

References	SMPTE 422-2014 - MXF - Mapping JPEG2000 Codestreams into the MXF Generic Container SMPTE 429-4-2006 - DCP - MXF JPEG2000 Application SMPTE 429-3-2007 - DCP - Sound And Pictures Track File ¹ SMPTE 377-1-2011 - MXF - File Format Specification - RGBA Essence Descriptor (p148) SMPTE 384M-2005 - MXF - Mapping of Uncompressed Pictures into the Generic Container
KLVS Model	Data Item
Universal Label	060e2b34.01020101.0d010301.15010801 - Picture Essence, non-chiffré 060e2b34.02530101.0d010101.01012900 - RGBA Essence Descriptor 060e2b34.02530101.0d010101.01015a00 - JPEG2000 Picture Sub-Descriptor

PREFACE

In a DCP, images are stored one by one in their specific KLVs placed in the **Body** part: each frame is separated from the others and stored in a KLV named **Picture Essence**.

SMPTE chose to store each movie frame separately, without any link between the previous and the next image.

To simplify the concept, in encoding systems like H254, H265, MPEG, and other, several frames are just [extrapolations of the previous frames](#) in order to save space. At fixed interval, a complete reference frame is inserted. Picture this: to create movement from a still image, we overlay several transparent layers on top of the previous ones, each containing only the changes from the layer below.

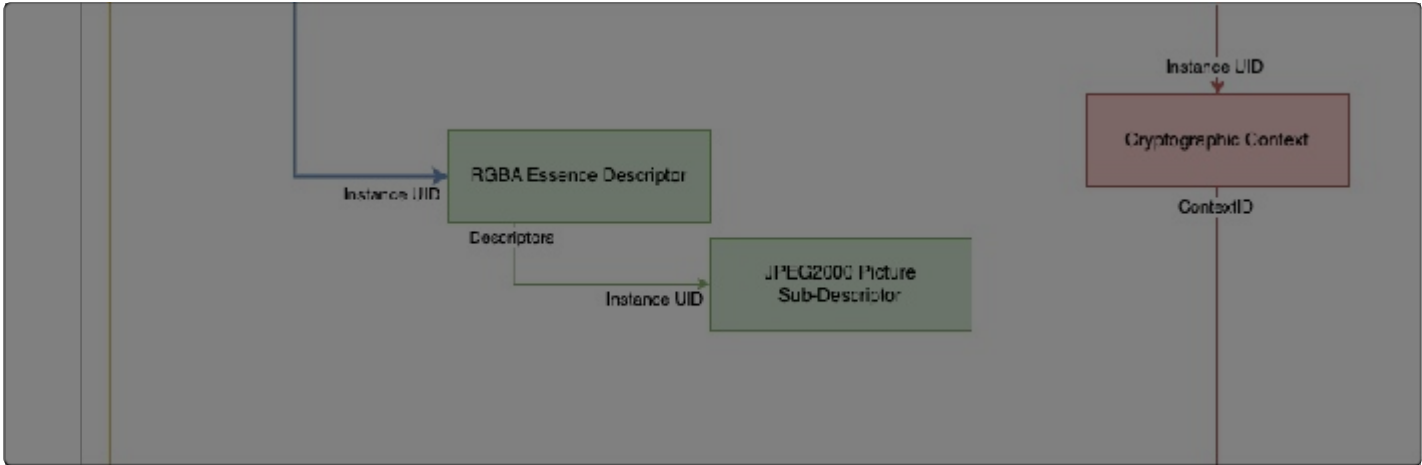
This technical feature has been discarded for DCP because, if a frame is dropped, unacceptable artifacts will appear during theater projection. Each frame must be independent : if a frame is dropped, no big deal, the next one will be there to compensate this issue.

Note that in this chapter, we will focus exclusively on unencrypted MXF files. Thus, KLVs are of a specific type ([Data Item](#)) and each frame is unencrypted. In encrypted form, KLVs are slightly different. You may find explanations in the chapter on [MXF Cryptography](#).

KLVS HEADERS

To store JPEG2000, we need some additional KLVs in the MXF headers :

The relationship between the **RGBA Essence Descriptor** & **JPEG2000 Picture Sub-Descriptor** is described below :



You can find the chapters on these two KLV headers :

- [RGBA Essence Descriptor](#)
- [JPEG2000 Picture Sub-Descriptor](#)

KLV BODY : PICTURE ESSENCE

Each frame is stored in a single KLV named "Picture Essence" : thus, we will have several (thousands for a complete movie) of KLV named "Picture Essence".

