- Overview
- Status Update
- Use Case
- Roadmap
- Discussion
Overview
OTIO Overview

- Open Source API and interchange format for editorial timeline information.
OTIO Overview

- Tracks, Clip order, timing, references to media
- Markers, Transitions
- Metadata!
- Eventually: Timing Effects
OTIO Overview

- References to media, *not* actual media
OTIO Overview

- “Has this shot changed duration since yesterday”
- “How many extra frames of this shot are used in the crossfade”
- “What shots precede and follow this shot”
Another Format?

Simple
EDL

“Just Right”
OTIO

Complex
AAF
Goal: Flexible Pipeline

Editorial

Production

Finaling

OTIO

OTIO

OTIO

XML

METADATA

UNREAL ENGINE
OTIO Overview

- Features:
  - JSON File Format
  - Python API
  - Editorial Math library ("opentime")
  - File Conversion Adapter Plugins
  - Media Linker Plugins
OpenTime

- RationalTime + TimeRange
- Conversion
- Math

```python
import otio

t = otio.opentime.RationalTime(10, 24)
t2 = otio.opentime.RationalTime(20, 48)
t = t.rescaled_to(t2)

timecode = "00:06:56:17"
t = otio.opentime.from_timecode(timecode, 24)

t1 = otio.opentime.RationalTime(15.2)
t2 = otio.opentime.RationalTime(15.6)
self.assertTrue(t1 < t2)
self.assertTrue(t1 <= t2)
self.assertFalse(t1 > t2)
self.assertFalse(t1 == t2)
```
OpenTime

- RationalTime + TimeRange
- Conversion
- Math

```python
1 t = otio.opentime.RationalTime(10, 24)
2 t2 = otio.opentime.RationalTime(20, 48)
3 t = t.rescaled_to(t2)

5 timecode = "00:06:56:17"
6 t = otio.opentime.from_timecode(timecode, 24)

8 t1 = otio.opentime.RationalTime(15.2)
9 t2 = otio.opentime.RationalTime(15.6)
10 self.assertTrue(t1 < t2)
11 self.assertTrue(t1 <= t2)
12 self.assertFalse(t1 > t2)
13 self.assertFalse(t1 >= t2)
```
OpenTime

- RationalTime + TimeRange
- Conversion
- Math

```python
1 t = otio.opentime.RationalTime(10, 24)
2 t2 = otio.opentime.RationalTime(20, 48)
3 t = t.rescaled_to(t2)
4 
5 timecode = "00:06:56:17"
6 t = otio.opentime.from_timecode(timecode, 24)
7 
8 t1 = otio.opentime.RationalTime(15.2)
9 t2 = otio.opentime.RationalTime(15.6)
10 self.assertTrue(t1 < t2)
11 self.assertTrue(t1 <= t2)
12 self.assertTrue(t1 > t2)
13 self.assertTrue(t1 >= t2)
```
OpenTime

- RationalTime + TimeRange
- Conversion
- Math

```python
import otio

t = otio.opentime.RationalTime(10, 24)
t2 = otio.opentime.RationalTime(20, 48)
t = t.rescaled_to(t2)

timecode = "00:06:56:17"
t = otio.opentime.from_timecode(timecode, 24)

t1 = otio.opentime.RationalTime(15.2)
t2 = otio.opentime.RationalTime(15.6)
self.assertTrue(t1 < t2)
self.assertTrue(t1 <= t2)
self.assertFalse(t1 > t2)
self.assertFalse(t1 >= t2)
```
OTIO Overview

