

1 Introduction

During writing software to change dates in MP4/MOV files I had a lot of moments that I thought: how is this data stored? There are a lot of special cases that have probably grown during the development and merging with other standards. This document (for now) focuses on the atoms/boxes where metadata is stored. The idea behind this document to make things more clear and explain some of the knotty terminology with things often meaning the same (or not).

About this *webpage design*: there is none. I am currently updating this information in an HTML file that I edit MS Word, with the main aim to also make it printable on A4/letter format. There are Excel tables linked in this document that I want to 'keep on the page'. I realise that it looks sub-optimal in a browser... Please resize the browser to a small width, or you can view the pdf version [index.pdf](#).

MP4 or Quicktime? What name to use: The MP4 format is based on the Quicktime format[1] but from here I will call it MP4 format since this has become more general.

2 Dates and data in MP4 files

This whole exercise started with the annoyance of having MP4 files wrongly dated because of wrong time-setting on the recording device and also by noticing strange sorting behavior during sorting in Google Photos and other applications. Simply, the first distinction is between the operating system file attributes (time created, modified, last opened) and the data stored within the MP4 file.

3 Atoms that contain metadata

3.1 Metadata atom

The 'meta' atom, full name 'metadata atom' is one container where metadata is stored in an MP4/MOV file. It can be either a so-called 'full atom' with version and flag bytes added, or a non-full atom without the latter two. The size of the atom is variable and depends on all the data inside of it (child atoms).

Metadata atom			
Variable	# Bytes	Type	Values
Size	4	uint32	<i>variable</i>
Type	4	uint32	'meta'
Version	1		1
Flags	3	3 bytes	

Position in file. The path of metadata atom can be either [\moov\meta](#) or [\moov\udta\meta](#), or both. No more than one atom is allowed in each path. The metadata atom can also be present in [:\[trak ?\]](#).

The meta atom contains many childatoms and data stored in keys. I will not describe the structure of each separate atom inside as this has been often described elsewhere, e.g. (1) and references therein. The idea of this document is to show the structure of the atoms using data from real-life examples in illustrations such as in Table 1. In the top-line the hexadecimal byte values are shown and below descriptive information.

Table 1: Example of metadata atom that has no 'keys' section. (file had no metadata before, metadata inserted by Windows property editor). Note that metadata atom is a **full-atom** here. File:[2]

Metadata atom	00 00 size CO	6D 65 74 type meta	61 00 v	00 00 flags	00 00 00 00																		
	Metadata handler atom	00 00 size	21 68 64 type hdr	6C 72 00 v	00 00 flags	00 00 00 00 predefined=0	6D 64 69 72 handler type mdir	00 00 00 00 Reserved=0	00 00 00 00 Reserved=0	00 00 00 00 Reserved=0	00 00 00 00 Reserved=0	00 Name											
	Metadata Item List Atom	00 00 size	93 69 6C type ilst	73 74																			
	Metadata Item Atom	00 00 size	2E A9 77 type @wrt	72 74																			
		Value atom	00 00 size	26 64 type data	61 74 61 DF	00 00 00 01 known types	00 00 00 00 locale indicator country language	57 6E 20 63 6F 6D 70 6F 73 65 72 ... type=1: string Wn composer1/c...															
	Metadata Item Atom	00 00 size	19 74 6D type tmpo	70 6F																			
		Value atom	00 00 size	11 64 type data	61 74 61 DF	00 00 00 15 known types	00 00 00 00 locale indicator country language	00 type=\$15=21: big-endian signed int in 1,2,3 or 4 bytes															
	Metadata Item Atom	00 00 size	20 A9 6E type @nam	61 6D																			
		Value atom	00 00 size	18 64 type data	61 74 61 DF	00 00 00 01 known types	00 00 00 00 locale indicator country language	57 69 6E 54 69 74 6C 65 type=1: string "WinTitle"															
	Metadata Item Atom	00 00 size	24 A9 63 type @cmt	6D 74																			
		Value atom	00 00 size	1C 64 type data	61 74 61 DF	00 00 00 01 known types	00 00 00 00 locale indicator country language	57 69 6E 20 43 6F 6D 6D 65 6E 74 73 type=1: string "Win Comments"															

Table 2: Example of metadata atom that has a ‘keys’ section. Note that metadata atom is a **non-full-atom** here. File: [3]

