

Quickstart

Installing

You can install pyaaf2 via:

```
pip install pyaaf2
```

or if you want to use the latest development git master:

```
git clone https://github.com/markreidvfx/pyaaf2
cd pyaaf2
python setup.py install
```

Reading

```
import aaf2

with aaf2. open ("path/to/file.aaf", "r") as f:

    # get the main composition
    main_compostion = next(f.content.toplevel())

    # print the name of the composition
    print(main_compostion.name)

    # AAFObjects have properties that can be
    # accessed just like a dictionary
    print(main_compostion['CreationTime'].value)

    # video, audio and other track types are
    # stored in slots on a mob object.
    for slot in main_compostion.slots:
        segment = slot.segment
        print(segment)
```

Embedding Footage

First lets generate some DNxHR media with ffmpeg:

```
ffmpeg -f lavfi -i testsrc=size=960x540 -frames:v 24 -vcodec dnxhd -pix_fmt yuv422p -profile:v dnxhr_lb sample.dnxhd
```

Now lets generate some audio media:

```
ffmpeg -f lavfi -i aevalsrc="sin(420*2*PI*t):s=48000:d=1.0" -acodec pcm_s16le sample.wav
```

Finally import the footage:

```
import aaf2

with aaf2. open ("example2.aaf", 'w') as f:

    # objects are create with a factory
    # on the AAFFile Object
    mob = f.create.MasterMob("Demo2")

    # add the mob to the file
    f.content.mobs.append(mob)

    edit_rate = 25

    # Lets also create a tape so we can add timecode (optional)
    tape_mob = f.create.SourceMob()
    f.content.mobs.append(tape_mob)

    timecode_rate = 25
    start_time = timecode_rate * 60 * 60 # 1 hour
    tape_name = "Demo Tape"

    # add tape slots to tape mob
    tape_mob.create_tape_slots(tape_name, edit_rate,
                              timecode_rate, media_kind='picture')

    # create sourceclip that references timecode
    tape_clip = tape_mob.create_source_clip(1, start_time)

    # now finally import the generated media
    mob.import_dnxhd_essence("sample.dnxhd", edit_rate, tape_clip)
    mob.import_audio_essence("sample.wav", edit_rate)
```