TRACK 3

- CLIP
- CLIP
- GAP

MARKER

TRACK 2

- GAP
- CLIP
- GAP

TRACK 1

- GAP
- CLIP

TRANSITION

MEDIAREFERENCE

URL reference to any media format
Media Reference

CLIP

MEDIA REFERENCE
OTIO File Format

```
"OTIO_SCHEMA": "Timeline.1",
"metadata": {},
"name": "Coco_Beginning.Exported.01",
"tracks": {
  "OTIO_SCHEMA": "Stack.1",
  "children": [
    {"OTIO_SCHEMA": "Track.1",
      "children": [
        {"OTIO_SCHEMA": "Clip.1",
          "effects": [],
          "markers": [],
          "media_reference": {
            "OTIO_SCHEMA": "ExternalReference.1",
            "available_range": null,
            "metadata": {},
            "name": null,
            "target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.take.1"
          },
          "metadata": {
            "AAF": {
              "ClassName": "MasterMob",
              "CreationTime": "2012-05-01 00:10:54.000",
              "LastModified": "2017-10-03 23:43:05.000",
              "MobID": "urn:smp:mid:060a2b34.0101010.01010f00.13000000.060e2b34.7f7f2a80.4f9f2a0e.760f75b7",
              "Name": "acad_head_leader_239 (REN1)",
              "UserComments": {
                "Profile": "PixarRec7092010",
                "ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
                "SourceImageSize": "1920x804",
                "prod": "acad"
              },
              "pixar": {
                "globaltake": "1",
                "prod": "acad",
                "shot": "acad_head_leader_239",
                "unit": "studio"
              }
            },
            "name": "acad_head_leader_239 (REN1)",
            "source_range": {
              "OTIO_SCHEMA": "TimeRange.1",
            }
          }
        }
      ]
    }
  ]
}
```
"OTIO_SCHEMA": "Timeline.1",
"metadata": {
  "name": "Coco_Begginning.Exported.01",
  "tracks": [
    {"OTIO_SCHEMA": "Stack.1",
      "children": [
        {"OTIO_SCHEMA": "Track.1",
         "children": [
           {"OTIO_SCHEMA": "Clip.1",
            "effects": [],
            "markers": [],
            "media_reference": {"OTIO_SCHEMA": "ExternalReference.1",
              "available_range": null,
              "metadata": {},
              "name": null,
              "target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.take_01"
            },
            "metadata": {
              "AAE": {
                "ClassName": "MasterMob",
                "CreationTime": "2012-05-01 00:10:54.000",
                "LastModified": "2017-10-03 23:43:05.000",
                "MobID": "urn:smp:uuid:060a2b34.01010101.0101000.13000000.060e2b34.7f7f2a80.4f9f2a0e.7600f707",
                "Name": "acad_head_leader_239 (REN1)",
                "UserComments": {
                  "Profile": "PixarRec7092010",
                  "ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
                  "SourceImageSize": "1920x804",
                  "prod": "acad"
                }
              },
              "pixar": {
                "globaltake": "1",
                "prod": "acad",
                "shot": "acad_head_leader_239",
                "unit": "studio"
              }
            }
          }
        }
      ]
    }
  ]
},
"name": "acad_head_leader_239 (REN1)",
"source_range": {
  "OTIO_SCHEMA": "TimeRange.1",
  "duration": "155.951"
}
"OTIO_SCHEMA": "Timeline.1",

"metadata": {},

"name": "Coco_Beginning Exported.01",

"OTIO_SCHEMA": "Stack.1",

"children": [

  "OTIO_SCHEMA": "Track.1",

  "children": [

    "OTIO_SCHEMA": "Clip.1",
    "effects": [],
    "markers": [],
    "media_reference": {
      "OTIO_SCHEMA": "ExternalReference.1",
      "available_range": null,
      "metadata": {},
      "name": null,
      "target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.take.01"
    },

    "metadata": {
      "AAF": {
        "ClassName": "MasterMob",
        "CreationTime": "2012-05-01 00:10:54.000",
        "LastModifiedDate": "2017-05-01 23:43:05.000",
        "MobID": "urn:uuid:060a2b34.01010101.0101000.0000000.060a2b34.7f7f7f7f7f7f7f7f7f7f7f7f7f7f7f7f",
        "Name": "acad_head_leader_239 (REN1)",
        "UserComments": {
          "Profile": "PixarRec7092010",
          "ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
          "SourceImageSize": "1920x804",
          "prod": "acad"
        }
      }
    },

    "pixar": {
      "globaltake": "1",
      "prod": "acad",
      "shot": "acad_head_leader_239",
      "unit": "studio"
    },

