

# Final Cut Pro X XML Format



# Contents

## About Final Cut Pro X XML Version 1.5 6

At a Glance 6

See Also 10

## Events and Projects 11

About Events and Projects 11

Collections 15

Import Options 16

## Story Elements 18

Structure 18

Timeline 19

Annotations and Notes 19

Timing Attributes 20

Time Maps 21

Rate Conforming 21

Audio Components 22

## Resources 25

Structure 25

Media Formats 26

Bookmark Data 27

Multicam Media 28

Using Multicam Media 30

Audio Components and Multicam 30

## Adjustments and Effects 32

Adjustments 32

Audible Adjustments 32

Visual Adjustments 34

Effects 36

Filters 36

Transitions 38

Text Styles 39

Adjustment Attributes and Effect Parameters 41

Animation 42

**Metadata** 45

Elements 45

Keys and Sources 47

**Appendix A: FCPXML Supported Identifiers** 50

Rendering Formats 50

Video Formats 50

Blend Modes 55

Pan Modes 56

Metadata Keys 57

**Appendix B: FCPXML Version 1.5 DTD** 60

**Document Revision History** 83

# Tables and Listings

## Events and Projects 11

Table 1-1	Library and resources elements	12
Table 1-2	Collection elements	15
Table 1-3	Smart collection match elements	15
Table 1-4	Option element attributes	16
Listing 1-1	A project referencing audio and video in an external event	12
Listing 1-2	An event with multiple clips	13

## Story Elements 18

Table 2-1	Story elements	18
Table 2-2	Timeline attributes	19
Table 2-3	Timeline audio attributes	19
Table 2-4	Annotation elements	20
Table 2-5	Note element	20
Table 2-6	Time value attributes	20
Table 2-7	Rate conforming	22
Table 2-8	Audio component elements	22

## Resources 25

Table 3-1	Resource elements	25
Table 3-2	Format element attributes	26
Table 3-3	Bookmark element	27
Table 3-4	Multicam media elements	29
Table 3-5	Multicam source element	30
Listing 3-1	Multicam definition with two angles	29
Listing 3-2	Multicam project with two multicam clips and reconfigured audio	31

## Adjustments and Effects 32

Table 4-1	Audible adjustment elements	33
Table 4-2	Visual adjustment elements	35
Table 4-3	Filter elements	37
Table 4-4	Mask elements	37
Table 4-5	Text style elements	39
Table 4-6	Parameter element attributes	41

Table 4-7      Data element   42

Table 4-8      Keyframe animation elements   42

**Metadata**   45

Table 5-1      Metadata elements   45

# About Final Cut Pro X XML Version 1.5

The Final Cut Pro X XML (FCPXML) format lets you import and export XML documents between Final Cut Pro X and third-party applications. FCPXML v1.5 requires Final Cut Pro X v10.2 or later.

FCPXML describes certain aspects of projects and events that are useful for exchanging with other applications. It does not describe all possible data, and therefore is not a substitute for the native project and event data organized in a library bundle.

In this document, it is assumed that you understand XML and have used Final Cut Pro X.

---

**Note:** Starting with Final Cut Pro X v10.1, project and event data is organized in a library bundle. Refer to the [Managing Media with Final Cut Pro X Libraries](#) for more information on Final Cut Pro Libraries.

---

## At a Glance

You use the Final Cut Pro X application to create, edit, and produce high-quality video. With the FCPXML file format, you can transfer the details of your events and projects between Final Cut Pro X and third-party applications, devices, and media asset management tools that do not natively recognize Final Cut Pro X events or projects.

You can use Final Cut Pro X to export and import FCPXML documents to accomplish the following tasks:

- Exchange Final Cut Pro X event and project data with other applications.
- Create new Final Cut Pro X events and projects.

FCPXML v1.5 includes the following changes:

- Collections—Added support for smart collections in Final Cut Pro X libraries.

---

**Relevant Section:** [Collections](#) (page 15)

---

- Video filter masks—Now applies video filters to a region defined by a collection of masks.

---

**Relevant Section:** [Masked Filters](#) (page 37)

---

- Color filtering and adjustments—Now describes color correction, color matching, and color balance effects using the <filter-video> element instead of the <adjust-color> and <color-filter> elements.

---

**Relevant Section:** [Color Filters](#) (page 38)

---

- 3D text—Now applies three dimensional text appearance to titles.

---

**Relevant Section:** [Text Styles](#) (page 39)

---

- Structured parameter data—Describes three dimensional text and other effects.

---

**Relevant Section:** [Parameters](#) (page 41)

---

- ASC CDL color information—Now exports as an XML comment the primary color correction definition in American Society of Cinematographers Color Decision List (ASC CDL) format.

---

**Relevant Content:** See the [Note](#) (page 38) in the Color Filters section.

---

FCPXML v1.4 included the following changes:

- Structural changes reflecting the organization of Final Cut Pro X libraries.

---

**Relevant Chapter:** [Events and Projects](#) (page 11)

---

- Change in specifying keyframe animations.

---

**Relevant Section:** [Animation](#) (page 42)

---

- Visual adjustments on a multicam clip must now be specified on individual angles instead of the entire clip.

---

**Relevant Section:** [Multicam Media](#) (page 28) and [Visual Adjustments](#) (page 34)

---

FCPXML v1.3 added support for the following Final Cut Pro X features:

- **Adjustments**
  - Color adjustments—Define and preserve color adjustments, such as balancing, matching, and correction.
  - Audio adjustments—Define audio component enhancements, such as loudness, noise and hum reduction, and equalization.
  - Video adjustments—Define the stabilization and rolling shutter reduction applied to a clip.

---

**Relevant Section:** [Adjustments](#) (page 32)

---

- **Effects**
  - Text styles—Define text style effects for titles, such as font, size, color, alignment, spacing, style, stroke, baseline, and kerning.
  - Color filtering—Apply color filter effects to a clip.

---

**Relevant Section:** [Effects](#) (page 36)

---

- **Import options**—Control certain options during XML import.

---

**Relevant Section:** [Import Options](#) (page 16)

---

- **Collections**—Group clips and projects using folders, keywords, and other search criteria.

---

**Relevant Section:** [Collections](#) (page 15)

---



---

**Note:** In FCPXML v1.3, some attributes and elements are renamed or removed. Take care if you plan to support both v1.3 and earlier versions. In particular, the `<filter>` element is replaced by the `<filter-audio>` and `<filter-video>` elements, depending on the type of the filter. Also the `<timeMap>` element representing rate conforming (with the `rateConform` attribute) is replaced by the `<conformRate>` element. Refer to the DTDs for prior versions in [Legacy DTDs for Final Cut Pro X XML](#) for specific changes.

---

FCPXML v1.2 added support for the following Final Cut Pro X features:

- Audio Component Editing—Define how audio components are used (for example, channel mappings, enabled, role, effects, and mute ranges).

---

**Relevant Section:** [Audio Components](#) (page 22)

---

- Metadata Export/Import—Import and export metadata associated with media assets or clips in an event or project.

---

**Relevant Chapter:** [Metadata](#) (page 45)

---

FCPXML v1.1 added support for the following Final Cut Pro X features:

- Multicam editing—Manage media from multiple cameras or other synchronized footage.

---

**Relevant Chapter:** [Resources](#) (page 25)

---

- Adjustments
  - Video and image transformation—Crop, trim, pan and zoom, distort, conform, move, resize, rotate, and blend videos and images.
  - Audio adjustment—Adjust audio volume and panning.

---

**Relevant Chapter:** [Adjustments and Effects](#) (page 32)

---

- Retrieve ASC CDL color information—Export the first primary color correction definition in American Society of Cinematographers Color Decision List (ASC CDL) format.

---

**Note:** In FCPXML v1.5, this feature was changed and ASC CDL color information is now exported as an XML comment. See the [Note \(page 38\)](#) in the Color Filters section.

---

## See Also

The following resources may be helpful as you work with the FCPXML format:

- [Final Cut Pro X Help](#)
- [Final Cut Pro X Resources](#)
- See [Legacy DTDs for Final Cut Pro X XML](#) for prior DTD versions.
- The [Extensible Markup Language \(XML\) 1.0](#) specification

# Events and Projects

In Final Cut Pro X, you use FCPXML (.fcpxml) documents to export and import events and projects in a library. An FCPXML document represents a library or a portion of a library.

Here is a list of Final Cut Pro X terms you'll encounter throughout this document:

- A *clip* is a reference to media, such as a video, audio, or still image file, that allows you to edit and annotate the media without directly modifying it. A clip controls which portions of the media you would like to use, and it allows you to organize the media based on keywords you have applied. Clips can also contain other clips to represent composite media.
- Use a Final Cut Pro X *event* to store and organize media. You can import media files into a new or existing event. You can choose to copy these files into an event's own media folder, or reference them in their original locations. Final Cut Pro X tracks each imported file as an asset and ensures your event contains at least one clip per asset.
- Use a Final Cut Pro X *project* to build a finished movie. A project has a primary container called a sequence that defines your movie's final appearance. You build a sequence by bringing clips into it from one or more events, or by creating new clips within the sequence. You adjust and arrange the clips, along with other story elements in the sequence, to produce your movie. Every clip in a project is unique to that project (not shared), but referenced media always resides in an event and may be shared across more than one project.
- Use a Final Cut Pro X *library* to organize your events. Think of a library as a container that you use to keep track of all events, projects, and media related to a particular assignment.

## About Events and Projects

An FCPXML document contains a root `<fcpxml>` element as the top-level container. That root element contains a `<resources>` element and a `<library>` element, with the `<library>` element containing a list of `<event>` elements.

---

**Note:** As of FCPXML v1.4, an FCPXML document is *self-contained* and allows multiple events, but does not allow references to objects in another library.

---

An `<event>` element can contain:

- A `<project>` element representing a timeline in a Final Cut Pro X project

---

**Note:** As of FCPXML v1.4, the `<project>` element now only defines a project timeline; events are now represented by an `<event>` element.

---

- A set of `<clip>`, `<audition>`, `<mc-clip>`, and `<ref-clip>` elements for a Final Cut Pro X event

See [Story Elements](#) (page 18) for further details on these elements.

The `<fcpxml>` element may include the child elements listed in Table 1-1.

**Table 1-1** Library and resources elements

Element	Description
<code>&lt;event&gt;</code>	A single event in a library.
<code>&lt;library&gt;</code>	A single library. The <code>location</code> attribute specifies the URL of the library on export, but is ignored during import. Refer to the <code>library location</code> import option in <a href="#">Import Options</a> (page 16).
<code>&lt;project&gt;</code>	A project timeline.
<code>&lt;resources&gt;</code>	A container for relevant data that events and projects depend on, such as media assets, video formats, and effects. See <a href="#">Resources</a> (page 25) for details on the supported resource types.

