

 eBook Gratuit

# APPRENEZ video

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#video

# Table des matières

À propos.....	1
<b>Chapitre 1: Commencer avec la vidéo.....</b>	<b>2</b>
Remarques.....	2
Exemples.....	2
Comprendre les fichiers multimédia autonomes.....	2
<b>Général.....</b>	<b>3</b>
<b>Vidéo.....</b>	<b>3</b>
<b>l'audio.....</b>	<b>5</b>
<b>Résumé de l'analyse.....</b>	<b>5</b>
<b>Plus.....</b>	<b>6</b>
<b>Chapitre 2: Comprendre une présentation multimédia.....</b>	<b>7</b>
Remarques.....	7
Exemples.....	7
Comprendre les fichiers multimédia autonomes.....	7
<b>Général.....</b>	<b>8</b>
<b>Vidéo.....</b>	<b>8</b>
<b>l'audio.....</b>	<b>10</b>
<b>Résumé de l'analyse.....</b>	<b>10</b>
<b>Plus.....</b>	<b>11</b>
Comprendre les composants de pistes vidéo.....	11
Comprendre les présentations en continu adaptatives DASH.....	14
<b>Chapitre 3: Ratios d'aspect vidéo.....</b>	<b>19</b>
Remarques.....	19
Exemples.....	19
Afficher les proportions (DAR).....	19
Ratio d'aspect d'image (PAR).....	20
Ratio d'aspect (SAR).....	20
Pixel aspect ratio.....	21
<b>Crédits.....</b>	<b>22</b>

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [video](#)

It is an unofficial and free video ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official video.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapitre 1: Commencer avec la vidéo

## Remarques

La lecture vidéo occupe une place centrale dans un large éventail de solutions modernes, les logiciels et les normes évoluant rapidement. Pour comprendre ce domaine, vous devez d'abord comprendre les multiples aspects de la vidéo:

- Les informations de couleur brute capturées dans le monde physique sont généralement *codées* à l'aide d'un codec - un algorithme dont le but est de représenter ces données sous une forme compressée, sacrifiant souvent des détails visuels en faveur d'une plus grande compression.
- Pour la lecture, l'algorithme inverse est exécuté - les données sont *décodées* pour redevenir des informations de couleur brute pouvant être fournies à un périphérique de sortie (par exemple un moniteur).
- Entre codage et décodage, les données compressées sont *regroupées* pour le stockage, ce qui peut impliquer la combinaison de pistes de différents types en un seul fichier ou la segmentation du contenu en un grand nombre de petits segments.
- La vidéo est *livrée* au périphérique de l'utilisateur final à l'aide d'une technologie de livraison, qui peut être aussi simple qu'un téléchargement de fichier via HTTP ou beaucoup plus complexe, impliquant un retour en direct de l'infrastructure réseau et une adaptation automatique des niveaux de qualité.
- Le contenu Premium est généralement *crypté* avant l'emballage et ne peut être lu que sur un lecteur équipé d'une technologie DRM qui garantit la sécurité de la clé de décryptage pendant l'utilisation et protège activement contre la capture de sortie.

Bien que la partie visuelle soit clairement dominante dans la visibilité, l'audio et le texte jouent également un rôle clé dans les présentations multimédias, fournissant des fonctionnalités multilingues qui rendent le contenu accessible à un large public. Dans la plupart des flux de travail, les pistes audio et texte sont traitées de manière équivalente aux pistes vidéo, codées, décodées, empaquetées et livrées selon les mêmes lignes.

Tous ces aspects - et plus encore - doivent être pris en compte dans une solution moderne, garantissant une expérience agréable aux utilisateurs finaux.

## Exemples

### Comprendre les fichiers multimédia autonomes

Le contenu de l'échantillon utilisé ici est Tears of Steel, de Blender Foundation. Plus précisément, nous utiliserons le téléchargement intitulé "HD 720p (~ 365 Mo, mov, 2.0)". Ceci est un fichier unique qui se termine par l'extension "mov" et jouera dans à peu près n'importe quel lecteur multimédia moderne.

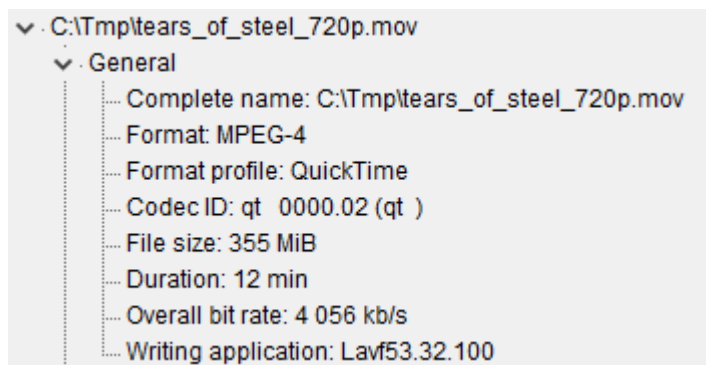
Notez que la page de téléchargement propose des sous-titres en tant que téléchargements de

fichiers SRT distincts. Dans cet exemple de contenu, il n'y a pas de sous-titres livrés ensemble dans le même fichier. Nous laissons donc l'analyse des sous-titres hors de la portée de cet exemple.

Un moyen simple d'analyser divers fichiers multimédias consiste à utiliser l'outil / la bibliothèque [MedialInfo](#) . Bien que la fonctionnalité d'analyse présentée ici utilise l'interface graphique pour plus de simplicité, toutes les fonctionnalités sont également disponibles via l'API MedialInfo.

