 66 00 00 00	\$dinf....dref...

Media Header Atoms - MDHD

Media header atom 定义了媒体的特性，例如 time scale 和 duration。它的类型是'mdhd'.

Bytes	
Media header atom	
Atom size	4
Type = 'mdhd '	4
Version	1
Flags	3
Creation time	4
Modification time	4
Time scale	4
Duration	4
Language	2
Quality	2

52RD.com

字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	mdhd
版本	1	这个 atom 的版本
标志	3	这里为 0
生成时间	4	Movie atom 的起始时间。基准时间是 1904-1-1 0:00 AM
修订时间	4	Movie atom 的修订时间。基准时间是 1904-1-1 0:00 AM
Time scale	4	A time value that indicates the time scale for this media—that is, the number of time units that pass per second in its time coordinate system.
Duration	4	The duration of this media in units of its time scale.
语言	2	媒体的语言码
质量	2	媒体的回放质量 ??? 怎样生成此质量，什么是参照点

Audio track 的值

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
000e0	00	00	00	00	40	00	00	00	00	00	00	00	00	00	00	00B.....
000f0	00	00	00	24	65	64	74	73	00	00	00	1e	65	6e	73	74	...\$edts...e1st
00100	00	00	00	00	00	00	00	01	00	00	a4	10	00	00	00	00
00110	00	01	00	00	00	00	0f	d4	6d	64	69	61	00	00	00	20mdie...
00120	6d	64	68	64	00	00	00	c1	b8	4a	63	c1	b8	4a	63		mdhd.... Jc Jc
00130	00	00	1f	40	00	08	8e	00	15	e7	00	00	00	00	35	B....J....5
00140	68	64	6e	72	00	00	00	00	00	00	00	00	73	6f	75	6e	hdr.....soun
00150	00	00	00	00	00	00	00	00	00	00	00	00	e6	bb	b9	2b
00160	c9	e9	c6	b5	c3	bd	cc	e5	b4	a6	cd	ed	b3	cc	d0	f2
00170	00	00	00	0f	77	6d	69	6e	66	00	00	00	10	73	6d	68wminf....smh
00180	64	00	00	00	00	00	00	00	00	00	00	00	24	64	69	6e	d.....\$din
00190	66	00	00	00	1e	64	72	65	66	00	00	00	00	00	00	00	f.....dref.....

52RD.com

52RD.com

Video track 的值

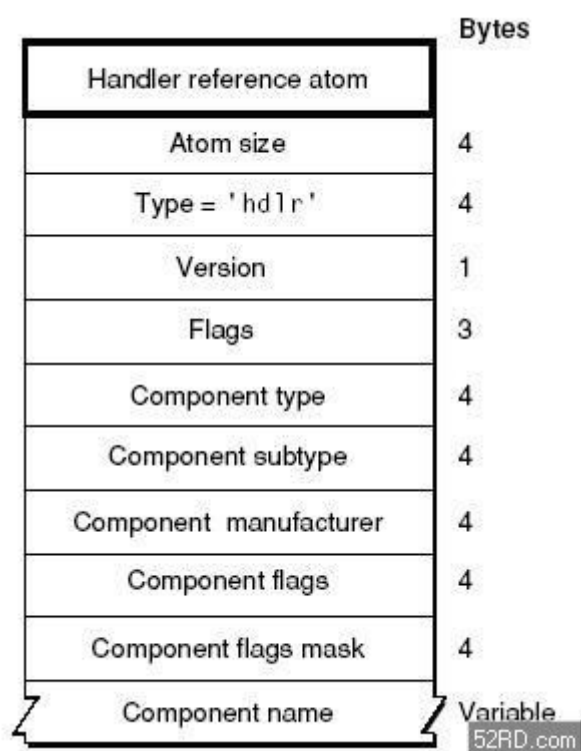
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
01180	65	6e	6f	66	00	00	00	00	00	00	00	00	00	f2	00	00	eof.....
01190	00	00	00	24	65	64	74	73	00	00	00	1e	65	6e	73	74	...\$edts...e1st
011a0	00	00	00	00	00	00	00	01	00	00	a4	10	00	00	00	00
011b0	00	01	00	00	00	00	1d	90	6d	64	69	61	00	00	00	20mdie...
011c0	6d	64	68	64	00	00	00	00	c1	b8	4a	63	c1	b8	4a	63	mdhd....Jc Jc
011d0	00	00	02	58	00	00	a4	10	15	e7	00	00	00	00	00	35	...X...J...S
011e0	68	64	6e	72	00	00	00	00	00	00	00	00	76	69	64	65	hdr.....vide
011f0	00	00	00	00	00	00	00	00	00	00	00	00	e6	bb	b9	2b
01200	ca	d3	c6	b5	c3	bd	cc	e5	b4	a6	cd	ed	b3	cc	d0	f2
01210	00	00	00	1d	33	6d	69	6e	66	00	00	00	14	76	6d	683wminf....vmh
01220	64	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00	d.....
01230	24	64	69	6e	66	00	00	00	1e	64	72	65	66	00	00	00	\$dinf....dref...
01240	00	00	00	00	01	00	00	00	0e	75	72	6e	20	00	00	00wrl...
01250	01	00	00	1e	d3	73	74	62	6e	00	00	00	ab	73	74	73stbl...sts
01260	64	00	00	00	00	00	00	01	00	00	00	9b	6d	70	34	d.....mp4	

52RD.com

52RD.com

Handler Reference Atoms - HDLR

Handler reference atom 定义了描述此媒体数据的 media handler component，类型是'hdlr'。在过去，handler reference atom 也可以用来数据引用，但是现在，已经不允许这样使用了。一个 media atom 内的 handler atom 解释了媒体流的播放过程。例如，一个视频 handler 处理一个 video track。



字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	hdlr
版本	1	这个 atom 的版本
标志	3	这里为 0
Component type	4	handler 的类型。当前只有两种类型 □ 'mhlr': media handlers □ 'dhlr': data handlers
Component subtype	4	media handler or data handler 的类型。 如果 component type 是 mhlr, 这个字段定义了数据的类型, 例如, 'vide'是 video 数据, 'soun'是 sound 数据 如果 component type 是 dhlr, 这个字段定义了数据引用的类型, 例如, 'alis'是文件的别名
Component manufacturer	4	保留字段, 缺省为 0
Component flags	4	保留字段, 缺省为 0
Component flags mask	4	保留字段, 缺省为 0
Component name	可变	这个 component 的名字, 也就是生成此 media 的 media handler。

t name		该字段的长度可以为 0
--------	--	-------------

Audio track 的值

	01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
000e0	00 00 00 00 40 00 00 00 00 00 00 00 00 00 000.....
000f0	00 00 00 24 65 64 74 73 00 00 00 1e 65 6e 73 74	...\$edts...e1st
00100	00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
00110	00 01 00 00 00 00 0f d4 6d 64 69 61 00 00 00 20mdia...
00120	6d 64 68 64 00 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd....Jc Jc
00130	00 00 1f 40 00 08 8c 00 15 c7 00 00 00 00 00 35	...8...5
00140	68 64 6e 72 00 00 00 00 00 00 00 00 73 6f 75 6e	hdlr.....soun
00150	00 00 00 00 00 00 00 00 00 00 00 00 c6 bb b9 fb
00160	c9 f9 c6 b5 c3 bd cc e5 b4 a6 c0 ed b3 cc d0 f2
00170	00 00 00 0f 77 6d 69 6e 66 00 00 00 10 73 6d 68	...uminf...smh
00180	64 00 00 00 00 00 00 00 00 00 00 00 24 64 69 6e	d.....\$din
00190	66 00 00 00 1c 44 72 65 66 00 00 00 00 00 00 00	f...dref.....
001a0	01 00 00 00 0c 75 72 6c 20 00 00 00 01 00 00 0furl
001b0	3b 73 74 62 6c 00 00 00 67 73 74 73 64 00 00 00	stbl...gsted...

52RD.com

Video track 的值

	01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
01180	65 6e 6f 66 00 00 00 00 c0 00 00 00 f3 00 00	enof....
01190	00 00 00 24 65 64 74 73 00 00 00 1e 65 6e 73 74	...\$edts...e1st
011a0	00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
011b0	00 01 00 00 00 00 1d 90 6d 64 69 61 00 00 00 20mdia...
011c0	6d 64 68 64 00 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd....Jc Jc
011d0	00 00 02 58 00 00 a4 10 15 c7 00 00 00 00 00 35	...X...5
011e0	68 64 6e 72 00 00 00 00 00 00 00 00 76 69 64 65	hdlr.....vide
011f0	00 00 00 00 00 00 00 00 00 00 00 00 c6 bb b9 fb
01200	ca d3 c6 b5 c3 bd cc e5 b4 a6 c0 ed b3 cc d0 f2
01210	00 00 00 1d 33 6d 69 6e 66 00 00 00 14 76 6d 68	...3minf...vmh
01220	64 00 00 00 01 00 00 00 00 00 00 00 00 00 00	d.....
01230	24 64 68 6e 66 00 00 00 1c 64 72 65 66 00 00 00	\$dinf...dref...
01240	00 00 00 00 01 00 00 00 0c 75 72 6c 20 00 00 00url
01250	01 00 00 1c f3 73 74 62 6c 00 00 00 ab 73 74 73stbl...sts