    "name": "acad_head_leader_239 (REN1)",
    "source_range": {
      "OTIO_SCHEMA": "TimeRange.1",
      "source_time_base": "25.000", ...
OTIO File Format

```
"OTIO_SCHEMA": "Timeline.1",
"metadata": {},
"name": "Coco Beginning.Exported.01",
"tracks": [
  "OTIO_SCHEMA": "Stack.1",
  "children": [
    "OTIO_SCHEMA": "Track.1",
    "children": [
      "OTIO_SCHEMA": "Clip.1",
      "effects": [],
      "markers": [],
      "media_reference": {"OTIO_SCHEMA": "ExternalReference.1",
        "available_range": null,
        "metadata": {},
        "name": null,
        "target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.take
      },
      "metadata": {
        "AAE": {
          "ClassName": "MasterMob",
          "CreationTime": "2012-05-01 00:10:54.00",
          "LastModified": "2017-10-03 23:43:05.00",
          "MobID": "urn:smp:mid:060a2b34.010101.01010f00.13000000.060e2b34.77f7f2a80.4f9f2a0e.7600f707",
          "Name": "acad_head_leader_239 (RENI1)",
          "UserComments": {
            "Profile": "PixarRec7092010",
            "ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
            "SourceImageSize": "1920x804",
            "prod": "acad"
          }
        },
        "pixar": {
          "globaltake": "1",
          "prod": "acad",
          "shot": "acad_head_leader_239",
          "unit": "studio"
        }
      },
      "name": "acad_head_leader_239 (RENI1)",
      "source_range": {
        "OTIO_SCHEMA": "TimeRange.1",
      }
    }]
  ]
]"
"OTIO_SCHEMA": "Timeline.1",
"metadata": {},
"name": "Coco_Begning.Exported.01",
"tracks": {
  "OTIO_SCHEMA": "Stack.1",
  "children": [
    {
      "OTIO_SCHEMA": "Track.1",
      "children": [
        {
          "OTIO_SCHEMA": "Clip.1",
          "metadata": {},
          "markers": [],
          "media_reference": {
            "OTIO_SCHEMA": "ExternalReference.1",
            "available_range": null,
            "metadata": {},
            "name": null,
            "target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.take.1AAF",
          },
          "metadata": {
            "AAF": {
              "ClassName": "MasterMob",
              "CreationTime": "2012-05-01 00:10:54.000",
              "LastModified": "2017-10-03 23:43:05.000",
              "MobID": "urn:smlt:umid:060a2b34.01010101.01010f00.13000000.060e2b34.7f7f2a80.4f9f2a0e.7600f707",
              "Name": "acad_head_leader_239 (REN1)",
              "UserComments": {
                "Profile": "PixarRec7092010",
                "ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
                "SourceImageSize": "1920x804",
                "prod": "acad"
              },
              "pixar": {
                "globaltake": "1",
                "prod": "acad",
                "shot": "acad_head_leader_239",
                "unit": "studio"
              },
            },
            "name": "acad_head_leader_239 (REN1)",
            "source_range": {
              "OTIO_SCHEMA": "TimeRange.1",
            }
          }
        }
      }
    }
  }
}
OTIO File Format

```json
"OTIO_SCHEMA": "Timeline.1",
"metadata": {},
"name": "Coco_Beginning.Exported.01",
"tracks": [
  "OTIO_SCHEMA": "Stack.1",
  "children": [
    "OTIO_SCHEMA": "Track.1",
    "children": [
      "OTIO_SCHEMA": "Clip.1",
      "effects": [],
      "markers": [],
      "media_reference": {
        "OTIO_SCHEMA": "ExternalReference.1",
        "available_range": null,
        "metadata": {},
        "name": null,
        "target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.take_1_mov"
      }
    }
  ]
]"metadata": {
  "AAF": {
    "ClassName": "MasterMob",
    "CreationTime": "2012-05-01 00:10:54.00",
    "LastModified": "2017-10-03 23:43:05.00",
    "MobID": "urn:smp:uuid:060a2b34.01010101.010100.1300000.060e2b34.7f7f2a80.4f9f2a0e.7600f707",
    "Name": "acad_head_leader_239 (RENI)",
    "UserComments": {
      "Profile": "PixarRec092010",
      "ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
      "SourceImageSize": "1920x804",
      "prod": "acad"
    }
  },
  "pixar": {
    "globaltake": "1",
    "prod": "acad",
    "shot": "acad_head_leader_239",
    "unit": "studio"
  }
},
"name": "GLOBHEAD_LEADER_239 (RENI)",
"source_range": {
  "OTIO_SCHEMA": "TimeRange.1",
  "begin": 0,
  "end": 120
}
```
OTIO
File Format

