

MXF : D-BOX : LE CINÉMA DYNAMIQUE SIMPLE, LES SIÈGES MOUVANTS, LA GERBE AU TOURNANT

Références	D-BOX - Technical Note - 124-915-0001-A05 ²⁰¹⁷ - D-BOX in SMPTE/DCI DCP D-BOX - Technical Note - 124-915-0002-A06 ²⁰¹⁷ - D-Cinema Distribution Requirements D-BOX - Technical Note - 124-915-0005-A02 ²⁰¹⁷ - Signaling D-BOX in a CPL
Modèle KLV	Local Sets (pour les descriptors)
Universal Label	<p>060e2b34.02530101.0d010101.01014800 - Wave Audio Essence Descriptor 060e2b34.02530101.0d010101.01016c00 - Multichannel Audio (MCA) Soundfield Group Label Sub-Descriptor 060e2b34.02530101.0d010101.01016b00 - Multichannel Audio (MCA) Channel Label Sub-Descriptor</p> <p>060e2b34.01020101.0d010301.16010101 - Sound Essence (non-chiffré) 060e2b34.02040101.0d010301.027e0100 - Encrypted Essence</p> <p>060e2b34.0401010d.0e160101.01010201 - D-BOX Motion Code Primary Stream 060e2b34.0401010d.0e160101.01010202 - D-BOX Motion Code Secondary Stream</p> <p>060e2b34.0401010d.0e160101.01010101 - D-BOX Motion Channel Configuration D1 (obsolète) 060e2b34.0401010d.0e160101.01010102 - D-BOX Motion Channel Configuration D2 (obsolète) 060e2b34.0401010d.0e160101.01010103 - D-BOX Motion Channel Configuration D3 (obsolète) 060e2b34.0401010d.0e160101.01010105 - D-BOX Motion Channel Configuration D5 (obsolète)</p>

PRÉFACE

Pour comprendre les différences majeures entre un MXF sonore classique et un MXF sonore avec DBox, reportez-vous déjà au chapitre [Configuration Audio & Multichannel Audio \(MCA\)](#) pour déjà comprendre le mécanisme des MXF Audio.

En résumé : les données D-BOX sont des compléments intégrés dans les MXF Audio, stockés sur une piste spécifique à côté des autres pistes sonores du film.

Une ou deux pistes DBOX ?

Il y a deux pistes DBOX normées (Primary & Secondary Stream) mais par simplicité dans la compréhension dans ce chapitre, je ne parlerai que d'une piste.

LES CONFIGURATIONS STATIQUES & DYNAMIQUES

Tout comme la configuration audio d'un MXF, vous aurez soit des **Channels Configurations Statiques**, soit des **Channels Configurations Dynamiques**.

D-BOX va réutiliser le même principe :

- **Pour les configurations statiques** : les données DBOX seront stockées sur une piste spécifique - définie au préalable dans les normes - et donc qui sera toujours la même.
- **Pour les configurations dynamiques** (ex. MCA) : les données DBOX seront stockées sur une piste spécifique qui sera définie dans un KLV [Multichannel Audio \(MCA\) Channel Label Sub-Descriptor](#) et qui indiquera que telle piste sera un canal DBOX.

LES CONFIGURATIONS STATIQUES

Les **Channel Configurations** définis dans la norme SMPTE 429-2 sont agrémentés des canaux **D-BOX Primary Stream** et **D-BOX Secondary Stream**.

D-BOX va utiliser les canaux non-utilisés pour s'intégrer.

#	Channel Config D1	Channel Config D2	Channel Config D3	Channel Config D4	Channel Config D5
1	L	L	L	CH1	L
2	R	R	R	CH2	R
3	C	C	C	CH3	C
4	LFE	LFE	LFE	CH4	LFE
5	Ls	Ls	Ls	CH5	Ls
6	Rs	Rs	Rs	CH6	Rs
7	HI	Cs	Lc	CH7	Rls
8	VIN	-	Rc	CH8	Rrs
9	D-BOX #1	HI	HI	CH9	HI
10	D-BOX #2	VIN	VIN	CH10	VIN
11	-	D-BOX #1	D-BOX #1	CH11	D-BOX #1
12	-	D-BOX #2	D-BOX #2	CH12	D-BOX #2
13-16	-	-	-	-	-
UL DBOX (obsolète)	06.0E.2B.34 04.01.01.0D 0E.16.01.01 01.01.01.01	06.0E.2B.34 04.01.01.0D 0E.16.01.01 01.01.01.02	06.0E.2B.34 04.01.01.0D 0E.16.01.01 01.01.01.03	-	06.0E.2B.34 04.01.01.0D 0E.16.01.01 01.01.01.05

Les pistes non utilisées (-) doivent contenir du silence.

La méthode statique sera utilisée principalement par les versions Interop. Les versions SMPTE utiliseront le mécanisme du [Multichannel Audio \(MCA\)](#), donc la configuration 4.

D-BOX et Interop ?

En Interop, nous n'aurons qu'un **Wave Audio Essence Descriptor** assez basique sans aucun item **Descriptors & Sub-Descriptors** et donc sans Soundfield Group Label Sub-Descriptor ni Audio Channel Label Sub-Descriptor.

Les pistes D-BOX se trouve en toute fin du fichier, dans les dernières pistes.

La norme D-Box indique que le signal **D-BOX Motion Code Primary Stream** doit être stocké - sans modification - dans le canal audio n°13. Et que le canal audio n°14 peut aussi des données D-BOX.

Sur un MXF Interop, seul le **D-BOX Motion Code Primary Stream** sera utilisé. Le secondaire semble ne pas être utilisé si on en croit la doc D-BOX.

Si aucun **Channel Assignment** dans **Wave Audio Essence Descriptor** par défaut, cela sera la **Channel Configuration 1**.