52RD.com

Media Information Atoms - MINF

Media information atoms 的类型是'minf'，存储了解释该track的媒体数据的 handler-specific 的信息。media handler用这些信息将媒体时间映射到媒体数据，并进行处理。它是一个容器 atom，包含其他的子 atom。

这些信息是与媒体定义的数据类型特别对应的，而且 media information atoms 的格式和内容也是与解释此媒体数据流的 media handler 密切相关的。其他的 media handler 不知道如何解释这些信息。

字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	minf

Audio track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
000e0	00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 008.....
000f0	00 00 00 24 65 64 74 73 00 00 00 1e 65 6e 73 74	...\$edts....elst
00100	00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
00110	00 01 00 00 00 00 0f d4 6d 64 69 61 00 00 00 20mdia...
00120	6d 64 68 64 00 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd....Jc Jc
00130	00 00 1f 40 00 00 8c 00 15 c7 00 00 00 00 00 35	...@.....5
00140	68 64 6e 72 00 00 00 00 00 00 00 73 6f 78 6e	hdlr.....soun
00150	00 00 00 00 00 00 00 00 00 00 00 c6 bb b9 fb
00160	e9 f9 e6 b5 e3 bd cc e5 b4 a6 e0 ed b3 cc d0 f2	
00170	00 00 00 0f 77 6d 69 6e 6d 00 00 00 10 73 6d 68	...sminf....smh
00180	64 00 00 00 00 00 00 00 00 00 00 24 64 69 6e	d.....\$din
00190	68 00 00 00 1e 64 72 65 6e 00 00 00 00 00 00 00	£....dref.....
001a0	01 00 00 00 0e 75 72 6e 20 00 00 00 01 00 00 0furl

52RD.com

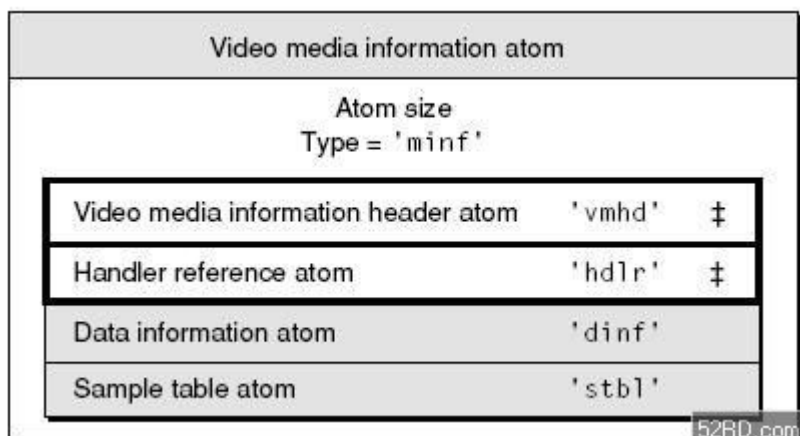
Video track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
01180	65 6e 6f 66 00 00 00 00 e0 00 00 00 f2 00 00	enof.....
01190	00 00 00 24 65 64 74 73 00 00 00 1e 65 6e 73 74	...\$edts....elst
011a0	00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
011b0	00 01 00 00 00 00 1d 90 6d 64 69 61 00 00 00 20mdia...
011c0	6d 64 68 64 00 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd....Jc Jc
011d0	00 00 02 58 00 00 a4 10 15 c7 00 00 00 00 00 35	...X.....5
011e0	68 64 6e 72 00 00 00 00 00 00 00 76 69 64 65	hdlr.....vide
011f0	00 00 00 00 00 00 00 00 00 00 00 c6 bb b9 fb
01200	ca d3 e6 b5 e3 bd cc e5 b4 a6 e0 ed b3 cc d0 f2	
01210	00 00 00 1d 33 6d 69 6e 6d 00 00 00 14 76 6d 68	...3minf....vmh
01220	64 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00	d.....
01230	24 64 69 6e 6d 00 00 1e 64 72 65 6e 00 00 00 00	\$dinf....dref...
01240	00 00 00 00 01 00 00 00 0e 75 72 6e 20 00 00 00url ...
01250	01 00 00 1e 73 74 62 6e 00 00 00 ab 73 74 73stbl...sts
01260	64 00 00 00 00 00 00 01 00 00 00 9b 6d 70 34	d.....mp4

52RD.com

Video Media Information Atoms

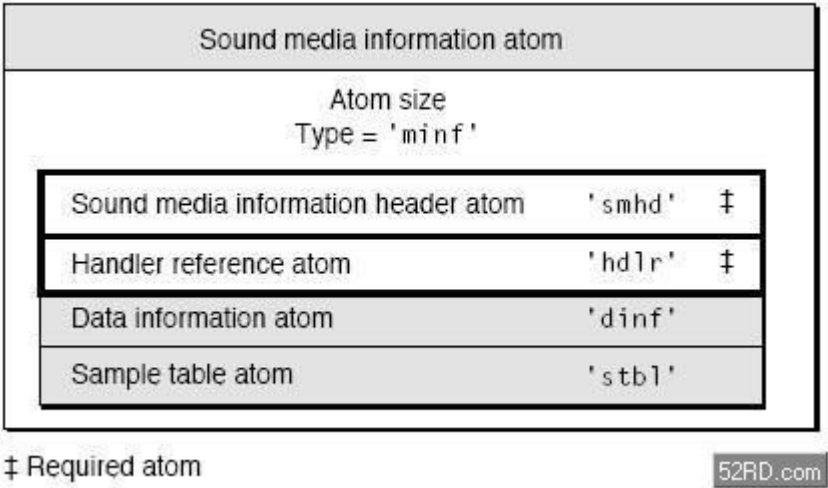
Video media information atoms 是视频媒体的第一层 atoms，包含其他的定义视频媒体数据的特性。



52RD.com

Sound Media Information Atoms

Sound media information atoms 是音频媒体的第一层 atoms，包含其他的定义音频媒体数据的特性。



Video Media Information Header Atoms

Video media information header atoms 定义颜色和图形模式信息。

Bytes	
Video media information header atom	
Atom size	4
Type = 'vmhd'	4
Version	1
Flags	3
Graphics mode	2
Opcolor	6

52RD.com

字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	vmhd
版本	1	这个 atom 的版本
标志	3	这里总是 0x000001
图形模式	2	The transfer mode. The transfer mode specifies which Boolean operation QuickDraw should perform when drawing or transferring an image from one location to another.
Opcolor	6	Three 16-bit values that specify the red, green, and blue colors for the transfer mode operation indicated in the graphics mode field.

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
01180	65 6e 6f 66 00 00 00 00 00 00 00 00 00 00 00 00	enof.....
01190	00 00 00 24 65 64 74 73 00 00 00 1c 65 6c 73 74	...\$edcs...elst
011a0	00 00 00 00 00 00 00 01 00 00 a4 10 00 00 00 00
011b0	00 01 00 00 00 00 1d 90 6d 64 69 61 00 00 00 20mdia...
011c0	6d 64 68 64 00 00 00 00 c1 b8 4a 63 c1 b8 4a 63	mdhd....dcjc
011d0	00 00 02 58 00 00 a4 10 15 c7 00 00 00 00 00 35	...X.....5
011e0	68 64 6c 72 00 00 00 00 00 00 00 00 76 69 64 65	hdlr.....vide
011f0	00 00 00 00 00 00 00 00 00 00 00 00 c6 bb b9 fb
01200	ca d3 c6 b5 c3 bd cc e5 b4 a6 c0 ed b3 cc d0 f2	
01210	00 00 00 1d 33 6d 69 6e 66 00 00 00 14 76 6d 683minf....vmh
01220	64 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00	d.....
01230	24 64 69 6e 66 00 00 00 1c 64 72 65 66 00 00 00	\$dinf....dref...
01240	00 00 00 00 01 00 00 00 0c 75 72 6c 20 00 00 00url...
01250	01 00 00 1c f3 73 74 62 6e 00 00 00 ab 73 74 73stbi...sts
01260	64 00 00 00 00 00 00 00 01 00 00 00 9b 6d 70 34	d.....mp4

