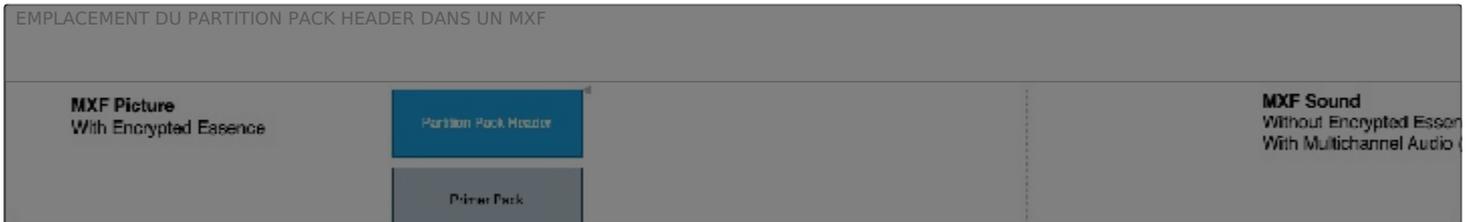


Références	SMPTE 377-1-2011 - MXF - File Format Specification Chapitre 6 - Overall Specification ^{P26} Chapitre 7 - Partitions ^{P40} SMPTE 429-3-2007 - DCP - Sound And Pictures Track File Chapitre 4.3.5 - Partitions ^{P5}
Modèle KLV	Fixed-Length Pack
Universal Label	06.0e.2b.34.02.05.01.01.0d.01.02.01.01.XX.XX.00 (SMPTE)

PRÉFACE



Les **Partitions Pack** sont des KLV spécifiques de séparations entre les différentes parties d'un KLV.

Voyez les **Partitions Packs** comme des intercalaires dans un classeur afin de séparer différentes parties de documents.

Dans un MXF de DCP, vous avez au minimum 3 KLV **Partition Packs** : un **Header**, un **Body** et un **Footer**. Chacun ayant une spécificité :

- **Partition Pack Header** intègre toutes les métadonnées du MXF
- **Partition Pack Body** intègre toutes les données utiles du MXF (par exemple, les images ou le son)
- **Partition Pack Footer** intègre toutes les métadonnées de fin

La **Partition Pack Header** est le **premier KLV** d'un MXF.

DESCRIPTION

Nos trois Partition Packs ont un [Universal Label](#) :

Nom du KLV Partition Pack	UUID
Partition Pack - Header	060e2b34.02050101.0d010201.01 02 04 00
Partition Pack - Body	060e2b34.02050101.0d010201.01 03 04 00
Partition Pack - Body Generic Stream	060e2b34.02050101.0d010201.01 03 11 00
Partition Pack - Footer	060e2b34.02050101.0d010201.01 04 04 00

Les différents **Partition Pack** possèdent une base commune dans leur Universal Label qui est : `060e2b34.02050101.0d010201.01XXXX00`

Les derniers octets définissent un type et un statut particulier :

- L'**octet 14** définit le **type de partition** :
 - 0x02 : Header (entête)
 - 0x03 : Body (corps)
 - 0x04 : Footer (fin)
- L'**octet 15** définit **son statut** :
 - 0x01 : Ouvert et Incomplet (Open & Incomplete)
 - 0x02 : Fermé et Incomplet (Closed & Incomplete)
 - 0x03 : Ouvert et Complet (Open & Complet)
 - 0x04 : Fermé et Complet (Closed & Complet)

Tous les MXF de DCP doivent être à 0x04, donc **Fermé et Complet** (Closed & Complet). A noter que **Partition Pack Footer** ne possède pas de status "Ouvert" (open) : Un **Footer** doit toujours être fermé (Closed).

C'EST QUOI « BODY GENERIC STREAM » ?

Leurs présences seront après un **Partition Pack Body** classique.

Vous ne les rencontrerez que dans des MXF sous-titres, ils serviront à séparer les différents types de données (sous-titres XML, sous-titres PNG, données Font, ...) à l'intérieur d'un **Partition Pack Body**.

Un exemple de pile KLV avec un MXF sous-titres:

```
- Partition Pack - Body
- TimedText - Essence (XML)
- Partition Pack - Body (<-- Generic Stream Partition)
- TimedText - Ancillary Resource (Font)
```

LES MÉTADONNÉES

Le KLV est de type [Fixed-Length Pack](#) : chaque métadonnées est à un emplacement précis avec une taille pré-définie.

Interprétations des données binaires et sortie du logiciel

Les données et intitulés que vous verrez ci-dessous sont des interprétations par rapport aux données binaires stockées dans le KLV.

Ainsi, vous ne trouverez pas dans les données brutes, les termes **Major Version**, **ThisPartition** ou encore **Operational Pattern** ou autres **JPEG2000 Picture Element**, mais seulement des codes et des identifiants (UUID, UL, LocalTag, etc..) qui sont référencés dans les documentations SMPTE.

