

## LES FICHIERS

Les fichiers évoqués dans le chapitre :

| Fichier                              | Description                                                                                        |
|--------------------------------------|----------------------------------------------------------------------------------------------------|
| <a href="#">ASSETMAP-ST429-9.xsd</a> | Fichier de définition XML AssetMap                                                                 |
| <a href="#">ASSETMAP.full.xml</a>    | Fichier AssetMap complet                                                                           |
| <a href="#">ASSETMAP.min.xml</a>     | Fichier AssetMap minimaliste                                                                       |
| <a href="#">parser.py</a>            | Parser AssetMap (Python) affichant quelques informations utiles                                    |
| <a href="#">parser.sh</a>            | Parser AssetMap (Shell) affichant quelques informations utiles avec différentes méthodes en shell. |
| <a href="#">path.smppte.txt</a>      | Fichier contenant des milliers de <a href="#">Path</a> respectant la norme SMPTE                   |
| <a href="#">path.non-smppte.txt</a>  | Fichier contenant des centaines de <a href="#">Path</a> ne respectant <i>pas</i> la norme SMPTE    |

## VALIDATION XML

Schéma XML de référence pour l'AssetMap : [ASSETMAP-ST429-9.xsd](#)

```
$ xmllint --noout --schema ASSETMAP-ST429-9.xsd ASSETMAP.xml
ASSETMAP.xml validates
```

## POUR GÉNÉRER UN UUID

### EN SHELL UNIX

```
$ uuidgen
D993CDA8-E3AE-47FA-B129-F573A1CEA524
```

### EN PYTHON

```
>>> import uuid
>>> uuid.uuid4()
UUID('2004320f-6772-4876-a4dd-b0dbd475b1a7')
```

## POUR GÉNÉRER UNE DATE AU FORMAT ISO

### AVEC SHELL UNIX RÉCENT

```
$ date --iso-8601=seconds
2022-01-15T18:30:23+01:00
```

### AVEC UN SHELL UNIX ANCIEN

```
$ date +"%Y-%m-%dT%H:%M:%S+01:00"
2021-11-03T17:37:39+01:00
```

### AVEC PYTHON

```
>>> import datetime
>>> datetime.datetime.now().astimezone().replace(microsecond=0).isoformat()
'2022-01-30T15:30:00+01:00'
```

```
>>> import datetime
>>> datetime.datetime.utcnow().strftime("%Y-%m-%dT%H:%M:%S+00:00")
'2022-01-30T20:21:01+00:00'
```

## AVEC UN OUTIL EN LIGNE

- [Timestamp Generator](#)
- [Unix Timestamp](#)

## POUR PARSER UNE DATE AU FORMAT ISO

```
>>> import datetime
>>> datetime.datetime.fromisoformat("2022-01-30T20:21:01+00:00")
datetime.datetime(2022, 1, 30, 20, 21, 1, tzinfo=datetime.timezone.utc)
```

## POUR PARSER UNE DATE ET LA RETOURNER AU FORMAT ISO

```
>>> import dateutil.parser as parser
>>> parser.parse("15 January 2022, 21:20:01, UTC").isoformat()
'2022-01-15T21:20:01+00:00'
```

ou utiliser [dateutil.parser.isoparse](#).

## RÉCUPÉRATION D'ÉLÉMENTS DANS UNE ASSETMAP

### RÉCUPÉRATION DE SON IDENTIFIANT (ID)

Avec xmllint :

```
$ xmllint --xpath '//*[local-name()="AssetMap"]/*[local-name()="Id"]/text()' ASSETMAP.xml
urn:uuid:606e9f2d-3a4c-49e2-ba4a-fc213caf1e64
```

Avec du shell - version courte et non strict :

```
$ grep -oE '<(am:)*Id>urn:uuid:[A-Za-z0-9\-\+</am:)*Id>' ASSETMAP.xml | head -n1 | awk -F'<|>' '{ print $3 }'
urn:uuid:606e9f2d-3a4c-49e2-ba4a-fc213caf1e64
```

Avec du shell - version longue et strict :

```
$ grep -oE '<(am:)*Id>urn:uuid:[A-Za-z0-9]{8}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{12}</am:)*Id>' ASSETMAP.xml
urn:uuid:606e9f2d-3a4c-49e2-ba4a-fc213caf1e64
```

Avec Python :

```
>>> from lxml import etree
>>> with open("ASSETMAP.xml", "rb") as xml:
    tree = etree.fromstring(
        text = xml.read()
    )
    tree.xpath("//*[local-name()='AssetMap']/*[local-name()='Id']/text()")

[urn:uuid:606e9f2d-3a4c-49e2-ba4a-fc213caf1e64]
```