52RD.com

Sound Media Information Header

Atoms - SMHD

The sound media information header atom 定义了声音媒体的控制信息，例如均衡。

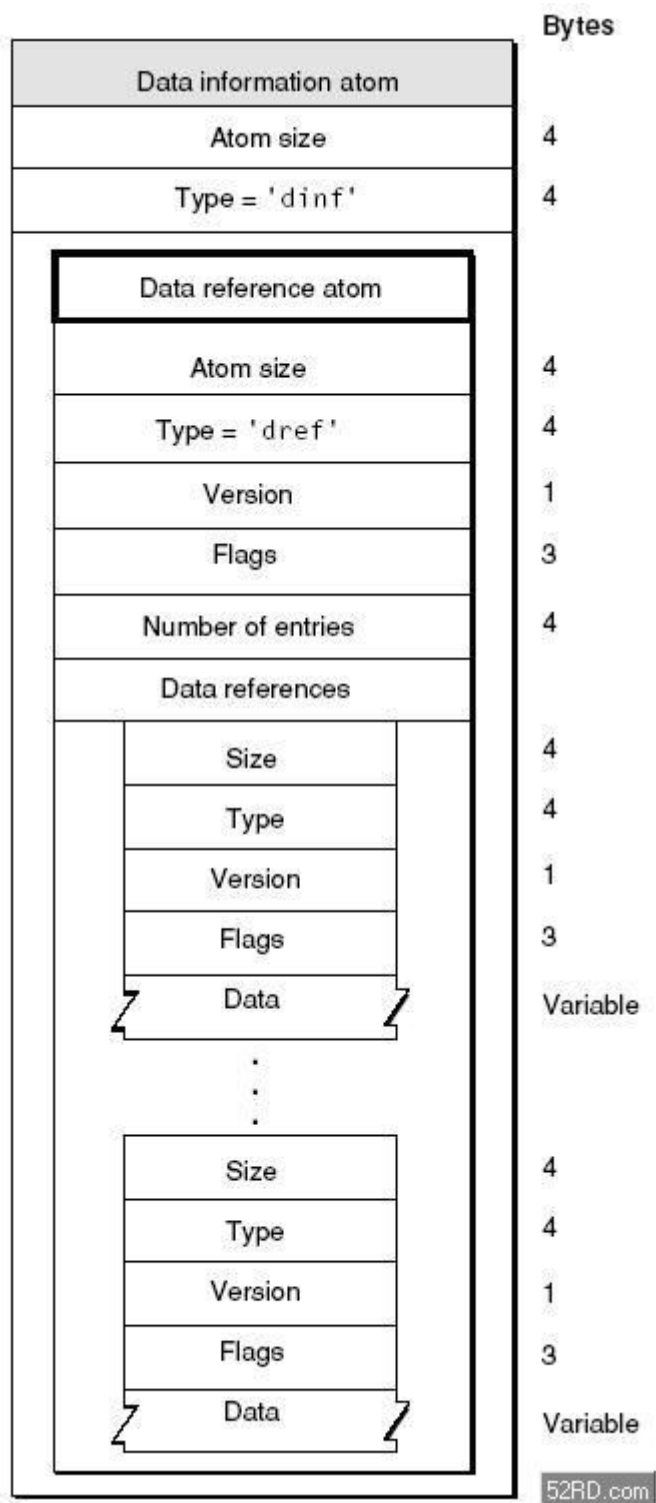
500)this.width=500'>

字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	smhd
版本	1	这个 atom 的版本
标志	3	这里为 0
均衡	2	音频的均衡是用来控制计算机的两个扬声器的声音混合效果，一般是 0。一般值是 0。
保留	2	保留字段，缺省为 0

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 0123456789abcdef
00140	88 84 6c 72 00 00 00 00 00 00 00 00 73 8f 75 6e hdir.....soun
00150	00 00 00 00 00 00 00 00 00 00 00 00 c6 bb b9 db
00160	c9 19 c6 b5 c3 bd cc e5 b4 a6 c0 ed b3 cc d0 f2wminf....smh
00170	00 00 00 0f 77 6d 69 6e 66 00 00 00 10 73 6d 65d.....\$din
00180	64 00 00 00 00 00 00 00 00 00 00 00 24 64 69 6ef.....dref.....
00190	66 00 00 00 1c 64 72 65 66 00 00 00 00 00 00 00ucl
001a0	01 00 00 00 0c 75 72 6c 30 00 00 00 01 00 00 0fjstbl...gated...
001b0	3b 73 74 62 6c 00 00 00 67 73 74 73 64 00 00 00Uupia...
001c0	00 00 00 00 01 00 00 00 57 6d 70 34 61 00 00 003esd
001d0	00 00 00 00 01 00 00 00 00 00 00 00 00 00 02 00\$.....N..N.
001e0	10 00 00 00 00 1f 40 00 00 00 00 00 33 65 73 64
001f0	73 00 00 00 00 03 80 80 80 22 00 00 00 04 80 80
00200	80 14 40 15 00 18 00 00 00 4e 20 00 00 4e 20 05

Data Information Atoms - DINF

handler reference 定义 data handler component 如何获取媒体数据，data handler 用这些数据信息来解释媒体数据。Data information atoms 的类型是'dinf'。它是一个容器 atom，包含其他的 atom。



Audio track 的值

00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
00140	68	64	6c	72	00	00	00	00	00	00	00	73	6f	75	6e	hdr.....soun
00150	00	00	00	00	00	00	00	00	00	00	00	c6	bb	b9	fb
00160	e2	f9	e6	b5	e3	bd	cc	a5	b4	a6	c0	ed	b3	cc	d0	f2
00170	00	00	00	0f	77	6d	69	6e	66	00	00	00	10	73	6d	68
00180	64	00	00	00	00	00	00	00	00	00	00	24	64	69	6e	d.....\$din
00190	6e	00	00	00	1c	64	72	65	66	00	00	00	00	00	00	\$.....dref.....
001a0	01	00	00	00	0c	75	72	6e	20	00	00	00	01	00	00ukl.....
001b0	3b	73	74	62	6c	00	00	00	67	73	74	73	64	00	00	stbl...gstsd...
001c0	00	00	00	00	01	00	00	00	57	6d	70	34	61	00	00Unp4a...
001d0	00	00	00	00	01	00	00	00	00	00	00	00	00	00	02
001e0	10	00	00	00	00	1f	40	00	00	00	00	00	33	65	73	64
001f0	73	00	00	00	00	03	80	80	80	22	00	00	00	04	80	80
00200	80	14	10	15	00	18	00	00	00	4e	20	00	00	4e	20	05

Video track 的值

00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
011e0	68	64	6c	72	00	00	00	00	00	00	00	76	69	64	65	hdr.....vide
011f0	00	00	00	00	00	00	00	00	00	00	00	c6	bb	b9	fb
01200	ca	d3	c6	b5	c3	bd	cc	a5	b4	a6	c0	ed	b3	cc	d0	f2
01210	00	00	00	1d	33	6d	69	6e	66	00	00	00	14	76	6d	68
01220	64	00	00	00	01	00	00	00	00	00	00	00	00	00	00	d.....\$din
01230	24	64	69	6e	66	00	00	00	1c	64	72	65	66	00	00	\$din\$.....dref...
01240	00	00	00	00	01	00	00	00	0c	75	72	6c	20	00	00ukl.....
01250	01	00	00	1c	f3	73	74	62	6c	00	00	00	ab	73	74	73
01260	64	00	00	00	00	00	00	01	00	00	00	9b	6d	70	34	d.....mp4
01270	76	00	00	00	00	00	00	00	01	00	00	00	00	00	00	v.....
01280	00	00	00	00	00	00	00	00	00	e0	00	f2	00	48	00H.
01290	00	00	48	00	00	00	00	00	00	01	00	00	00	00	00	..N.....
012a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Data Reference Atoms - DREF

Data reference atoms 包含列表数据，data handler component 可以用这些数据获取媒体数据。

字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	dref
版本	1	这个 atom 的版本
标志	3	这里为 0
条目数目	4	data references 的数目
Data references		每个 data reference 就像 atom 的格式一样，包含以下的数据成员
尺寸	4	这个 atom 的字节数
类型	4	见下表
版本	1	这个 data reference 的版本
标志	3	目前只有一个标志： Self reference This flag indicates that the media's data is in the same file as the movie atom. On the Macintosh, and other file systems with multifork files, set this flag to 1 even if the data resides in a different fork from the movie atom. This flag's value is 0x0001.
数据	可变	data reference 信息

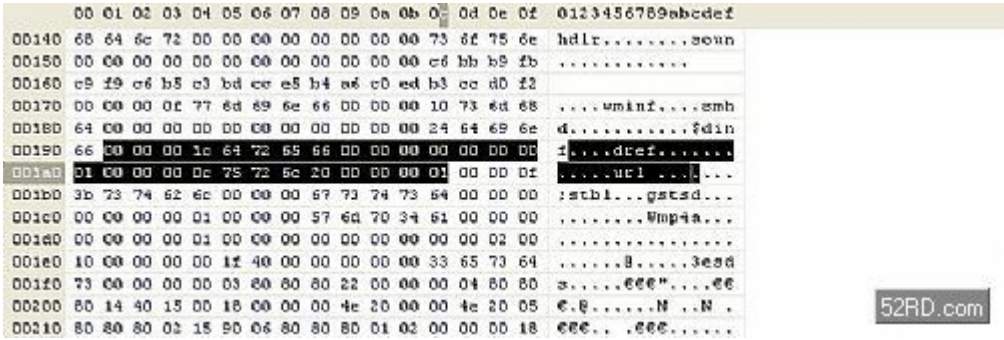
Following Table shows the currently defined data reference types that may be stored in a header

atom.