En ouvrant ce fichier dans l'interface graphique de MedialInfo et en passant à l'arborescence, vous verrez trois sections: Général, Vidéo et Audio. Le premier contient des informations de base sur le fichier, tandis que les deux autres décrivent chacun une piste multimédia trouvée dans ce fichier. Examinons les informations les plus pertinentes dans chaque section du résultat.

## Général



Les premiers paramètres d'intérêt sont le *profil Format* et *Format* . Le premier indique que le **format d'emballage** provient de la suite de normes MPEG-4. MPEG-4 définit le *format de fichier multimédia de base ISO* et le *format d'emballage MP4* . De plus, Apple a créé sa propre spécification qui en dérive, nommée dans MedialInfo comme profil "QuickTime".

Remarque: veuillez à ne pas confondre MP4 et MPEG-4 - le premier fait référence à un format d'emballage spécifique dans la suite de normes internationales MPEG-4, qui inclut également les codecs vidéo et audio. Cela peut entraîner de la confusion, évitez donc d'utiliser le terme MPEG-4 lorsque vous vous référez à autre chose que l'ensemble complet de normes.

Tous les formats d'emballage basés sur le format de fichier multimédia de base ISO, défini dans la famille de normes MPEG-4, sont très similaires et peuvent souvent être traités par les mêmes outils. ignoré. Ainsi, nous pouvons nous attendre à ce que l'exemple de vidéo ici soit hautement compatible avec tous les lecteurs vidéo modernes.

## Vidéo

```
Video
  ID: 1
  Format: AVC
  Format/Info: Advanced Video Codec
  Format profile: Main@L3.1
  Format settings, CABAC: No
  Format settings, ReFrames: 2 frames
  Format settings, GOP: M=4, N=18
  Codec ID: avc1
  Codec ID/Info: Advanced Video Coding
  Duration: 12 min
  Bit rate: 4 000 kb/s
  Width: 1 280 pixels
  Height: 534 pixels
  Display aspect ratio: 2.40:1
  Frame rate mode: Constant
  Frame rate: 24.000 FPS
  Color space: YUV
  Chroma subsampling: 4:2:0
  Bit depth: 8 bits
  Scan type: Progressive
  Bits/(Pixel*Frame): 0.244
  Stream size: 338 MiB (95%)
  Writing library: x264 core 122
  Encoding settings: cabac=0 / ref=2 / deblock=1:0:0 / ar
  Language: English
```

Le détail le plus crucial de la piste vidéo est le codec utilisé pour transformer les données de couleur brutes en une forme compressée. Le nom du codec est fourni par le paramètre *Format*.

AVC est également connu sous le nom de H.264 et c'est le codec vidéo le plus répandu actuellement, supporté sur pratiquement tous les appareils et plates-formes logicielles modernes. Une piste vidéo encodée avec AVC est sûre de jouer sur presque tous les lecteurs.

Les codecs ont souvent plusieurs *profils* qui permettent de diviser les fonctionnalités du codec en niveaux, ce qui permet une évolution de la technologie contrôlée. Le paramètre *Format profile* indique que cette vidéo utilise le profil principal. Ce profil est relativement rare, car à peu près tous les appareils modernes prennent en charge le profil élevé, qui offre une plus grande efficacité de compression.

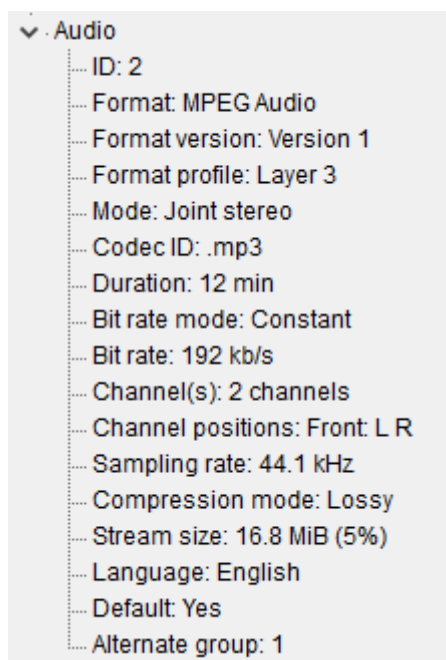
La qualité de la piste vidéo est souvent d'une importance primordiale. Nous voyons ici les facteurs critiques exprimés par les paramètres *Débit binaire*, *Largeur* et *Hauteur*. Ces deux derniers indices suggèrent qu'il s'agit d'une piste vidéo 720p, considérée comme une qualité HD inférieure. L'image est en réalité plus courte verticalement que le cadre standard 720p de 1280x720 pixels.

Le débit binaire mesure la quantité de données que la forme compressée du flux vidéo occupe, en moyenne, pour chaque seconde de lecture. C'est un paramètre crucial pour l'optimisation, car la quantité de données livrées est une source majeure de coût dans les solutions vidéo à grande échelle.

Les points de données ci-dessus sur la qualité vidéo sont simplement des faits que nous obtenons de l'analyse - tout jugement sur la pertinence de ces paramètres est un sujet qui nécessiterait

beaucoup plus d'analyses travailler avec des pistes vidéo.

## l'audio



Encore une fois, connaître le codec utilisé pour encoder les données audio revêt une importance cruciale. Ceci est exprimé par les paramètres de *profil Format* et *Format*. "MPEG Audio Layer 3" est plus connu sous le nom de MP3 et il s'agit d'un format audio universellement pris en charge auquel on peut s'attendre à jouer partout.

Comme pour la vidéo, les paramètres de qualité audio sont les deuxièmes points de données les plus importants, exprimés principalement par le paramètre *Débit binaire*.