LES CONFIGURATIONS DYNAMIQUES

Dans notre KLV **Wave Audio Essence Descriptor**, l'item **Channel Assignment** sera à **Channel Configuration 4**

Nous n'utilisons pas l'Universal Label pour le MXF Multichannel Audio Framework (voir chapitre [Configuration Audio & Multichannel Audio \(MCA\)](#)) mais l'**Universal Label** pour la **Configuration 4**, dites ouverte.

Voici les différents KLV Audio que nous avons déjà vu dans le chapitre [Configuration Audio & Multichannel Audio \(MCA\)](#), le seul ajout sera un **Audio Channel Label Sub-Descriptor** en supplément pour notre DBOX :

```
060e2b34.02530101.0d010101.01014800 | Wave Audio Essence Descriptor
-----
3C0A - Instance UID | 67a4c301.85fd40c8.ad956f09.d8f9cab1
FFF9 - Descriptors & Sub-Descriptors | 9 item(s):
| - 5d37f679.37a04bfb.bdc1c2e5.d732ad4b (Soundfield Group Label Sub-Descriptor)
| - 0e371c28.821d440f.83fc4ef7.f663561b (Audio Channel Label Sub-Descriptor)
| - c37ebbb8.9ebc418f.a98edd0c.726856a7 (Audio Channel Label Sub-Descriptor)
| - 8efec86a.db944dd8.9a40d86c.d4409d7e (Audio Channel Label Sub-Descriptor)
| - f49ac040.cba944c5.9c3ebc86.1f98ffe1 (Audio Channel Label Sub-Descriptor)
| - a34424b7.4d7440cd.916dad63.68411470 (Audio Channel Label Sub-Descriptor)
| - 62a637a9.74334bd9.b6613e96.0fd272d4 (Audio Channel Label Sub-Descriptor)
```

	- 828d7c89.e0674e70.808a8703.134bb15a	(Audio Channel Label Sub-Descriptor)
	- 935ab599.f6384581.b9ddd65c.db46c41e	(Audio Channel Label Sub-Descriptor)
3006 - Linked Track ID	2	
3001 - Sample Rate	24/1	
3002 - Container Duration	3404	
3004 - Essence Container	060e2b34.04010101.0d010301.02060100	(Broadcast Wave audio - frame-based mapping)
3D03 - Audio sampling rate	48000/1	
3D02 - Locked/Unlocked	False	
3D04 - Audio Ref Level	0	
3D07 - ChannelCount	16	
3D01 - Quantization bits	24	
3D0C - Dial Norm	0	
3D0A - Block Align	48	
3D0B - Sequence Offset	0	
3D09 - Average Bytes Per Second	2304000	
3D32 - Channel Assignment	060e2b34.0401010b.04020210.03010400	(SMPTE-429-2 Channel Configuration 4)

060e2b34.02530101.0d010101.01016c00 | Soundfield Group Label Sub-Descriptor

3C0A - Instance UID	5d37f679.37a04bfb.bdc1c2e5.d732ad4b	
FFF8 - MCA Label Dictionary ID	060e2b34.0401010d.03020201.00000000	(Soundfield Group - 5.1 (L, C, R, Ls, Rs, LFE))
FFF7 - MCA Link ID	d5757361.c2924faf.ae8f5b46.4809aa66	
FFF6 - MCA Tag Symbol	sg51	
FFF5 - MCA Tag Name	5.1	
FFF4 - RFC 5646 Spoken Language	fr	
FFF3 - MCA Title		
FFF2 - MCA Title Version		
FFF1 - MCA Audio Content Kind		
FFF0 - MCA Audio Element Kind		

060e2b34.02530101.0d010101.01016b00 | Audio Channel Label Sub-Descriptor

3C0A - Instance UID	0e371c28.821d440f.83fc4ef7.f663561b	
FFF8 - MCA Label Dictionary ID	060e2b34.0401010d.03020101.00000000	(Audio Channel - Left (L))
FFF7 - MCA Link ID	f92da821.6326461c.a736e243.a0548fd4	
FFF6 - MCA Tag Symbol	chL	
FFF5 - MCA Tag Name	Left	
FFEF - MCA Channel ID	1	
FFF4 - RFC 5646 Spoken Language	fr	
FFEE - SoundfieldGroupLinkID	d5757361.c2924faf.ae8f5b46.4809aa66	

060e2b34.02530101.0d010101.01016b00 | Audio Channel Label Sub-Descriptor (Right)

3C0A - Instance UID	c37ebbb8.9ebc418f.a98edd0c.726856a7	
FFF8 - MCA Label Dictionary ID	060e2b34.0401010d.03020102.00000000	(Audio Channel - Right (R))
FFF7 - MCA Link ID	dde4944c.f02748e1.ab2e84ed.161b8e65	
FFF6 - MCA Tag Symbol	chR	
FFF5 - MCA Tag Name	Right	
FFEF - MCA Channel ID	2	
FFF4 - RFC 5646 Spoken Language	fr	
FFEE - SoundfieldGroupLinkID	d5757361.c2924faf.ae8f5b46.4809aa66	

060e2b34.02530101.0d010101.01016b00 | Audio Channel Label Sub-Descriptor (Center)

3C0A - Instance UID	8efec86a.db944dd8.9a40d06c.d4409d7e	
FFF8 - MCA Label Dictionary ID	060e2b34.0401010d.03020103.00000000	(Audio Channel - Center (C))
FFF7 - MCA Link ID	21c8259d.fa6e4bc7.9ebfb14b.4c3ceb0a	
FFF6 - MCA Tag Symbol	chC	
FFF5 - MCA Tag Name	Center	
FFEF - MCA Channel ID	3	
FFF4 - RFC 5646 Spoken Language	fr	
FFEE - SoundfieldGroupLinkID	d5757361.c2924faf.ae8f5b46.4809aa66	

060e2b34.02530101.0d010101.01016b00 | Audio Channel Label Sub-Descriptor (LFE)