```json
"OTIO_SCHEMA": "Timeline.1",
"metadata": {},
"name": "Coco_Beginning.Export.01",
"tracks": {
  "OTIO_SCHEMA": "Stack.1",
  "children": [
    { 
      "OTIO_SCHEMA": "Track.1",
      "children": [
        {
          "OTIO_SCHEMA": "Clip.1",
          "effects": [],
          "markers": [],
          "media_reference": {
            "OTIO_SCHEMA": "ExternalReference.1",
            "available_range": null,
            "metadata": {},
            "name": null,
            "target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.take_1.mov",
            "metadata": {}  
          },
          "metadata": {
            "AAF": {
              "ClassName": "MasterMob",
              "CreationTime": "2012-05-01 00:10:54.00",
              "LastModified": "2017-10-03 23:43:05.00",
              "MobID": "urn:spp:uuid:060a2b34.0101010.0101f00.13000000.00e2b34.7f7f2a80.4f9f2a0e.7600f707",
              "Name": "acad_head_leader_239 (REN1)",
              "UserComments": {
                "Profile": "PixarRec7092010",
                "ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
                "SourceImageSize": "1920x804",
                "prod": "acad"
              },
              "PIXI": {
                "globaltake": "1",
                "prod": "acad",
                "shot": "acad_head_leader_239",
                "unit": "studio"
              },
              "name": "acad_head_leader_239 (REN1)",
              "source_range": {
                "OTIO_SCHEMA": "TimeRange.1",
                "duration": "1.00",
                "start_timecode": "00:00:00:00.000",
                "end_timecode": "00:00:00:01.000"
              }
```
```
OTIO File Format

```
"OTIO_SCHEMA": "Timeline.1",
"metadata": {},
"name": "Coco_Beginning Exported.01",
"tracks": {
  "OTIO_SCHEMA": "Stack.1",
  "children": [
    {
      "OTIO_SCHEMA": "Track.1",
      "children": [
        {
          "OTIO_SCHEMA": "Clip.1",
          "effects": [],
          "markers": [],
          "media_reference": {
            "OTIO_SCHEMA": "ExternalReference.1",
            "available_range": null,
            "metadata": {},
            "name": null,
            "target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.take.1"
          },
          "metadata": {
            "AAF": {
              "ClassName": "MasterMob",
              "CreationTime": "2012-05-01 00:10:54.00",
              "LastModified": "2017-10-03 23:43:05.00",
              "MobID": "urn:smp:uuid:060a2b34.01010101.0101f00.13000000.060e2b34.7f7f2a80.4f9f2a0e.7600f787",
              "Name": "acad_head_leader_239 (REM1)",
              "UserComments": {
                "Profile": "PixarRec7092010",
                "ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
                "SourceImageSize": "1920x804",
                "prod": "acad"
              }
            },
            "pixar": {
              "globaltake": "1",
              "prod": "acad",
              "shot": "acad_head_leader_239",
              "unit": "studio"
            }
          },
          "name": "acad_head_leader_239 (REM1)",
          "source_range": {
            "OTIO_SCHEMA": "TimeRange.1",
```
"OTIO_SCHEMA": "Timeline.1",
"metadata": {},
"name": "Coco_Beginning.Exported.01",
"tracks": {
"OTIO_SCHEMA": "Stack.1",
"children": [
{ "OTIO_SCHEMA": "Track.1",
"children": [
{ "OTIO_SCHEMA": "Clip.1",
"effects": [],
"markers": [],
"media_reference": { "OTIO_SCHEMA": "ExternalReference.1",
"available_range": null,
"metadata": {},
"name": null,
"target_url": "/shows/studio/diskfarm/TAKES_acad_head_leader_239/take_1_mov/TAKES_acad_head_leader_239.tak"
},
"metadata": {
"AAF": {
"ClassName": "MasterMob",
"CreationTime": "2012-05-01 00:10:54.000",
"LastModified": "2017-10-03 23:43:05.000",
"MobID": "urn:smp:uuid:060a2b34.01010111.01010f00.13000000.060e2b34.7f7f2a80.4f9f2a0e.6f00f707",
"Name": "acad_head_leader_239 (REN1)",