Listing 1-1 shows an FCPXML project.

**Listing 1-1** A project referencing audio and video in an external event

```
<fcpxml version="1.5">
  <!-- Resources -->
  <resources>
    <asset id="r1" src="file://Volumes/Media/MyVideo.mov"/>
    <format id="r2" name="FFVideoFormat1080p30"/>
  </resources>
  <library>
    <event name="MyEvent">
      <project name="MyProject">
```

```
        <!-- Project Story Elements -->
        <sequence format="r2">
            <spine>
                <video ref="r1" duration="5s">
                    <audio lane="-1" ref="r1" duration="5s"/>
                </video>
            </spine>
        </sequence>
    </project>
</event>
</library>
</fcpxml>
```

In Listing 1-1, the `MyProject` project uses two resources:

- A reference to an external asset located at `/Volumes/Media/MyVideo.mov`
- A built-in video format, `FFVideoFormat1080p30`

When Final Cut Pro X imports the file in Listing 1-1, it:

- Creates a new event called `MyEvent` in the library the user chooses.
- Imports `MyVideo.mov` into `MyEvent` by either copying or linking, depending on the application preference.
- Creates a new project called `MyProject` with a sequence using a 1080p, 30 fps video format specified by `FFVideoFormat1080p30`.
- Inserts the first 5 seconds of video from `MyVideo.mov` into the project timeline.
- Anchors the first 5 seconds of audio from `MyVideo.mov` one lane below the video. See [Story Elements](#) (page 18) for details on the `lane` attribute.

---

**Note:** You can use import options to specify the library in which the new event is created or override the preference for copying versus linking media. See [Import Options](#) (page 16).

---

Listing 1-2 shows an FCPXML event.

**Listing 1-2** An event with multiple clips

```
<fcpxml version="1.5">
```

```
<!-- Resources -->
<resources>
  <format id="r1" name="FFVideoFormat1080p30"/>
  <asset id="r2" src="file://Volumes/Media/MyVideo1.mov"/>
  <asset id="r3" src="file://Volumes/Media/MyVideo2.mov"/>
  <asset id="r4" src="file://Volumes/Media/MyVideo3.mov"/>
</resources>
<library>
  <event name="MyEvent">
    <!-- Clip Elements -->
    <clip name="MyVideo1" duration="45s" format="r1">
      <video ref="r2" duration="45s">
        <audio lane="-1" ref="r2" duration="45s"/>
      </video>
    </clip>
    <clip name="MyVideo2" duration="15s" format="r1">
      <video ref="r3" duration="15s">
        <audio lane="-1" ref="r3" duration="15s"/>
      </video>
    </clip>
    <clip name="MyVideo3" duration="30s" format="r1">
      <video ref="r4" duration="30s">
        <audio lane="-1" ref="r4" duration="30s"/>
      </video>
    </clip>
  </event>
</library>
</fcpxml>
```

In Listing 1-2, the event `MyEvent` uses these resources:

- Three external assets named `MyVideo1.mov`, `MyVideo2.mov`, and `MyVideo3.mov` located at `/Volumes/Media/`
- A built-in video format, `FFVideoFormat1080p30`

When Final Cut Pro X imports this file, it:

- Creates a new event called `MyEvent` in the library you choose.
- Imports `MyVideo1.mov`, `MyVideo2.mov`, and `MyVideo3.mov` into `MyEvent` by either copying or linking, depending on the application preference.
- Creates three clips (`MyVideo1`, `MyVideo2`, and `MyVideo3`) containing audio and video data from the imported media and using a 1080p, 30 fps video format specified by `FFVideoFormat1080p30`.

---

**Note:** You can use import options to specify the library in which the new event is created or override the preference for copying versus linking media. See [Import Options](#) (page 16).

---

## Collections

An event may group clips into keyword and smart collections, and a library may group clips into smart collections. Collections are represented by the elements in Table 1-2 as children of the `<event>` and `<library>` elements.

---

**Note:** Only `<smart-collection>` elements can appear as child elements of the `<library>` element.

---

**Table 1-2** Collection elements

Element	Description
<code>&lt;collection-folder&gt;</code>	A container to group other collection elements.
<code>&lt;keyword-collection&gt;</code>	Group clips and projects using keywords.
<code>&lt;smart-collection&gt;</code>	Group clips and projects matching various search criteria. Matching criteria elements are listed in <a href="#">Table 1-3</a> (page 15).

Smart collections use the elements listed in Table 1-3 to specify matching criteria for grouping clips and projects.

**Table 1-3** Smart collection match elements

Element	Description
<code>&lt;match-clip&gt;</code>	Match clips based on type (one of <code>audition</code> , <code>synchronized</code> , <code>compound</code> , <code>multicam</code> , <code>layeredGraphic</code> , or <code>project</code> ).
<code>&lt;match-keywords&gt;</code>	Match clips based on assigned keywords.

Element	Description
<match-media>	Match clips or projects based on media type (one of audioOnly, stills, videoOnly, or videoWithAudio).
<match-property>	Match clips or projects based on format/property information.
<match-ratings>	Match clips based on assigned ratings.
<match-roles>	Match clips based on assigned roles.
<match-shot>	Match clips based on assigned shot type.
<match-stabilization>	Match clips based on the stabilization type.
<match-text>	Match clips or projects based on the specified text.
<match-time>	Match clips or projects created or imported based on a specific date.
<match-timeRange>	Match clips or projects created or imported based on a range of time.

## Import Options

The <import-options> element, as a child of the <fcpxml> element, can contain zero or more <option> elements pertaining to importing events and projects.

The <option> element attributes are defined in Table 1-4.

**Table 1-4** Option element attributes

Attribute	Description
key	<p>A string that identifies one of the following import options:</p> <ul style="list-style-type: none"><li>• <code>copy assets</code>—Copy (1) or link (0) assets referenced in the imported XML.</li><li>• <code>copy assets location</code>—The location (URL) to copy the imported assets to, if <code>copy assets</code> is set to 1. If the specified URL is invalid, the assets are not copied.</li><li>• <code>library location</code>—The location (URL) of the library to which to add the event or project. If the specified URL represents a directory, the default library name is used. If no library exists as the location specified, a new library is created.</li><li>• <code>suppress warnings</code>—Suppress (<math>\geq 1</math>) or show (0) warnings generated during import.</li></ul>



Attribute	Description
value	The value for the import option.

# Story Elements

Story elements are the building blocks of your Final Cut Pro X events and projects.

## Structure

An event or project may include the story elements listed in Table 2-1.

**Table 2-1** Story elements

Element	Description
<clip>	A container for editing/compositing media and other story elements.
<audio>	A reference to audio data from an <asset> or <effect>.
<video>	A reference to video data from an <asset> or <effect>.
<mc-clip>	A reference to audio/video data from a <multicam> media source.
<ref-clip>	A reference to audio/video data from a <sequence> media source.
<gap>	A placeholder element with no intrinsic audio or video data.
<transition>	An effect that combines zero, one, or two neighboring elements.
<title>	An effect with custom text elements.
<spine>	A container that schedules elements sequentially in time.
<audition>	A container of alternative elements, exactly one of which is currently active. When exported, the XML lists the currently active item as the first child in the audition container.
<sequence>	A container representing the top-level sequence for a Final Cut Pro X project or compound clip.

You can *anchor* one or more other elements to each story element. An anchored item has a positive or negative lane index that positions the item either above or below its base element in the timeline.

For video elements, lane order also implies compositing order—items with higher lane indexes composite over elements with lower lane indexes. For audio elements, lane order does not affect compositing. Items that reside inside (rather than above or below) a container are called *contained items* and have an implied lane index of zero.

Events and projects can also contain filters that apply effects to certain elements. See [Effects](#) (page 36).

## Timeline

The `<sequence>` and `<clip>` story elements, and the `<multicam>` media element create a new timeline and use the attributes listed in Table 2-2 to define its characteristics.

**Table 2-2** Timeline attributes

Attribute	Description
format	A reference to the video format defined by the <code>&lt;format&gt;</code> element.
tcFormat	The timecode display format, either drop frame (DF) or nondrop frame (NDF, the default).
tcStart	The timecode origin represented as a time value. See <a href="#">Timing Attributes</a> (page 20).

The `<sequence>` element can also contain the attributes in Table 2-3 to define the audio characteristics of the new timeline:

**Table 2-3** Timeline audio attributes

Attribute	Description
audioLayout	The audio channel layout, one of <code>mono</code> , <code>stereo</code> (the default), or <code>surround</code> .
audioRate	The audio sample rate (one of 32k, 44.1k, 48k, 88k, 96k, 176.4k, or 192k); the default is 48k.

---

**Note:** The `audioLayout` attribute for the top-level `<sequence>` element in a project must be either `stereo` or `surround`.

---

## Annotations and Notes

Events and projects can contain the annotations (keyword, markers, and so on) in Table 2-4. Many story elements can contain annotations over a range of time, specified with the `start` and `duration` attributes.

**Table 2-4** Annotation elements

Element	Description
<keyword>	An annotation that applies a keyword to a range of time.
<marker>	An annotation (with optional <i>to-do</i> flag) that applies a text marker to a range of time.
<chapter-marker>	An annotation that applies to a range of time and indicates a chapter.
<analysis-marker>	An annotation that indicates the result of scene analysis to determine the number of people present and the shot and stabilization types.
<rating>	A favorite or reject annotation that applies to a range of time.

Descriptive notes apply to an entire element and are defined by the <note> element.

**Table 2-5** Note element

Element	Description
<note>	An annotation that applies to an entire element.

## Timing Attributes

Each story element has up to three time values, listed in Table 2-6, that schedule it (and its contained or anchored items) in a timeline.

**Table 2-6** Time value attributes

Attribute	Description
offset	An element's location in parent time (or base element's time for an anchor).
duration	An element's extent in parent time.
start	The start of an element's local timeline (to schedule its contained and anchored items).

The time range for contained items is limited by their parent. The time range for anchored items is not limited by their parent; however, anchored items may be limited by an ancestor that either directly or indirectly contains the parent.

Time values are expressed as a rational number of seconds with a 64-bit numerator and a 32-bit denominator. Frame rates for NTSC-compatible media, for example, use a frame duration of  $1001/30000$ s (29.97 fps) or  $1001/60000$ s (59.94 fps). If a time value is equal to a whole number of seconds, the fraction may be reduced into whole seconds (for example, 5s).

Timing attribute values are expected to be a multiple of the frame duration for the respective timeline. Otherwise, Final Cut Pro X has to insert a gap to maintain the specified timing upon import. A warning message appears when this happens.