## Résumé de l'analyse

Le contenu est emballé en utilisant un format d'emballage très populaire, basé sur la suite de normes MPEG-4. Il est codé à l'aide de codecs vidéo et audio universellement adoptés. De ce fait, il est clair que la vidéo est destinée à être facilement accessible à tous les spectateurs - la compatibilité et la disponibilité étaient essentielles pour ses auteurs.

L'utilisation de MP3 montre l'âge du contenu de l'exemple, car il n'est plus considéré comme comparable aux concurrents modernes - au lieu de cela, AAC (Advanced Audio Coding) est le soutien de la famille des codecs audio.

On peut en dire autant de l'utilisation du profil principal H.264. Il est très rare qu'un profil H.264 en dehors de High soit utilisé, étant donné que presque tous les décodeurs le prennent en charge, permettant à chacun de tirer parti de l'efficacité accrue qu'offrent les fonctionnalités de profil élevé.

Les débits utilisés sont légèrement supérieurs aux prévisions pour l'environnement actuel. Cela peut s'expliquer par le désir des auteurs de qualité ou simplement par les limitations des

encodeurs disponibles lors de la création du contenu.

---

## Plus

Les autres outils utiles pour l'analyse des fichiers multimédias sont [FFprobe](#) , qui fait partie du progiciel FFmpeg, et les [outils Bento4](#) pour travailler avec les fichiers MP4. Les deux sont également disponibles sous forme de bibliothèque. Ils sont capables d'une analyse de bas niveau supérieure à celle de MediaInfo, dans les situations où vous devez examiner des éléments individuels constituant des fichiers multimédias.

Lire Commencer avec la vidéo en ligne: <https://riptutorial.com/fr/video/topic/5690/commencer-avec-la-video>



---

# Chapitre 2: Comprendre une présentation multimédia

## Remarques

La lecture vidéo occupe une place centrale dans un large éventail de solutions modernes, les logiciels et les normes évoluant rapidement. Pour comprendre ce qui constitue une présentation multimédia, vous devez d'abord comprendre les multiples aspects de la vidéo:

- Les informations de couleur brute capturées dans le monde physique sont généralement *codées à l'aide* d'un codec - un algorithme dont le but est de représenter ces données sous une forme compressée, sacrifiant souvent des détails visuels en faveur d'une plus grande compression.
- Pour la lecture, l'algorithme inverse est exécuté - les données sont *décodées* pour redevenir des informations de couleur brute pouvant être fournies à un périphérique de sortie (par exemple un moniteur).
- Entre codage et décodage, les données compressées sont *regroupées* pour le stockage, ce qui peut impliquer la combinaison de pistes de différents types en un seul fichier ou la segmentation du contenu en un grand nombre de petits segments.
- La vidéo est *livrée* au périphérique de l'utilisateur final à l'aide d'une technologie de livraison, qui peut être aussi simple qu'un téléchargement de fichier via HTTP ou beaucoup plus complexe, impliquant un retour en direct de l'infrastructure réseau et une adaptation automatique des niveaux de qualité.
- Le contenu Premium est généralement *crypté* avant l'emballage et ne peut être lu que sur un lecteur équipé d'une technologie DRM qui garantit la sécurité de la clé de décryptage pendant l'utilisation et protège activement contre la capture de sortie.

Bien que la partie visuelle soit clairement dominante dans la visibilité, l'audio et le texte jouent également un rôle clé dans les présentations multimédias, fournissant des fonctionnalités multilingues qui rendent le contenu accessible à un large public. Dans la plupart des flux de travail, les pistes audio et texte sont traitées de manière équivalente aux pistes vidéo, codées, décodées, empaquetées et livrées selon les mêmes lignes.

Tous ces aspects - et plus encore - ont un impact sur la composition, le formatage et l'utilisation d'une présentation multimédia. Ils doivent être compris dans leur ensemble afin d'utiliser efficacement les technologies des médias.

## Exemples

### Comprendre les fichiers multimédia autonomes

Le contenu de l'échantillon utilisé ici est [Tears of Steel](#), de Blender Foundation. Plus précisément, nous utiliserons le téléchargement intitulé "HD 720p (~ 365 Mo, mov, 2.0)". Ceci est un fichier unique qui se termine par l'extension "mov" et jouera dans à peu près n'importe quel lecteur

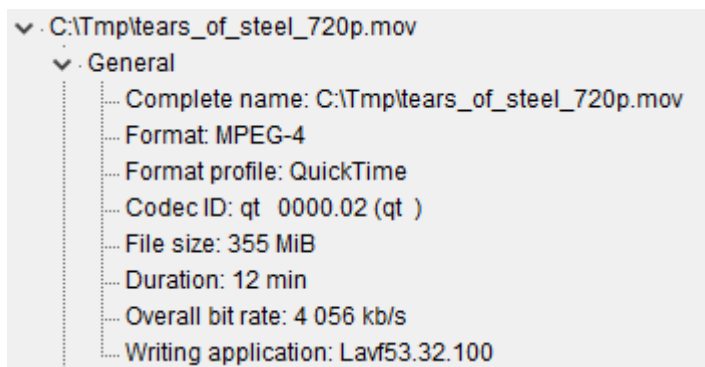
multimédia moderne.

Notez que la page de téléchargement propose des sous-titres en tant que téléchargements de fichiers SRT distincts. Dans cet exemple de contenu, il n'y a pas de sous-titres livrés ensemble dans le même fichier. Nous laissons donc l'analyse des sous-titres hors de la portée de cet exemple.

Un moyen simple d'analyser divers fichiers multimédias consiste à utiliser l'outil / la bibliothèque [MediaInfo](#) . Bien que la fonctionnalité d'analyse présentée ici utilise l'interface graphique pour plus de simplicité, toutes les fonctionnalités sont également disponibles via l'API MediaInfo.