3C0A - Instance UID	f49ac040.cba944c5.9c3ebc86.1f98ffe1	
FFF8 - MCA Label Dictionary ID	060e2b34.0401010d.03020104.00000000	(Audio Channel - LFE (LFE))
FFF7 - MCA Link ID	08262570.33b84c07.acbfc205.225a01a2	
FFF6 - MCA Tag Symbol	chLFE	
FFF5 - MCA Tag Name	LFE	
FFEF - MCA Channel ID	4	
FFF4 - RFC 5646 Spoken Language	fr	
FFEE - SoundfieldGroupLinkID	d5757361.c2924faf.ae8f5b46.4809aa66	

060e2b34.02530101.0d010101.01016b00 | Audio Channel Label Sub-Descriptor (Left Surround)

3C0A - Instance UID	a34424b7.4d7440cd.916dad63.68411470	
---------------------	-------------------------------------	--

```

FFF8 - MCA Label Dictionary ID | 060e2b34.0401010d.03020105.00000000 (Audio Channel - Left Surround (Ls))
FFF7 - MCA Link ID | 52151320.014147fa.8dacc209.2583cdcc
FFF6 - MCA Tag Symbol | chLs
FFF5 - MCA Tag Name | Left Surround
FFEF - MCA Channel ID | 5
FFF4 - RFC 5646 Spoken Language | fr
FFEE - SoundfieldGroupLinkID | d5757361.c2924faf.ae8f5b46.4809aa66

060e2b34.02530101.0d010101.01016b00 | Audio Channel Label Sub-Descriptor (Right Surround)

3C0A - Instance UID | 62a637a9.74334bd9.66613e96.0fd272d4
FFF8 - MCA Label Dictionary ID | 060e2b34.0401010d.03020106.00000000 (Audio Channel - Right Surround (Rs))
FFF7 - MCA Link ID | 1d500b9e.9a4c424c.b3f8eb2f.e950c51c
FFF6 - MCA Tag Symbol | chRs
FFF5 - MCA Tag Name | Right Surround
FFEF - MCA Channel ID | 6
FFF4 - RFC 5646 Spoken Language | fr
FFEE - SoundfieldGroupLinkID | d5757361.c2924faf.ae8f5b46.4809aa66

060e2b34.02530101.0d010101.01016b00 | Audio Channel Label Sub-Descriptor (Visually Impaired-Narrative)

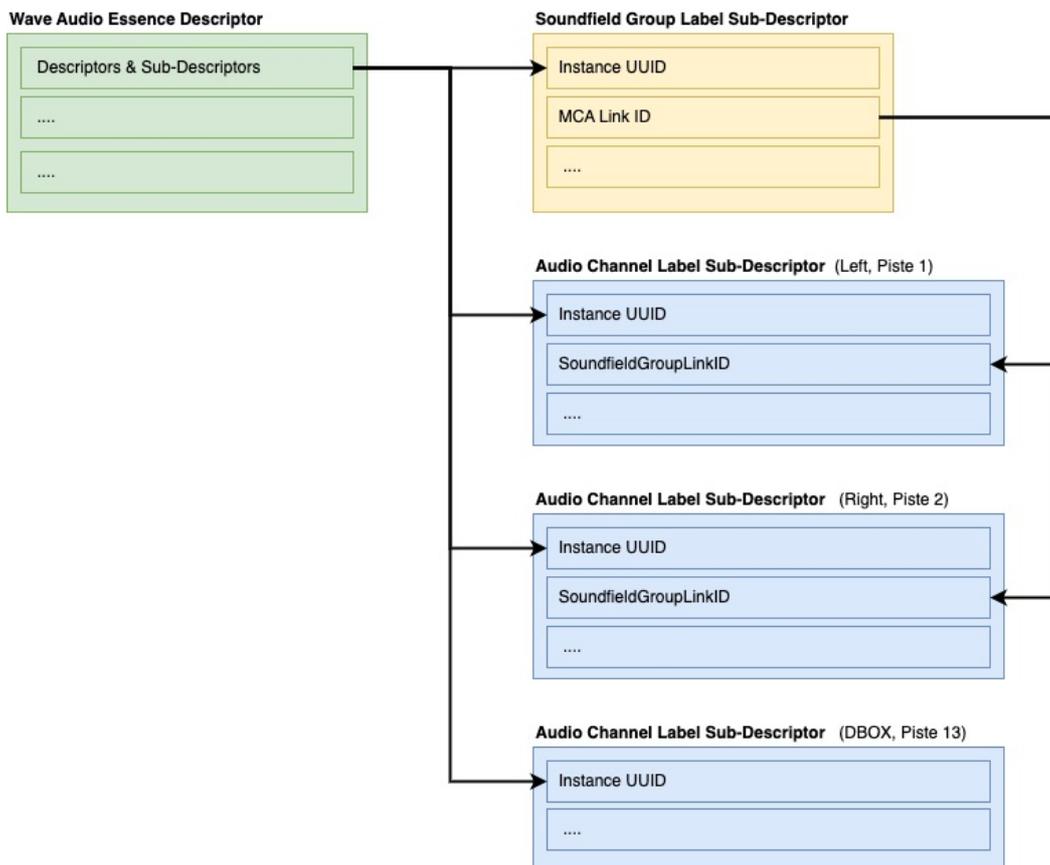
3C0A - Instance UID | 820d7c89.e0674e70.800a8703.134bb15a
FFF8 - MCA Label Dictionary ID | 060e2b34.0401010d.0302010f.00000000 (Audio Channel - Visually Impaired Narrative)
FFF7 - MCA Link ID | e104e863.56194948.91431863.a3623110
FFF6 - MCA Tag Symbol | chVIN
FFF5 - MCA Tag Name | Visually Impaired Narrative
FFEF - MCA Channel ID | 8
FFF4 - RFC 5646 Spoken Language | fr

060e2b34.02530101.0d010101.01016b00 | Audio Channel Label Sub-Descriptor (D-BOX)

3C0A - Instance UID | 935ab599.f6384581.b9ddd65c.db46c41e
FFF8 - MCA Label Dictionary ID | 060e2b34.0401010d.0e160101.01010201 (D-BOX Motion Code Primary Stream)
FFF7 - MCA Link ID | 4f664746.7d3544c2.8600f387.07a8cde6
FFF6 - MCA Tag Symbol | DBOX
FFF5 - MCA Tag Name | D-BOX Motion Code Primary Stream
FFEF - MCA Channel ID | 13
FFF4 - RFC 5646 Spoken Language | fr