Its **Universal Label** is `060e2b34.01020101.0d010301.15010801` which is structured as follows :

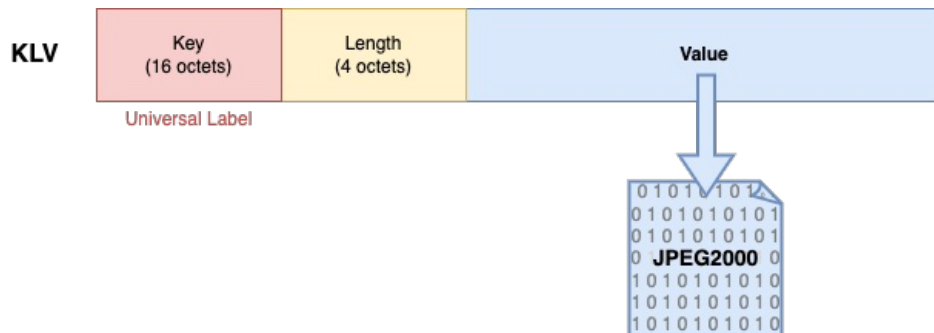
```
060e2b34.01020101.0d010301.15010801
```

Each last octet (byte) represents a status :

- **15** represents a **Generic Container Picture Item** ¹. In other words, a generic container that stores pictures
- **01** represents the number of picture stored in the container. Here, we have a single picture in the container.
- **08** represents the status named **Not Clip-wrapped JPEG 2000 Picture Element**.
- **01** represents a kind of index number for the picture. Don't worries, the value is always 1.

The **Picture Essence** KLV is the simplest of the KLVs: all you need to do is read the entire **Value** of the KLV which integrates the entire frame, from the JPEG2000 headers to the picture data. You have the entire JPEG2000 content file ! No processing needed, no conversion required. Thus, the JPEG2000 file is an **12-bits X'Y'Z'** image file.

You can save the content of the **Value** of KLV directly as a **JPEG2000 J2C** file.



CODE : READ AND EXTRACT THE KLV "PICTURE ESSENCE"

[Source code](#) to read the KLV **Picture Essence** and extract them :

```

import sys

# Integer Conversion
def to_int(length : bytes = b'') -> int:
    return int.from_bytes(length, byteorder='big')

if len(sys.argv) < 2:
    print("Usage: %s <mxfile>" % sys.argv[0])
    sys.exit(1)

mxfile = sys.argv[1]

with open(mxfile, "rb") as file:

    while True:

        # Key : Universal Label
        key = file.read(16)

        # End of file
        if not key:
            break

        # Length (BER format) (quick & dirty, see KLV chapter why)
        length = to_int(file.read(4)[1:]) # BER format - read last 3 bytes

        # Value
        value = file.read(length)

        # Filter by KLV "Picture Essence"
        if key.hex() != "060e2b34010201010d01030115010801":
            continue

        # Show each KLV
        print("{key} - {length:>6d} bytes - {value}...".format(
            key = key.hex(),
            length = length,
            value = value[0:16].hex()
        ))

        # write Plaintext to file
        filename = "output_%d.j2c" % file.tell()
        print(f"Extract {filename}")
        with open(filename, "wb") as f:
            f.write(value)

```

The source code isn't complicated, all you need to do is step through each KLV, and when a KLV **Picture Essence** is detected, just read the **Length** field and read the **Value** field.

The advantage of the KLV **Picture Essence JPEG2000** is that the entire content of the JPEG2000 is stored in the **Value** field. All you need to do is extract the raw data and save it to a file to have a complete, ready-to-use JPEG2000 file.

CODE : WRITE OWN KLV PICTURE ESSENCE

Coming soon...

CRYPTOGRAPHY

Note that everything you have read in this chapter - including the linked KLV [RGBA Essence Descriptor](#) & [JPEG2000 Picture Sub-Descriptor](#) - will apply to the chapter on [MXF Cryptography](#) as well (except that **Essence Container** should be replaced by **Encrypted Essence Container**)

XYZ COLORIMETRY

See the chapter [XYZ Colorimetry](#)

JPEG2000 ENCODING

See the chapter [JPEG2000](#)

STEREOSCOPIC (3D)

See the chapter [Essence Picture : Stereoscopic \(3D\)](#)

HIGH FRAME RATE (HFR)

See the chapter [Essence Picture : High Frame Rate \(HFR\)](#)

NOTES

- =====
1. « *Track Files shall use the MXF Generic Container (GC) SMPTE 379M. Track Files shall use GC frame-based essence mappings* » -- SMPTE 429-3-2007 - DCP - Sound And Pictures Track File [↩](#)