En ouvrant ce fichier dans l'interface graphique de MediaInfo et en passant à l'arborescence, vous verrez trois sections: Général, Vidéo et Audio. Le premier contient des informations de base sur le fichier, tandis que les deux autres décrivent chacun une piste multimédia trouvée dans ce fichier. Examinons les informations les plus pertinentes dans chaque section du résultat.

## Général



Les premiers paramètres d'intérêt sont le *profil Format* et *Format* . Le premier indique que le **format d'emballage** provient de la suite de normes MPEG-4. MPEG-4 définit le *format de fichier multimédia de base ISO* et le *format d'emballage MP4* . De plus, Apple a créé sa propre spécification qui en dérive, nommée dans MediaInfo comme profil "QuickTime".

Remarque: veillez à ne pas confondre MP4 et MPEG-4 - le premier fait référence à un format d'emballage spécifique dans la suite de normes internationales MPEG-4, qui inclut également les codecs vidéo et audio. Cela peut entraîner de la confusion, évitez donc d'utiliser le terme MPEG-4 lorsque vous vous référez à autre chose que l'ensemble complet de normes.

Tous les formats d'emballage basés sur le format de fichier multimédia de base ISO, défini dans la famille de normes MPEG-4, sont très similaires et peuvent souvent être traités par les mêmes outils. ignoré. Ainsi, nous pouvons nous attendre à ce que l'exemple de vidéo ici soit hautement compatible avec tous les lecteurs vidéo modernes.

## Vidéo

```
Video
  ID: 1
  Format: AVC
  Format/Info: Advanced Video Codec
  Format profile: Main@L3.1
  Format settings, CABAC: No
  Format settings, ReFrames: 2 frames
  Format settings, GOP: M=4, N=18
  Codec ID: avc1
  Codec ID/Info: Advanced Video Coding
  Duration: 12 min
  Bit rate: 4 000 kb/s
  Width: 1 280 pixels
  Height: 534 pixels
  Display aspect ratio: 2.40:1
  Frame rate mode: Constant
  Frame rate: 24.000 FPS
  Color space: YUV
  Chroma subsampling: 4:2:0
  Bit depth: 8 bits
  Scan type: Progressive
  Bits/(Pixel*Frame): 0.244
  Stream size: 338 MiB (95%)
  Writing library: x264 core 122
  Encoding settings: cabac=0 / ref=2 / deblock=1:0:0 / ar
  Language: English
```

Le détail le plus crucial de la piste vidéo est le codec utilisé pour transformer les données de couleur brutes en une forme compressée. Le nom du codec est fourni par le paramètre *Format*.

AVC est également connu sous le nom de H.264 et c'est le codec vidéo le plus répandu actuellement, supporté sur pratiquement tous les appareils et plates-formes logicielles modernes. Une piste vidéo encodée avec AVC est sûre de jouer sur presque tous les lecteurs.

Les codecs ont souvent plusieurs *profils* qui permettent de diviser les fonctionnalités du codec en niveaux, ce qui permet une évolution de la technologie contrôlée. Le paramètre *Format profile* indique que cette vidéo utilise le profil principal. Ce profil est relativement rare, car à peu près tous les appareils modernes prennent en charge le profil élevé, qui offre une plus grande efficacité de compression.

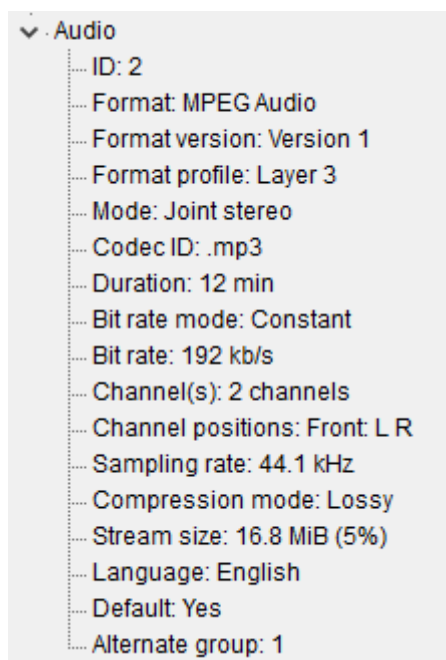
La qualité de la piste vidéo est souvent d'une importance primordiale. Nous voyons ici les facteurs critiques exprimés par les paramètres *Débit binaire*, *Largeur* et *Hauteur*. Ces deux derniers indices suggèrent qu'il s'agit d'une piste vidéo 720p, considérée comme une qualité HD inférieure. L'image est en réalité plus courte verticalement que le cadre standard 720p de 1280x720 pixels.

Le débit binaire mesure la quantité de données que la forme compressée du flux vidéo occupe, en moyenne, pour chaque seconde de lecture. C'est un paramètre crucial pour l'optimisation, car la quantité de données livrées est une source majeure de coût dans les solutions vidéo à grande échelle.

Les points de données ci-dessus sur la qualité vidéo sont simplement des faits que nous obtenons de l'analyse - tout jugement sur la pertinence de ces paramètres est un sujet qui nécessiterait

beaucoup plus d'analyses travailler avec des pistes vidéo.

## l'audio



Encore une fois, connaître le codec utilisé pour encoder les données audio revêt une importance cruciale. Ceci est exprimé par les paramètres de *profil Format* et *Format*. "MPEG Audio Layer 3" est plus connu sous le nom de MP3 et il s'agit d'un format audio universellement pris en charge auquel on peut s'attendre à jouer partout.

Comme pour la vidéo, les paramètres de qualité audio sont les deuxièmes points de données les plus importants, exprimés principalement par le paramètre *Débit binaire*.