```

En images, les différentes interconnexions des descripteurs, le dernier Sub-Descriptor est notre DBOX :



Le signal **D-BOX Motion Code Primary Stream** sera donc placé dans un des canaux sonores du fichier sonore principal défini dans la CPL par **MainSound**, et pour connaître sa position, il faudra lire chaque KLV **Audio Channel Label Sub-Descriptor**

pour déterminer celui qui sera un D-BOX Motion Code Stream, il faudra lire l'attribut **MCA Label Dictionary ID** :

Attribut	Universal Label	Description
MCA Label Dictionary ID	060e2b34.0401010d.0e160101.01010201	D-BOX Motion Code Primary Stream
MCA Label Dictionary ID	060e2b34.0401010d.0e160101.01010202	D-BOX Motion Code Secondary Stream

Comme cela, on pourrait se dire qu'on peut mapper les données utiles DBOX dans n'importe quel canal.

Cependant, comme vu dans le chapitre [Configuration Audio & Multichannel Audio \(MCA\)](#), paragraphe **Channel Layout / Soundfield Groups Configuration : le statique dynamique statique**, on ne peut pas vraiment faire n'importe quoi, DBOX va utiliser la piste **Motion Data** - qui se trouve sur le canal 13 ¹ - pour stocker les informations utiles à DBOX.

KDM

Aucun élément supplémentaire, la piste est DBOX est intégrée au MXF audio, il sera donc intégré au workflow cryptographique avec les clefs de type [MDAK](#).

Cependant, sur des très vieux systèmes, il peut être nécessaire de désactiver les watermarkings audios en utilisant les options `ForensicMarkFlag` et l'item `mrkflag-audio-disable` afin de désactiver au delà du canal 12.

Voir le chapitre [KDM AuthenticatedPublic](#), Paragraphe **ForensicMarkFlag**.

CODES

Ces exemples de codes marchent avec sur notre [MXF DBox INTEROP](#), 14 channels, 24 bits, 48 kHz. Il faudra adapter ces exemples suivant le profil du MXF en vous reportant à son KLV **Wave Audio Essence Descriptor** et de ses attributs **ChannelCount** et **Quantization bits**.

CRÉATION D'UN HEADER WAV 14 CHANNELS, 24 BITS, 48 KHZ

Les données dans les KLV sont au format BWF ([Broadcast Wave Format](#), un .wav si vous préférez), cependant, ils ne contiennent aucun header Wave, il faut donc en construire un, compatible avec les informations disponibles dans les KLV **Wave Audio Essence Descriptor**.

Dans notre MXF, nous avons 14 channels en 24 bits et 48.000 hz :

```
#!/usr/bin/env python3
import wave
with wave.open("header.wav", "wb") as file:
    file.setnchannels(14) # 14 channels
    file.setsampwidth(3) # bit-depth (3 bytes : 24 bits)
    file.setframerate(48000) # sampling-rate (48kHz)
```

EXTRACTION DES KLV WAVE-ESSENCE ET CONCATÉINATION POUR LA CRÉATION D'UN FICHER WAV

Nous allons extraire tous les KLV SoundEssence de notre [MXF DBOX INTEROP](#) à l'aide de `mxf-analyzer` et nous ajouterons un header WAVE [header-14ch-24bits-48khz.wav](#) pour en faire un fichier WAVE lisible :

```
# Extraction des KLV
$ mxf-analyzer -f "DBox_v1-Atmos-Synchronization-Signal_INTEROP.mxf" -x extract/

# Concaténation de tous les KLV SoundEssence
$ cat extract/*SoundEssenceWave*.value.bin > "wave-essences.raw"

# Concaténation du header WAV avec les données WAV
$ cat "header-14ch-24bits-48khz.wav" "wave-essences.raw" > wave-essences.wav
```

Dans [wave-essences.raw](#), vous n'aurez que les données brutes BWF sans header. Avec la concaténation d'un [header BWF](#), nous produisons un Wave lisible : vous aurez 14 pistes dans un seul et [unique fichier WAV](#) en 24 bits et 48 Khz.

Vous pouvez le lire dans un VLC. Par défaut, il lira les canaux Left et Right (channel 1 et 2).

EXTRACTION DES 14 PISTES :

METHODE 1 : EXTRACTION DES 14 PISTES DEPUIS LE FICHER WAV DU DESSUS AVEC FFMPEG

Nous allons maintenant extraire chaque channel depuis le fichier [wave-essences.wav](#) pour créer un fichier wav pour chaque piste :

```
ffmpeg \  
-i "wave-essences.wav" \  
-y -map_channel 0.0.0 -c:a pcm_s24le "ch01.wav" \  
-y -map_channel 0.0.1 -c:a pcm_s24le "ch02.wav" \  
-y -map_channel 0.0.2 -c:a pcm_s24le "ch03.wav" \  
-y -map_channel 0.0.3 -c:a pcm_s24le "ch04.wav" \  
-y -map_channel 0.0.4 -c:a pcm_s24le "ch05.wav" \  
-y -map_channel 0.0.5 -c:a pcm_s24le "ch06.wav" \  
-y -map_channel 0.0.6 -c:a pcm_s24le "ch07.wav" \  
-y -map_channel 0.0.7 -c:a pcm_s24le "ch08.wav" \  
-y -map_channel 0.0.8 -c:a pcm_s24le "ch09.wav" \  
-y -map_channel 0.0.9 -c:a pcm_s24le "ch10.wav" \  
-y -map_channel 0.0.10 -c:a pcm_s24le "ch11.wav" \  
-y -map_channel 0.0.11 -c:a pcm_s24le "ch12.wav" \  
-y -map_channel 0.0.12 -c:a pcm_s24le "ch13.wav" \  
-y -map_channel 0.0.13 -c:a pcm_s24le "ch14.wav"
```