## Time Maps

Use a time map to adjust, or remap, the playback speed of an element. A time map is defined with the `<timeMap>` element, which contains one or more `<timept>` elements that specify the new timing. The parent element's playback speed is adjusted over a range of time using points on a curve to interpolate values. For example, if the local time range for a clip is originally 0-5s, then the time map for playing this clip at 50 percent speed, followed by playback at -50 percent speed, might look as follows:

```
<timeMap>
  <timept time="0s" value="0s" interp="linear"/>
  <timept time="10s" value="5s" interp="linear"/>    <!-- In 0-10s, play the
original 0-5s -->
  <timept time="20s" value="0s" interp="linear"/>    <!-- In 10-20s, play the
original 5-0s -->
</timeMap>
```

A time map affects only an element's own duration and the offsets of the element's anchored items; it does not modify the offsets of the element's contained items (such as items with `lane=0`).

## Rate Conforming

Final Cut Pro X automatically applies rate conforming when the timeline frame rate and the media frame rate are certain combinations (see Table 2-7) by converting the media frame rate to match the timeline frame rate. As a result, the duration is also adjusted. When this occurs, the Rate Conform section appears in the application's Video Inspector. In XML, rate conforming is indicated using the `<conform-rate>` element.

---

**Note:** In FCPXML v1.2, rate conforming was applied using a `<timeMap>` element with a `rateConform` attribute value of 1.

---

Table 2-7 illustrates how Final Cut Pro X applies rate conforming.

**Table 2-7** Rate conforming

Media Frame Rate	Timeline Frame Rate				
	23.98p	24p	25p 25i 50p	29.97p 59.94p 29.97i	30p 60p
23.98p	-	24p	25p	-	-
24p	23.98p	-	25p	-	-
25p	23.98p	24p	-	-	-
29.97p	-	-	-	-	30p
30p	-	-	-	29.97p	-
50p	47.96p	48p	-	-	-
59.94p	-	-	-	-	60p
60p	-	-	-	59.94p	-
25i	47.96p	48p	-	-	-
29.97i	-	-	-	-	60p

## Audio Components

For clips with audio, specify the configuration (such as channel mapping, effects, roles, mute ranges, and whether the configuration is enabled) at the component level using the audio component elements listed in Table 2-8.

**Table 2-8** Audio component elements

Element	Description
<audio-source>	Defines the configuration for a single audio component from the contained items.
<audio-aux-source>	Defines the configuration for a single audio component from the anchored items.
<mute>	Suppresses audio output for the audio component over a range of time.

The `<audio-source>` and `<audio-aux-source>` elements are very similar, the only difference being that the `<audio-source>` element specifies the configuration for the primary timeline items, while the `<audio-aux-source>` element applies to the items anchored to the primary timeline. The configuration, either specified as an attribute or child element of the `<audio-source>` element, applies to the primary timeline items of the contained clip item. The configuration of the `<audio-aux-source>` element applies to the items anchored to the primary timeline of the contained clip.

You can use `<mute>` or `<filter-audio>` elements as children to specify mute ranges or audio effects.

The channel configuration is specified through the `srcCh` and `outCh` attributes of the `<audio-source>` and `<audio-aux-source>` elements.

- The `srcCh` attribute specifies the source channel indices in the order they appear in the output, such as "1, 2".
- The `outCh` attribute specifies the channel tags of the output in the order they appear in the output, such as "L, R".

In the following example, the audio from the primary timeline is reconfigured as dual mono, while the audio from the anchored timeline is mapped to reverse stereo. Additionally, the last 20 seconds of the anchored audio is muted.

```
<clip duration="120s">
  <clip duration="120s">
    <!-- Primary Timeline -->
    <video ref="r2" duration="120s">
      <audio lane="-1" ref="r2" duration="120s" outCh="L, R"/>
    </video>
    <!-- Anchored Item -->
    <clip lane="-1" duration="120s">
      <audio ref="r3" duration="120s" outCh="L, R"/>
    </clip>
  </clip>
  <audio-source name="mono 1" srcCh="1"/>
  <audio-source name="mono 2" srcCh="2"/>
  <audio-aux-source name="reverse stereo" srcCh="2, 1">
    <mute start="100s" duration="20s"/>
  </audio-aux-source>
</clip>
```

---

**Note:** In the example above, the `<audio-aux-source>` element does not have an `outCh` attribute because reversing the order of the source channels is sufficient to reconfigure the channel configuration to reverse stereo. In more general cases, both `srcCh` and `outCh` are used to describe the channel mapping.

Also, the `<audio-aux-source>` element applies to the items anchored to the contained item, not the items anchored to the parent element.

---



# Resources

An FCPXML document contains multiple references to shared resources, such as media files, video formats, and effects. You define references to shared resources in the `<resources>` element, a child of the `<fcpxml>` element.

---

**Note:** In FCPXML v1.4, the `<resources>` element is an immediate child of the `<fcpxml>` element, and contains the resources used by events included in the XML.

In FCPXML v1.3 and earlier, the `<resources>` element is a child of the `<project>` element, and contains the resources used by that project.

---

## Structure

The `<resources>` element can include the child elements in Table 3-1.

**Table 3-1** Resource elements

Element	Description
<code>&lt;asset&gt;</code>	A reference to a media file managed by a library.
<code>&lt;effect&gt;</code>	A reference to an effect plug-in (for example, FxPlug, Motion document, or Audio Unit).
<code>&lt;format&gt;</code>	A reference to a Final Cut Pro X video format definition.
<code>&lt;media&gt;</code>	A reference to a new or existing media definition in a library. The child element, either <code>&lt;multicam&gt;</code> or <code>&lt;sequence&gt;</code> , specifies whether this is a multicam or compound clip.

Each resource has an `id` attribute that specifies a *local identifier* for use within the FCPXML document. For example, an asset resource with a local identifier of `r1` might be declared as follows:

```
<asset id="r1" name="MyMovieFile" src="file://path/to/MyMovieFile.mov"/>
```

The `src` attribute is expected to be a string that specifies an absolute file URL conforming to RFC 2396.

Later in the same FCPXML document, a video or audio element that uses the asset can refer to it by its local identifier.

```
<video ref="r1" ... >
```

---

**Note:** Third-party tools may use any string for a local identifier as long as the string begins with a letter and no two resources in the same FCPXML document share the same local identifier.

---

For most resources, Final Cut Pro X also assigns a lengthier *unique identifier* that it writes when exporting the FCPXML document. This unique identifier is stored as a `uid` attribute in the resource.

---

**Note:** Unique identifiers managed by Final Cut Pro X should not be generated or altered by third-party tools.

---

Final Cut Pro X has both a local identifier and a unique identifier so that it can distinguish between resources that already exist and resources that are imported for the first time. Exported FCPXML documents that refer to existing resources have unique identifiers assigned by Final Cut Pro X, whereas imported FCPXML documents containing new resources only have local identifiers. A local identifier provides an identity for a new resource in an FCPXML document before Final Cut Pro X assigns it a permanent unique identifier.

## Media Formats

The `<format>` element specifies the video format for a timeline. This element has a reference to one of the predefined video formats listed in [Video Formats](#) (page 50) through the `name` attribute, or it can describe a custom format using the attributes in Table 3-2.

**Table 3-2** Format element attributes

Attribute	Description
<code>fieldOrder</code>	The field order for interlaced (upper first or lower first), or progressive video (progressive).
<code>frameDuration</code>	The frame duration as a time value.
<code>height</code>	The video frame height, in pixels.
<code>pasph</code>	The relative width of a pixel when video is encoded with non-square pixels.

Attribute	Description
paspv	The relative height of a pixel when video is encoded with non-square pixels.
width	The video frame width, in pixels.

## Bookmark Data

A security-scoped bookmark, associated with an `<asset>` element, is required for sandboxed applications that need to access a media asset (file-system resource) outside the sandboxed environment. Refer to the *Security-Scoped Bookmarks and Persistent Resource Access* section in the *App Sandbox Design Guide* for more information.

Because of this requirement, Final Cut Pro X creates a document-scoped bookmark (see Table 3-3) for each media asset the user grants access to, and includes this bookmark data in the asset element when exporting an FCPXML event or project.

**Table 3-3**     Bookmark element

Element	Description
<code>&lt;bookmark&gt;</code>	A Base64-encoded string that represents security-scoped bookmark data. It is used by a sandboxed application to reference an asset outside the sandboxed environment.  A bookmark is associated with an <code>&lt;asset&gt;</code> element.

Follow these steps to use a security-scoped bookmark in a sandboxed application:

1. Decode the bookmark data.  
  
The bookmark data in the bookmark element is Base64-encoded so you must first decode the bookmark before it can be used. You can use the *Security Transforms API* to encode or decode Base64 data—refer to *Security Transforms Programming Guide* for more details.
2. Resolve the decoded bookmark data into a security-scoped URL that points to the media asset.
3. Indicate that you want to use the security-scoped URL to access the file-system resource.
4. Access the file-system resource using the security-scoped URL in your application.
5. Release the file-system resource.

These steps are outlined in the following code snippet (refer to *NSURL Class Reference* for details on the NSURL methods used):

```
// Decode the Base64 data using the Security Transforms API.
NSData* decodedBookmark = yourBase64DataDecodeRoutine(bookmark);

// Resolve the decoded bookmark data into a security-scoped URL.
NSError* err = nil;
NSURL* url = [NSURL URLWithString:decodedBookmark
                    options:NSURLBookmarkResolutionWithSecurityScope
                    relativeToURL:sourceURL
                    bookmarkDataIsStale:nil
                    error:&err];

if (url) {
    // Indicate that you want to access the file-system resource.
    [url startAccessingSecurityScopedResource];

    // Use the resolved security scoped URL.
    ...

    // Release the file-system resource when you are done.
    [url stopAccessingSecurityScopedResource];
}
```

---

**Note:** To use security-scoped bookmarks, your application must have the `com.apple.security.files.bookmarks.document-scope` entitlement set to `true`. For more details see [Enabling Security-Scoped Bookmark and URL Access](#).

---

## Multicam Media

You can use a `<multicam>` media element to assemble footage from multiple cameras, or angles, that are synchronized in time. A `<multicam>` element contains one or more `<mc-angle>` elements that each manage a series of other story elements.

These multicam media elements are listed in Table 3-4.

**Table 3-4** Multicam media elements

Element	Description
<multicam>	A root container for <i>angles</i> of related media, organized as <mc-angle> elements.
<mc-angle>	A container of story elements organized sequentially in time.

The <multicam> element appears as a child element under a <media> element (as a resource).