Metadata atom	00 00 xx xx	6D 65 74 type meta 61																																							
	Metadata handler atom	00 00 size	22 68 64 type hdr 6C 72	00 v	00 00 flags	predefined=0				6D 64 74 61 handler type mdtA	Reserved=0																00 00 Name														
	Metadata Item Keys Atom	00 00 size	93 68 65 type keys 79 73	00 v	00 00 flags	entry_count																																			
		Key value 1	00 00 key_size	28 6D 64 74 61 key_namespace mdtA	key_name (Apple calls it key_value)																																				
		Key value 2	00 00 key_size	21 64 61 74 61 key_namespace mdtA	com.apple.quicktime.creationdate																																				
		Key value 3	00 00 key_size	24 A9 6E 61 6D key_namespace mdtA	com.apple.quicktime.model																																				
					key_name (Apple calls it key_value)																																				
	Metadata Item List Atom	00 00 size	69 6C 73 type ilst 74	key_name (Apple calls it key_value)																																					
	Metadata Item Atom	00 00 size	30 00 type =key 1 01																																						
		Value atom	00 00 size	28 64 type data 61	type indicator				locale indicator				type=1: string																												
					DF				known types				country				language				2014-07-05T13:02:04+0200																				
	Metadata Item Atom	00 00 size	21 00 type =key 2 02																																						
		Value atom	00 00 size	19 64 type data 61	type indicator				locale indicator				type=1: string																												
					DF				known types				country				language				iPhone 5s																				
	Metadata Item Atom	00 00 size	32 00 type =key 3 03																																						
		Value atom	00 00 size	2A 64 type data 61	type indicator				locale indicator				type=1: string																												
					DF				known types				country				language				+43.6521+003.3638+148.202/																				
	Free Atom	00 00 size	04 00 66 72 type free 65 65																																						

3.2 User data atom

A User Data Atom whose immediate parent is a movie atom contains data relevant to the movie as a whole. A User Data Atom whose parent is a track atom contains information relevant to that specific track. A QuickTime movie file may contain many user data atoms, but only one User Data Atom is allowed as the immediate child of any given movie atom or track atom.

A User Data Atom whose immediate parent is a movie atom contains data relevant to the movie as a whole. A User Data Atom whose parent is a track atom contains information relevant to that specific track. A QuickTime movie file may contain many user data atoms, but only one user data atom is allowed as the immediate child of any given movie atom or track atom (2).

The series of atoms inside a User Data Atom is also referred to as 'user data list'. Each data element in the user data list contains size and type information along with its data. An example is shown in Table 3.

Table 3: Example of user data atom that also contains a meta data atom. File: [4]

User data atom	00 00 01 CA 75 64 74 61																												
User data Item Atom	00 00 00 10 53 44 4C 4E	SEQ_PLAY																											
	00 00 00 10 73 6D 72 64	TRUEBLUE																											
	00 00 00 1E A9 78 79 7A	00 12 15 C7 2B 35 30 2E	39 36 37 38 2D 31 31 34 2E 30 36 39 30 2F																										
	00 00 00 1A 73 6D 74 61	00 00 00 00 00 00 00 00 0E 73 61 75 74 00 00 00 00 00 00																											
	00 00 00 97 6D 65 74 61	00 v 00 00 00 flags																											
Metadata atom	00 00 00 97 6D 65 74 61	00 v 00 00 00 flags																											
Metadata handler atom	00 00 00 20 68 64 6C 72	00 v 00 00 00 flags	predefined=0				handler type mdir				Reserved=0				Reserved=0				Reserved=0										
Metadata Item List Atom	00 00 00 6B 69 6C 73 74																												
Metadata Item Atom	00 00 00 35 A9 6E 61 6D																												
Value atom	00 00 00 2D 64 61 74 61	type indicator				locale indicator				type=1: string "comment should be here"																			
Metadata Item Atom	00 00 00 2E A9 63 6D 74																												
Value atom	00 00 00 26 64 61 74 61	type indicator				locale indicator				type=1: string "comment should be here"																			

A User Data Atom can also be the container of a Meta Data Atom as is shown in Table 3. Now, an important part that is *different* from the data inside the Meta Data Atom (that also can contain atomtypes starting with ©) is the following:

All user data list entries whose type begins with the © character (ASCII 169) are defined to be international text. These list entries must contain a list of text strings with associated language codes. By storing multiple versions of the same text, a single user data text item can contain translations for different languages. The list of text strings uses a small integer atom format, which is identical to the QuickTime atom format, except that it uses 16-bit values for size and type instead of 32-bit values. **The first value is the size of the string, including the size and type***, and the second value is the language code for the string.