"UserComments": {
"Profile": "PixarRec7092010",
"ShotTakeRange": "acad_head_leader_239 (1) 1 - 192",
"SourceImageSize": "1920x804",
"prod": "acad"
},
"pixar": {
"globaltake": "1",
"prod": "acad",
"shot": "acad_head_leader_239",
"unit": "studio"
}
},
"name": "acad_head_leader_239 (REN1)",
"source_range": {
"OTIO_SCHEMA": "TimeRange.1",
"duration": 192,
"start_time": 0
}
```python
import opentimelineio as otio

def main():
    args = parse_args()

    timeline = otio.adapters.read_from_file(args.input)
    count = _conform_timeline(timeline, args.folder)
    otio.adapters.write_to_file(timeline, args.output)

def _conform_timeline(timeline, folder):
    """Look for replacement media for each clip in the given timeline.
    The clips are relinked in place if media with a matching name is found.
    """

    for clip in timeline.each_clip():
        # look for a media file that matches the clip's name
        new_path = _find_matching_media(clip.name, folder)

        # if no media is found, keep going
        if not new_path:
            continue

        # if we found one, then relink to the new path
        clip.media_reference = otio.schema.ExternalReference(
            target_url="file://" + new_path,
            available_range=None  # we don't know the available range
        )
```
```python
import opentimelineio as otio

def main():
    args = parse_args()

    timeline = otio.adapters.read_from_file(args.input)
    count = _conform_timeline(timeline, args.folder)
    otio.adapters.write_to_file(timeline, args.output)

def _conform_timeline(timeline, folder):
    """Look for replacement media for each clip in the given timeline. The clips are relinked in place if media with a matching name is found."
    
    for clip in timeline.each_clip():
        # look for a media file that matches the clip's name
        new_path = _find_matching_media(clip.name, folder)

        # if no media is found, keep going
        if not new_path:
            continue

        # if we found one, then relink to the new path
        clip.media_reference = otio.schema.ExternalReference(
            target_url="file://" + new_path,
            available_range=None  # we don't know the available range
        )
```

```python
import opentimelineio as otio

def main():
    args = parse_args()

    timeline = otio.adapters.read_from_file(args.input)
    count = _conform_timeline(timeline, args.folder)
    otio.adapters.write_to_file(timeline, args.output)

def _conform_timeline(timeline, folder):
    """Look for replacement media for each clip in the given timeline. The clips are relinked in place if media with a matching name is found."
    for clip in timeline.each_clip():
        # look for a media file that matches the clip's name
        new_path = _find_matching_media(clip.name, folder)
        # if no media is found, keep going
        if not new_path:
            continue

        # if we found one, then relink to the new path
        clip.media_reference = otio.schema.ExternalReference(
            target_url="file:///" + new_path,
            available_range=None  # we don't know the available range
        )
```

```python
import opentimelineio as otio

def main():
    args = parse_args()

    timeline = otio.adapters.read_from_file(args.input)
    count = _conform_timeline(timeline, args.folder)
    otio.adapters.write_to_file(timeline, args.output)

def _conform_timeline(timeline, folder):
    """Look for replacement media for each clip in the given timeline. The clips are relinked in place if media with a matching name is found."

    for clip in timeline.each_clip():
        new_path = _find_matching_media(clip.name, folder)

        if not new_path:
            continue

        clip.media_reference = otio.schema.ExternalReference(
            target_url="file://" + new_path,
            available_range=None  # we don't know the available range
        )