Avec un autre logiciel, vous pourrez avoir d'autres noms et intitulés. Normalement, les différences seront mineures car basées sur les informations des documentations SMPTE.

Pour en savoir si les données binaires brutes, référez-vous au paragraphe en fin de ce chapitre

Voici [un exemple](#) de données après parsing et interprétations des trois Partition Pack (Header, Body, Footer)

```
0 | 060e2b34.02050101.0d010201.01020400 | Partition Pack - Header - Closed & Complete
-----
Major Version      || 1
Minor Version     || 2
KAGSize           || 1
ThisPartition      || 0
PreviousPartition || 0
FooterPartition    || 56844
HeaderByteCount   || 16244
IndexByteCount    || 0
IndexSID          || 0
BodyOffset        || 0
BodySID           || 0
Operational Pattern || 060e2b34.04010102.0d010201.10000000
                  || Operational Pattern - Specialized Pattern - OP Atom - SMPTE
EssenceContainers || 2 item(s):
                  || - 060e2b34.04010103.0d010301.027f0100
                  ||   MXF Generic Container
                  || - 060e2b34.04010107.0d010301.020b0100
                  ||   JPEG2000 Picture Element - Frame Wrapped
```

```
16384 | 060e2b34.02050101.0d010201.01030400 | Partition Pack - Body - Closed & Complete
-----
Major Version      || 1
Minor Version     || 2
KAGSize           || 1
ThisPartition      || 16384
PreviousPartition || 0
FooterPartition    || 0
HeaderByteCount   || 0
IndexByteCount    || 0
IndexSID          || 0
BodyOffset        || 0
BodySID           || 1
Operational Pattern || 060e2b34.04010102.0d010201.10000000
                  || Operational Pattern - Specialized Pattern - OP Atom - SMPTE
EssenceContainers || 2 item(s):
                  || - 060e2b34.04010103.0d010301.027f0100
                  ||   MXF Generic Container
                  || - 060e2b34.04010107.0d010301.020b0100
                  ||   JPEG2000 Picture Element - Frame Wrapped
```

```
56844 | 060e2b34.02050101.0d010201.01040400 | Partition Pack - Footer - Closed & Complete
-----
Major Version      || 1
Minor Version     || 2
KAGSize           || 1
ThisPartition      || 56844
PreviousPartition || 16384
FooterPartition    || 56844
HeaderByteCount   || 0
IndexByteCount    || 151
IndexSID          || 129
BodyOffset        || 0
BodySID           || 0
Operational Pattern || 060e2b34.04010102.0d010201.10000000
                  || (Operational Pattern - Specialized Pattern - OP Atom - SMPTE)
EssenceContainers || 2 item(s):
                  || - 060e2b34.04010103.0d010301.027f0100
                  ||   MXF Generic Container
                  || - 060e2b34.04010107.0d010301.020b0100
                  ||   JPEG2000 Picture Element - Frame Wrapped
```

Voici les informations à propos des métadonnées :

Major et **Minor** Version : Normalement, Minor devrait être maintenant à 3 (SMPTE 377-2011) mais pour garder la compatibilité (SMPTE 377-2004), on reste en **Minor Version 2**.

KAGSize indique la taille du **KLV Alignment Grid** pour la partition. Ici, il est à 1, cela veut dire que... bah en fait il y a pas d'alignement. Ne cherchez pas, c'est SMPTE. Si elle était à 0, c'est qu'elle serait indéfinie. Et si elle était entre 2 et 1048576 (ça fait 1Mo). Vous ne verrez -normalement- jamais autre chose que la valeur 1. Juste pour expliquer le KLV Alignment Grid, on rajoute des octets de bourrage pour aligner les KLV. C'est seulement utile si on veut avoir des tailles parfaitement calibrées.

ThisPartition identifie sa position dans le fichier, par exemple, pour le premier, nous avons 0, ce qui est normal, c'est la première partition, donc elle est à l'offset 0 dans le fichier. Le second - Body - se trouve à l'offset 16384 et le dernier le Footer se trouve à l'offset 56844.

PreviousPartition définit où se trouve la partition précédente dans le fichier.

Pour le premier, toujours normal, c'est 0, nous n'avons pas de partition précédente (ou bien c'est lui-même).

Pour Body, on a un **PreviousPartition** à 0, donc identifiant le Header.

Et enfin, le Footer va indiquer l'offset de Body, qui est 16384. C'est comme une liste chaînée.

FooterPartition définit où se trouve la **Partition Footer**.

Sur certains **Partition Body**, le **FooterPartition** se trouvera très souvent à 0.

Ne me demande pas pourquoi, la norme impose pourtant un chiffre si le **Partition Footer** est bien présent.

HeaderByteCount indique la taille de l'entête entier, sans la taille du Partition Pack Header.

Cela permet de jumper directement à Body.

IndexSID indique (et identifie) simplement si un KLV "Index Table Segment" est présent.