Listing 3-1 shows a multicam definition with two angle definitions a1 and a2:

**Listing 3-1** Multicam definition with two angles

```
<resources>
  <asset id="r1" name="angle1" src="angle1.mov"/>
  <asset id="r2" name="angle2" src="angle2.mov"/>
  <format id="r3" name="FFVideoFormat720p2997"/>
  <media id="r4" name="MC Clip">
    <multicam format="r3">
      <mc-angle name="Angle1" angleID="a1">
        <clip duration="10s">
          <video ref="r1" duration="10s">
            <audio lane="-1" ref="r1" duration="10s"/>
          </video>
        </clip>
      </mc-angle>
      <mc-angle name="Angle2" angleID="a2">
        <clip duration="10s">
          <video ref="r2" duration="10s">
            <audio lane="-1" ref="r2" duration="10s"/>
          </video>
        </clip>
      </mc-angle>
    </multicam>
  </media>
</resources>
```

## Using Multicam Media

To use the multicam media as a clip, add an `<mc-clip>` element that references the media in the timeline.

Each `<mc-clip>` element can use audio and video from the same angle, or combine the audio and video from separate angles from the same `<multicam>` container. You use `<mc-source>` elements to specify which angle the audio, video, or both come from.

**Table 3-5** Multicam source element

Element	Description
<code>&lt;mc-source&gt;</code>	Specifies which angle a particular part of the clip comes from. This element has the following attributes: <ul style="list-style-type: none"><li><code>angleID</code>—Specifies the angle.</li><li><code>srcEnable</code>—Indicates which source to use, if any, from the angle (one of <code>audio</code>, <code>video</code>, <code>all</code>, or <code>none</code>).</li></ul>

The following example references the multicam media defined in [Listing 3-1](#) (page 29):

```
<mc-clip ref="r4" name="MC Clip" duration="10s">  
  <mc-source angleID="a1" srcEnable="all"/>  
</mc-clip>
```

---

**Note:** In FCPXML v1.1, you specified the angle using the `videoAngleID` and `audioAngleID` attributes of the `<mc-clip>` element. In FCPXML v1.2, you used the `<mc-source>` element's `angleID` attribute.

---

## Audio Components and Multicam

When you use multicam clips in your project, you can also specify the configuration of audio components in each clip, such as channel mapping, effects, roles, mute ranges, and whether the configuration is enabled. To do so, you use the `<audio-source>` element just as you do with regular clip elements.

Listing 3-2 is an example multicam project based on [Listing 3-1](#) (page 29), with two multicam clips whose audio components are reconfigured. The first clip has its audio reconfigured to dual mono, and the second clip has its audio reconfigured to reverse stereo with the last 2 seconds muted.

**Listing 3-2** Multicam project with two multicam clips and reconfigured audio

```
<project name="Multicam Project">
  <sequence format="r3">
    <spine>
      <mc-clip ref="r4" offset="0s" duration="5s" start="0s">
        <mc-source angleID="a1" srcEnable="all">
          <audio-source name="mono1" srcCh="1" outCh="L"/>
          <audio-source name="momo2" srcCh="2" outCh="R"/>
        </mc-source>
      </mc-clip>
      <mc-clip ref="r4" offset="5s" duration="5s" start="5s">
        <mc-source angleID="a1" srcEnable="audio">
          <audio-source name="reverse stereo" srcCh="2, 1">
            <mute start="3s" duration="2s"/>
          </audio-source>
        </mc-source>
        <mc-source angleID="a2" srcEnable="video"/>
      </mc-clip>
    </spine>
  </sequence>
</project>
```

---

**Note:** As of FCPXML v1.2, <mc-source> elements no longer support audio filters. Instead, you apply audio filters to an <mc-clip> element, which applies the effect to all active audio source angles, or to an <audio-source> element, which only applies the effect to the individual audio source, if active. Existing FCPXML v1.1 projects with audio angles on an <mc-clip> element are converted to v1.2 and later format on import.

---

# Adjustments and Effects

This chapter describes the FCPXML elements you use to apply adjustments and effects, alter parameters, and create animations.

## Adjustments

FCPXML supports adjustment elements to alter the audio and video output of story elements.

You can apply adjustment elements to control:

- Audible adjustments, such as audio volume and panning
- Visual adjustments, including geometrical video transformations, such as position, scale, rotation, crop, pan and zoom, blend mode, and size conforming

Each adjustment element has attributes that affect the adjustment you want to make. You specify the values to override, and all other values remain at their default settings. You can also create animated adjustments by varying attribute values over time. See [Animation](#) (page 42) for more details.

Some adjustment attributes expect a multidimensional (or compound) value, such as an (x,y) point. These attributes are specified with a space separating each value. For example, the two-dimensional values (0,0) or (10,-25) would be specified as "0 0" or "10 -25".

---

**Note:** When you work with multicam clips, audio adjustments are applied to the entire multicam clip (on to the `<mc-clip>` element), while visual adjustments are applied to each angle (on to the `<mc-source>` element). In FCPXML v1.3 and earlier, both visual adjustments and audio adjustments were applied to the entire multicam clip (on to the `<mc-clip>` element).

---

## Audible Adjustments

Audible adjustments control the audible aspects, such as volume and panning, of a story element. Table 4-1 lists the audible adjustment elements.



**Table 4-1** Audible adjustment elements

Element	Description
<adjust-EQ>	Specifies the equalization applied to a clip (one of <code>flat</code> , <code>voice_enhance</code> , <code>music_enhance</code> , <code>loudness</code> , <code>hum_reduction</code> , <code>bass_boost</code> , <code>bass_reduce</code> , <code>treble_boost</code> , or <code>treble_reduce</code> ).
<adjust-humReduction>	Applies hum reduction at the frequency, either 50 or 60, specified by the <code>frequency</code> attribute.
<adjust-loudness>	Modifies the loudness using the following attributes: <ul style="list-style-type: none"><li>• <code>amount</code>—The overall loudness amount.</li><li>• <code>uniformity</code>—The dynamic range to adjust.</li></ul>
<adjust-matchEQ>	Specifies the equalization, in an internal format, applied to match another clip's frequency characteristics.
<adjust-noiseReduction>	Applies noise reduction with the amount [0 . . 100] specified by the <code>amount</code> attribute.

Element	Description
<adjust-panner>	<p>Modifies the audio panning levels using the following attributes:</p> <ul style="list-style-type: none"><li>• <code>mode</code>—The pan mode; see <a href="#">Pan Modes</a> (page 56) for a list of supported values.</li><li>• <code>amount</code>—The pan amount.</li><li>• <code>original_decoded_mix</code>—The balance between the original and decoded signals.</li><li>• <code>ambient_direct_mix</code>—The balance between the decoded surround and center signals.</li><li>• <code>surround_width</code>—The separation, in dB, between the decoded surround signals.</li><li>• <code>left_right_mix</code>—The balance between the left and right speakers.</li><li>• <code>front_back_mix</code>—The balance between the front and back speakers.</li><li>• <code>LFE_balance</code>—The low-frequency effects signal.</li><li>• <code>rotation</code>—The rotation amount for the surround signals.</li><li>• <code>stereo_spread</code>—The stereo effect amount from the left and right channels to the center and surround speakers.</li><li>• <code>attenuate_collapse_mix</code>—The panning effect amount.</li><li>• <code>center_balance</code>—The balance between the center and surround signals.</li></ul>
<adjust-volume>	<p>Modifies the volume of a clip in dB (for example, <math>-6\text{dB}</math>).</p>

## Visual Adjustments

Visual adjustments control the visual aspects of a story element, such as cropping and other geometrical transformations, and degree of stabilization or rolling shutter reduction applied to the video. Table 4-2 lists the visual adjustment elements.

**Table 4-2** Visual adjustment elements

Element	Description
<adjust-blend>	<p>Modifies the compositing blend mode and opacity percentage [0.0 . 1.0] of the visible image using the following attributes:</p> <ul style="list-style-type: none"><li>• <code>amount</code>—A fractional value that specifies the opacity amount (for example, 0.75).</li><li>• <code>mode</code>—An integer value corresponding to a built-in FCP X blend mode; see <a href="#">Blend Modes</a> (page 55) for a list of supported values.</li></ul>
<adjust-conform>	<p>Modifies the image size of a clip to fill, fit, or remain unchanged within its container’s frame size. The <code>type</code> attribute specifies the conform method to apply, which defaults to <code>fit</code>.</p> <p>With the absence of this adjustment, the clip is <i>fit</i> into the frame size.</p>
<adjust-corners>	<p>Modifies the corners of the visible image, adding a distort (or skew) effect using the <code>botLeft</code>, <code>topLeft</code>, <code>topRight</code>, and <code>botRight</code> attributes. These attributes are two-dimensional (xy) values expressed as a percentage of frame height.</p>
<adjust-crop>	<p>Modifies the visible image width and height by cropping, trimming, or panning. This element has the following attributes:</p> <ul style="list-style-type: none"><li>• <code>enabled</code>—Enables (1, the default) or disables (0) this crop adjustment.</li><li>• <code>mode</code>—Specifies the active crop mode (one of <code>trim</code>, <code>crop</code>, or <code>pan</code>). Only one crop mode is active at any given time. Crop modes are defined via the following child elements, whose attribute values (<code>left</code>, <code>top</code>, <code>right</code>, and <code>bottom</code>) are expressed as a percentage of the original frame height:<ul style="list-style-type: none"><li>• <code>&lt;crop-rect&gt;</code>—Defines the crop values.</li><li>• <code>&lt;trim-rect&gt;</code>—Defines the trim values.</li><li>• <code>&lt;pan-rect&gt;</code>—Defines the pan and zoom animation (“Ken Burns” effect) values.</li></ul></li></ul>
<adjust-rollingShutter>	<p>Modifies the rolling shutter reduction applied to a clip. The <code>amount</code> attribute specifies the reduction amount, which defaults to <code>none</code>.</p>
<adjust-stabilization>	<p>Modifies the stabilization applied to a clip. The <code>type</code> attribute specifies the stabilization type, which defaults to <code>automatic</code>.</p>

Element	Description
<code>&lt;adjust-transform&gt;</code>	<p>Modifies the visible image by resizing, moving, or rotating using the following attributes:</p> <ul style="list-style-type: none"><li>• <code>enabled</code>—Enables (1, the default) or disables (0) this transform adjustment.</li><li>• <code>position</code>—A two-dimensional (<i>xy</i>) value that specifies the amount to move the frame in each dimension from its original location. These values are percentages based on the original frame height. For example, if the original frame height is 1,080 pixels, specifying <code>position="10 10"</code> moves the frame 108 pixels to the right and 108 pixels up.</li><li>• <code>scale</code>—A two-dimensional (<i>xy</i>) value that specifies the percentage amount to resize the frame in each dimension from its original frame size (for example, "25 25").</li><li>• <code>rotation</code>—The amount, in degrees, to rotate the frame. A positive value rotates counterclockwise and a negative value rotates clockwise (for example, <code>-180</code>).</li><li>• <code>anchor</code>—A two-dimensional (<i>xy</i>) value that offsets the origin of the video image and its center of rotation. This value is expressed as a percentage of the scaled image height and defaults to "0 0".</li></ul>

## Effects

Use filters, text styles, and transitions to apply effects to certain story elements in your events and projects. For a list of story elements, see [Table 2-1](#) (page 18).