Dans ce cas de figure, la piste 13 sera D-Box et la piste 14 sera un sync signal.

ffmpeg va séparer chaque canal et les insérer dans un fichier séparé. Nous forçons un `pcm_s24le` au cas où, pour bien indiquer à ffmpeg que le format de sortie sera en 24 bits.

METHODE 2 : EXTRACTION DES 14 PISTES DEPUIS LE FICHER RAW KLV AVEC NOTRE PROPRE PROGRAMME

Autre méthode d'extraction sans passer par ffmpeg, nous pouvons extraire les différentes pistes depuis notre concaténation de tous les KLV SoundEssence.

Pour cela, nous n'allons utiliser que notre fichier [wave-essences.raw](#) qui est une simple concaténation de tous les KLV SoundEssence sans aucune modification.

COMMENT SONT STOCKÉS NOS DONNÉES DANS NOS KLV :

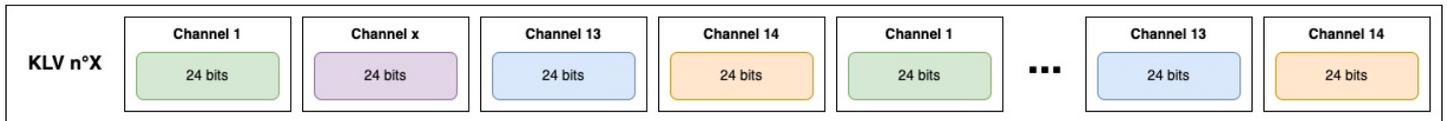
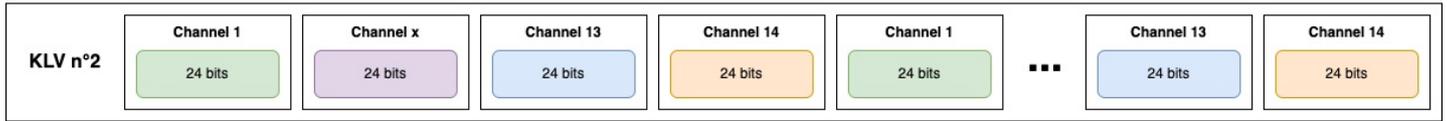
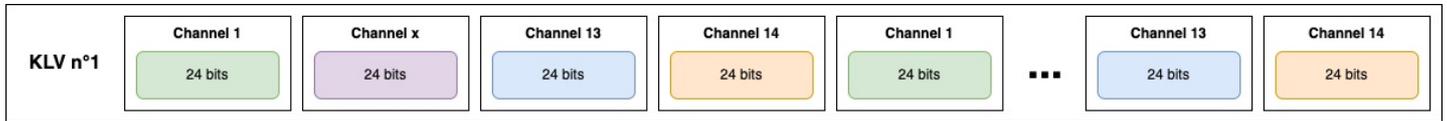
Si vous avez déjà lu le chapitre [MXF - KLV - Sound](#), vous connaissez déjà un peu le principe.

Dans notre exemple, notre **Wave Audio Essence Descriptor** indique un **Quantization Bits** en 24 bits :

```
060e2b34.02530101.0d010101.01014800 | Wave Audio Essence Descriptor  
||  
3C0A - Instance UID || ac7e6e98.037b44bf.b27452cc.16359c38  
3006 - Linked Track ID || 2  
3001 - Sample Rate || 24/1  
3002 - Container Duration || 720  
3004 - Essence Container || 060e2b34.04010101.0d010301.02060100 (Broadcast Wave audio - frame-based r  
3D03 - Audio sampling rate || 48000/1  
3D02 - Locked/Unlocked || False  
3D07 - ChannelCount || 14  
3D01 - Quantization bits || 24 <==== ICI  
3D06 - Sound Essence Coding || 00000000000000000000000000000000  
3D0A - Block Align || 42  
3D09 - Average Bytes Per Second || 2016000  
||
```

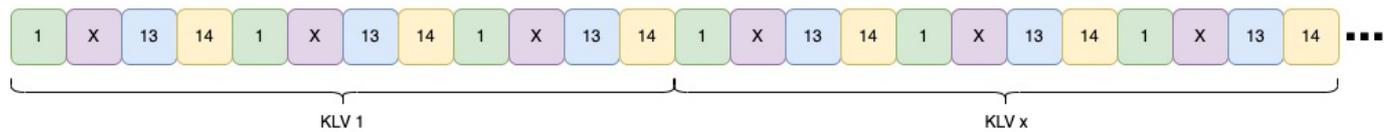
Nous savons donc que chaque portion (sample) de piste sera de 24 bits, soit 3 octets. Si nous lisons les 3 premiers octets, nous aurons un premier sample de la piste 1, les 3 octets suivants, un premier sample de la piste 2 et ainsi de suite jusqu'à la piste 14, puis nous reviendrons à la piste 1 et nous passerons au sample 2 de la piste 1.

