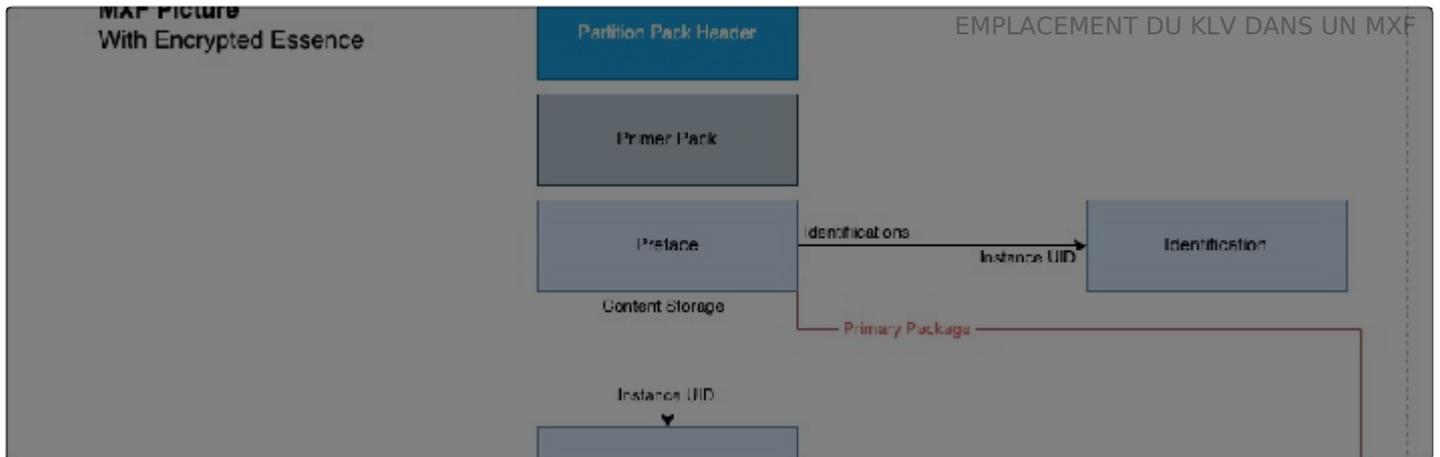


Références	SMPTE 377-1-2011 - MXF - File Format Specification Chapitre A.2 - Preface ^{P113}
Modèle KLV	Local Sets
Universal Label	06.0e.2b.34.02.53.01.01.0d.01.01.01.01.01.2f.00 (SMPTE)

PRÉFACE



La **Preface** est une sorte de passage du seuil d'un MXF, c'est lui qui va faire lien avec la fiche d'identité du MXF, le "paquetage" principal (Primary Package) et le KLV Content Storage qui fera lien avec les autres éléments.

DESCRIPTION

```
|| 3C0A - Instance ID || 929bbcfcd.cb794af5.99225662.61f9228b
|| 3B02 - Last Modified Date || 2022-04-08 12:55:36.0000+00:00
|| 3B05 - Version || 258
|| 3B07 - Object Model Version || 1
|| 3B08 - Primary Package || 6d2f324a.de9a45e4.abc8ee3b.ae364233
|| 3B06 - Identifications || 1 item(s): 81305bab.afdc40f3.8177385d.dddf0778
|| 3B03 - Content Storage || b9dac636.51e04124.88e99265.9bbeceb
|| 3B09 - Operational Pattern || 060e2b34.04010102.0d010201.10000000
|| || (Operational Pattern - version 2 - SMPTE)
|| 3B0A - EssenceContainers || 2 item(s):
|| || - 060e2b34.04010103.0d010301.027f0100
|| || (MXF Generic Container)
|| || - 060e2b34.04010107.0d010301.020b0100
|| || (Encrypted Essence Container - Frame Wrapped)
|| 3B0B - Descriptive Metadata Schemes || 1 item(s):
|| || - 060e2b34.04010107.0d010401.02010100
|| || (Cryptographic Framework Label)
```

Instance ID est l'UUID de la Preface

Last Modified Date est ... la date de la dernière modification du MXF :)

Version est un peu particulier, elle dépend de quelle norme on respecte :

- SMPTE 377-1-2004 : le chiffre doit être **258** et le Major/Minor de la [Partition Pack Header](#) doit être à 1.2
- SMPTE 377-1-2011 : le chiffre doit être **259** et le Major/Minor de la [Partition Pack Header](#) doit être à 1.3

Pour la compatibilité, en reste en 1.2, donc 258.

Alors, vous allez me dire, pourquoi des chiffres de version aussi grands ? Et bien, la norme SMPTE définit une équation pour créer un nombre de Version : $\text{Major} * 256 + \text{Minor}$. Donc pour la dernière version, nous avons $1 * 256 + 3 = 259$.

Object Model Version spécifie le numéro de version du mécanisme pour le stockage des objets (Object Storage Mechanism). Voir la [SMPTE RP-210](#) pour en savoir plus.