## Filters

Filter effects, such as audio and video, are applied to certain story elements using the filter elements listed in Table 4-3.

---

**Note:** In Final Cut Pro X 10.2, color correction behaves as other video filters do and can be inserted anywhere in a series of video filters.

---

**Table 4-3** Filter elements

Element	Description
<filter-audio>	A reference to an audio effect.
<filter-video>	A reference to a video effect.

Refer to the Document Type Definition (DTD) for specific information on the filter types each parent element supports.

---

**Note:** In FCPXML v1.3 and later, the <filter-audio> and <filter-video> elements represent audio or video filters respectively. Prior to FCPXML v1.3, a single <filter> element represented both audio and video filters.

In FCPXML v1.5, color filters are described with the <filter-video> element and its subelements. (In FCPXML v1.4 and v1.3, the <adjust-color> element was used.)

---

You apply video filters to elements that represent visual media (<video>, <text>, <clip>, and <ref-clip> elements). For multicam clips, apply the video filters on the individual angles (represented by <mc-source> elements).

You apply audio filters to elements that represent audible media (<audio>, <clip>, <ref-clip>, <audio-source>, and <audio-aux-source> elements). For multicam clips, apply the audio filters to the entire multicam clip (represented by <mc-clip> elements).

## Masked Filters

A video filter can be applied to a region designated by a collection of masks. You use the <filter-video-mask> element to apply the masks. Table 4-4 lists the available mask elements used to define masks.

**Table 4-4** Mask elements

Element	Description
<mask-isolation>	Specifies a color isolation mask.
<mask-shape>	Specifies a mask for a region using a super ellipse shape.

The mask elements are combined to define a single region to which the effect applies. The actual video filter is described with a `<video-filter>` element as a child of the `<filter-video-mask>` element.

## Color Filters

Color correction, as well as color matching and color balance effects, are described by the `<filter-video>` element. That element references an associated `<effect>` element in the resource section that designates the particular effect.

Color correction may be applied with a collection of video filter masks. The `<filter-video-mask>` element describes both the filter and the masks, and can have up to two `<filter-video>` child elements. Different color correction parameters can be applied to the inside (the first `<filter-video>` element) and the outside (the second `<filter-video>` element) of the mask.

---

**Note:** Final Cut Pro X supports the American Society of Cinematographers Color Decision List (ASC CDL) format for the exchange of basic primary color grading information between equipment and software from different manufacturers.

Final Cut Pro X v10.2 and later export ASC CDL information as an XML comment (as a child of the associated `<filter-video>` element) if a color correction is applied to a clip in your project. (Prior versions of Final Cut Pro X exported ASC CDL information with the `<info-asc-cdl>` element as a child of the associated `<clip>` element.)

The ASC CDL information describes the primary color correction in ASC CDL format as three components: slope, offset, and power. Each component is represented as a vector of red, green, and blue adjustment values (for example, “1.0 1.0 1.0”).

The following is an example export of ASC CDL information:

```
<filter-video ref="r1" name="Color Correction">
  <!-- info-asc-cdl: slope="1.05 1.05 1.05" offset="0.0275 0.0275 0.0275"
  power="1.25 1.2 1" -->
</filter-video>
```

---

## Transitions

A transition is an effect that combines zero, one or two neighboring elements. You apply transition effects with the `<transition>` story element. The `offset` and `duration` attributes define the position and extent of the transition effect. The `<filter-video>` and `<filter-audio>` filter elements specify the effect to apply. See [Table 4-3](#) (page 37).

**Note:** When an audio filter specifies an audio cross fade, the cross fade does not take effect on either side if the clip has audio specific timing set by the `audioStart` or `audioDuration` attributes. This is known as a J- or L-cut, depending on which side of the transition it occurs.

## Text Styles

Title text, represented by the `<title>` element—see [Table 2-1](#) (page 18)—can have text styles, such as font size and alignment. You apply these styles with the elements listed in [Table 4-5](#).

**Table 4-5** Text style elements

Element	Description
<code>&lt;text-style&gt;</code>	<p>Defines the text formatting to apply to a <code>&lt;title&gt;</code> or <code>&lt;text&gt;</code> element. Use the <code>ref</code> attribute to reference a text style definition. Use the other attributes to specify the text style settings.</p> <p>If a text style definition is referenced, any attributes specified in this element override the values in the text style definition.</p>
<code>&lt;text-style-def&gt;</code>	<p>Defines a set of text style attributes that can be shared with (referenced by) multiple <code>&lt;text-style&gt;</code> elements. This element has the following attributes:</p> <ul style="list-style-type: none"><li><code>id</code>—The local identifier for this text style definition.</li><li><code>name</code>—The name for this text style definition.</li></ul> <p>Each <code>&lt;text-style&gt;</code> child element specifies the text style to apply.</p>

You use the `<text>` element to create a series of text segments in a title. For each segment you can apply different text styles using the `<text-style>` element.

The following example XML uses text style definitions to create a text effect:

```
<title offset="0s" ref="r2" name="Continuous: Title with styles, including font
and size." duration="10s" start="0s">
  <text>
    <text-style ref="ts1">Title with styles, including </text-style>
    <text-style ref="ts2">font</text-style>
    <text-style ref="ts1"> and </text-style>
    <text-style ref="ts3">size</text-style>
    <text-style ref="ts1">.</text-style>
  </text>
```

```
<text-style-def id="ts1">
  <text-style font="Helvetica" fontSize="72" fontFace="Regular" fontColor="1
1 1 1" strokeColor="1 0 0 1" strokeWidth="1" alignment="center"/>
</text-style-def>
<text-style-def id="ts2">
  <text-style font="Capitals" fontSize="72" fontFace="Regular" fontColor="1
1 1 1" bold="1" strokeColor="1 0 0 1" strokeWidth="1" alignment="center"/>
</text-style-def>
<text-style-def id="ts3">
  <text-style font="Helvetica" fontSize="120" fontFace="Regular" fontColor="1
1 1 1" strokeColor="1 0 0 1" strokeWidth="1" alignment="center"/>
</text-style-def>
</title>
```

3D text geometry and appearance are described with one or more `<param>` elements as children of the `<text-style>` element, each specifying a respective parameter. See the following example XML snippet:

```
<text-style-def id="ts1">
  <text-style font="Helvetica" fontSize="63" fontFace="Regular" fontColor="1
0.999974 0.999991 1" alignment="center">
    <param name="Lighting" key="527">
      <param name="Environment" key="512">
        <param name="Intensity" key="514" value="0.8"/>
        <param name="Rotation" key="516" value="0 247 0 0 (Use Rotation)"/>
        <param name="Contrast" key="517" value="62"/>
        <param name="Saturation" key="525" value="47"/>
      </param>
    </param>
    <param name="Material" key="material:Basic">
      <param name="Layer" key="layer:Finish:1 (Enamel)">
        <param name="Highlight Color" key="127" value="0.838246 0.965254
0.503887"/>
      </param>
      <param name="Layer" key="layer:Substance:3 (Plastic)">
        <param name="Type" key="4" value="0 (Shiny)"/>
        <param name="Color" key="106" value="0.813144 0.729297 0.994205"/>
      </param>
    </param>
  </text-style>
</text-style-def>
```



```
        </param>
    </param>
</text-style>
</text-style-def>
```

## Adjustment Attributes and Effect Parameters

You can specify the value of certain adjustment attributes and effect parameters and change them for the entire duration of the parent element. You do this by adding a parameter (`<param>`) element for each adjustment attribute or effect parameter you would like to change.

You can also use the `<param>` element to create keyframe animations by changing adjustment attributes and effect parameters over time. See [Animation](#) (page 42) for more details.

The following elements may contain `<param>` child elements:

- Adjustment elements—`<adjust-blend>`, `<adjust-corners>`, `<adjust-EQ>`, `<adjust-panner>`, `<adjust-stabilization>`, `<adjust-transform>`, `<adjust-volume>`, `<crop-rect>`, and `<trim-rect>`
- Effect elements—`<filter-audio>` and `<filter-video>`
- Story elements—`<title>` and `<video>`

The `<param>` element attributes are defined in Table 4-6.

**Table 4-6** Parameter element attributes

Attribute	Description
name	For adjustments, this case-sensitive string identifies the parent attribute to override. For an effect parameter, this is for informational purposes only.
key	A string that identifies a parameter in the effect implementation. This is ignored when the <code>&lt;param&gt;</code> element is used for an adjustment.
value	The new value to apply to the adjustment attribute (optional) or the effect parameter. This value overrides the value specified by the parent adjustment attribute.

For example, the following XML adjusts the audio volume of a clip by  $-3\text{dB}$ :

```
<audio>
```

```
<adjust-volume>
  <param name="amount" value="-3dB"/>
</adjust-volume>
</audio>
```

Some adjustments and effects allow preserving parameter information in an internal binary format using the `<data>` element.

**Table 4-7** Data element

Element	Description
<code>&lt;data&gt;</code>	Specifies binary parameter data.

Some effect parameters allow structured data. This information is represented with a series of nested `<param>` elements as children of the effect parameter you want to override. See the *Lighting* parameter of the 3D text XML snippet in the [Text Styles](#) (page 39) section for an example.

## Animation

Create animations by changing certain adjustment attributes and effect parameters over time. You do this by adding a `<param>` element—see [Parameters](#) (page 41)—and the keyframe related elements, for each adjustment attribute or effect parameter you want to override. Table 4-8 describes the keyframe animation elements.

**Table 4-8** Keyframe animation elements

Attribute	Description
<code>&lt;keyframe&gt;</code>	<p>Specifies a keyframe point along an animation path. This element has the following attributes:</p> <ul style="list-style-type: none"><li><code>curve</code>—Specifies whether the interpolated values around the keyframe are linear or smooth. The default is <code>smooth</code>.</li><li><code>interp</code>—The animation speed over the keyframe. The default is <code>linear</code>.</li><li><code>time</code>—The keyframe time (for example, “0s”).</li><li><code>value</code>—The keyframe value (for example, “-5 10”).</li></ul>

Attribute	Description
<keyframe- Animation>	Specifies an animation path using points described by child <keyframe> elements.

To produce a keyframe animation, you specify the time points and values for each keyframe using the `time` and `value` attributes of the <keyframe> element. The Final Cut Pro X application interpolates the values between them.