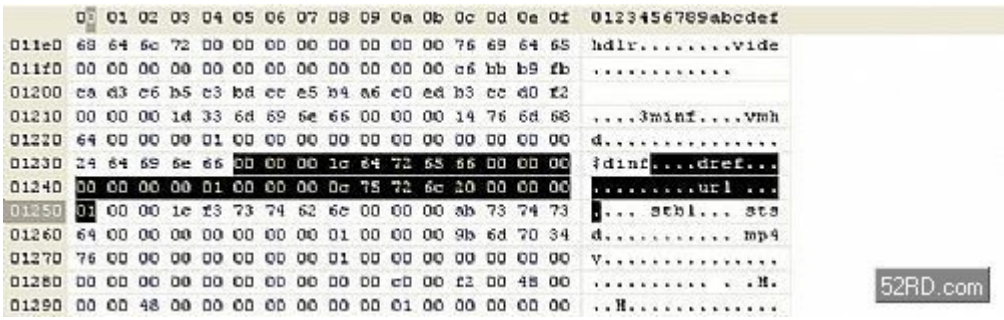
表：Data reference 类型

类型	描述
alis	Data reference 是一个 Macintosh alias。一个 alias 包含文件信息，例如全路径名。
rsrc	Data reference 是一个 Macintosh alias。Alias 末尾是文件使用的资源类型（32bit 整数）和 ID（16bit 带符号的整数）
url	一个 C 类型的字符串，表示一个 URL。字符串后可以有其他的的数据。

Audio track 的值

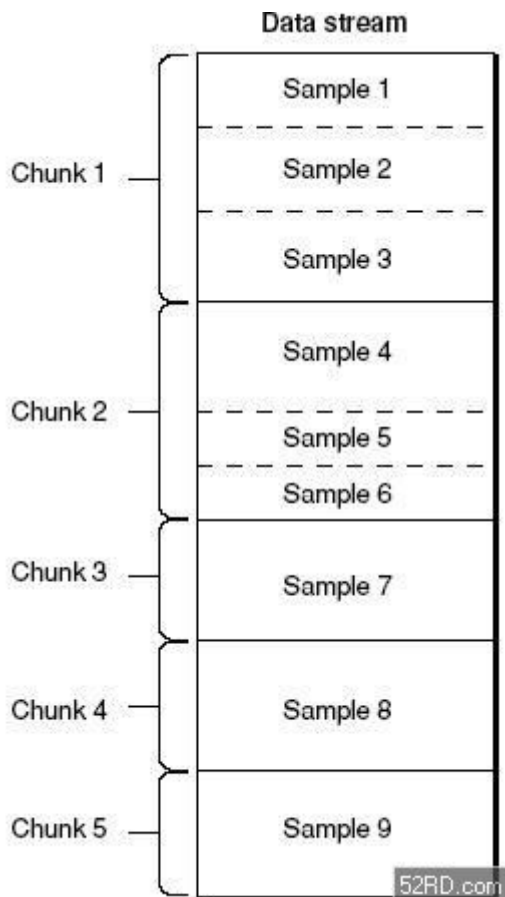


Video track 的值



Sample Atoms

存储媒体数据的单位是 samples。一个 sample 是一系列按时间顺序排列的数据的一个 element。Samples 存储在 media 中的 chunk 内，可以有不同的 durations。Chunk 存储一个或者多个 samples，是数据存取的基本单位，可以有不同的长度，一个 chunk 内的每个 sample 也可以有不同的长度。例如如下图，chunk 2 和 3 不同的长度，chunk 2 内的 sample 5 和 6 的长度一样，但是 sample 4 和 5，6 的长度不同。



sample table atom 描述 sample 的所有信息以及一些不同类型的 atoms，media handler 可以用这些信息正确的按顺序解析所有的 samples，而不需要强迫这些数据按 movie 的时间顺序存放到实际数据流中。

□ 如何得到 chunk 的数目和每个 chunk 包含多少个 sample，每个 chunk 的 description 是如何

解析 stsc - Sample-to-Chunk table。这个表类似于行程编码，第一个 first chunk 减去第二个 first chunk 就是一共有多少个 trunk 包含相同的 sample 数目，这样通过不断的叠加，就可以得到一共有 280 个 trunk，每个 trunk 包含多少个 sample，以及每个 trunk 对应的 description。

□ 如何计算每个 sample 的 duration

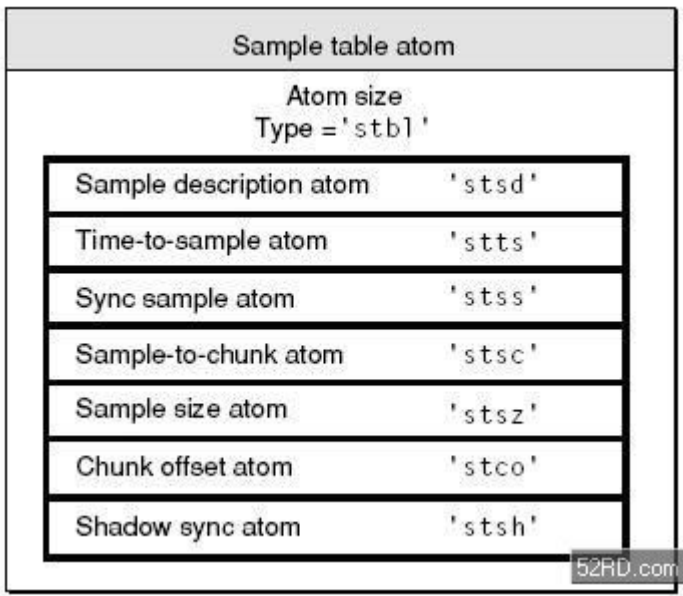
□ 如何计算每个 sample 的长度

解析 stsz - Sample Size table。这个表包含了每个 sample 的长度，找到 sample 的序号，就可以找到对应 sample 的长度了。

Sample Table Atoms - STBL

sample table atom 包含转化媒体时间到实际的 sample 的信息，他也说明了解释 sample 的信息，例如，视频数据是否需要解压缩，解压缩算法是什么？它的类型是'stbl'，是一个容器

atom, 包含 sample description atom, time-to-sample atom, sync sample atom, sample-to-chunk atom, sample size atom, chunk offset atom 和 shadow sync atom. sample table atom 包含 track 中 media sample 的所有时间和数据索引, 利用这个表, 就可以定位 sample 到媒体时间, 决定其类型, 大小, 以及如何在其他容器中找到紧邻的 sample。如果 sample table atom 所在的 track 没有引用任何数据, 那么它就不是一个有用的 media track, 不需要包含任何子 atom。如果 sample table atom 所在的 track 引用了数据, 那么必须包含以下的子 atom: sample description, sample size, sample to chunk 和 chunk offset。所有的子表有相同的 sample 数目。sample description atom 是必不可少的一个 atom, 而且必须包含至少一个条目, 因为它包含了数据引用 atom 检索 media sample 的目录信息。没有 sample description, 就不可能计算出 media sample 存储的位置。sync sample atom 是可选的, 如果没有, 表明所有的 samples 都是 sync samples。



Audio track 的值

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
00140	66	64	6c	72	00	00	00	00	00	00	00	00	73	6f	75	6e	hdr.....soun
00150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00160	c9	e9	c6	b5	c3	b4	cc	e5	b4	a6	c0	ed	b3	cc	d0	f2
00170	00	00	00	0f	77	6d	69	6e	66	00	00	00	10	73	6d	68uninf....swh
00180	64	00	00	00	00	00	00	00	00	00	00	00	24	64	69	6e	d.....Edin
00190	66	00	00	00	1c	64	72	65	66	00	00	00	00	00	00	00	E.....dtef.....
001a0	01	00	00	00	0e	75	72	6e	20	00	00	00	01	00	00	0euel.......
001b0	5b	73	74	62	6c	00	00	00	67	73	74	73	64	00	00	00	pstbl...getsd...
001c0	00	00	00	00	01	00	00	00	57	6d	70	34	61	00	00	00Uppla...
001d0	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00
001e0	10	00	00	00	00	1f	40	00	00	00	00	00	33	65	73	64@.....3esd
001f0	73	00	00	00	00	03	80	80	80	22	00	00	00	04	80	80	s.....eef"....e0

52RD.com

Video track 的值

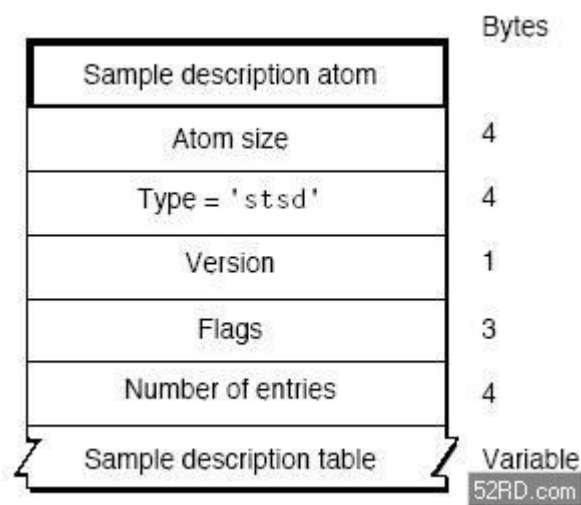
	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
01210	00 00 00 1d 33 6d 6e 6e 00 00 00 14 76 6d 68	...3mainf...vmlh
01220	64 00 00 00 01 00 00 00 00 00 00 00 00 00 00	d.....
01230	24 64 69 6e 66 00 00 00 1c 64 72 65 66 00 00 00	\$dinf...dref...
01240	00 00 00 00 01 00 00 00 0e 75 72 6e 20 00 00 00url...
01250	01 00 00 1c f3 73 74 62 6c 00 00 00 ab 73 74 73	...stbl...sta
01260	64 00 00 00 00 00 00 00 01 00 00 00 9b 6d 70 34	d.....mp4
01270	76 00 00 00 00 00 00 00 01 00 00 00 00 00 00	v.....
01280	00 00 00 00 00 00 00 00 00 c0 00 f2 00 48 00H.
01290	00 00 48 00 00 00 00 00 00 01 00 00 00 00 00	..H.....
012a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
012b0	00 00 00 00 00 00 00 00 00 00 00 18 ff ff 00