What is meant is probably the following: In the 'Quicktime atom format' (Table 3c) the data is stored in a Value Atom with type 'data' including a 4-byte type indicator (1 for string) and a 4-byte locale indicator. When you reshuffle the bytes of the User Data Item Atom you can place the data in a 'pseudo' Value Atom like in (Table 3b).

***The first value is the size of the string, including the size and type:** type indicator is 1 by definition, type of atom is 'data' by definition.

Table 4: Difference between an Item Atom in User data and Meta data (see text)

a. atom rooted in \udta\

User data Item Atom	00 00 00 1E	A9 78 79 7A	00 12	15 C7	2B 35 30 2E 39 36 37 38 2D 31 31 34 2E 30 36 39 30 2F
	size	type ©xyz	size 18	langua. eng'	+50.9678-114.0690/

b. same atom rooted in \udta\, but data reorganised to match pattern of 'Quicktime atom format' shown under c.

User data Item Atom	00 00 00 1E	A9 78 79 7A			
	size	type ©xyz			
Pseudo Value atom	- - 00 12	- - - -	- - - -	- - 15 C7	2B 35 30 2E 39 36 37 38 2D 31
	size	type 'data' by defin.	type indicator 1 by defin.	locale indicator	type=1: string +50.9678-114.0690/

c. another atom rooted in \udta\meta\ilst\ which is referred to as 'Quicktime atom format'

Metadata Item Atom	00 00 00 35	A9 6E 61 6D			
	size	type ©nam			
Value atom	00 00 00 2D	64 61 74 61	00 00 00 01	00 00 00 00	41 6E 64 72 6F 69 64 20 36 2E
	size	type data	type indicator DF	locale indicator known types	country language "Android 6.0.1 tags by Windows"

User data text strings may use either Macintosh text encoding or Unicode text encoding. The format of the language code determines the text encoding format. Macintosh language codes are followed by Macintosh-encoded text. If the language code is specified using the ISO language codes listed in specification ISO 639-2/T, the text uses Unicode text encoding. When Unicode is used, the text is in UTF-8 unless it starts with a byte-order-mark (BOM, 0xFEFF), in which case the text is in UTF-16. Both the BOM and the UTF-16 text should be big-endian. Multiple versions of the same text may use different encoding schemes.

3.3 Microsoft Xtra atom

When an MP4 or MOV file is edited by Windows Properties in the 'Details' tab of (right mouse), an 'Xtra' atom is added or changed as \\moov\udta\Xtra. Besides this, several values are also added or changed in the \\moov\udta\meta atom.

Property	Value	Origin	Content
Description			
Title	MyTitle	Directors MyDirect	Parental rating MyParRating
Subtitle	MySubt	Producers MyProd	Parental rating reason
Rating	★ ★ ☆ ☆ ☆	Writers MyWriter	Composers MyComp
Tags	MyTag1; MyTag2	Publisher MyPubl	Conductors MyCond
Comments	MyComments	Content provider MyContProv	Period MyPeriod
Media		Media created 2012-07-11 07:16	Mood MyMood
Contributing artists	MyContrArtists	Encoded by MyEncBy	Part of set 4
Year	2020	Author URL MyAuthURL	Initial key MyInitKey
Genre	MyGenre	Promotion URL MyPromoURL	Beats-per-minute 0
		Copyright	Protected No

Figure 1: Properties displayed in the Windows property editor by Windows.

Table 5: Properties displayed in the Windows property editor and the atoms where they are stored.