The following example performs a position and scale transform adjustment animation using keyframes:

```
<adjust-transform position="-10 10">
  <param name="position">
    <keyframeAnimation>
      <keyframe time="0s" value="-5 10"/>
      <keyframe time="10s" value="10 20"/>
    </keyframeAnimation>
  </param>
  <param name="scale">
    <keyframeAnimation>
      <keyframe time="0s" value="1 1"/>
      <keyframe time="10s" value="0.75 0.75"/>
    </keyframeAnimation>
  </param>
</adjust-transform>
```

Initially, at `time=0s`, the frame is moved left 5 percent and up 10 percent ("`-5 10`") of the project's height from the original location, with no change in size ("`1 1`"). At `time=10s`, the frame has moved 10 percent right and 20 percent up ("`10 20`") of the project's height from the original location, with a 25 percent ("`0.75 0.75`") reduction in size from the original ("`1 1`"). See [Table 4-2](#) (page 35) for more information on the <adjust-transform> element.

Final Cut Pro X determines the parameter value based on the following priority (with 1 being the highest priority):

1. The `value` and `time` attributes of the <keyframe> element (for a particular time)
2. The `value` attribute of the <param> element
3. The respective attribute of the adjustment, if applicable

In the previous example, it does not matter that the `position` attribute of the `<adjust-transform>` element has a value of `"-10 10"` because the keyframe defined at `time=0s` (via the `<keyframe>` element) takes priority and overrides the value to `"-5 10"`.

# Metadata

Final Cut Pro X maintains various metadata item values (for example, Camera Name, Reel, Copyright). Metadata typically comes from the media and may be of interest to other applications, but is not critical for Final Cut Pro X to perform its tasks.

On export, Final Cut Pro X includes only the metadata items in the selected metadata view. On import, Final Cut Pro X uses the values specified in the FCPXML for editable metadata items; the values for non-editable metadata items are ignored during import, and are instead retrieved from the media asset.

See [Metadata Keys](#) (page 57) for a complete list of metadata items that are editable through Final Cut Pro X.

## Elements

Table 5-1 lists the FCPXML metadata elements.

**Table 5-1** Metadata elements

Element	Description
<metadata>	A container for a collection of <md> elements.

Element	Description
<md>	<p>Specifies an individual metadata item. This element has the following attributes:</p> <ul style="list-style-type: none"> <li><b>key</b>—A structured Final Cut Pro X key string that identifies the metadata item (for example, <code>com.apple.proapps.mio.cameraName</code>).</li> <li><b>value</b>—The value of the metadata item. When the value is an array, an &lt;array&gt; element is used instead.</li> <li><b>editable</b>—Indicates whether the metadata item is editable (1=Yes, 0=No). The default is 0. Most metadata items are not editable, and this attribute may be omitted. If editable, Final Cut Pro X applies the new value for the metadata item on import.</li> <li><b>type</b>—Defines the metadata item value type.</li> <li><b>displayName</b>—The metadata item name that appears in Final Cut Pro X.</li> <li><b>description</b>—The metadata item description that appears in Final Cut Pro X.</li> <li><b>source</b>—The metadata item origin.</li> </ul>
<array>	A container for a collection of <string> elements that define the value(s) for an array of strings.
<string>	Defines one metadata value in an array of strings.

The following is an example of metadata with primitive data types:

```
<metadata>
  <md key="com.apple.proapps.mio.cameraName" value="The Big Camera"/>
  <md key="com.apple.proapps.spotlight.kMDItemCopyright"
    value="© 2009 All Rights Reserved"/>
  <md key="com.apple.proapps.studio.reel" value="R945"/>
  <md key="com.apple.proapps.studio.scene" value="Deep Water"/>
</metadata>
```

The following is an example of metadata with complex data types:

```
<metadata>
  <md key="com.apple.proapps.spotlight.kMDItemCodecs" displayName="Codecs">
    <array>
```

```
<string>AAC</string>
<string>H.264</string>
</array>
</md>
</metadata>
```

## Keys and Sources

Metadata items Final Cut Pro X maintains are grouped based on where they come from (their source). The source is indicated by the `<md>` element's `source` attribute (for example, `studio`) or the prefix of the key attribute (for example, `com.apple.proapps.studio`).

The sources and key prefixes currently used are:

**Camera** Metadata from a camera or capturing device that Final Cut Pro X or the device driver/plugin picks up. These are not editable, except for `com.apple.proapps.mio.cameraName`.

*Key Prefixes:*

- `com.apple.proapps.mio`
- `org.smpte.mxf`

*Example:*

- `com.apple.proapps.mio.cameraName`

**Exif** Metadata from files with Exif information (for example, a JPEG file).

*Key Prefix:*

- `com.apple.proapps.exif.{Exif}`

*Example:*

- `com.apple.proapps.exif.{Exif}.FocalLength`

**Image** Metadata from an image file (for example, TIFF or PNG), or camera specific metadata saved with the image.

*Key Prefixes:*

- `com.apple.proapps.image.{TIFF}`
- `com.apple.proapps.image.{CIFF}`
- `com.apple.proapps.image.{GIF}`

- `com.apple.proapps.image.{PNG}`
- `com.apple.proapps.image.{DNG}`
- `com.apple.proapps.image.{GPS}`
- A prefix that identifies the camera manufacturer.

*Examples:*

- `com.apple.proapps.image.{TIFF}.WhitePoint`
- `com.apple.proapps.image.{CIFF}.FocusMode`
- `com.apple.proapps.image.{PNG}.Gamma`
- `com.apple.proapps.image.{GPS}.MapDatum`

**IPTC** International Press Telecommunication Council defined photo metadata.

*Key Prefix:*

- `com.apple.proapps.iptc.{IPTC}`

*Example:*

- `com.apple.proapps.iptc.{IPTC}.Headline`

**Share** Metadata added to a share output.

*Key Prefix:*

- `com.apple.proapps.share`
- `com.apple.quicktime`

*Example:*

- `com.apple.proapps.share.id`
- `com.apple.quicktime.copyright`

**Spotlight** Spotlight metadata.

*Key Prefix:*

- `com.apple.proapps.spotlight`

*Example:*

- `com.apple.proapps.spotlight.kMDItemCopyright`

**Studio** Final Cut Pro X-defined metadata. See [Metadata Keys](#) (page 57) for a list of editable Studio keys. Additional *Studio* keys exist that are not editable.



*Key Prefix:*

- `com.apple.proapps.studio`

*Examples:*

- `com.apple.proapps.studio.reel`
- `com.apple.proapps.studio.scene`

**Custom**

Metadata the user added in Final Cut Pro X (has prefix: `com.apple.proapps.custom`), or metadata that comes from a third-party application or camera (prefix is a Reverse DNS string that identifies a third-party application or a camera manufacturer).

*Key Prefixes:*

- `com.apple.proapps.custom`
- A prefix that identifies a third-party application or a camera manufacturer.

*Examples:*

- `com.apple.proapps.custom.mycustommetadata`
- `com.yourCompany.yourApp.yourCustomMetadata`

The following is an example of custom metadata items:

```
<metadata>
  <md key="com.yourCompany.yourApp.yourCustomMetadata"
    value="This is your custom metadata" type="string"
    displayName="Your Custom Metadata" source="Your Application"/>
  <md key="com.apple.proapps.custom.mycustommetadata"
    value="This is my custom metadata" type="string" editable="1"
    displayName="myCustomMetadata" source="custom source"/>
</metadata>
```

# Appendix A: FCPXML Supported Identifiers

This appendix lists the supported identifiers you can use to specify rendering and video formats, and blend and pan modes in your FCPXML documents.

## Rendering Formats

```
FFRenderFormatProRes4444  
FFRenderFormatProRes422HQ  
FFRenderFormatProRes422  
FFRenderFormatUncompressed10bit422
```

## Video Formats

```
FFVideoFormatRateUndefined  
FFVideoFormat1080i50  
FFVideoFormat1080i5994  
FFVideoFormat1080p2398  
FFVideoFormat1080p24  
FFVideoFormat1080p25  
FFVideoFormat1080p2997  
FFVideoFormat1080p30  
FFVideoFormat1080p50  
FFVideoFormat1080p5994  
FFVideoFormat1080p60  
FFVideoFormat1280x1080i50  
FFVideoFormat1280x1080i5994  
FFVideoFormat1280x1080p2398  
FFVideoFormat1280x1080p24  
FFVideoFormat1280x1080p25
```

FFVideoFormat1280x1080p2997  
FFVideoFormat1280x1080p30  
FFVideoFormat1280x1080p50  
FFVideoFormat1280x1080p5994  
FFVideoFormat1280x1080p60  
FFVideoFormat1440x1080i50  
FFVideoFormat1440x1080i5994  
FFVideoFormat1440x1080p2398  
FFVideoFormat1440x1080p24  
FFVideoFormat1440x1080p25  
FFVideoFormat1440x1080p2997  
FFVideoFormat1440x1080p30  
FFVideoFormat1440x1080p50  
FFVideoFormat1440x1080p5994  
FFVideoFormat1440x1080p60  
FFVideoFormat2048x1024p2398  
FFVideoFormat2048x1024p24  
FFVideoFormat2048x1024p25  
FFVideoFormat2048x1024p2997  
FFVideoFormat2048x1024p30  
FFVideoFormat2048x1024p50  
FFVideoFormat2048x1024p5994  
FFVideoFormat2048x1024p60  
FFVideoFormat2048x1080p2398  
FFVideoFormat2048x1080p24  
FFVideoFormat2048x1080p25  
FFVideoFormat2048x1080p2997  
FFVideoFormat2048x1080p30  
FFVideoFormat2048x1080p50  
FFVideoFormat2048x1080p5994  
FFVideoFormat2048x1080p60  
FFVideoFormat2048x1152p2398  
FFVideoFormat2048x1152p24  
FFVideoFormat2048x1152p25  
FFVideoFormat2048x1152p2997

FFVideoFormat2048x1152p30  
FFVideoFormat2048x1152p50  
FFVideoFormat2048x1152p5994  
FFVideoFormat2048x1152p60  
FFVideoFormat2048x1536p2398  
FFVideoFormat2048x1536p24  
FFVideoFormat2048x1536p25  
FFVideoFormat2048x1536p2997  
FFVideoFormat2048x1536p30  
FFVideoFormat2048x1536p50  
FFVideoFormat2048x1536p5994  
FFVideoFormat2048x1536p60  
FFVideoFormat2048x1556p2398  
FFVideoFormat2048x1556p24  
FFVideoFormat2048x1556p25  
FFVideoFormat2048x1556p2997  
FFVideoFormat2048x1556p30  
FFVideoFormat2048x1556p50  
FFVideoFormat2048x1556p5994  
FFVideoFormat2048x1556p60  
FFVideoFormat3840x2160p2398  
FFVideoFormat3840x2160p24  
FFVideoFormat3840x2160p25  
FFVideoFormat3840x2160p2997  
FFVideoFormat3840x2160p30  
FFVideoFormat3840x2160p50  
FFVideoFormat3840x2160p5994  
FFVideoFormat3840x2160p60  
FFVideoFormat4096x2048p2398  
FFVideoFormat4096x2048p24  
FFVideoFormat4096x2048p25  
FFVideoFormat4096x2048p2997  
FFVideoFormat4096x2048p30  
FFVideoFormat4096x2048p50