## RÉCUPÉRATION L'ANNOTATIONTEXT

Avec shell :

```
$ grep -oE "<AnnotationText>(.*?)</AnnotationText>" ASSETMAP.xml | awk -F'<|>' '{ print $3 }'
```

```
DCP-INSIDE_TST-2D-24_C_FR-XX_51_4K_20220102_SMPTE_OV
```

Avec xmllint :

```
$ xmllint --xpath '//*[local-name()="AnnotationText"]/text()' ASSETMAP.xml
```

```
DCP-INSIDE_TST-2D-24_C_FR-XX_51_4K_20220102_SMPTE_OV
```

Avec Python :

```
>>> from lxml import etree
>>> with open("ASSETMAP.xml", "rb") as xml:
    tree = etree.fromstring(
        text = xml.read()
    )
    tree.xpath("//*[local-name()='AssetMap']//*[local-name()='AnnotationText']/text()")

['DCP-INSIDE_TST-2D-24_C_FR-XX_51_4K_20220102_SMPTE_OV']
```

## RÉCUPÉRATION DES FICHIERS DES DIFFÉRENTS ASSETS

Avec xmllint :

```
$ xmllint --xpath '//*[local-name()="Path"]/text()' ASSETMAP.xml
```

```
PKL.xml
CPL.xml
picture.mxf
audio.mxf
```

Avec shell (respect du SMPTE) :

```
$ grep -oE "<(am:)*Path>(file:\\\\)*[A-Za-z0-9\\/_\\.\\-]+<\\/(am:)*Path>" ASSETMAP.xml | awk -F'<|>' '{ print $3 }'
```

```
PKL.xml
CPL.xml
picture.mxf
audio.mxf
```

Avec shell (non respect du SMPTE, par des labos bourrés qui mettent n'importe quoi dans les noms des fichiers) :

```
$ grep -oE "<(am:)*Path>(file:\\\\)*[A-Za-z0-9#\\;\\&\\,\\ \\:\\'\\@+\\(\\)\\/_\\.\\-]+<\\/(am:)*Path>" ASSETMAP.xml | awk -F'<|>' '{ print $3 }'
```

Avec Python :

```
>>> from lxml import etree
>>> with open("ASSETMAP.xml", "rb") as xml:
    tree = etree.fromstring(
        text = xml.read()
    )
    tree.xpath("//*[local-name()='Path']/text()")

['PKL.xml',
 'CPL.xml',
 'picture.mxf',
 'audio.mxf']
```

## RÉCUPÉRATION DES IDENTIFIANTS (ID) DES DIFFÉRENTS ASSETS

Avec xmllint :

```
$ xmllint --xpath '//*[local-name()="Asset"]//*[local-name()="Id"]/text()' ASSETMAP.xml
```

```
urn:uuid:ce5c22c8-a640-428c-9004-2107e1f3c94e
urn:uuid:d2e32d20-1f85-4d86-b09f-2ee40ac097a0
urn:uuid:af457798-e34e-4233-9fe1-3cc974ca33e9
urn:uuid:6e007158-b706-4168-964d-53f35e7d2b74
```

Avec shell :

```
$ grep -oE '<(am:)*Id>urn:uuid:[A-Za-z0-9]{8}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{12}</(am:)*Id>' ASSETMAP.xml
urn:uuid:ce5c22c8-a640-428c-9004-2107e1f3c94e
urn:uuid:d2e32d20-1f85-4d86-b09f-2ee40ac097a0
urn:uuid:af457798-e34e-4233-9fe1-3cc974ca33e9
urn:uuid:6e007158-b706-4168-964d-53f35e7d2b74
```

Avec shell (variante) :

```
$ grep -A1 -E "<(am:)*Asset>" ASSETMAP.xml | grep -oE '<(am:)*Id>urn:uuid:[A-Za-z0-9]{8}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{12}</(am:)*Id>' ASSETMAP.xml
urn:uuid:ce5c22c8-a640-428c-9004-2107e1f3c94e
urn:uuid:d2e32d20-1f85-4d86-b09f-2ee40ac097a0
urn:uuid:af457798-e34e-4233-9fe1-3cc974ca33e9
urn:uuid:6e007158-b706-4168-964d-53f35e7d2b74
```

Avec Python :

```
>>> from lxml import etree
>>> with open("ASSETMAP.xml", "rb") as xml:
    tree = etree.fromstring(
        text = xml.read()
    )
    tree.xpath("//*[local-name()='Asset']/*[local-name()='Id']/text()")

['urn:uuid:ce5c22c8-a640-428c-9004-2107e1f3c94e',
 'urn:uuid:d2e32d20-1f85-4d86-b09f-2ee40ac097a0',
 'urn:uuid:af457798-e34e-4233-9fe1-3cc974ca33e9',
 'urn:uuid:6e007158-b706-4168-964d-53f35e7d2b74']
```

=====