```
```python
import opentimelineio as otio

def main():
    args = parse_args()

    timeline = otio.adapters.read_from_file(args.input)
    count = _conform_timeline(timeline, args.folder)
    otio.adapters.write_to_file(timeline, args.output)

def _conform_timeline(timeline, folder):
    """Look for replacement media for each clip in the given timeline.
    The clips are relinked in place if media with a matching name is found.
    """

    for clip in timeline.each_clip():
        # look for a media file that matches the clip's name
        new_path = _find_matching_media(clip.name, folder)

        # if no media is found, keep going
        if not new_path:
            continue

        # if we found one, then relink to the new path
        clip.media_reference = otio.schema.ExternalReference(
            target_url="file://" + new_path,
            available_range=None  # we don't know the available range
        )
```

Python API Example
```python
import opentimelineio as otio

def main():
    args = parse_args()

    timeline = otio.adapters.read_from_file(args.input)
    count = _conform_timeline(timeline, args.folder)
    otio.adapters.write_to_file(timeline, args.output)

def _conform_timeline(timeline, folder):
    """Look for replacement media for each clip in the given timeline. The clips are relinked in place if media with a matching name is found."

    for clip in timeline.each_clip():
        new_path = _find_matching_media(clip.name, folder)

        if not new_path:
            continue

        clip.media_reference = otio.schema.ExternalReference(
            target_url="file://" + new_path,
            available_range=None  # we don't know the available range
        )
```

Python API Example

```python
import opentimelineio as otio

def main():
    args = parse_args()

    timeline = otio.adapters.read_from_file(args.input)
    count = _conform_timeline(timeline, args.folder)
    otio.adapters.write_to_file(timeline, args.output)

def _conform_timeline(timeline, folder):
    """ Look for replacement media for each clip in the given timeline.
    The clips are relinked in place if media with a matching name is found.
    """

    for clip in timeline.each_clip():
        # look for a media file that matches the clip's name
        new_path = _find_matching_media(clip.name, folder)

        # if no media is found, keep going
        if not new_path:
            continue

        # if we found one, then relink to the new path
        clip.media_reference = otio.schema.ExternalReference(
            target_url="file://" + new_path,
            available_range=None  # we don't know the available range
        )
```
```python
import opentimelineio as otio

def main():
    args = parse_args()

    timeline = otio.adapters.read_from_file(args.input)
    count = _conform_timeline(timeline, args.folder)
    otio.adapters.write_to_file(timeline, args.output)

def _conform_timeline(timeline, folder):
    """Look for replacement media for each clip in the given timeline. The clips
    are relinked in place if media with a matching name is found."

    for clip in timeline.each_clip():
        # look for a media file that matches the clip's name
        new_path = _find_matching_media(clip.name, folder)

        # if no media is found, keep going
        if not new_path:
            continue

        # if we found one, then relink to the new path
        clip.media_reference = otio.schema.ExternalReference(
            target_url="file://" + new_path,
            available_range=None  # we don't know the available range
        )
```
Adapter Plugins

- Adapters
  - Read and/or Write formats
  - 4 functions
  - Associate w/ file suffixes
OTIO Adapters

Core

Contrib
OTIO Adapters

Contrib

AAF

RV Session

Maya Sequencer

ALE

FFmpeg Burnins

HLS playlist
OTIO Adapters

source: https://xkcd.com/927/
Adapter Plugins

- Adapters
  - Read and/or Write formats
  - 4 functions
  - Associate w/ file suffixes
Media Linker Plugins

- Adapter may not have enough information to link to media
- Logic to link to actual media on disk is usually proprietary
  - EDL Comments?
  - Production database lookup based on metadata?
adapter.read_from_file

find adapter \rightarrow call read function \rightarrow call media linker
Media Linker Plugins

- Adapter creates ‘missing references’
- calls $OTIO_DEFAULT_MEDIA_LINKER before returning resulting object
Media Linker Plugins