FFVideoFormat4096x2048p5994  
FFVideoFormat4096x2048p60  
FFVideoFormat4096x2160p2398  
FFVideoFormat4096x2160p24  
FFVideoFormat4096x2160p25  
FFVideoFormat4096x2160p2997  
FFVideoFormat4096x2160p30  
FFVideoFormat4096x2160p50  
FFVideoFormat4096x2160p5994  
FFVideoFormat4096x2160p60  
FFVideoFormat4096x2304p2398  
FFVideoFormat4096x2304p24  
FFVideoFormat4096x2304p25  
FFVideoFormat4096x2304p2997  
FFVideoFormat4096x2304p30  
FFVideoFormat4096x2304p50  
FFVideoFormat4096x2304p5994  
FFVideoFormat4096x2304p60  
FFVideoFormat4096x3112p2398  
FFVideoFormat4096x3112p24  
FFVideoFormat4096x3112p25  
FFVideoFormat4096x3112p2997  
FFVideoFormat4096x3112p30  
FFVideoFormat4096x3112p50  
FFVideoFormat4096x3112p5994  
FFVideoFormat4096x3112p60  
FFVideoFormat5120x2160p2398  
FFVideoFormat5120x2160p24  
FFVideoFormat5120x2160p25  
FFVideoFormat5120x2160p2997  
FFVideoFormat5120x2160p30  
FFVideoFormat5120x2160p50  
FFVideoFormat5120x2160p5994  
FFVideoFormat5120x2160p60  
FFVideoFormat5120x2560p2398

FFVideoFormat5120x2560p24  
FFVideoFormat5120x2560p25  
FFVideoFormat5120x2560p2997  
FFVideoFormat5120x2560p30  
FFVideoFormat5120x2560p50  
FFVideoFormat5120x2560p5994  
FFVideoFormat5120x2560p60  
FFVideoFormat5120x2700p2398  
FFVideoFormat5120x2700p24  
FFVideoFormat5120x2700p25  
FFVideoFormat5120x2700p2997  
FFVideoFormat5120x2700p30  
FFVideoFormat5120x2700p50  
FFVideoFormat5120x2700p5994  
FFVideoFormat5120x2700p60  
FFVideoFormat640x360p2997  
FFVideoFormat640x480p2398  
FFVideoFormat640x480p24  
FFVideoFormat640x480p25  
FFVideoFormat640x480p2997  
FFVideoFormat640x480p30  
FFVideoFormat720p2398  
FFVideoFormat720p24  
FFVideoFormat720p25  
FFVideoFormat720p2997  
FFVideoFormat720p30  
FFVideoFormat720p50  
FFVideoFormat720p5994  
FFVideoFormat720p60  
FFVideoFormat720x486i5994  
FFVideoFormat720x486i5994\_16x9  
FFVideoFormat720x486p2398  
FFVideoFormat720x486p2398\_16x9  
FFVideoFormat720x486p2997

```
FFVideoFormat720x486p2997_16x9
FFVideoFormat720x576i50
FFVideoFormat720x576i50_16x9
FFVideoFormat720x576p25
FFVideoFormat720x576p25_16x9
FFVideoFormat960x540p2398
FFVideoFormat960x540p24
FFVideoFormat960x540p25
FFVideoFormat960x540p2997
FFVideoFormat960x540p30
FFVideoFormat960x720p2398
FFVideoFormat960x720p24
FFVideoFormat960x720p25
FFVideoFormat960x720p2997
FFVideoFormat960x720p30
FFVideoFormat960x720p50
FFVideoFormat960x720p5994
FFVideoFormat960x720p60
FFVideoFormatDV720x480i5994
FFVideoFormatDV720x480i5994_16x9
FFVideoFormatDV720x480p2398
FFVideoFormatDV720x480p2398_16x9
FFVideoFormatDV720x480p2997
FFVideoFormatDV720x480p2997_16x9
FFVideoFormatDV720x576i50
FFVideoFormatDV720x576i50_16x9
```

## Blend Modes

FCPXML v1.1 and later support the following blend modes.

---

**Note:** XML generators can omit the value in parentheses, and parsers should scan for the numeric value only.

---

```
0 (Normal)
2 (Subtract)
3 (Darken)
4 (Multiply)
5 (Color Burn)
6 (Linear Burn)
8 (Add)
9 (Lighten)
10 (Screen)
11 (Color Dodge)
12 (Linear Dodge)
14 (Overlay)
15 (Soft Light)
16 (Hard Light)
17 (Vivid Light)
18 (Linear Light)
19 (Pin Light)
20 (Hard Mix)
22 (Difference)
23 (Exclusion)
25 (Stencil Alpha)
26 (Stencil Luma)
27 (Silhouette Alpha)
28 (Silhouette Luma)
29 (Behind)
31 (Alpha Add)
32 (Premultiplied Mix)
```

## Pan Modes

FCPXML v1.1 and later support the following pan modes.



**Note:** XML generators can omit the value in parentheses, and parsers should scan for the numeric value only.

0 (Default)  
1 (Stereo Left/Right)  
2 (Create Space)  
3 (Dialogue)  
4 (Music)  
5 (Ambience)  
6 (Circle)  
7 (Rotate)  
8 (Back to Front)  
9 (Left Surround to Right Front)  
10 (Right Surround to Left Front)

## Metadata Keys

Final Cut Pro X imports/exports the following metadata keys.

Key	Display Name	Type	Source
com.apple.proapps.mio.cameraName	Camera Name	String	Camera
com.apple.proapps.studio.alphaHandling	Alpha Handling	Integer	Studio
com.apple.proapps.studio.angle	Camera Angle	String	Studio
com.apple.proapps.studio.colorSpaceOverride	ColorSpace Override	Integer	Studio
com.apple.proapps.studio.metadataAnamorphicType	Anamorphic Override	Integer	Studio
com.apple.proapps.studio.metadataDeinterlaceType	Deinterlace	Boolean	Studio

Key	Display Name	Type	Source
com.apple.proapps.studio.metadataFieldDominanceOverride	Field Dominance Override	Integer	Studio
com.apple.proapps.studio.metadataLocation	Location	String	Studio
com.apple.proapps.studio.reel	Reel	String	Studio
com.apple.proapps.studio.scene	Scene	String	Studio
com.apple.proapps.studio.shot	Take	String	Studio

The supported values for *Integer* types are provided in the following table.

Integer Type	Value	Description
Alpha Handling	0	Premultiply
	1	Straight
	2	None/Ignore Alpha
ColorSpace Override	0	None Set
	1	601 170M (NTSC)
	2	601 EBU (PAL)
	3	709 (HD)
	4	Adobe RGB (1998)
	5	sRGB (IEC 61966-2-1)
Anamorphic Override	0	None Set
	1	Standard
	2	Widescreen
Field Dominance Override	0	None Set
	1	Progressive

Integer Type	Value	Description
	2	Upper First
	3	Lower First

# Appendix B: FCPXML Version 1.5 DTD

The following DTD applies to Final Cut Pro X XML Interchange Format v1.5.

---

**Note:** Even though an imported FCPXML document matches this DTD, import errors may still occur due to invalid data. If the document does not match the DTD, Final Cut Pro X rejects the import operation completely.

---

```
<!-- FCP X XML Interchange Format, Version 1.5 -->
<!-- Copyright (c) 2011–2015 Apple Inc. All rights reserved. -->

<!-- FCPXML -->
<!ELEMENT fcpxml (import-options?, resources?, library)>
<!ATTLIST fcpxml version CDATA #FIXED "1.5">

<!-- COMMON ENTITIES -->

<!-- 'time' attributes are expressed as a rational number of seconds (e.g.,
"1001/30000s") -->
<!-- with a 64-bit numerator and 32-bit denominator. -->
<!-- Integer 'time' values, such as 5 seconds, may be expressed as whole numbers
(e.g., '5s'). -->
<!ENTITY % time "CDATA">

<!-- A 'timelist' is a semi-colon-separated list of time values -->
<!ENTITY % timelist "CDATA">

<!ENTITY % collection_item "collection-folder | keyword-collection |
smart-collection">

<!-- IMPORT OPTIONS -->
```

```

<!-- Element that contain options on how the import should be conducted (e.g.
whether to copy assets, etc.) -->
<!ELEMENT import-options (option*)>

<!ELEMENT option EMPTY>
<!ATTLIST option key CDATA #REQUIRED>
<!ATTLIST option value CDATA #REQUIRED>

<!-- LIBRARY ELEMENT -->
<!ELEMENT library (event | smart-collection)*>
<!ATTLIST library location CDATA #IMPLIED>

<!-- EVENT ELEMENT -->
<!ELEMENT event (clip | audition | mc-clip | ref-clip | %collection_item; |
project)*>
<!ATTLIST event name CDATA #IMPLIED>
<!ATTLIST event uid CDATA #IMPLIED>

<!-- PROJECT ELEMENT -->
<!ELEMENT project (sequence)>
<!ATTLIST project name CDATA #IMPLIED>
<!ATTLIST project uid CDATA #IMPLIED>
<!ATTLIST project id ID #IMPLIED>

<!-- RESOURCE ELEMENTS -->
<!-- A 'resource' is a project element potentially referenced by other project
elements. -->
<!-- To support such references, all resource instances require a local ID attribute.
-->
<!ELEMENT resources (asset | effect | format | media)*>

<!-- A 'media' defines a reference to new or existing media via an FCP-assigned
unique identifier ('uid'). -->
<!-- If 'uid' is not specified, FCP creates a new media object as specified by the
optional child element. -->
<!-- If 'projectRef' is not specified, FCP uses the default instance. -->

