MXF : MATERIAL EXCHANGE FORMAT

The MXF is a format designed for audiovisual use.

It's a binary file and, by convention, its extension is **.mxf**

At the first sight, the MXF format seems to be obscure compared to other files in a DCP. We move from XML files, readable by humans with a simple text editor, to a binary file completely unreadable to the average person. And yet, you will see that the MXF format is very conceptually simple.

Keep in mind, we will only look at the MXF used for DCPs [1], there are many other types of MXF each with its own specificities.

Preface

In this chapter, we will use terms such as "binary", "hexadecimal", "bit" or "byte/octet". The term "octet" will be used by default instead of "byte" which can easily be confused with the term "bit" (reminder : 8 bits = 1 octet / byte). I will also try to be as didactic as possible for those who are not familiar with these terms. If that's not the case, feel free to let me know.

AN MXF IS A SIMPLE CONTAINER

A container is only a wrapper around an asset. An asset is - for example - a PNG, a MPEG-4, a JPEG.

The MXF is a wrapper around an asset with some additional information. Without this asset inside the MXF, an MXF would just be an empty wrapper with metadata.

See it as a cardboard box : put various items inside the box. There can be image, sound, text, metadata or other types of data (such as assets). Close the box, stick a label on it : you have your own "container".

The MXF is not restricted to a specific asset format, we can integrate everything. That's why we have various MXF files that integrate different assset formats, such as JPEG2000, DPX, Prores or MPEG-4.

Thanks to its malleability, we can have specific MXF files for image, sounds, subtitles or even for anccilary data like the kind used in 4DX cinema (you know, the moving chairs ! :)

THE SMPTE STANDARDS

The MXF DCP format is mainly defined in the following standards :

Norm	Titlte
SMPTE 377	Material Exchange Format — File Format Specification
SMPTE 410	Material Exchange Format — Generic Stream Partition

Additional standards found in various SMPTE documents:

Norm	Title
SMPTE EG-377-3	MXF — Engineering Guideline
SMPTE EG-41	MXF — Engineering Guideline
SMPTE EG-42	MXF — Descriptive Metadata
SMPTE 379-1	MXF — MXF Generic Container
SMPTE 379-2	MXF — MXF Constrained Generic Container
SMPTE 380M	MXF — Descriptive Metadata Scheme-1
SMPTE 381-x	MXF — Mapping xxx Streams into the MXF
SMPTE 382M	MXF — Mapping Wave Audio into the MXF Generic Container
SMPTE 390	MXF — Specialized Operational Pattern OP-Atom
SMPTE 422	MXF — Mapping JPEG2000 Codestreams into the MXF Generic Container
DCP	
SMPTE 429-2	DCP - DCP Operational Constraints
SMPTE 429-3	DCP - Sound And Pictures Track File
SMPTE 429-4	DCP — MXF JPEG2000 Application
SMPTE 429-6	DCP — MXF Track File Essence Encryption
SMPTE 429-20	DCP - MXF Constraints
Misc	
SMPTE 336	Data Encoding Protocol using Key-Length-Value (KLV)
SMPTE 298	Universal Labels for Unique Identification of Digital Data
SMPTE RP-224	SMPTE Universal Labels
SMPTE ST-400	SMPTE Labels Structure
SMPTE ST-384	Container : Uncompressed pictures
SMPTE ST-382	Container : Audio
SMPTE ST-422	Container : JPEG2000 (DCinema)
SMPTE ST-394	Container : MXF Generic Container (~AuxData)
SMPTE ST-405	Container : MXF Generic Container : Elements and Individual Data Items (~AuxData)
Extras	
The MXF Book	Introduction to the Material Exchange Format (ISBN: 9780240806938)

AN MXF IS SIMPLY A SERIES OF BUILDING BRICKS

Picture the inside of an MXF like this :

A series of building bricks of Lego.

Each brick represents a specific piece of data : a brick for headers, a brick for metadata, a brick for image, etc.

An MXF is therefore a simple concatenation of many bricks one after the other (or one on top the other, if you prefer)

Each brick is called **KLV**, which we'll study in the next chapter: **KLV : Key-Length-Value**

ANALYZE OF MXF

Through the chapters, we'll often analyze MXF in depth. For that, we'll use several tools that are located in the asset MXF directory, which we will analyze throughout the following chapters.

Meanwhile, let me introduce two tools that help you :

- The first is an easy-to-use online visualisator, no need to install.
- The second, much more complete, an MXF analyzer which will be used throughout the MXF chapters.

ONLINE MXF VISUALISATOR

To help you, you can visualize the different KLVs in an MXF using this online MXF analyzer (still a prototype) :

DRAG YOUR MXF HERE

0 elements found
0 %

You can test it with this MXF file or one of your MXF file: no upload on server, everything is analyzed in your computer.

The project is available at this address : https://github.com/sherpadown/js-mxf (demo)

MXF ANALYZER :

You can also test it with the `mxf-reader`. This tool is much more comprehensive than the previous tool. It allows deeper analysis of MXF files, giving more information about KLVs, the ability to extract each KLV and its data (such as JPEG2000, Wave, ...), decrypt MXF, and more... :

```
$ mxf-reader --help
usage:
  -h, --help            show this help message and exit
  -f FILENAME           <mxf> = mxf filename
  -x EXTRACT            <directory> = extract each KLV into files
  -k KEY, --key KEY     AES Key, ex. --key 00000000000000000000000000000000
  -v, --verbose         increase output verbosity
  -n, --no-resolv       Do not resolv UL (speed)
  --filter FILTER       filter by name
  --fuzzy               Fuzzy mode only (very slow)
  --limit LIMIT         stop after x klv parsed
  --slow                Slowdown parse to avoid flood loadavg
```

To keep this tool on hand while reading of following chapter on MXF is a plus, **but not essential**, I've included output samples and analysis from this tool at each step to present every part of an MXF.

TABLE OF CONTENTS

There are various MXF sections :

NOTES

1. With the possibility to go beyond on the **Interoperable Master Format** (IMF) ↩