## Résumé de l'analyse

Le contenu est emballé en utilisant un format d'emballage très populaire, basé sur la suite de normes MPEG-4. Il est codé à l'aide de codecs vidéo et audio universellement adoptés. De ce fait, il est clair que la vidéo est destinée à être facilement accessible à tous les spectateurs - la compatibilité et la disponibilité étaient essentielles pour ses auteurs.

L'utilisation de MP3 montre l'âge du contenu de l'exemple, car il n'est plus considéré comme comparable aux concurrents modernes - au lieu de cela, AAC (Advanced Audio Coding) est le soutien de la famille des codecs audio.

On peut en dire autant de l'utilisation du profil principal H.264. Il est très rare qu'un profil H.264 en dehors de High soit utilisé, étant donné que presque tous les décodeurs le prennent en charge, permettant à chacun de tirer parti de l'efficacité accrue qu'offrent les fonctionnalités de profil élevé.

Les débits utilisés sont légèrement supérieurs aux prévisions pour l'environnement actuel. Cela peut s'expliquer par le désir des auteurs de qualité ou simplement par les limitations des

encodeurs disponibles lors de la création du contenu.

## Plus

Les autres outils utiles pour l'analyse des fichiers multimédias sont [FFprobe](#), qui fait partie du progiciel FFMpeg, et les [outils Bento4](#) pour travailler avec les fichiers MP4. Les deux sont également disponibles sous forme de bibliothèque. Ils sont capables d'une analyse de bas niveau supérieure à celle de MediaInfo, dans les situations où vous devez examiner des éléments individuels constituant des fichiers multimédias.

### Comprendre les composants de pistes vidéo

Cet exemple explorera comment voir la disposition d'une piste vidéo et comment extraire les images individuelles.

Le contenu de l'échantillon utilisé ici est [Tears of Steel](#), de Blender Foundation. Plus précisément, nous utiliserons le téléchargement intitulé "HD 720p (~ 365 Mo, mov, 2.0)". Ceci est un fichier unique qui se termine par l'extension "mov" et jouera dans à peu près n'importe quel lecteur multimédia moderne.

Nous utiliserons les outils mp4info et mp4dump de la [suite Bento4](#) pour la disposition de la piste et l'analyse de la structure, ainsi que [FFmpeg](#) pour extraire les images individuelles constituant la piste vidéo.

L'exemple de film utilise le format de packaging "QuickTime" (MOV), basé sur le format de fichier multimédia de base ISO - une norme internationale qui sous-tend tous les formats de packaging de la famille de formats de fichiers MP4. Cela le rend hautement compatible avec la majorité des outils disponibles et permet une analyse facile.

Examinons d'abord la structure globale du fichier. Tous les fichiers multimédias basés sur le format de fichier multimédia de base ISO sont structurés comme une hiérarchie de boîtes - un mini-système de fichiers. Utilisez l'utilitaire mp4dump pour extraire la structure de la boîte en exécutant la commande suivante:

```
mp4dump tears_of_steel_720p.mov
```

La sortie sera similaire à la suivante:

```
[ftyp] size=8+12
  major_brand = qt
  minor_version = 200
  compatible_brand = qt
[wide] size=8+0
[mdat] size=8+371579623
[moov] size=8+598972
  [mvhd] size=12+96
    timescale = 1000
    duration = 734167
    duration(ms) = 734167
```

```
[trak] size=8+244250
[tkhd] size=12+80, flags=f
  enabled = 1
  id = 1
  duration = 734167
  width = 1280.000000
  height = 534.000000
...
```

Cela représente la structure interne du fichier. Par exemple, vous voyez ici une boîte de *moov* qui a un en-tête de 8 octets et 598972 octets de contenu. Cette boîte est un conteneur pour différentes zones de métadonnées décrivant le contenu du fichier. Pour plus d'informations sur la signification des différentes cases et de leurs propriétés, voir [ISO / IEC 14496-12](#) .

Les échantillons de média eux-mêmes - les images compressées et les formes d'ondes audio - sont stockés dans la boîte *mdat* , dont le contenu est opaque pour l'utilitaire *mp4dump*.

Pour exclure les données non pertinentes et simplifier le flux de travail d'analyse - cet exemple se concentre sur la piste vidéo - nous supprimons maintenant la piste audio de notre exemple de film. Exécutez la commande suivante:

```
ffmpeg -i tears_of_steel_720p.mov -an -vcodec copy video_track.mov
```

Notez que l'étape ci-dessus supprimera également divers éléments d'extension personnalisés de la vidéo d'entrée, regroupant l'essence du contenu visuel dans un nouveau fichier de conteneur et supprimant toute autre chose. Si vous faites cela dans un scénario de production, assurez-vous que vous êtes vraiment libre de supprimer tous les autres éléments du fichier d'entrée!

Les pistes vidéo codées sont une séquence d'images. Avec le codec H.264 utilisé ici - et tous les autres codecs modernes couramment utilisés - les images peuvent être de différents types:

- I-images- ce sont des images indépendantes, décodables uniquement en utilisant les données contenues dans l'image.
- P-pictures - ceux-ci prennent une autre image comme base et appliquent une transformation à cette image (par exemple, "déplacez ces pixels particuliers vers la droite de 5 pixels").
- B-images - similaires aux images P, mais bidirectionnelles - elles peuvent également référencer des images du futur et définir des transformations telles que "ces pixels particuliers qui seront entièrement visibles dans 5 images sont maintenant visibles à 10%".

La combinaison exacte des types d'image peut être librement choisie par le workflow d'encodage, créant de nombreuses opportunités d'optimisation, bien que certains cas d'utilisation puissent limiter la flexibilité disponible, par exemple en exigeant qu'une image I soit présente à 2 secondes d'intervalle.

