# ASSETMAP : CODES & FILES

## FILES

Files mentioned in this chapter :

| Filename | Description |
|---|---|
| `ASSETMAP-ST429-9.xsd` | XML Definition (XSD) for AssetMap |
| `ASSETMAP.full.xml` | Complete AssetMap file (example) |
| `ASSETMAP.min.xml` | Minimal AssetMap file (example) |
| `parser.py` | AssetMap Parser (Python) displaying some useful information |
| `parser.sh` | AssetMap Parser (Shell) displaying some useful information using different shell methods |
| `path.smpte.txt` | File containing thousands `Path` values following the SMPTE standard |
| `path.non-smpte.txt` | File containing thousand `Path` values *NOT* following the SMPTE standard |

## XML VALIDATION

XML Reference Schema for AssetMap : ASSETMAP-ST429-9.xsd

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

## GENERATE AN UUID

### IN UNIX SHELL

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

### IN PYTHON

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

## GENERATE DATE IN ISO FORMAT

### USING RECENT UNIX SHELL

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

### USING OLD UNIX SHELL

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

### IN 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'
```

### USING ONLINE TOOLS

- Timestamp Generator
- Unix Timestamp

## TO PARSE DATE IN ISO FORMAT

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

## TO PARSE DATE AND RETURNS IT IN ISO FORMAT

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

or use dateutil.parser.isoparse.

## GET ELEMENTS WITHIN AN ASSETMAP

### GET ITS IDENTIFIER (ID)

Using xmllint :

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

Using shell - short non-strict version :

```
$ 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
```

Using shell - long strict version :

```
$ 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>'
urn:uuid:606e9f2d-3a4c-49e2-ba4a-fc213caf1e64
```

Using 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]
```

## GET ANNOTATIONTEXT

Using shell :

```
$ grep -His -oE "<AnnotationText>(.*)</AnnotationText>" ASSETMAP.xml | awk -F'<|>' '{ print $3 }'
DCP-INSIDE_TST-2D-24_C_FR-XX_51_4K_20220102_SMPTE_OV
```

Using xmllint :

```
$ xmllint --xpath '//*[local-name()="AnnotationText"]/text()' ASSETMAP.xml
DCP-INSIDE_TST-2D-24_C_FR-XX_51_4K_20220102_SMPTE_OV
```

Using 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']
```

## GET FILENAMES OF THE DIFFERENTS ASSETS

Using xmllint :

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

Using shell (SMPTE compliant) :

```
$ 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
```

Using shell (SMPTE non-compliant, by drunk labs that put random stuff in filenames) :

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

Using 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']
```

## GET IDENTIFIERS (ID) OF EACH ASSETS

Using 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
```

Using 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>'
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
```

Using 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}-
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
```

Using 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']
```