- Adapter creates ‘missing references’
- calls $OTIO_DEFAULT_MEDIA_LINKER before returning resulting object
Media Linker Plugins

- Adapter creates ‘missing references’
- calls `$OTIO_DEFAULT_MEDIA_LINKER` before returning resulting object
Media Linker Example

```
metadata: {
  pixar: {
    globaltake: "1",
    prod: "PXR",
    shot: "PXR_HeadLeader_AllRatios",
    unit: "studio"
  }
},

media_reference: {
  OTIO_SCHEMA: "ExternalReference.1",
  available_range: null,
  metadata: {},
  name: null,
  target_url: "/shows/studio/diskfarm/T"
},
```
def link_media_reference(in_clip, media_linker_argument_map):

Media Linker Plugins

- Media Linkers
  - Optional
  - Run after the adapter has read the file
  - Alter the media references
Python API

- JSON Backed File Format
- Python API
  - Opentime Library
  - Adapter Plugins
  - Media Linker Plugins
Status Update
OTIO Status Update

- Since 2016
  - Contributions from 10 different groups
  - 8 releases
- Used in production
  - Pixar, Marvel Studios, ILM, PIX System, and a few more
OTIO Status Update

- “Open Beta”
OTIO Status Update

- Public Beta 0.7.1
  - [https://github.com/PixarAnimationStudios/OpenTimelineIO](https://github.com/PixarAnimationStudios/OpenTimelineIO)
  - [http://opentimeline.io](http://opentimeline.io)
  - pip install OpenTimelineIO
License

- Pixar Open Source License
- Modified Apache 2.0 License
- Same as OpenSubdiv & USD
- Need a Contributor License Agreement (CLA) to contribute code
OTIO Use Case: Pixar Coco (2017)

Editorial → EDL → OTIO → Review
OTIO Use Case: Pixar Coco (2017)

- Avid EDL to RV
- Relink media to latest renders
- Transitions
OTIO Use Case: Pixar 2018

Editorial → AAF → OTIO → Reviews → OTIO → Animation → OTIO → Grading

Cross Dissolves
Roadmap
OTIO Roadmap

- Driven by studio use cases
- Community contributions
OTIO Roadmap

- EDL * raise EDLParseError
  #244 opened 2 days ago by col-one

- Add a `__version__` field to the top level __init__.py modules
  #242 opened 7 days ago by ssteinbach  Public Beta 8

- Put OTIO on PyPi
  #239 opened on Mar 13 by swallitsch  Public Beta 8

- AAF adapter should support writing compositions
  #236 opened on Feb 22 by jminor  1.0 Release

- Investigate whether types in otio.schema should be treated like reference types rather than value types
  #228 opened on Feb 8 by ssteinbach  Public Beta 8

- Convenience for creating/adding a Gap, black, etc.
  #226 opened on Feb 7 by jminor

- It should be easy to work with a Track that starts at non-zero time.
  #225 opened on Feb 7 by jminor
OTIO Roadmap

- Next Steps:
  - AAF Adapter
  - Speed Effects
OTIO Roadmap

- Longer Term:
  - AAF Creation
  - Subtitles, more effects, etc.
  - C++ API w/ Python Bindings
  - Native Integration w/ 3rd Party Apps (NLEs, Shotgun, etc.)
OTIO Contributions

- Axis Animation: Premiere adapter
- Lucasfilm: ffmpeg burnin adapter
- PIX System: HLS Live Stream adapter
OTIO Call for Contributions

- Use cases written up on the wiki
- Sample EDLs, AAFs, XMLs, etc.
- Engagement on C++ API for native support in 3rd party tools
- Adapters for IMF/CPL, FCP X, etc.
Discussion
Follow up

- We’re eager to help you get started and/or hear your use cases.
- To schedule something, contact opentimelineio@pixar.com
Thank You

- Stephan Steinbach - steinbach@pixar.com - @stephan_gfx
- Joshua Minor - joshm@pixar.com - @jminor
- Kyle McDaniel - kmcdaniel@pixar.com
- opentimelineio@pixar.com
- http://opentimeline.io