52RD.com

Sample Description Atoms - STS

D

利用 sample description atom 存储的信息可以正确的解码 media sample。不同的媒体类型存储不同的 sample description，例如，视频媒体，sample description 就是图像的结构。第四章解释了不同媒体类型对应的 sample description 信息。



sample description atom 的类型是'ststd', 包含了一个 sample description 表。根据不同的编码方案和存储数据的文件数目，每个 media 可以有一个到多个 sample description。sample-to-chunk atom 通过这个索引表，找到合适 medai 中每个 sample 的 description。

字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	stsd
版本	1	这个 atom 的版本
标志	3	这里为 0
条目数目	4	sample descriptions 的数目
Sample		不同的媒体类型有不同的 sample description，但是每个 sample

description		description 的前四个字段是相同的，包含以下的数据成员
尺寸	4	这个 sample description 的字节数
数据格式	4	存储数据的格式。
保留	6	
数据引用索引	2	利用这个索引可以检索与当前 sample description 关联的数据。数据引用存储在 data reference atoms。

Audio track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
00170	00 00 00 0f 77 6d 69 8e 66 00 00 00 10 73 6d 68wainf...smh
00180	64 00 00 00 00 00 00 00 00 00 00 00 24 64 69 6e	d.....\$din
00190	66 00 00 00 1c 64 72 65 66 00 00 00 00 00 00 00	f.....dref.....
001a0	01 00 00 00 0c 75 72 6c 20 00 00 00 01 00 00 0furl.....
001b0	3b 73 74 62 6c 00 00 00 67 73 74 73 64 00 00 00	;stbl...gstsd...
001c0	00 00 00 00 01 00 00 00 57 6d 70 34 61 00 00 00mp4a...
001d0	00 00 00 00 01 00 00 00 00 00 00 00 00 00 02 00
001e0	10 00 00 00 00 1f 40 00 00 00 00 00 33 65 73 64E.....3esd
001f0	73 00 00 00 00 03 80 80 80 21 00 00 00 04 80 80	S.....@@@...@
00200	80 14 40 15 00 16 00 00 00 4e 20 00 00 4e 20 05	E..E.....N..N
00210	80 80 80 02 15 80 06 80 80 80 01 02 00 00 00 18	@@@...@@@...@
00220	73 74 74 73 00 00 00 00 00 00 00 01 00 00 02 23	stts.....#
00230	00 00 04 00 00 00 01 9c 73 74 73 63 00 00 00 00stsc...
00240	00 00 00 21 00 00 00 01 00 00 00 02 00 00 00 01	...!.....
00250	00 00 00 16 00 00 00 01 00 00 00 01 00 00 00 17

可以看出这个 sample 只有一个 description，对应得的数据格式是'mp4a'，14496-12 定义了这种结构，mp4 解码器会识别此 description。XXX

Video track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
01240	00 00 00 00 01 00 00 00 75 72 6c 20 00 00 00 00url...
01250	01 00 00 1c 23 73 74 62 6c 00 00 00 ab 73 74 73	...stbl...stc
01260	64 00 00 00 00 00 00 00 01 00 00 00 9b 6d 70 34	d.....mp4
01270	76 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00	v.....
01280	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00H.
01290	00 00 48 00 00 00 00 00 00 00 01 00 00 00 00 00	..H.....
012a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
012b0	00 00 00 00 00 00 00 00 00 00 00 00 18 ff ff 00
012c0	00 00 45 65 73 64 73 00 00 00 00 03 37 00 00 1f	..Eads.....7...
012d0	04 2f 20 11 00 6d 6d 00 01 24 f8 00 01 24 f8 05	/..m...\$...\$..
012e0	20 00 00 01 b0 f0 00 00 01 b5 0e a0 40 c0 cf 00@..
012f0	00 01 00 00 00 01 20 00 84 40 fa 28 30 20 f2 a2@ (0
01300	1f 06 01 03 00 00 00 18 73 74 74 73 00 00 00 00	...@.....stts...
01310	00 00 00 01 00 00 04 1a 00 00 00 28 00 00 00 9e(...
01320	73 74 73 73 00 00 00 00 00 00 00 23 00 00 00 01	stss.....#....
01330	00 00 00 1f 00 00 00 3d 00 00 00 5b 00 00 00 79"....[...y
01340	00 00 00 97 00 00 00 b5 00 00 00 d3 00 00 00 f1
01350	00 00 01 0f 00 00 01 2d 00 00 01 4b 00 00 01 69-...K...i
01360	00 00 01 87 00 00 01 a5 00 00 01 c3 00 00 01 e1

可以看出这个 sample 只有一个 description，对应得的数据格式是'mp4v'，14496-12 定义了这种结构，mp4 解码器会识别此 description。

Time-to-Sample Atoms - STTS

Time-to-sample atoms 存储了 media sample 的 duration 信息，提供了时间对具体 data sample 的映射方法，通过这个 atom，你可以找到任何时间的 sample，类型是'stts'。这个 atom 可以包含一个压缩的表来映射时间和 sample 序号，用其他的表来提供每个 sample 的长度和指针。表中每个条目提供了在同一个时间偏移量里面连续的 sample 序号，以

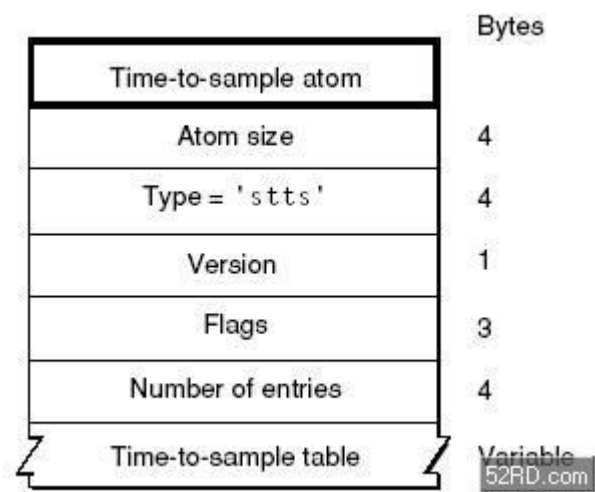
及 samples 的偏移量。递增这些偏移量，就可以建立一个完整的 time-to-sample 表，计算公式如下

$DT(n+1) = DT(n) + STTS(n)$

其中 STTS(n)是没有压缩的 STTS 第 n 项信息，DT 是第 n 个 sample 的显示时间。Sample 的排列是按照时间戳的顺序，这样偏移量永远是非负的。DT 一般以 0 开始，如果不为 0，edit list atom 设定初始的 DT 值。DT 计算公式如下

$DT(i) = \text{SUM (for } j=0 \text{ to } i-1 \text{ of } \delta(j))$

所有偏移量的和就是 track 中 media 的长度，这个长度不包括 media 的 time scale，也不包括任何 edit list。



字段	长度 (字节)	描述	
尺寸	4	这个 atom 的字节数	
类型	4	stts	
版本	1	这个 atom 的版本	
标志		4	time-to-sample 的数目
time-to-sample		Media 中每个 sample 的 duration。包含如下结构	
Sample count	4	有相同 duration 的连续 sample 的数目	
Sample duration	4	每个 sample 的 duration	

如果多个 sample 有相同的 duration，可以只用一项描述所有这些 samples，数量字段说明 sample 的个数。例如，如果一个视频媒体的帧率保持不变，整个表可以只有一项，数量就是全部的帧数。

Audio track 的值

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
00170	00	00	00	0f	77	6d	89	6e	66	00	00	00	10	73	6d	68	...uninf...smh
00180	64	00	00	00	00	00	00	00	00	00	00	00	24	64	69	6a	d.....\$din
00190	66	00	00	00	1e	64	72	65	66	00	00	00	00	00	00	00	f....dref.....
001a0	01	00	00	00	0c	75	72	6c	20	00	00	00	01	00	00	0furl.....
001b0	3b	73	74	62	6e	00	00	00	67	73	74	73	64	09	00	00	:stbl...gstsd...
001c0	00	00	00	00	01	00	00	00	57	6d	70	34	61	00	00	00Unp4a...
001d0	00	00	00	00	01	00	00	00	00	00	00	00	00	00	02	00
001e0	10	00	00	00	00	1f	40	00	00	00	00	00	33	65	73	640.....3esd
001f0	73	00	00	00	00	03	80	80	80	22	00	00	00	04	80	80	s.....eSe"....eSe
00200	80	14	40	15	00	18	00	00	00	4e	20	00	00	4e	20	05	e.0.....N...N...
00210	80	80	80	02	15	30	06	80	80	80	01	02	00	00	00	18	eSe...eSe.....
00220	73	74	74	73	00	00	00	00	00	00	00	01	00	00	02	23	stts.....#
00230	00	00	04	00	00	00	01	9e	73	74	73	63	00	00	00	00stsc....
00240	00	00	00	21	00	00	00	01	00	00	00	02	00	00	00	01!
00250	00	00	00	16	00	00	00	01	00	00	00	01	00	00	00	17
00260	00	00	00	02	00	00	00	01	00	00	00	2c	00	00	00	01
00270	00	00	00	01	00	00	00	2d	00	00	00	02	00	00	00	01=.....