Dans notre exemple ci-dessous, nous voyons les 3 premiers octets du KLV n°1 qui constitue notre premier sample pour notre Channel 1 :



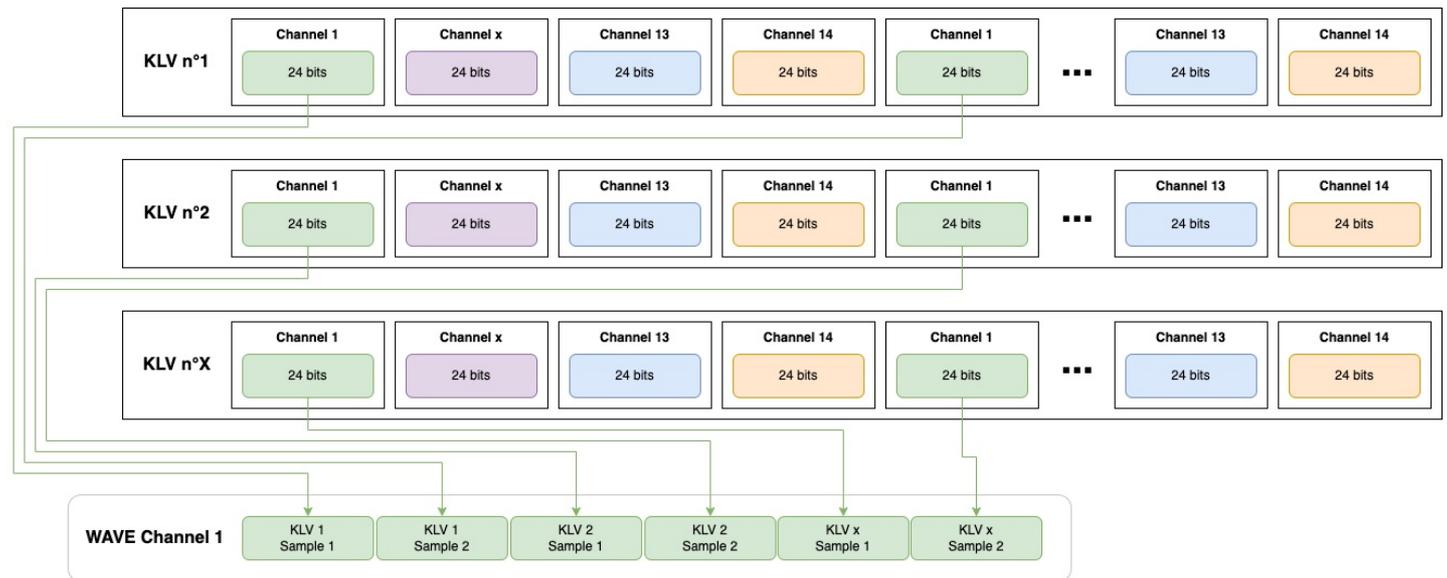
Dans le KLV n°1, vous aurez les 3 premiers octets du channel 1, puis on passe au 3 octets du channel 2, et ainsi de suite pour faire un bloc KLV, puis on passe au bloc KLV suivant qui sera la suite directe du précédent KLV.

Si on concatène tous les KLV Essence, nous aurions ceci :



Pour des raisons de lecture, les KLV sont simplifiés à mort :)

Pour reconstruire chaque channel, il suffit de prendre les bons blocs de données dans chaque KLV :



Voyez cela comme une distribution de cartes durant une partie de jeu de cartes : vous avez un jeu de cartes et vous devez distribuer toutes les cartes à tous les participants au jeu : vous donnez la première carte au 1er joueur, la seconde au 2eme joueur et ainsi de suite jusqu'au dernier joueur, puis vous revenez au premier joueur et continuez jusqu'à ce que vous n'ayez plus de cartes dans votre main.

Maintenant qu'on a compris le principe, il suffit de faire un parseur qui va lire tous les 24 bits et les dispatcher dans chaque channel :

```
#!/usr/bin/python3

# create all output handler
output = {}
for i in range(0, 14):
    output[i] = open("ch%02d.raw" % (i+1), "wb")

with open("wave-essences.raw", "rb") as file:
    index = 0
    while True:
        content = file.read(3) # read 24 bits
        output[index].write(content) # write in specific channel file
        if not content:
            break
        index = (index + 1) % 14

# close all output handler
for i in range(0, 14):
    output[i].close()
```

Au tout début, on va créer nos différents handlers pour nos 14 fichiers channels. Puis, on ouvre le fichier [wave-essences.raw](#) (concaténation brute de tous les KLV Essence), on va lire par portion (sample) de 24 bits tout le fichier et écrire ces samples dans chaque fichier channels.

Le résultat : 14 fichiers contenant des données au format WAV **mais** sans entêtes WAV.

On peut rajouter un header simplement en concaténant un [simple fichier header WAV](#) avec des métadonnées pour 1 channel, 24 bits et 48 kHz:

```
cat "header-1ch-24bits-48khz.wav" "ch01.raw" > ch01.wav
```

Sinon, nous pouvons agrémenter notre précédent programme et rajouter la librairie wav pour créer directement les headers WAVE :

```
#!/usr/bin/python3

import wave

# create all output handler
output = {}
for i in range(0, 14):
    output[i] = wave.open("ch%02d.wav" % (i+1), "wb")
    output[i].setnchannels(1) # 1 channel
    output[i].setsampwidth(3) # bit-depth (3 bytes : 24 bits)
    output[i].setframerate(48000) # sampling-rate (48kHz)

with open("wave-essences.raw", "rb") as file:
    index = 0
    while True:
        content = file.read(3) # read 24 bits
        output[index].writeframesraw(content) # write in specific channel file
        if not content:
            break
        index = (index + 1) % 14

# close all output handler
for i in range(0, 14):
    output[i].close()
```

Les seules modifications seront sur les `setnchannels()`, `setsampwidth()`, `setframerate()` et l'utilisation de `writeframesraw()` à la place du classique `write()` (notez qu'on aurait pu utiliser juste le `write()` mais il aurait fallu clôturer d'abord les différents handlers ouverts avec `wave.open()` puis les rouvrir avec le classique `open()`, un peu lourd ... mais sur le principe `writeframesraw()` ne fait rien de plus qu'écrire les données brutes sans modifications, à l'exception de la taille des données dans le header, mais même sans, les players peuvent lire le fichier)

