ITL 3 PRACTICAL - 06

**AIM**

**Design a tag to perform the necessary editing in a given report.**

**Media metadata can be roughly broken down into two categories: the one that describes how media is stored and the one that describes the content of the media. The first type of metadata are things like: video codec, clip duration, video bitrate etc. while the second type can be things like language, title, artist and even the album cover image. The reason to distinguish between them is because the first is mostly computer processable while the second makes sense to humans. This article focuses on the second type of metadata, i.e. the one that describes the content itself.**

**The early formats such as MP3 had a pre-defined set of metadata fields specified by the creators of the format. However since there are basically infinite ways to describe something the new generation of formats supports an extensible metadata storage instead.**

**In MP4 metadata is an infinite set of key,value pairs where both keys and values are arbitrary. It's up to a specific application to interpret the value of a certain key. For instance iPhone, Android or iTunes all have different set of keys they are aware of.**

**JCodec provides a way to read and modify the key,value pairs inside the MP4 files without actually understanding what they mean. This functionality is implemented in the class** [**org.jcodec.movtool.MetadataEditor**](https://github.com/jcodec/jcodec/blob/master/src/main/java/org/jcodec/movtool/MetadataEditor.java)**. There's also a command line interface to this functionality in the class** [**org.jcodec.movtool.MetadataEditorMain**](https://github.com/jcodec/jcodec/blob/master/src/main/java/org/jcodec/movtool/MetadataEditorMain.java)**.**

## **MetadataEditor CLI**

## **The most basic syntax is:**

## **./metaedit <file.mp4>**

## **in which case it will print out all the content metadata it can find in the file. You can also query for a specific metadata field:**

## **./metaedit -q <field name> <file.mp4>**

## **this will print out just the value of a specific metadata field. For instance to save the album cover image from the song to a file one could do:**

## **./metaedit -q covr file.mp4 > cover.jpg**

## **To update metadata in a file one can use either of the two options -- '-sk name' or '-si name'. These stand for keyed metadata and iTunes metadata. The difference is the location inside the file and the way they are stored. iTunes metadata is used by iTunes, hence the name, and keyed metadata is mostly used by modern applications, including Android phones. It's important to know that the keys for the iTunes metadata are limited to 4 characters, i.e. they are FOURCCs, however the keys for keyed metadata can have arbitrary length and usually start with a domain prefix like 'com.google.android' or 'com.apple'. So to update a specific metadata field use:**

## **./metaedit -sk com.apple.foo=one:com.google.bar=two -si @ART=Whatever\ this\ might\ be file.mp4**

## **The above will set or update the key 'com.apple.foo' to have the value of 'one' and the key 'com.google.bar' to have the value of 'two'. This will also update the iTunes artist field (fourcc '@ART') to 'Whatever'.**

## **Metadata values can also have types other then string. Applications are free to define their own data types however there's a set of standard types** [**defined here**](https://developer.apple.com/library/content/documentation/QuickTime/QTFF/Metadata/Metadata.html#//apple_ref/doc/uid/TP40000939-CH1-SW35)**. MetadataEditor CLI allows you to specify the type for the metadata field by putting it into parentheses after the value:**

## **./metaedit -sk com.android.capture.fps,float=25.0:com.apple.somethingelse,int=10 file.mp4**

## **JCodec only supports 'float', 'int' and 'string' types right now.**

## **By default metaedit will attempt to put the new movie header in place of the old one, however the header may grow in size as a result of adding more stuff to it.**

## **When this happens it can no longer be put into the old place hence the space will be made to accomodate the new (bigger) header. This requires rewriting the whole file which can be slow for large files.**

## **In cases when it's not critical that the header stays in it's original place a 'fast' mode can be used. In this mode instead of rewriting the file metaedit will simply place the header at the end of the file. This mode is useful when working with huge files for which it's not important that they remain web-optimized (header at the beginning).**

## **The 'fast' mode is triggered by giving '-f' or '--fast' option on the command line:**

## **./metaedit -f -si ©ART=Super\ cool file.mov**

## **Some fields can have binary value that can't be printed out or typed in the command line. Such BLOB values can be put into metadata fields from a file or stdin. You can only do one value at a time by specifying '-sib key,type=file\_name' argument on a command line:**

## **./metaedit -sib covr,jpeg=super.jpg file.mov**

## **The above will read the contents of the file 'super.jpg' and set it as a value of the metadata field 'covr', the type of the field will be set to 'jpeg'.**

## **If the value is not specified as in '-sib key,type' then the bytes will be read from stdin, for example:**

## **cat super.jpg | ./metaedit -f -sib covr,jpeg star\_trek.mov**

## **At the moment metaedit supports the following BLOB types:**

|  |  |
| --- | --- |
| **Name** | **Comment** |
| **utf8** | **UTF-8 encoded string** |
| **utf16** | **UTF-16 encoded string** |
| **float** | **4 bytes representing a float in big endian byte order** |
| **int, integer** | **4 bytes representing an int in big endian byte order** |
| **jpeg, jpg** | **A JPEG image** |
| **png** | **A PNG image** |
| **bmp** | **A BMP image** |

## **MetadataEditor API**

## [**org.jcodec.movtool.MetadataEditor**](https://github.com/jcodec/jcodec/blob/master/src/main/java/org/jcodec/movtool/MetadataEditor.java) **class is the one that implements the MP4 manipulation logic used in the CLI. Below is the code that prints out the metadata from a file:**

## **MetadataEditor mediaMeta = MetadataEditor.createFrom(new File("file.mp4"));**

## **Map<String, MetaValue> keyedMeta = mediaMeta.getKeyedMeta();**

## **if (keyedMeta != null) {**

## **System.out.println("Keyed metadata:");**

## **for (Entry<String, MetaValue> entry : keyedMeta.entrySet()) {**

## **System.out.println(entry.getKey() + ": " + entry.getValue());**

## **}**

## **}**

## 

## **Map<Integer, MetaValue> itunesMeta = mediaMeta.getItunesMeta();**

## **if (itunesMeta != null) {**

## **System.out.println("iTunes metadata:");**

## **for (Entry<Integer, MetaValue> entry : itunesMeta.entrySet()) {**

## **System.out.println(fourccToString(entry.getKey()) + ": " + entry.getValue());**

## **}**

## **}**

## **OR to edit the metadata:**

## **MetadataEditor mediaMeta = MetadataEditor.createFrom(new File("file.mp4"));**

## **Map<String, MetaValue> keyedMeta = mediaMeta.getKeyedMeta();**

## **keyedMeta.put("com.google.foo", MetaValue.createInt(42));**

## **mediaMeta.save(false); // fast mode is off**

## **Known issues**

## **The iTunes metadata added to a non-iTunes file (the one that came out of FFMpeg for instance) will not show up in Apple tools (shows up fine in mediainfo). However changing metadata on an existing iTunes file will work.**

## 