可以看出这个 mp4a 类型的音频 track 只有一项 time-to-sample，一共有 547 个 sample，每个 sample 的 duration 是 1024。

Video track 的值

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	0123456789abcdef
012e0	20	00	00	01	b0	10	00	00	01	b5	0e	e0	40	c0	ef	000..
012f0	00	01	00	00	00	01	20	00	94	40	fa	28	30	20	82	a20(0
01300	1f	06	01	02	00	00	00	18	73	74	74	73	00	00	00	00stts...
01310	00	00	00	01	00	00	04	1a	00	00	00	26	00	00	00	9c[...
01320	73	74	73	73	00	00	00	00	00	00	00	23	00	00	00	01	stts.....#....
01330	00	00	00	1f	00	00	00	3d	00	00	00	5b	00	00	00	79[...Y
01340	00	00	00	97	00	00	00	b5	00	00	00	d3	00	00	00	11
01350	00	00	01	0f	00	00	01	2d	00	00	01	4b	00	00	01	69-...K...i
01360	00	00	01	97	00	00	01	a5	00	00	01	c3	00	00	01	e1
01370	00	00	01	1f	00	00	02	1d	00	00	02	3b	00	00	02	89Y
01380	00	00	02	77	00	00	02	95	00	00	02	b3	00	00	02	d1	...W... ..

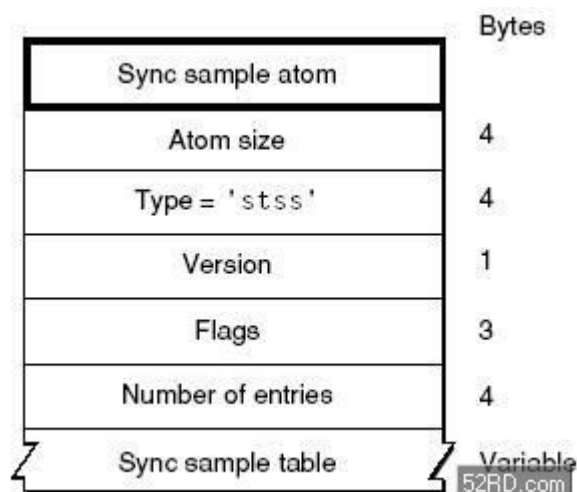
可以看出这个 mp4v 类型的 video track 只有一项 time-to-sample，一共有 1050 个 sample，每个 sample 的 duration 是 40。

Sync Sample Atoms - STSS

sync sample atom 确定 media 中的关键帧。对于压缩的媒体，关键帧是一系列压缩序列的开始帧，它的解压缩是不依赖于以前的帧。后续帧的解压缩依赖于这个关键帧。

sync sample atom 可以非常紧凑的标记媒体内的随机存取点。它包含一个 sample 序号表，表内的每一项严格按照 sample 的序号排列，说明了媒体中的哪一个 sample 是关键帧。如果此表不存在，说明每一个 sample 都是一个关键帧，是一个随机存取点。

Sync sample atoms 的类型是'stss'。



字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	stss
版本	1	这个 atom 的版本
标志	3	这里为 0
条目数目	4	sync sample 的数目
sync sample		sync sample 表的结构
Sample 序号	4	是关键帧的 sample 序号

Video track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
01300	1f 06 01 02 00 00 00 18 73 74 74 73 00 00 00 00stss....
01310	00 00 00 01 00 00 04 1a 00 00 00 28 00 00 00 9c{....
01320	73 74 73 73 00 00 00 00 00 00 00 23 00 00 00 01	stss.....#....
01330	00 00 00 1f 00 00 00 3d 00 00 00 5b 00 00 00 79=...[...]
01340	00 00 00 97 00 00 00 b5 00 00 00 d3 00 00 00 f1<....
01350	00 00 01 0f 00 00 01 2d 00 00 01 4b 00 00 01 69-...K...]
01360	00 00 01 87 00 00 01 a5 00 00 01 c3 00 00 01 e1<....
01370	00 00 01 ff 00 00 01 1d 00 00 02 3b 00 00 02 59f...Y
01380	00 00 02 77 00 00 02 95 00 00 02 b3 00 00 02 d1W...<....
01390	00 00 02 ef 00 00 03 0d 00 00 03 2b 00 00 03 49<...+...I
013a0	00 00 03 67 00 00 03 85 00 00 03 a3 00 00 03 c1g...<....
013b0	00 00 03 df 00 00 03 fd 00 00 06 40 73 74 73 63<...stsc
013c0	00 00 00 00 00 00 00 8c 00 00 00 01 00 00 00 04<....
013d0	00 00 00 01 00 00 00 04 00 00 00 03 00 00 00 01<....
013e0	00 00 00 05 00 00 00 04 00 00 00 01 00 00 00 08<....

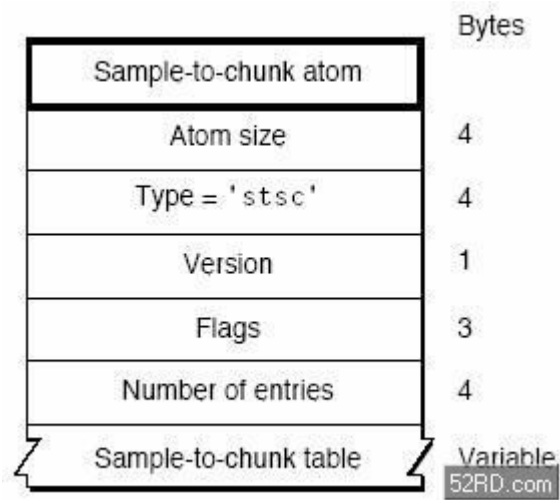
52RD.com

可以看出这个 video 片断共有 35 个关键帧。

Sample-to-Chunk Atoms - STSC

当添加 samples 到 media 时，用 chunks 组织这些 sample，这样可以方便优化数据获取。一个 trunk 包含一个或多个 sample，chunk 的长度可以不同，chunk 内的 sample 的长度也可以不同。sample-to-chunk atom 存储 sample 与 chunk 的映射关系。

Sample-to-chunk atoms 的类型是'stsc'。它也有一个表来映射 sample 和 trunk 之间的关系，查看这张表，就可以找到包含指定 sample 的 trunk，从而找到这个 sample。



字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	stsc
版本	1	这个 atom 的版本
标志	3	这里为 0
条目数目	4	sample-to-chunk 的数目
sample-to-chunk		sample-to-chunk 表的结构
First chunk	4	这个 table 使用的第一个 chunk 序号
Samples per chunk	4	当前 trunk 内的 sample 数目
Sample description ID	4	与这些 sample 关联的 sample description 的序号

Audio track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
00230	00 00 04 00 00 00 01 9c 73 74 73 63 00 00 00 00	... stsc...
00240	00 00 00 11 00 00 00 01 00 00 00 02 00 00 00 01
00250	00 00 00 16 00 00 00 01 00 00 00 01 00 00 00 17
00260	00 00 00 02 00 00 00 01 00 00 00 2c 00 00 00 03
00270	00 00 00 01 00 00 00 3d 00 00 00 02 00 00 00 03
00280	00 00 00 40 00 00 00 01 00 00 00 01 00 00 00 43
00290	00 00 00 02 00 00 00 01 00 00 00 56 00 00 00 03V....
002a0	00 00 00 01 00 00 00 57 00 00 00 02 00 00 00 03W....
002b0	00 00 00 6c 00 00 00 01 00 00 00 01 00 00 00 6d	...l.....m
002c0	00 00 00 02 00 00 00 01 00 00 00 80 00 00 00 03E....
002d0	00 00 00 01 00 00 00 81 00 00 00 02 00 00 00 03
002e0	00 00 00 96 00 00 00 01 00 00 00 01 00 00 00 96
002f0	00 00 00 02 00 00 00 01 00 00 00 99 00 00 00 03
00300	00 00 00 01 00 00 00 9a 00 00 00 02 00 00 00 03
00310	00 00 00 aa 00 00 00 01 00 00 00 01 00 00 00 ae
00320	00 00 00 02 00 00 00 01 00 00 00 c1 00 00 00 03
00330	00 00 00 01 00 00 00 c2 00 00 00 02 00 00 00 03
00340	00 00 00 d7 00 00 00 01 00 00 00 01 00 00 00 d8
00350	00 00 00 02 00 00 00 01 00 00 00 e8 00 00 00 03
00360	00 00 00 01 00 00 00 ee 00 00 00 02 00 00 00 03
00370	00 00 00 fc 00 00 00 01 00 00 00 01 00 00 00 fd
00380	00 00 00 02 00 00 00 01 00 00 00 fe 00 00 00 03
00390	00 00 00 01 00 00 00 ff 00 00 00 02 00 00 00 03
003a0	00 00 01 02 00 00 00 01 00 00 00 01 00 00 01 03
003b0	00 00 00 02 00 00 00 01 00 00 01 19 00 00 00 03
003c0	00 00 00 01 00 00 01 19 00 00 00 02 00 00 00 03
003d0	00 00 08 a0 73 74 73 7a 00 00 00 00 00 00 00 00	... 0000...
003e0	00 00 02 23 00 00 00 07 00 00 00 07 00 00 01 3e	...#.....<
003f0	00 00 01 8e 00 00 01 8e 00 00 01 1e 00 00 01 17