Vous remarquerez que ce champ n'est pas à 0 seulement le Footer, contenant justement un KLV "Index Table Segment"

BodyOffset indique le nombre d'octet où démarre le segment Essence Container dans cette partition (pour l'instant, toujours à 0)

BodySID indique (et identifie) simplement si un KLV de données sera présent.

Vous le verrez à 1 que dans Partition Pack Body (avec les sous-titres, vous aurez plusieurs Body avec des BodySID supérieur à 1)

Operational Pattern désigne le type de modèle de MXF.

Pour en savoir plus, voyez du côté du chapitre [Operational Pattern](#) (mais ce n'est 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**

BodySID indique s'il y a des conteneurs d'essence (**Essence Container**).

Vu que **Partition Body** possède des essences, il est normal que ce soit le seul qui soit à 1.

IndexByteCount indique la taille de l'[Index Table Segment](#)

RELATION ENTRE LES DIFFÉRENTS PARTITIONS PACKS

Voici la relation entre les 3 **Partitions Packs** :

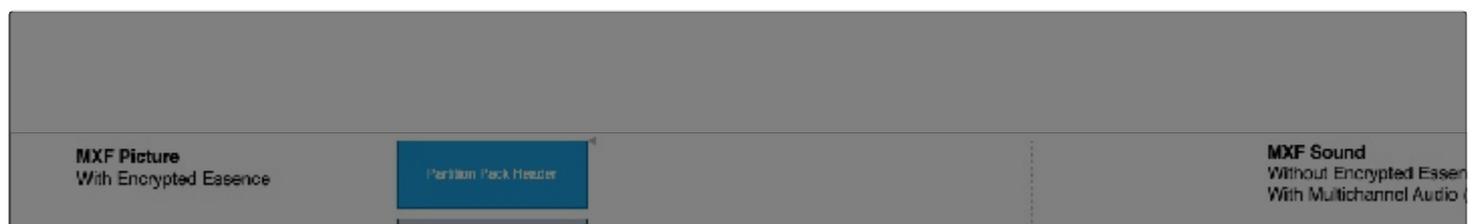
Item name	Header	Body	Footer
Major Version	1	1	1
Minor Version	2	2	2
KAGSize	1	1	1
ThisPartition	0	16384	56844
PreviousPartition	0	0	16384
FooterPartition	56844	0	56844
HeaderByteCount	16244	0	0
IndexByteCount	0	0	151
IndexSID	0	0	129
BodyOffset	0	0	0
BodySID	0	1	0
Operational Pattern	060e2b34.04010102.0d010201.10000000 (SMPTE)	060e2b34.04010102.0d010201.10000000 (SMPTE)	060e2b34.04010102.0d010201.10000000 (SMPTE)
EssenceContainers	2 item(s): 060e2b34.04010103.0d010301.027f0100 060e2b34.04010107.0d010301.020b0100	2 item(s): 060e2b34.04010103.0d010301.027f0100 060e2b34.04010107.0d010301.020b0100	2 item(s): 060e2b34.04010103.0d010301.027f0100 060e2b34.04010107.0d010301.020b0100

POSITION DE CES KLV DANS UN MXF

Les **Partitions Pack** se trouvent à différents endroits du MXF. Le **Header** se trouve en tout premier, le **Footer** en (presque) fin. En règle général, il n'y aura qu'un **Body** dans la plupart des assets, à l'exception de certains comme les sous-titres qui peuvent intégrer des Body supplémentaires pour séparer les différents assets stockés (sous-titres XML, sous-titres PNG, les données de Fonts, ...)

PARTITION PACK HEADER

La **Partition Pack Header** se trouve toujours en haut du MXF.



Nom	Taille	Type
Major Version	2 octets	uint16
Minor Version	2 octets	uint16
KAGSize	4 octets	uint32
ThisPartition	8 octets	uint64
PreviousPartition	8 octets	uint64
FooterPartition	8 octets	uint64
HeaderByteCount	8 octets	uint64
IndexByteCount	8 octets	uint64
IndexSID	4 octets	uint32
BodyOffset	8 octets	uint64
BodySID	4 octets	uint32
OperationalPattern	16 octets	UUID (UL)
EssenceContainers	variable	Batch-UL

Batch UL = Nombre d'éléments (4 octets) + Taille d'un élément (4 octets) + (Nombre d'élément * UL (16 octets))

ETUDE RAPIDE DE L'UNIVERSAL LABEL DE PARTITION PACK

```

UL = 06.0E.2B.34.02.05.01.vv.0D.01.02.01.01.xx.yy.00
    ^^----- Item Designator : Organizationally registered
      ^^----- Organization : AAF
        ^^----- Application : MXF File Structure
          ^^----- Structure Version : Version 1
            ^^----- Structure Kind : MXF File Structure Sets & Packs
              ^^----- Set/Pack Kind : (see below)
                ^^----- Partition Status : (see below)

```