Section	Property	Xtra key	moov/udta/meta key
Description	Title	-	©nam
	Subtitle	WM/SubTitle	-
	Rating	WM/SharedUserRating	-
	Tags	-	-
	Comments	-	©cmt
Media	Contributing artists	-	©ART
	Year	-	©day
	Genre	-	©gen
Origin	Directors	WM/Director	-
	Producers	WM/Producer	-
	Writers	WM/Writer	-
	Publisher	WM/Publisher	-
	Content provider	WM/ContentDistributor	-
	Media created	cannot be set	-

Content	Encoded by	WM/EncodedBy	-
	Author URL	WM/AuthorURL	-
	Promotion URL	WM/PromotionURL	-
	Copyright	<i>cannot be set</i>	-
	Parental rating	WM/ParentalRating	-
	Parental rating reason	<i>cannot be set</i>	-
	Composers	-	©wrt
	Conductors	WM/Conductor	-
	Period	WM/Period	-
	Mood	WM/Mood	-
	Part of set	-	disk
	Initial key	WM/InitialKey	-
	Beats-per-minute	-	tmpo
	Protected	<i>cannot be set</i>	-

Information is not easy to find. The data can be stored in various types that are indicated by a type enumeration (like well-know types in the meta atom).

I cannot find this information as you would expect it on the Microsoft website. Although many ‘keys’ that are used in the Xtra atom (e.g. WM/Composer) are described in the WMF Attribute list (3) including a type enumeration (WMT_ATTR_DATATYPE enumeration); part of this enumeration is as follows: WMT_TYPE_DWORD = 0, WMT_TYPE_STRING = 1, WMT_TYPE_BINARY = 2 etc. However, this is not the enumeration used in the Xtra atom.

The actual enumeration used I found in Exiftool source code (4) and in an Xtrabox Java script (5), and is shown in Table 6. As noted in (4), an implementation has existed in a branch of mp4v2 but has been removed. This is discussed in (6).

Table 6: Type enumeration of Xtra values

Const Name	Decimal	Hexadecimal
MP4_XTRA_BT_UNICODE	8	\$8
MP4_XTRA_BT_INT64	19	\$13
MP4_XTRA_BT_FILETIME	21	\$15
MP4_XTRA_BT_GUID	72	\$48

Table 7: Example of an Xtra atom

Xtra atom	00 00 03 14 58 74 72 61	size 788	type Xtra
Key 1	00 00 00 2D 00 00 00 13	key_size 45	key_name_len 13
	00 00 00 0E 00 13	val_type 19	key_name WM/SharedUserRating
Value 1	00 00 00 0E 00 13	value_size 19	value
Key 2	00 00 00 2B 00 00 00 0B	key_size 43	key_name_len 11
	00 00 00 14 00 08	val_type 8	key_name WM/SubTitle
Value 1	00 00 00 14 00 08	value_size 20	value
Key 3	00 00 00 3F 00 00 00 0B	key_size 63	key_name_len 11
	00 00 00 14 00 08	val_type 8	key_name WM/Category
Value 1	00 00 00 14 00 08	value_size 20	value
Value 2	00 00 00 14 00 08	value_size 20	value
Key 4	00 00 00 27 00 00 00 07	key_size 27	key_name_len 7
	00 00 00 14 00 08	val_type 8	key_name WM/Mood
Value 1	00 00 00 14 00 08	value_size 20	value

1 https://en.wikipedia.org/wiki/MPEG-4_Part_14#

2 MOV_0234-windowscomments.mp4

3 Apple-Iphone5s.mov

4 Android 6.0.1 - with tags added by Windows Explorer.mp4

1. **Wikipedia.** Wikipedia - QuickTime File Format. [Online] https://en.wikipedia.org/wiki/QuickTime_File_Format.

2. **Apple Computer, Inc.** *QuickTime File Format*. Cupertino, CA : s.n., 2002.

3. **Microsoft.** Windows Media Format 11 Attribute List. [Online] [Cited: 6 14, 2020.] <https://docs.microsoft.com/en-us/windows/win32/wmformat/attribute-list>.

4. **Harvey, Phil.** Perlscript "Microsoft.pm". [Online] [Cited: 6 14, 2020.] <https://github.com/exiftool/exiftool/blob/master/lib/Image/ExifTool/Microsoft.pm>.

5. "XtraBox.java" script. [Online] [Cited: 6 14, 2020.] <http://www.java2s.com/example/java-src/pkg/com/googlecode/mp4parser/boxes/microsoft/xtrabox-3706e.html>.

6. mp4v2 - issue #113. [Online] 8 5, 2011. [Cited: 6 14, 2020.] <https://code.google.com/archive/p/mp4v2/issues/113>.