ANALYSE ET DONNÉES

Les données utiles :

- [Fichier MXF D-BOX + VIN + DBox MCA \(SMPTE\) - avec chiffrement](#) : C'est ce fichier qui nous sert d'exemple pour les analyses des métadonnées dans cette documentation (les données étant chiffrés, nous n'avons accès qu'aux métadonnées du MXF)
- [Fichier MXF D-BOX + Atmos Sync Signal \(INTEROP\) - sans chiffrement](#) : C'est ce fichier qui nous sert pour les analyses des

données (les données n'étant pas chiffrés, les données sont stockées dans les KLV SoundEssence)

- **Fichier RAW KLV** : Extraction des données venant des KLV SoundEssence du fichier INTEROP, nous avons donc toutes les pistes audios, le données brutes, sans aucun traitement (sans header WAV par exemple)
- **Fichier WAV KLV** : Fichier au dessus avec en plus, un header WAV 24bits - 14 channels - 48khz. Le fichier est jouable avec un **VLC** ou sur l'**analyseur spectral en ligne**.
- **Fichier WAV - Piste 13** : La piste 13 du fichier WAV KLV (sa version **RAW**)
- **Fichier WAV - Piste 14** : La piste 14 du fichier WAV KLV (sa version **RAW**)
- **Fichier Header WAVE** : 1 channel, 24 bits, 48 kHz
- **Fichier Header WAVE** : 14 channels, 24 bits, 48 kHz
- Le repository des données de ce chapitre [ici](#).

Les analyseurs :

- Analyseur spectral en ligne : <https://www.dcode.fr/spectral-analysis>
- Spek — Acoustic spectrum analyser (opensource) : <https://www.spek.cc/>