Primary Package fait référence au KLV [Source Package](#) via son **Instance UID**

Identifications fait référence au KLV [Identification](#) via son **Instance UID**. **Identification** est la fiche d'identité du MXF.

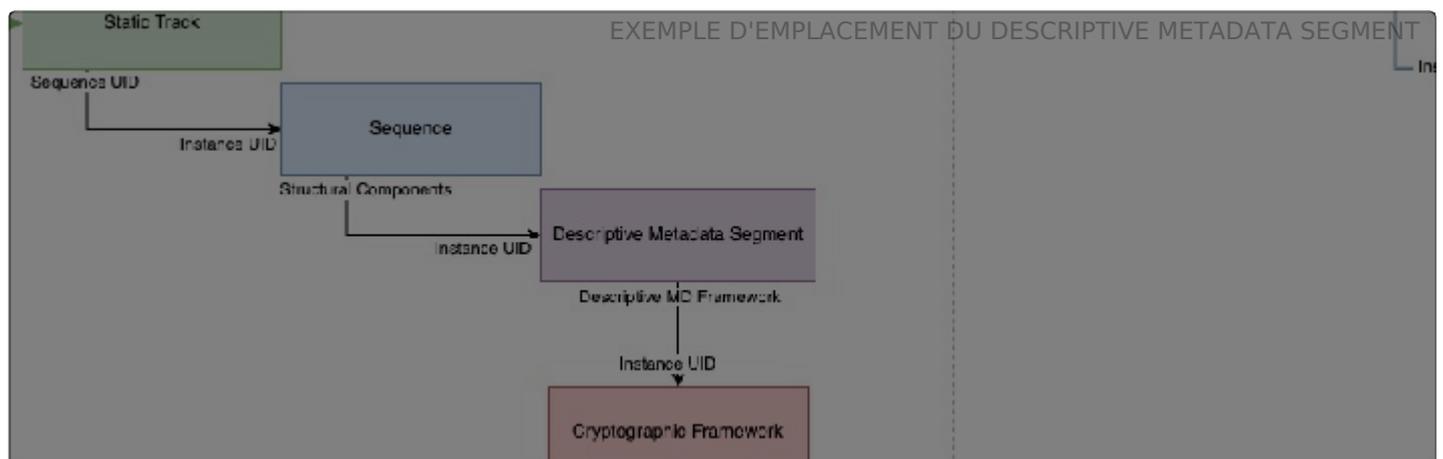
Content Storage fait référence au KLV [Content Storage](#) via son **Instance UID**. **Content Storage** fait lien avec tous les packages (**Source Package** & **Material Package**) et du Essence Container Data.

Operational Pattern désigne le type de modèle de MXF. Pour en savoir plus, voyez du côté du chapitre [Operational Pattern](#) (toujours pas obligatoire :)

Essence Containers sont les **Universal Labels** identifiant le type des différentes essences. Ici, nous voyons que nous un **MXF Generic Container** et un **JPEG2000 Picture Element**.

Descriptive Metadata Schemes donne un identifiant pour un potentiel KLV **Descriptive Metadata Segment**. Ici, vu que nous avons un MXF chiffré, nous aurons un **DM Segment** sur la cryptographie un peu plus tard (du côté de [Source Package](#) → [Static Track](#)) - avec l'ensemble de ses KLV de métadonnées cryptographiques qui seront liés à lui (comme [Cryptographic Framework](#) et [Cryptographic Context](#)). Sur un MXF non-chiffré, il n'y aura aucune valeur.

Voici un exemple de [Descriptive Metadata Segment](#) :



LES DONNÉES BRUTES DU KLV

Le KLV **Preface** est un type **Local Set**, cela veut dire que nous aurons une foultitude de bébé-KLV.

Exemple avec la **Value** du KLV :

```
3c0a0010929bbcfdcb794af59922566261f9228b3b02000807e604080c3724003b0500020102
3b070004000000013b0800106d2f324ade9a45e4abc8ee3bae3642333b060018000000010000
001081305babafdc40f38177385dddf07783b030010b9dac63651e0412488e992659bbecedb
3b090010060e2b34040101020d010201100000003b0a0028000000200000010060e2b340401
01030d010301027f0100060e2b34040101070d010301020b01003b0b00180000000100000010
060e2b34040101070d01040102010100
```

Les bébé-KLV seront des items comprenant une clef de 2 octets, une taille de 2 octets et le reste (basé sur la valeur dans la taille).

Prenons notre premier item que j'ai pré-découpé en connaissant déjà sa taille complète afin d'aider à la compréhension :

```
3c0a0010929bbcfdbc794af59922566261f9228b
```

Nous voyons notre petit **Local Tag** `3C0A` : après analyse de la table de correspondances **Primer Pack**, nous constatons que notre **Local Tag** correspond à une **Instance UID** : nos données seront un identifiant au format UUID.