可以建立这个 sample-to-chunk 表，共有 33 项。

Video track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0123456789abcdef
013b0	00 00 03 df 00 00 03 fd 00 00 06 a0 73 74 73 63	... stsc
013c0	00 00 00 00 00 00 00 8c 00 00 00 01 00 00 00 04
013d0	00 00 00 01 00 00 00 04 00 00 00 03 00 00 00 01
013e0	00 00 00 05 00 00 00 04 00 00 00 01 00 00 00 08
013f0	00 00 00 03 00 00 00 01 00 00 00 09 00 00 00 04
01400	00 00 00 01 00 00 00 0c 00 00 00 03 00 00 00 01
01410	00 00 00 0d 00 00 00 04 00 00 00 01 00 00 00 10
01420	00 00 00 03 00 00 00 01 00 00 00 11 00 00 00 04
01430	00 00 00 01 00 00 00 14 00 00 00 03 00 00 00 01
01440	00 00 00 15 00 00 00 04 00 00 00 01 00 00 00 18
01450	00 00 00 03 00 00 00 01 00 00 00 19 00 00 00 04
01460	00 00 00 01 00 00 00 1c 00 00 00 03 00 00 00 01
01470	00 00 00 1d 00 00 00 04 00 00 00 01 00 00 00 20
01480	00 00 00 03 00 00 00 01 00 00 00 21 00 00 00 04!
01490	00 00 00 01 00 00 00 34 00 00 00 03 00 00 00 01\$.....
014a0	00 00 00 25 00 00 00 04 00 00 00 01 00 00 00 289.....(
014b0	00 00 00 03 00 00 00 01 00 00 00 29 00 00 00 04)....
014c0	00 00 00 01 00 00 00 2c 00 00 00 03 00 00 00 01
014d0	00 00 00 2d 00 00 00 04 00 00 00 01 00 00 00 30-.....0
014e0	00 00 00 03 00 00 00 01 00 00 00 31 00 00 00 041....
014f0	00 00 00 01 00 00 00 34 00 00 00 03 00 00 00 014.....
01500	00 00 00 35 00 00 00 04 00 00 00 01 00 00 00 36	...5.....6
01510	00 00 00 03 00 00 00 01 00 00 00 39 00 00 00 049.....
01520	00 00 00 01 00 00 00 3c 00 00 00 03 00 00 00 01

可以建立 Video track 的 sample-to-chunk 表，共有 140 项。

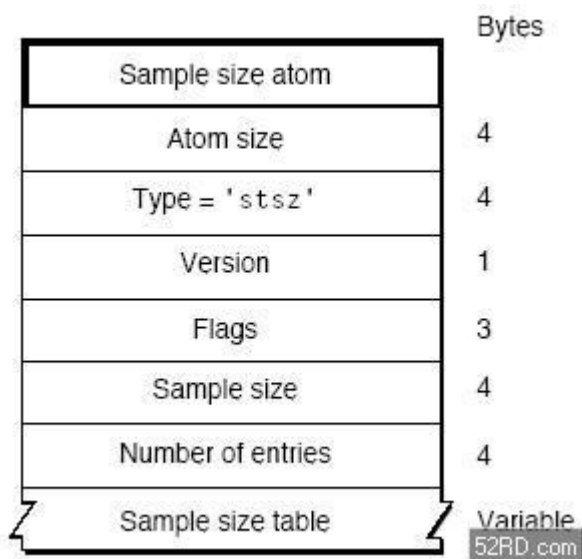
First chunk	Samples per chunk	Sample description ID
1	4	1
4	3	1
5	4	1
8	3	1

9	4	1
12	3	1
13	4	1
16	3	1
17	4	1
20	3	1
21	4	1
...
277	4	1
280	3	1

这个表类似于行程编码，第一个 first chunk 减去第二个 first chunk 就是一共有多少个 trunk 包含相同的 sample 数目，这样通过不断的叠加，就可以得到一共有 280 个 trunk，每个 trunk 包含多少个 sample，以及每个 trunk 对应的 description。

Sample Size Atoms - STSZ

sample size atoms 定义了每个 sample 的大小，它的类型是'stsz'，包含了媒体中全部 sample 的数目和一张给出每个 sample 大小的表。这样，媒体数据自身就可以没有边框的限制



字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	stsz
版本	1	这个 atom 的版本
标志	3	这里为 0
Sample size	4	全部 sample 的数目。如果所有的 sample 有相同的长度，这个字段就是这个值。否则，这个字段的值就是 0。那些长度存在 sample size 表中
条目数目	4	sample size 的数目

sample size		sample size 表的结构。这个表根据 sample number 索引，第一项就是第一个 sample，第二项就是第二个 sample
大小	4	每个 sample 的大小

Audio track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0 1 2 3 4 5 6 7 8 9 a b c d e f
003d0	00 00 08 a0 73 74 73 7a 00 00 00 00 00 00 00 00	...stst.....
003e0	00 00 02 28 00 00 00 07 00 00 00 07 00 00 01 3c	...#.....<
003f0	00 00 01 9e 00 00 01 9e 00 00 01 3e 00 00 01 17
00400	00 00 01 16 00 00 00 eb 00 00 01 18 00 00 01 07
00410	00 00 01 14 00 00 00 eb 00 00 01 28 00 00 00 1c(...
00420	00 00 01 1f 00 00 01 20 00 00 01 0f 00 00 01 16
00430	00 00 01 17 00 00 01 29 00 00 01 04 00 00 00 ee].....
00440	00 00 01 35 00 00 01 35 00 00 01 3d 00 00 01 39	...S...S...#...9
00450	00 00 01 2e 00 00 01 40 00 00 01 39 00 00 01 400...0...0
00460	00 00 01 3b 00 00 01 37 00 00 01 23 00 00 01 39	...j...7...#...9
00470	00 00 01 3e 00 00 01 38 00 00 01 63 00 00 01 6d	...>...8...c...h
00480	00 00 01 5d 00 00 01 28 00 00 01 61 00 00 01 98	...]...[...a...
00490	00 00 01 88 00 00 01 1d 00 00 01 2f 00 00 01 1e/...
004a0	00 00 01 2e 00 00 01 21 00 00 01 36 00 00 01 3b!...6...7
004b0	00 00 01 39 00 00 01 34 00 00 01 36 00 00 01 28	...9...4...6...{
004c0	00 00 01 34 00 00 01 4e 00 00 01 4d 00 00 01 49	...4...L...H...I
004d0	00 00 01 4e 00 00 01 3b 00 00 01 39 00 00 01 51	...L...z...9...c
004e0	00 00 01 46 00 00 01 4c 00 00 01 45 00 00 01 58	...F...L...E...X
004f0	00 00 01 4e 00 00 01 34 00 00 01 4d 00 00 01 1c	...N...4...H...
00500	00 00 01 43 00 00 01 43 00 00 01 36 00 00 01 4a	...C...C...6...J

这个表是最大的一个表，可以看到这个 audio track 的 sample 的长度都不一样，一共有 547 项。

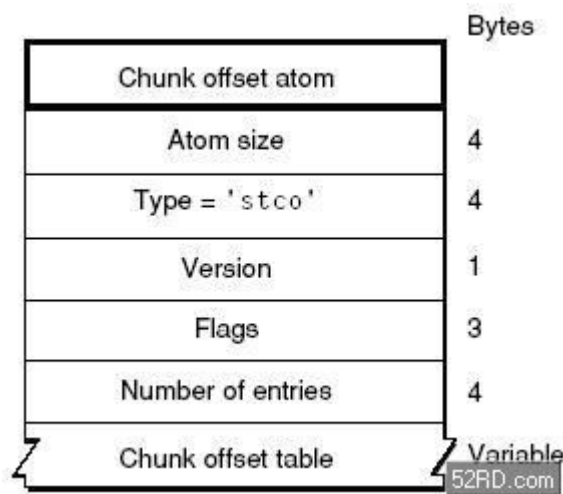
Video track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0 1 2 3 4 5 6 7 8 9 a b c d e f
01a50	00 00 00 03 00 00 00 01 00 00 10 7e 73 74 73 7a stst
01a60	00 00 00 00 00 00 00 00 00 00 04 1a 00 00 02 f4
01a70	00 00 01 68 00 00 01 2d 00 00 01 21 00 00 01 90	...h...-...!
01a80	00 00 00 ce 00 00 00 ed 00 00 00 d0 00 00 00 ab
01a90	00 00 01 96 00 00 01 8e 00 00 01 cc 00 00 01 55t
01aa0	00 00 01 54 00 00 01 b4 00 00 01 72 00 00 01 60	...T...e...^
01ab0	00 00 01 b6 00 00 01 d9 00 00 02 39 00 00 02 3c9...<
01ac0	00 00 02 ef 00 00 02 59 00 00 02 61 00 00 02 09	...Y...a...
01ad0	00 00 02 78 00 00 02 7f 00 00 02 5e 00 00 02 5d	...X...^...]
01ae0	00 00 02 7b 00 00 04 5f 00 00 03 32 00 00 02 e9	...f...2...
01af0	00 00 02 88 00 00 02 e7 00 00 02 82 00 00 02 f8
01b00	00 00 03 78 00 00 04 0a 00 00 04 59 00 00 04 8c	...x...Y...
01b10	00 00 04 6f 00 00 04 65 00 00 03 e6 00 00 04 1e	...o...e...
01b20	00 00 04 0d 00 00 03 9c 00 00 03 c5 00 00 03 aa
01b30	00 00 03 98 00 00 03 5b 00 00 03 18 00 00 02 c2[.....
01b40	00 00 02 99 00 00 02 4f 00 00 02 2e 00 00 02 0f	...o...z...
01b50	00 00 01 d6 00 00 01 ce 00 00 01 a0 00 00 07 93
01b60	00 00 03 d9 00 00 03 4f 00 00 03 0c 00 00 02 e0	...o.....
01b70	00 00 02 dc 00 00 02 ef 00 00 02 93 00 00 02 4e	...K...N...
01b80	00 00 02 4b 00 00 01 f1 00 00 01 e3 00 00 01 c9	...K... ..
01b90	00 00 01 9e 00 00 01 60 00 00 01 97 00 00 01 e8
01ba0	00 00 01 99 00 00 01 d9 00 00 01 c2 00 00 01 65