Exécutez la commande suivante pour voir la structure de l'image de la piste vidéo:

```
mp4info --show-layout video_track.mov
```

En plus de présenter une forme lisible par l'homme de l'ensemble des métadonnées du fichier, vous verrez une impression détaillée de la disposition de l'image de la piste vidéo.

```
...
00000959 [V] (1) size= 7615, offset=15483377, dts=491008 (39958 ms)
00000960 [V] (1) * size=104133, offset=15490992, dts=491520 (40000 ms)
00000961 [V] (1) size= 16168, offset=15595125, dts=492032 (40042 ms)
00000962 [V] (1) size= 4029, offset=15611293, dts=492544 (40083 ms)
00000963 [V] (1) size= 24615, offset=15615322, dts=493056 (40125 ms)
00000964 [V] (1) size= 4674, offset=15639937, dts=493568 (40167 ms)
00000965 [V] (1) size= 18451, offset=15644611, dts=494080 (40208 ms)
00000966 [V] (1) size= 95800, offset=15663062, dts=494592 (40250 ms)
00000967 [V] (1) size= 30271, offset=15758862, dts=495104 (40292 ms)
00000968 [V] (1) size= 10997, offset=15789133, dts=495616 (40333 ms)
00000969 [V] (1) size= 28458, offset=15800130, dts=496128 (40375 ms)
00000970 [V] (1) size= 9593, offset=15828588, dts=496640 (40417 ms)
00000971 [V] (1) size= 24548, offset=15838181, dts=497152 (40458 ms)
00000972 [V] (1) size= 6853, offset=15862729, dts=497664 (40500 ms)
00000973 [V] (1) size= 27698, offset=15869582, dts=498176 (40542 ms)
00000974 [V] (1) size= 7565, offset=15897280, dts=498688 (40583 ms)
00000975 [V] (1) size= 24682, offset=15904845, dts=499200 (40625 ms)
00000976 [V] (1) size= 5535, offset=15929527, dts=499712 (40667 ms)
00000977 [V] (1) size= 38360, offset=15935062, dts=500224 (40708 ms)
00000978 [V] (1) * size= 82466, offset=15973422, dts=500736 (40750 ms)
00000979 [V] (1) size= 13388, offset=16055888, dts=501248 (40792 ms)
00000980 [V] (1) size= 2315, offset=16069276, dts=501760 (40833 ms)
00000981 [V] (1) size= 21983, offset=16071591, dts=502272 (40875 ms)
00000982 [V] (1) size= 3384, offset=16093574, dts=502784 (40917 ms)
00000983 [V] (1) size= 22225, offset=16096958, dts=503296 (40958 ms)
...
```

Chaque ligne de cette impression est une image contenue dans la piste vidéo. Ceux marqués d'un astérisque comme (1) \* sont des images I. Vous pouvez voir comment ils sont les plus grands en taille, les autres permettant une plus grande compression en référençant les images existantes et en décrivant uniquement les différences.

La liste contient également le décalage des données d'image dans le fichier vidéo et l'horodatage de décodage de l'image, ce qui permet une corrélation et une analyse supplémentaires. Notez que l'ordre / le calendrier de décodage des images n'est pas nécessairement le même que l'ordre / le moment de la présentation! Si des images B existent dans la vidéo, elles ne peuvent être décodées *qu'après* les images auxquelles elles font référence, même si elles sont présentées *avant* les images référencées!

Après avoir pris connaissance de la structure de la piste vidéo, exécutez la commande suivante pour extraire les 30 images à partir de la marque 40 secondes sous la forme de fichiers PNG:

```
ffmpeg -i video_track.mov -ss 00:00:40 -vframes 30 picture%04d.png
```

Les images extraites seront entièrement décodées, comme elles apparaîtront dans un lecteur vidéo - il n'est pas possible (sans outillage extrêmement spécialisé) d'obtenir une représentation visuelle des données brutes dans les images P ou les images B.

Observez comment la 7ème image générée est un changement complet de scène dans la vidéo.



Vous pouvez facilement corrélérer cela avec la sortie mp4info ci-dessus - la 7ème image à partir de la marque 40 secondes (numéro 00000966) est de taille beaucoup plus grande que celles proches. Les changements de scène sont difficiles à encoder, car ils rafraîchissent l'image entière et contiennent beaucoup de nouvelles données. Si le codeur ne dispose pas de suffisamment de clarté pour optimiser les changements de scène (c.-à-d. Qu'il n'est pas autorisé à générer une grande image), la sortie visuelle sera de mauvaise qualité ou "bloquée" jusqu'à la prochaine image I. En examinant l'allocation de la bande passante (octets) à diverses images, vous pouvez obtenir des informations sur les artefacts visuels susceptibles d'apparaître soudainement dans une vidéo.

## Comprendre les présentations en continu adaptatives DASH

DASH est la technologie de diffusion adaptative la plus largement déployée dans les solutions modernes, utilisée pour diffuser de la vidéo dans une grande variété de scénarios. La meilleure façon de comprendre les présentations DASH est d'observer l'activité du réseau pendant la lecture.

Cet exemple utilise [Fiddler](#) pour capturer et analyser le trafic réseau du navigateur, même si un outil similaire suffit. Nous utiliserons le [lecteur](#) open source [dash.js](#) pour la lecture vidéo.

Pour notre contenu de démonstration, nous utiliserons les [vecteurs de test Axinom DASH](#), en particulier la variante 1080p du vecteur de test "Clear".