Puis, pour la taille de nos données, nous avons la valeur `0x0010` qui correspond à une valeur décimale de 16. Nous savons donc que notre UUID sera d'une taille de 16 octets (ce qui semble logique :)

Sitôt fini, nous passons à l'item suivant :

```
3b02000807e604080c372400
```

Même procédé :

Notre **Local Tag** `3C0A` correspond à un attribut **Last Modified Date**, nous aurons donc affaire à une date au format Timestamp (YYYYMMDDhhmmssms).

Sa taille est de `0x0008` donc 8 octets.

La valeur est ... `07e604080c372400`. Oui, effectivement, vous n'aurez pas une date lisible directement :)

Avec cette donnée, vous devez découper par tranche, prenons l'année, elle est encodée sur 2 octets, prenons donc les deux premiers octets `0x07e6`, en décimal cela donne... `2022` :)

Vous découpez et convertissez d'hexadécimal en décimal l'ensemble des valeurs pour arriver à une date :

```
07e6 04 08 0c 37 24 00  <= valeur hexadecimale
-----
YYYY MM DD hh mm ss ms
-----
2022 04 08 12 55 36 00  <= valeur décimale
```

Notre **Last Modified Date** est donc `2022-04-08 12:55:36.00`

Nous allons prendre un cas plus complexe, sautons directement sur la partie **EssenceContainers** dont le **Local Tag** est `3B0A` :

```
3c0a0010929bbcfdbc794af59922566261f9228b3b02000807e604080c3724003b0500020102
3b070004000000013b0800106d2f324ade9a45e4abc8ee3bae3642333b060018000000010000
001081305babafdc40f38177385dddff07783b030010b9dac63651e0412488e992659bbecebd
3b090010060e2b34040101020d010201100000003b0a0028000000200000010060e2b340401
01030d010301027f0100060e2b34040101070d010301020b01003b0b00180000000100000010
060e2b34040101070d01040102010100
```

Analysons maintenant cette partie :

```
3b0a00280000000200000010060e2b34040101030d010301027f0100060e2b34040101070d010301020b0100
```

Si vous avez l'oeil, vous voyez déjà les **Universal Label** en fin du jeu de données :

```
3b0a00280000000200000010060e2b34040101030d010301027f0100060e2b34040101070d010301020b0100
```

Mais à quoi correspondent les valeurs en blanc ? Et bien, cela correspond simplement au nombre d'items (`0x00000002`) et de la longueur de chaque item (`0x00000010`). Avec cette information, nous savons que nous aurons 2 items avec une taille de 16 octets chacun, ce qui correspond à nos deux valeurs en roses.

Les **Local Sets** sont intéressants car ils ne nécessitent pas de grand développement, une simple boucle de lecture et des méthodes d'interprétations par type de données, et vous pourrez lire une grande partie des autres KLV qui utilisent le même principe.

STRUCTURES DES DONNÉES DE PREFACE

Local Tag	Nom de l'attribut	Type	Taille (*)	Fixe/Variable SMPTE	Obligatoire
3C0A	Instance UID	UID	16 octets	Fixe	Obligatoire
3B02	Last Modified Date	Timestamp	8 octets	Fixe	Obligatoire
3B05	Version	uint16	2 octets	Fixe	Obligatoire
3B07	Object Model Version	uint32	4 octets	Fixe	Non
3B08	Primary Package	UID	16 octets	Fixe	Non
3B06	Identifications	Batch-UID	(4+4) + 16x	Variable	Obligatoire
3B03	Content Storage	UID	16 octets	Fixe	Obligatoire
3B09	Operational Pattern	UL	16 octets	Fixe	Obligatoire
3B0A	EssenceContainers	Batch-UL	(4+4) + 16x	Variable	Obligatoire
3B0B	Descriptive Metadata Schemes	Batch-UL	(4+4) + 16x	Variable	Obligatoire

La norme possède plus d'items mais ne seront jamais utilisés dans un MXF DCP.

Pour les types étranges :

- **Batch-UL / Batch-UID** : 4 octets pour le nombre d'item, 4 octets pour la taille d'un item, et enfin 0 ou plusieurs identifiant de 16 octets selon le nombre d'item. Le parsing des Batch-UL a été vu dans [Content Storage](#)
- **Timestamp** = Année (int16) + Mois (uint8) + Jour (uint8) + Heure (uint8) + Minute (uint8) + Second (uint8) + Millisecondes (uint8)

HIÉRARCHIE DE FORMAT

Interchange Object → Preface

ETUDE RAPIDE DE L'UNIVERSAL LABEL DE PREFACE

```
UL = 06.0E.2B.34.02.53.01.vv.0D.01.01.01.01.01.2F.00
    ^^----- Item Designator      : Organizationally registered
      ^^----- Organization        : AAF
        ^^----- Application       : MXF / AAF Association Structural Me
          ^^----- Structure Version : Structure Version 1
            ^^----- Structure Kind  : MXF / AAF Association compatible se
              ^^----- MXF Set Definition : Preface
                ^^----- MXF Set Definition : Preface
```