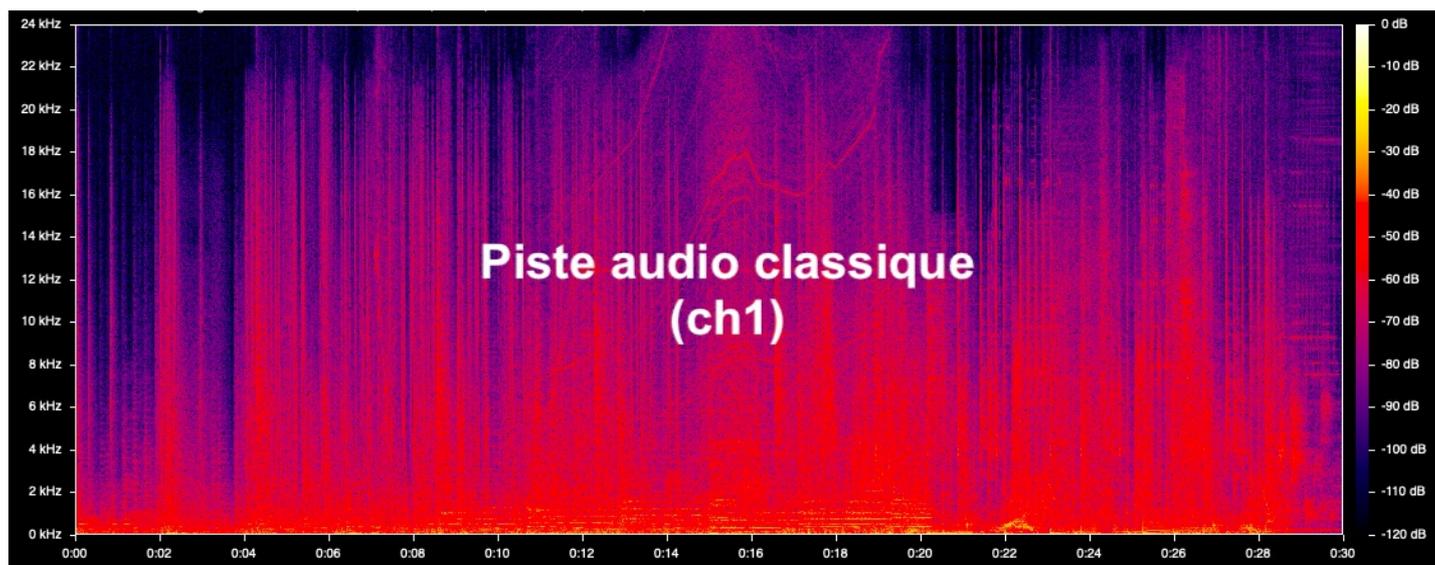
ANALYSE DES DONNÉES D-BOX

Ce paragraphe est en cours d'analyse

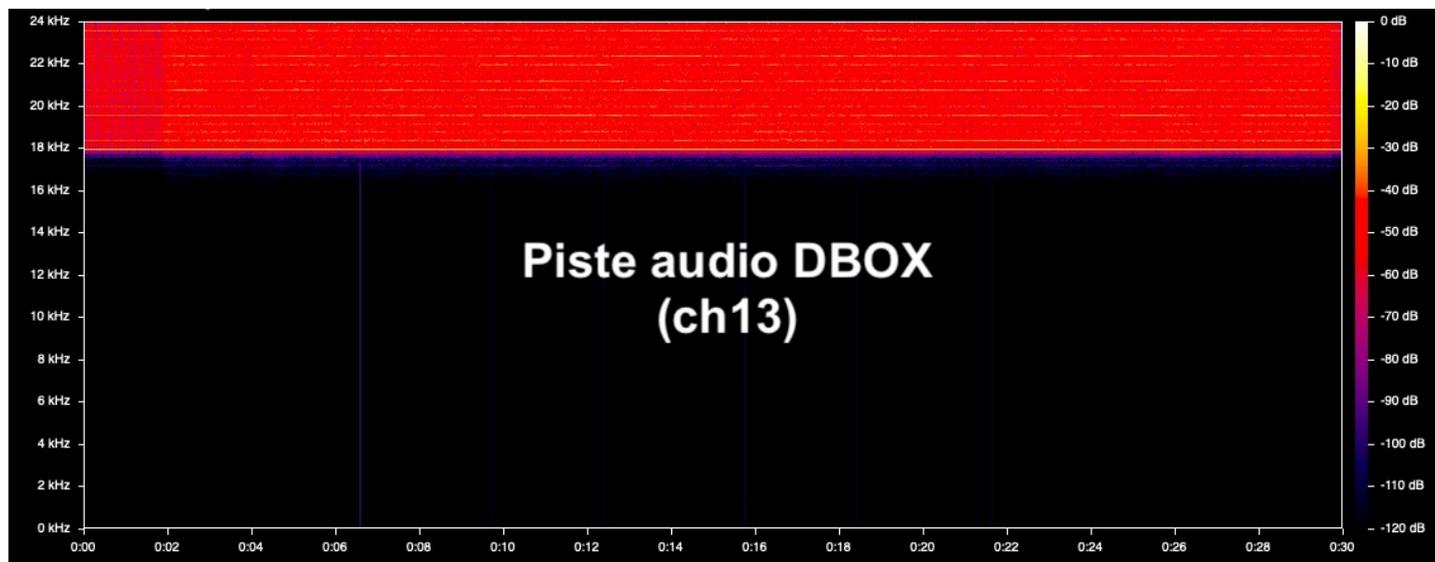
Les données MXF intégrant une piste DBOX sont en encore à l'état d'étude préliminaire

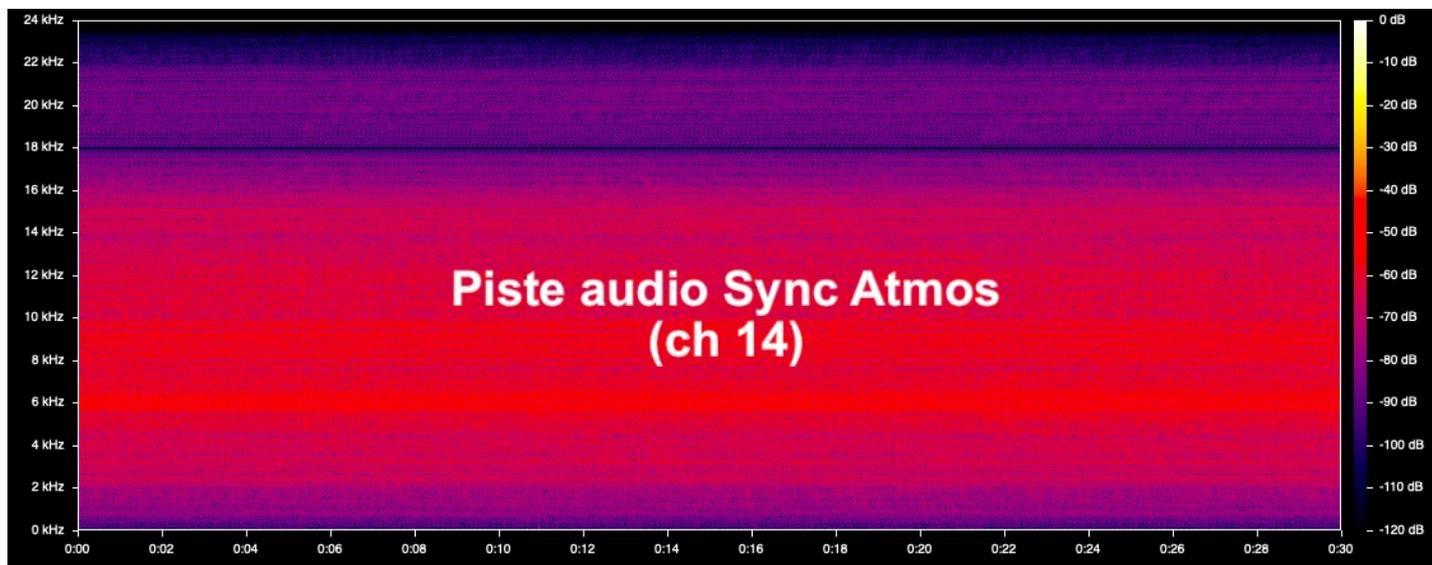
Voici l'extraction et l'analyse spectral sur un DCP **Interop** (donc sans **Multichannel Audio (MCA)**, les canaux sont donc fixes) :

Channel 1 : Le canal sonore Left :



Channel 13 : Les données liées à D-BOX :





Premières analyses : L'analyse spectro de données n'a peut-être pas de sens car ce ne sont pas forcément des données "sonores" mais peut-être des données bruts (binaires) qui sont envoyées par le canal sonore puis reconverties en données binaires par le DBOX Controller. Il se peut également que la piste DBOX du MXF ne fournisse pas les données utiles (exemple déplacement siège) mais seulement une piste de synchro entre le MXF et le matériel DBOX; et donc que les données utiles pour les mouvements soient stockées autre part (par exemple dans la boîte noire DBOX). En discutant avec un propriétaire de DBOX Home, il se peut que les playlists des évènements se trouvent uniquement dans la boîte DBOX et que ce dernier puisse piocher sur les serveurs DBOX.

NOTES

- Selon un propriétaire de DBOX Home interrogé, il existe un décalage entre DBOX et images venant du projecteur DCI, à priori, ce décalage serait de 2 frames (80ms)
1. « *The D-BOX Motion Code Primary Stream signal shall be stored without modification in audio channel 13 of the Main Sound Track File.* » - D-BOX in SMPTE/DCI DCP, Technical Note 124-915-0001-A05 ↩