pycmx Documentation

Release 1.2.2

Jamie Hardt

Nov 08, 2023

API REFERENCE

1 Parse Function 1

2 pycmx Classes 3

3 Indices and tables 7

Index 9

CHAPTER

ONE

PARSE FUNCTION

pycmx.parse_cmx_events.**parse_cmx3600**(*f: TextIO*)

Parse a CMX 3600 EDL.

Parameters

f (*TextIO*) – a file-like object, anything that's readlines-able.

Returns

An *pycmx.edit_list.EditList*.

CHAPTER

TWO

PYCMX CLASSES

class pycmx.edit_list.**EditList**(*statements*)

Represents an entire edit decision list as returned by `parse_cmx3600()`.

property channels: *ChannelMap*

Return the union of every channel channel.

property events: Generator[*Event*, None, None]

A generator for all the events in the edit list

property format: str

The detected format of the EDL. Possible values are: “3600”, “File32”, “File128”, and “unknown”.

property sources: Generator[Source, None, None]

A generator for all of the sources in the list

property title: str

The title of this edit list.

property unrecognized_statements: Generator[Unrecognized, None, None]

A generator for all the unrecognized statements in the list.

class pycmx.event.**Event**(*statements*)

Represents a collection of *Edit* s, all with the same event number.

property edits: List[*Edit*]

Returns the edits. Most events will have a single edit, a single event will have multiple edits when a dissolve, wipe or key transition needs to be performed.

property number: int

Return the event number.

property unrecognized_statements: Generator[Unrecognized, None, None]

A generator for all the unrecognized statements in the event.

class pycmx.edit.**Edit**(*edit_statement*, *audio_ext_statement*, *clip_name_statement*, *source_file_statement*, *trans_name_statement=None*)

An individual source-to-record operation, with a source roll, source and recorder timecode in and out, a transition and channels.

property aux_source: bool

An auxiliary source is the source of this event.

property black: bool

Black video or silence should be used as the source for this event.

property channels: *ChannelMap*

Get the ChannelMap object associated with this Edit.

property clip_name: **str | None**

Get the clip name, as attested by a "* FROM CLIP NAME" or "* TO CLIP NAME" remark on the EDL. This will return None if the information is not present.

property line_number: **int**

Get the line number for the "standard form" statement associated with this edit. Line numbers a zero-indexed, such that the "TITLE:" record is line zero.

property record_in: **str**

Get the record in timecode.

property record_out: **str**

Get the record out timecode.

property source: **str**

Get the source column. This is the 8, 32 or 128-character string on the event record line, this usually references the tape name of the source.

property source_file: **str | None**

Get the source file, as attested by a "* SOURCE FILE" remark on the EDL. This will return None if the information is not present.

property source_in: **str**

Get the source in timecode.

property source_out: **str**

Get the source out timecode.

property transition: *Transition*

Get the Transition object associated with this edit.

class pycmx.transition.**Transition**(*transition*, *operand*, *name=None*)

A CMX transition: a wipe, dissolve or cut.

property cut: **bool**

True if this transition is a cut.

property dissolve: **bool**

True if this traansition is a dissolve.

property effect_duration: **int**

The duration of this transition, in frames of the record target.

In the event of a key event, this is the duration of the fade in.

property key_background: **bool**

True if this edit is a key background.

property key_foreground: **bool**

True if this edit is a key foreground.

property key_out: **bool**

True if this edit is a key out. This material will removed from the key foreground and replaced with the key background.

property kind: str | None

Return the kind of transition: Cut, Wipe, etc

property wipe: bool

True if this transition is a wipe.

property wipe_number: int | None

Wipes are identified by a particular number.

class pycmx.channel_map.**ChannelMap**(*v=False*, *audio_channels={}*)

Represents a set of all the channels to which an event applies.

property a1: bool

True if A1 is included

property a2: bool

True if A2 is included

property a3: bool

True if A3 is included

property a4: bool

True if A4 is included

property audio: bool

True if an audio channel is included

property channels: Generator[int, None, None]

A generator for each audio channel

get_audio_channel(*chan_num*) → bool

True if chan_num is included

set_audio_channel(*chan_num*, *enabled: bool*)

If enabled is true, chan_num will be included

property video: bool

True if video is included

CHAPTER

THREE

INDICES AND TABLES

- genindex
- search

INDEX

A

a1 (*pycmx.channel_map.ChannelMap property*), 5
a2 (*pycmx.channel_map.ChannelMap property*), 5
a3 (*pycmx.channel_map.ChannelMap property*), 5
a4 (*pycmx.channel_map.ChannelMap property*), 5
audio (*pycmx.channel_map.ChannelMap property*), 5
aux_source (*pycmx.edit.Edit property*), 3

B

black (*pycmx.edit.Edit property*), 3

C

ChannelMap (*class in pycmx.channel_map*), 5
channels (*pycmx.channel_map.ChannelMap property*), 5
channels (*pycmx.edit.Edit property*), 3
channels (*pycmx.edit_list.EditList property*), 3
clip_name (*pycmx.edit.Edit property*), 4
cut (*pycmx.transition.Transition property*), 4

D

dissolve (*pycmx.transition.Transition property*), 4

E

Edit (*class in pycmx.edit*), 3
EditList (*class in pycmx.edit_list*), 3
edits (*pycmx.event.Event property*), 3
effect_duration (*pycmx.transition.Transition property*), 4
Event (*class in pycmx.event*), 3
events (*pycmx.edit_list.EditList property*), 3

F

format (*pycmx.edit_list.EditList property*), 3

G

get_audio_channel() (*pycmx.channel_map.ChannelMap method*), 5

K

key_background (*pycmx.transition.Transition property*), 4
key_foreground (*pycmx.transition.Transition property*), 4
key_out (*pycmx.transition.Transition property*), 4
kind (*pycmx.transition.Transition property*), 4

L

line_number (*pycmx.edit.Edit property*), 4

N

number (*pycmx.event.Event property*), 3

P

parse_cmx3600() (*in module pycmx.parse_cmx_events*), 1

R

record_in (*pycmx.edit.Edit property*), 4
record_out (*pycmx.edit.Edit property*), 4

S

set_audio_channel() (*pycmx.channel_map.ChannelMap method*), 5
source (*pycmx.edit.Edit property*), 4
source_file (*pycmx.edit.Edit property*), 4
source_in (*pycmx.edit.Edit property*), 4
source_out (*pycmx.edit.Edit property*), 4
sources (*pycmx.edit_list.EditList property*), 3

T

title (*pycmx.edit_list.EditList property*), 3
Transition (*class in pycmx.transition*), 4
transition (*pycmx.edit.Edit property*), 4

U

unrecognized_statements (*pycmx.edit_list.EditList property*), 3

unrecognized_statements (*pycmx.event.Event property*), 3

V

video (*pycmx.channel_map.ChannelMap property*), 5

W

wipe (*pycmx.transition.Transition property*), 5
wipe_number (*pycmx.transition.Transition property*), 5