[Live document: last update 20200619]

# 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 <u>index.pdf</u>.

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.

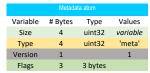
## 2 Dates and data in MP4 files

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).



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]



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



#### 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.

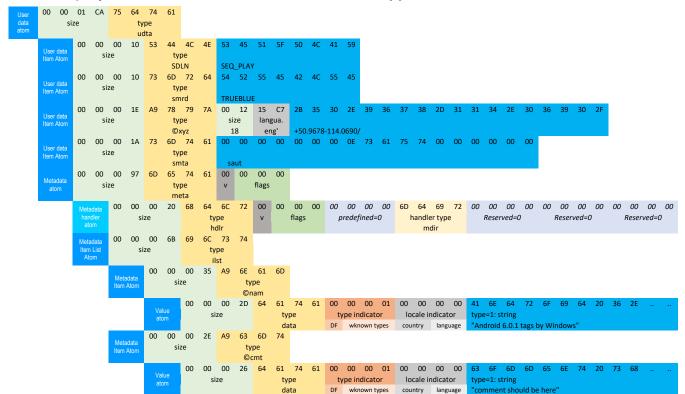


Table 3: Example of user data atom that also contains a meta data atom. File: [4]

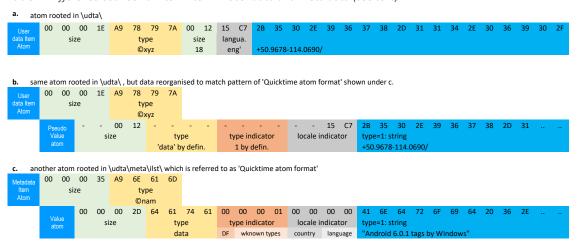
A User Data Atom can also be the container of a Meta Data Atom as is shown in Table 3. Now, an important part that is *different* from the data inside the Meta Data Atom (that also can contain atomtypes starting with ©) is the following:

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)



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-ordermark (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.



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	-	

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

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).

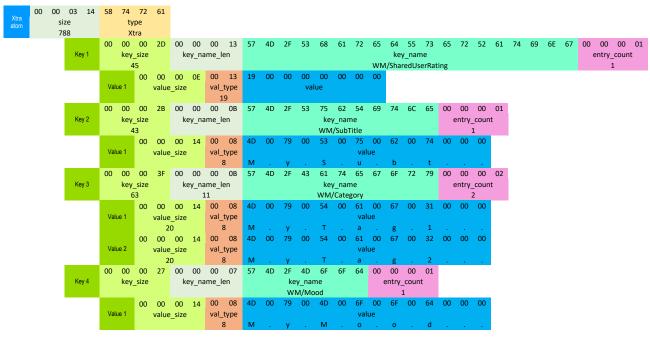
I cannot find this information as you would expect it on the Microsoft website. Although many 'keys' that are used in the Xtra atom (e.g. WM/Composer) are described in the WMF Attribute list (3) including a type enumeration (WMT\_ATTR\_DATATYPE enumeration); part of this enumeration is as follows: WMT\_TYPE\_DWORD = 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



1 https://en.wikipedia.org/wiki/MPEG-4 Part 14#

- 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.

<sup>2</sup> MOV 0234-windowscomments.mp4

<sup>3</sup> Apple-Iphone5s.mov