这个表包含了每个 sample 的长度，找到 sample 的序号，就可以找到对应 sample 的长度了。可以看到 video track 共有 1050 个 sample。

Chunk Offset Atoms - STCO

Chunk offset atoms 定义了每个 trunk 在媒体流中的位置，它的类型是'stco'。位置有两种可能，32 位的和 64 位的，后者对非常大的电影很有用。在一个表中只会有一种可能，这个位置是在整个文件中的，而不是在任何 atom 中的，这样做就可以直接在文件中找到媒体数据，而不用解释 atom。需要注意的是一旦前面的 atom 有了任何改变，这张表都要重新建立，因为位置信息已经改变了。



字段	长度(字节)	描述
尺寸	4	这个 atom 的字节数
类型	4	stco
版本	1	这个 atom 的版本
标志	3	这里为 0
条目数目	4	chunk offset 的数目
chunk offset		字节偏移量从文件开始到当前 chunk。这个表根据 chunk number 索引，第一项就是第一个 trunk，第二项就是第二个 trunk
大小	4	每个 sample 的大小

Audio track 的值

	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f	0 1 2 3 4 5 6 7 8 9 a b c d e f
00e70	00 00 04 78 73 74 63 62 00 00 00 00 00 00 01 1a	...stco.....
00e80	00 00 2f 6c 00 00 2f 7a 00 00 3d 09 00 00 3f b5	.../1.../a...?...
00e90	00 00 4b da 00 00 4d da 00 00 5f 16 00 00 63 09	...K...H... ..b...
00ea0	00 00 74 bc 00 00 76 2b 00 00 92 23 00 00 94 50	...e...v... ..f...P...
00eb0	00 00 b4 3a 00 00 b6 ac 00 00 d4 f8 00 00 d7 71	...f...e...
00ec0	00 00 e8 53 00 00 ea ad 00 01 09 f9 00 01 0c 94	...S...
00ed0	00 01 1c 91 00 01 1f 1a 00 01 2d 3e 00 01 2f e3-?.../...
00ee0	00 01 3a 9f 00 01 3e ee 00 01 4c 1e 00 01 4e 6b	...t...<...l...N...
00ef0	00 01 5e be 00 01 61 3e 00 01 7b de 00 01 7e 68	...^...a>... ..-e...

小结

其实 mp4 的文件格式就是在 quicktime 的 mov 文件格式上演变出来的。原文可以参见
<http://developer.apple.com/documentation/QuickTime/QTFF/qtff.pdf>

我只是把最重要的一些 atom 取出来，翻译了一部分，加上一部分自己的心得体会。还有一部分关于 streaming 的 hint atom 的格式，过两天再传上来。

获取 mp4 文件信息 1 - 计算电影长度

方法 1

从 mvhd - movie header atom 中找到 time scale 和 duration，duration 除以 time scale 即是整部电影的长度。

time scale 相当于定义了标准的 1 秒在这部电影里面的刻度是多少。

例如 audio track 的 time scale = 8000, duration = 560128，所以总长度是 70.016，video track 的 time scale = 600, duration = 42000，所以总长度是 70

方法 2

首先计算出共有多少个帧，也就是 sample（从 sample size atoms 中得到），然后整部电影的 duration = 每个帧的 duration 之和（从 Time-to-sample atoms 中得出）

例如 audio track 共有 547 个 sample，每个 sample 的长度是 1024，则总 duration 是 560128，电影长度是 70.016；video track 共有 1050 个 sample，每个 sample 的长度是 40，则总 duration 是 42000，电影长度是 70

获取 mp4 文件信息 2 - 计算电影图像宽

度和高度

从 tkhd - track header atom 中找到宽度和高度即是。

获取 mp4 文件信息 3 - 计算电影声音采样频率

从 tkhd - track header atom 中找出 audio track 的 time scale 即是声音的采样频率。

获取 mp4 文件信息 4 - 计算视频帧率

首先计算出整部电影的 duration，和帧的数目然后
帧率 = 整部电影的 duration / 帧的数目

获取 mp4 文件信息 5 - 计算电影的比特率

整部电影的尺寸除以长度，即是比特率，此电影的比特率为 $846623/70 = 12094$ bps

获取 mp4 文件信息 6 - 查找 sample

当播放一部电影或者一个 track 的时候，对应的 media handler 必须能够正确的解析数据流，对一定的时间获取对应的媒体数据。如果是视频媒体，media handler 可能会解析多个 atom，才能找到给定时间的 sample 的大小和位置。具体步骤如下：

1. 确定时间，相对于媒体时间坐标系统
2. 检查 time-to-sample atom 来确定给定时间的 sample 序号。
3. 检查 sample-to-chunk atom 来发现对应该 sample 的 chunk。
4. 从 chunk offset atom 中提取该 trunk 的偏移量。
5. 利用 sample size atom 找到 sample 在 trunk 内的偏移量和 sample 的大小。

例如，如果要找第 1 秒的视频数据，过程如下：

1. 第 1 秒的视频数据相对于此电影的时间为 600
2. 检查 time-to-sample atom，得出每个 sample 的 duration 是 40，从而得出需要寻找第 $600/40 = 15 + 1 = 16$ 个 sample
3. 检查 sample-to-chunk atom，得到该 sample 属于第 5 个 chunk 的第一个 sample，该 chunk 共有 4 个 sample
4. 检查 chunk offset atom 找到第 5 个 trunk 的偏移量是 20472
5. 由于第 16 个 sample 是第 5 个 trunk 的第一个 sample，所以不用检查 sample size atom，trunk 的偏移量即是该 sample 的偏移量 20472。如果是这个 trunk 的第二个 sample，则从 sample size atom 中找到该 trunk 的前一个 sample 的大小，然后加上偏移量即可得到实际位置。
6. 得到位置后，即可取出相应数据进行解码，播放

获取 mp4 文件信息 7 - 查找关键帧

查找过程与查找 sample 的过程非常类似，只是需要利用 sync sample atom 来确定 key frame 的 sample 序号

1. 确定给定时间的 sample 序号
2. 检查 sync sample atom 来发现这个 sample 序号之后的 key frame
3. 检查 sample-to-chunk atom 来发现对应该 sample 的 chunk
4. 从 chunk offset atom 中提取该 trunk 的偏移量
5. 利用 sample size atom 找到 sample 在 trunk 内的偏移量和 sample 的大小

获取 mp4 文件信息 8 - Random acce

SS

Seeking 主要是利用 sample table box 里面包含的子 box 来实现的，还需要考虑 edit list 的影响。可以按照以下步骤 seek 某一个 track 到某个时间 T，注意这个 T 是以 movie header box 里定义的 time scale 为单位的：

1. 如果 track 有一个 edit list，遍历所有的 edit，找到 T 落在哪个 edit 里面。将 Edit 的开始时间变换为以 movie time scale 为单位，得到 EST，T 减去 EST，得到 T'，就是在这个 edit 里面的 duration，注意此时 T' 是以 movie 的 time scale 为单位的。然后将 T' 转化成 track 媒体的 time scale，得到 T''。T'' 与 Edit 的开始时间相加得到以 track 媒体的 time scale 为单位的时间点 T'''。

2. 这个 track 的 time-to-sample 表说明了该 track 中每个 sample 对应的的时间信息，利用这个表就可以得到 T 对应的 sample N_T 。
3. sample N_T 可能不是一个 random access point，这样就需要其他表的帮助来找到最近的 random access point。一个表是 sync sample 表，定义哪些 sample 是 random access point。使用这个表就可以找到指定时间点最近的 sync sample。如果没有这个表，就说明所有的 sample 都是 synchronization points，问题就变得更容易了。另一个 shadow sync box 可以帮助内容作者定义一些特殊的 samples，它们不用在网络中传输，但是可以作为额外的 random access point。这就改进了 random access，同时不会影响正常的传输比特率。这个表指出了非 random access point 和 random access point 之间的关系。如果要寻找指定 sample 之前最近的 shadow sync sample，就需要查询这个表。总之，利用 sync sample 和 shadow sync 表，就可以 seek 到 N_T 之前的最近的 access point sample N_{apo} 。
4. 找到用于 access point 的 sample N_{ap} 之后，利用 sample-to-chunk 表来确定 sample 位于哪个 chunk 内。
5. 找到 chunk 后，使用 chunk offset 找到这个 chunk 的开始位置。
6. 使用 sample-to-chunk 表和 sample size 表中的数据，找到 N_{ap} 在此 chunk 内的位置，再加上此 chunk 的开始位置，就找到了 N_{ap} 在文件中的位置。