Sample Streams ▾

[http://media.axprod.net/TestVectors/v7-Clear/Manifest\\_1080p.mpd](http://media.axprod.net/TestVectors/v7-Clear/Manifest_1080p.mpd)

00:02:37.583  
Frame 3782 @ 24.00 fps  
H.264 High Profile 1920x1080 @ 3000 kbps



Avec la capture de votre réseau en cours d'exécution, ouvrez le [lecteur d'exemple de génération nocturne dash.js](#) dans tout navigateur moderne et entrez l'URL

[http://media.axprod.net/TestVectors/v6-Clear/Manifest\\_1080p.mpd](http://media.axprod.net/TestVectors/v6-Clear/Manifest_1080p.mpd) dans la zone de texte.

Appuyez sur Load pour lancer la lecture. Vous observerez les fichiers suivants en cours de téléchargement:

```
http://media.axprod.net/TestVectors/v7-Clear/Manifest_1080p.mpd
http://media.axprod.net/TestVectors/v7-Clear/2/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/15/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/18/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/18/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/2/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/15/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/15/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/5/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/5/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/18/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/5/0003.m4s
...
```

Le premier fichier est le manifeste de présentation - un document XML dont le format est défini dans [ISO / IEC 23009-1](#) . Cela décrit la présentation DASH à une profondeur suffisante pour permettre au joueur de comprendre comment le lire.

Si vous regardez à l'intérieur du manifeste, vous verrez divers éléments d' `AdaptationSet` , chacun décrivant une seule adaptation du contenu. Par exemple, il y a un jeu d'adaptation pour la piste vidéo, trois jeux d'adaptation pour trois langues audio et cinq jeux d'adaptation pour cinq langues de sous-titres.

Les ensembles d'adaptation internes sont des éléments de `Representation` . Pour l'ensemble d'adaptation vidéo, il en existe plusieurs: chaque représentation contient le même contenu visuel encodé en utilisant un niveau de qualité différent. Chaque ensemble d'adaptation audio et texte ne comporte qu'une représentation.

```
AdaptationSet [ segmentAlignment=true group=1 maxWidth=1920 maxHeight=1080 maxFrameRate=24 par=16:9 ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="1200000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="8" mimeType="video/mp4" codecs="hev 1.2.4.L63.90" width="512" height="288" frameRate="24" />
  <Representation id="9" mimeType="video/mp4" codecs="hev 1.2.4.L63.90" width="640" height="360" frameRate="24" />
  <Representation id="10" mimeType="video/mp4" codecs="hev 1.2.4.L90.90" width="852" height="480" frameRate="24" />
  <Representation id="11" mimeType="video/mp4" codecs="hev 1.2.4.L93.90" width="1280" height="720" frameRate="24" />
  <Representation id="12" mimeType="video/mp4" codecs="hev 1.2.4.L120.90" width="1920" height="1080" frameRate="24" />
AdaptationSet [ segmentAlignment=true group=2 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  Representation [ id=15 mimeType=audio/mp4 codecs=mp4a.40.29 audioSamplingRate=48000 startWithSAP=1 ]
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2" />
AdaptationSet [ segmentAlignment=true group=2 lang=en-AU ]
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  Representation [ id=16 mimeType=audio/mp4 codecs=mp4a.40.29 audioSamplingRate=48000 startWithSAP=1 ]
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2" />
AdaptationSet [ segmentAlignment=true group=2 lang=et-ET ]
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  Representation [ id=17 mimeType=audio/mp4 codecs=mp4a.40.29 audioSamplingRate=48000 startWithSAP=1 ]
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2" />
AdaptationSet [ segmentAlignment=true group=3 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="subtitle" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="1000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="18" mimeType="application/mp4" codecs="wvtt" startWithSAP="1" bandwidth="428" xmlns="urn:mpeg:dash:schema:mpd:2011" />
```

Pour effectuer la lecture, un joueur devra décider des adaptations à présenter au spectateur. Il peut prendre cette décision en fonction de toute logique métier personnalisée ou intégrée qu'il souhaite (par exemple, ordre de préférence linguistique). Les adaptations que l'auteur du contenu considère comme primaires ont un élément `Role` dans le manifeste déclarant le rôle "principal".

De plus, le joueur devra décider quelle représentation présenter au spectateur (si un ensemble d'adaptations offre plusieurs représentations). La plupart des lecteurs commencent de manière conservatrice et appliquent un algorithme heuristique qui tente de présenter le niveau de qualité maximal que la connexion réseau du spectateur peut supporter.

Le lecteur est libre de modifier à tout moment l'ensemble actif de représentations et / ou de jeux d'adaptation, en réponse à l'action de l'utilisateur (sélection d'une langue différente) ou à la logique automatisée (l'heuristique de bande passante entraîne un changement de niveau de

qualité).

L'élément `SegmentTemplate` définit la structure d'URL que le lecteur peut utiliser pour accéder aux différentes représentations. Un facteur clé des présentations DASH est que le contenu est divisé en petits segments de quelques secondes chacun (4 secondes pour notre exemple de film), qui sont téléchargés indépendamment. Chaque représentation a également un segment d'initialisation, nommé "init.mp4" pour cet exemple de film, qui contient une configuration de décodeur spécifique à la représentation et doit donc être chargé avant que tout autre segment de cette représentation puisse être traité.

Le comportement décrit ici est précis pour le profil DASH Live, qui est la variante la plus utilisée de DASH. Il existe également d'autres profils avec des comportements légèrement différents, non couverts ici. Faites attention à l'attribut "profile" sur l'élément racine du manifeste DASH pour vous assurer que cette description s'applique à vos vidéos!

Lorsque vous examinez la liste des URL obtenues à partir de la capture du trafic réseau et que vous les comparez aux informations fournies par le manifeste, vous concluez que le lecteur a effectué les actions suivantes après avoir téléchargé le manifeste:

1. Téléchargez les segments d'initialisation pour les représentations 2 (vidéo 360p), 15 (audio anglais) et 18 (sous-titres anglais).
2. Téléchargez les premiers segments des trois représentations ci-dessus (0001.m4s).
3. Téléchargez le deuxième segment de la représentation audio.
4. À partir du deuxième segment vidéo, passez au flux vidéo 1080p! Ceci est indiqué en téléchargeant le segment d'initialisation et le deuxième segment de représentation 5 (vidéo 1080p).
5. Continuez à télécharger d'autres segments des représentations actives.

En observant l'activité du réseau, il devient facile d'observer les décisions prises par un lecteur de streaming adaptatif DASH. Un tel lecteur est simplement un mécanisme qui télécharge des segments de différentes pistes et les fournit consécutivement à un moteur de lecture multimédia, en changeant de piste si nécessaire.

Les [vecteurs de test Axinom DASH](#) contiennent également des fichiers d'archive qui vous permettent de télécharger l'intégralité de la présentation pour une analyse au niveau du système de fichiers. Vous constaterez que les fichiers sur le disque sont exactement tels qu'ils sont au niveau du réseau. Cela signifie que les présentations DASH peuvent être servies par des serveurs HTTP arbitraires, sans aucune logique côté serveur personnalisée.

Un aspect de DASH de profil en direct qui complique l'analyse est que les échantillons de médias sont répartis sur un grand nombre de segments. La plupart des outils d'analyse de médias ne peuvent pas traiter des segments individuels, fonctionnant uniquement sur des pistes entières. Vous pouvez souvent contourner cette limitation en concaténant simplement les segments d'une seule représentation, en commençant par le segment d'initialisation. Par exemple, sous Windows, vous pouvez utiliser la commande suivante:

```
copy init.mp4 /b + 0001.mp4 /b + 0002.mp4 /b + 0003.mp4 /b track.mp4
```

Cela créera un fichier track.mp4 contenant les trois premiers segments de média d'une représentation. Bien que sa structure ne soit pas identique à un fichier MP4 autonome, un tel fichier peut toujours être analysé par la plupart des outils (tels que mp4info et FFmpeg) sans perte significative de fonctionnalité.

Lire Comprendre une présentation multimédia en ligne:

<https://riptutorial.com/fr/video/topic/5879/comprendre-une-presentation-multimedia>



# Chapitre 3: Ratios d'aspect vidéo

## Remarques

Les rapports d'aspect sont souvent exprimés sous la forme d'un rapport largeur: hauteur qui est souvent - mais pas toujours - simplifié et parfois aussi sous la forme d'un entier simple à virgule flottante.

Tous les ratios d'aspect suivants ont la même valeur, exprimée de différentes manières:

- 1280: 720
- 16: 9
- 1.7777777777777777777777777777778

## Exemples

Afficher les proportions (DAR)



1280

720

Display Aspect Ratio (DAR) 16:9

Ceci est une capture d'écran d'une vidéo en cours de lecture. Vous voyez une vidéo normale 16: 9 comme vous vous en doutez dans une solution vidéo moderne. C'est ce que l'on appelle le *format d'affichage* ou DAR.

À partir des paramètres illustrés, nous voyons que  $DAR = 1280: 720 = 16: 9 = 1,7777777777777777777777777777778$ .

## Ratio d'aspect d'image (PAR)

En interne, toutes les vidéos ne sont que des séries d'images. Jetons un coup d'oeil à une telle image.



Cela semble étrange, non? Effectivement. Les images qui composent une vidéo peuvent avoir un rapport d'aspect différent du DAR, le plus souvent pour des raisons algorithmiques (par exemple, seules les tailles multiples de 16 peuvent être compressées par l'algorithme choisi). C'est ce qu'on appelle le *format d'image* ou PAR.

Dans cet exemple, les dimensions de l'image sont exactement inversées entre la forme affichée (16: 9) et l'image réelle, de sorte que le paramètre PAR est 9:16. Normalement, les différences sont plus petites mais cet exemple est exagéré pour plus de clarté.

A partir des paramètres illustrés, on voit que  $PAR = 720: 1280 = 9:16 = 0,5625$

## Ratio d'aspect (SAR)

Comme l'indique l'exemple de *rapport d'image*, les vidéos sont des séries d'images qui n'ont pas nécessairement le même rapport d'aspect que le résultat final à afficher pour l'utilisateur.

Alors, comment obtenez-vous de ces images étirées à la sortie normalement affichée? Vous avez besoin d'un facteur d'étirement! Ce facteur d'étirement est appliqué à l'image afin de l'amener au format d'image correct pour l'affichage. C'est le *format d'image* ou SAR.



1280

Sample Aspect Ratio (SAR) 81:256



1280

720

Picture Aspect Ratio (PAR) 9:16

Le facteur d'étirement est souvent exprimé par un rapport de deux nombres entiers. Vous le calculez comme  $SAR = PAR / DAR$ .

À partir des paramètres illustrés, nous voyons que  $SAR = 9:16 / 16:9 = (9/16) / (16/9) = 81/256 = 3.1604938271604938271604938271605$

## Pixel aspect ratio

Ceci est un autre nom pour le *format de l'échantillon* et devrait être évité, car l'acronyme naturel (PAR) est en conflit avec le *rapport d'aspect de l'image*.

Lire Ratios d'aspect vidéo en ligne: <https://riptutorial.com/fr/video/topic/5713/ratios-d-aspect-video>

---

# Crédits

S. No	Chapitres	Contributeurs
1	Commencer avec la vidéo	<a href="#">Community</a> , <a href="#">Sander</a>
2	Comprendre une présentation multimédia	<a href="#">Sander</a>
3	Ratios d'aspect vidéo	<a href="#">Sander</a>