```

```

<!ELEMENT media (multicam | sequence)?>
<!ATTLIST media id ID #REQUIRED>
<!ATTLIST media name CDATA #IMPLIED>
<!ATTLIST media uid CDATA #IMPLIED>
<!ATTLIST media projectRef IDREF #IMPLIED>

<!-- A 'format' describes video properties. -->
<!ELEMENT format EMPTY>
<!ATTLIST format id ID #REQUIRED>
<!ATTLIST format name CDATA #IMPLIED>
<!ATTLIST format frameDuration %time; #IMPLIED>
<!ATTLIST format fieldOrder CDATA #IMPLIED>          <!-- (progressive | upper
first | lower first ) -->
<!ATTLIST format width CDATA #IMPLIED>
<!ATTLIST format height CDATA #IMPLIED>
<!ATTLIST format paspH CDATA #IMPLIED>
<!ATTLIST format paspV CDATA #IMPLIED>

<!-- An 'asset' defines a reference to external source media (i.e., a local file).
-->
<!-- 'uid' is an FCP-assigned unique ID; if not specified, FCP creates a new default
clip for the asset. -->
<!ELEMENT asset (bookmark?,metadata?)>
<!ATTLIST asset id ID #REQUIRED>
<!ATTLIST asset name CDATA #IMPLIED>
<!ATTLIST asset uid CDATA #IMPLIED>
<!ATTLIST asset src CDATA #REQUIRED>                <!-- file: URL -->
<!ATTLIST asset start %time; #IMPLIED>
<!ATTLIST asset duration %time; #IMPLIED>
<!ATTLIST asset hasVideo CDATA #IMPLIED>
<!ATTLIST asset format IDREF #IMPLIED>
<!ATTLIST asset hasAudio CDATA #IMPLIED>
<!ATTLIST asset audioSources CDATA #IMPLIED>
<!ATTLIST asset audioChannels CDATA #IMPLIED>
<!ATTLIST asset audioRate CDATA #IMPLIED>

```

```

<!ATTLIST asset colorOverride CDATA #IMPLIED>

<!ENTITY % md-type "( string | boolean | integer | float | date | timecode )">

<!ELEMENT metadata (md*)>

<!ELEMENT md (array?)>
<!ATTLIST md key CDATA #REQUIRED>
<!ATTLIST md value CDATA #IMPLIED>
<!ATTLIST md editable (0 | 1) "0">
<!ATTLIST md type %md-type; #IMPLIED>
<!ATTLIST md displayName CDATA #IMPLIED>
<!ATTLIST md description CDATA #IMPLIED>
<!ATTLIST md source CDATA #IMPLIED>

<!-- An 'effect' defines a reference to a built-in or user-defined Motion effect,
      FxPlug plug-in, audio bundle, or audio unit. -->
<!ELEMENT effect EMPTY>
<!ATTLIST effect id ID #REQUIRED>
<!ATTLIST effect name CDATA #IMPLIED>
<!ATTLIST effect uid CDATA #REQUIRED>

<!-- STORY ELEMENTS -->

<!-- The 'ao_attrs' entity declares the attributes common to 'anchorable' objects.
      -->
<!-- The 'lane' attribute specifies where the object is contained/anchored relative
      to its parent: -->
<!--      0 = contained inside its parent (default) -->
<!--      >0 = anchored above its parent -->
<!--      <0 = anchored below its parent -->
<!-- The 'offset' attribute defines the location of the object in the parent
      timeline (default is '0s'). -->
<!ENTITY % ao_attrs "
      lane CDATA #IMPLIED

```

```

        offset %time; #IMPLIED
">

<!-- The 'clip_attrs' entity declares the attributes common to all story elements.
-->

<!-- The 'start' attribute defines a local timeline to schedule contained and
anchored items. -->

<!-- The default start value is '0s'. -->
<!ENTITY % clip_attrs "
    %ao_attrs;
    name CDATA #IMPLIED
    start %time; #IMPLIED
    duration %time; #REQUIRED
    enabled (0 | 1) '1'
">

<!ENTITY % audioHz "( 32k | 44.1k | 48k | 88.2k | 96k | 176.4k | 192k )">

<!-- The 'media_attrs' entity declares the attributes common to media instances.
-->

<!-- 'format' specifies a <format> resource ID. -->
<!-- 'tcStart' specifies the timecode origin of the media. -->
<!-- 'tcFormat' specifies the timecode display format (DF=drop frame; NDF=non-drop
frame). -->
<!ENTITY % media_attrs "
    format IDREF #REQUIRED
    duration %time; #IMPLIED
    tcStart %time; #IMPLIED
    tcFormat (DF | NDF) #IMPLIED
">

<!ENTITY % outputChannel "( L | R | C | LFE | Ls | Rs | X )">

<!ENTITY % fadeType "(linear | easeIn | easeOut | easeInOut)">

```



```

<!-- A 'fadeIn' element animates a parameter from its min value to its implied
value over a specified duration. -->
<!--ELEMENT fadeIn EMPTY>
<!--ATTLIST fadeIn type %fadeType; #IMPLIED>           <!-- default is 'easeIn' -->
<!--ATTLIST fadeIn duration %time; #REQUIRED>

<!-- A 'fadeOut' element animates a parameter from its implied value to its min
value over a specified duration. -->
<!--ELEMENT fadeOut EMPTY>
<!--ATTLIST fadeOut type %fadeType; #IMPLIED>           <!-- default is 'easeOut'
-->
<!--ATTLIST fadeOut duration %time; #REQUIRED>

<!-- A 'keyframeAnimation' element describes an animation curve using the contained
'keyframe' elements -->
<!--ELEMENT keyframeAnimation (keyframe*)>

<!-- A 'keyframe' element describes an point along an animation curve -->
<!--ELEMENT keyframe EMPTY>
<!--ATTLIST keyframe time %time; #REQUIRED>
<!--ATTLIST keyframe value CDATA #REQUIRED>
<!--ATTLIST keyframe interp (linear | ease | easeIn | easeOut) "linear"> <!--
interpolation type for point. -->
<!--ATTLIST keyframe curve (linear | smooth) "smooth"> <!-- whether the value changes
smoothly over the keyframe, or linearly. -->

<!-- A 'mute' element suppresses audio output for a range of source media time -->
<!--ELEMENT mute (fadeIn?, fadeOut?)>
<!--ATTLIST mute start %time; #IMPLIED>
<!--ATTLIST mute duration %time; #IMPLIED>

<!-- A 'param' specifies the range for a parameter over time, optionally including
key-framed values. -->
<!-- Fade-in and fade-out are optional. -->
<!--ELEMENT param (fadeIn?, fadeOut?, keyframeAnimation?, param*)>
<!--ATTLIST param name CDATA #REQUIRED>

```

```

<!ATTLIST param key CDATA #IMPLIED>           <!-- key is used to identify
the parameter (can include a path (delimited by /) depending on the containing
element -->

                                                    <!-- some parameters have
localizable names which is why key is necessary (if the key attribute does not
exist, the name is used) -->
<!ATTLIST param value CDATA #IMPLIED>          <!-- initial value -->
<!ATTLIST param enabled (0 | 1) "1">

<!-- A 'data' specifies a chunk of data -->
<!ELEMENT data (#PCDATA)>
<!ATTLIST data key CDATA #IMPLIED>

<!-- A 'crop-rect' specifies crop values as a percentage of original frame height.
-->
<!ELEMENT crop-rect (param*)>
<!ATTLIST crop-rect left CDATA "0">
<!ATTLIST crop-rect top CDATA "0">
<!ATTLIST crop-rect right CDATA "0">
<!ATTLIST crop-rect bottom CDATA "0">

<!-- A 'trim-rect' specifies trim values as a percentage of original frame height.
-->
<!ELEMENT trim-rect (param*)>
<!ATTLIST trim-rect left CDATA "0">
<!ATTLIST trim-rect top CDATA "0">
<!ATTLIST trim-rect right CDATA "0">
<!ATTLIST trim-rect bottom CDATA "0">

<!-- A 'pan-rect' specifies the initial or final crop values for a "Ken Burns"
animation. -->
<!-- The attributes of a pan-rect cannot be keyframed. -->
<!ELEMENT pan-rect EMPTY>
<!ATTLIST pan-rect left CDATA "0">
<!ATTLIST pan-rect top CDATA "0">
<!ATTLIST pan-rect right CDATA "0">
<!ATTLIST pan-rect bottom CDATA "0">

```

```

<!-- An 'info-asc-cdl' describes a primary color correction in the form of an ASC
      CDL (American Society of Cinematographers Color Decision List). -->
<!-- In FCPX XML v1.5, the ASC CDL information is placed under 'filter-video'
      element as comment, for example: -->
      <!-- info-asc-cdl: slope="1.0 1.0 1.0" offset="0 0 0" power="1.0 1.0 1.0" -->
<!-- Each component is a vector of "red green blue" adjustments. -->

<!-- The 'adjust-crop' element modifies the visible image width and height. -->
<!-- This element contains an optional adjustment for each crop mode, although
      only one mode is active. -->
<!ELEMENT adjust-crop (crop-rect?, trim-rect?, (pan-rect, pan-rect)?)>
<!ATTLIST adjust-crop mode (trim | crop | pan) #REQUIRED>
<!ATTLIST adjust-crop enabled (0 | 1) "1">

<!ELEMENT adjust-corners (param*)>
<!ATTLIST adjust-corners enabled (0 | 1) "1">
<!ATTLIST adjust-corners botLeft CDATA "0 0">
<!ATTLIST adjust-corners topLeft CDATA "0 0">
<!ATTLIST adjust-corners topRight CDATA "0 0">
<!ATTLIST adjust-corners botRight CDATA "0 0">

<!-- The absence 'adjust-conform' element implies "fit" -->
<!ELEMENT adjust-conform EMPTY>
<!ATTLIST adjust-conform type (fit | fill | none) "fit">

<!ELEMENT adjust-transform (param*)>
<!ATTLIST adjust-transform enabled (0 | 1) "1">
<!ATTLIST adjust-transform position CDATA "0 0">
<!ATTLIST adjust-transform scale CDATA "1 1">
<!ATTLIST adjust-transform rotation CDATA "0">
<!ATTLIST adjust-transform anchor CDATA "0 0">

<!ELEMENT adjust-blend (param*, reserved*)>
<!ATTLIST adjust-blend amount CDATA "1.0">
<!ATTLIST adjust-blend mode CDATA #IMPLIED>

```

```
<!ELEMENT adjust-stabilization (param*)>
<!ATTLIST adjust-stabilization type (automatic | inertiaCam | smoothCam) "automatic">

<!ELEMENT adjust-rollingShutter EMPTY>
<!ATTLIST adjust-rollingShutter amount (none | low | medium | high | extraHigh)
"none">

<!ELEMENT adjust-loudness EMPTY>
<!ATTLIST adjust-loudness amount CDATA #REQUIRED>
<!ATTLIST adjust-loudness uniformity CDATA #REQUIRED>

<!ELEMENT adjust-noiseReduction EMPTY>
<!ATTLIST adjust-noiseReduction amount CDATA #REQUIRED>

<!ELEMENT adjust-humReduction EMPTY>
<!ATTLIST adjust-humReduction frequency (50 | 60) #REQUIRED>

<!ELEMENT adjust-EQ (param*)>
<!ATTLIST adjust-EQ mode (flat | voice_enhance | music_enhance | loudness |
hum_reduction | bass_boost | bass_reduce | treble_boost | treble_reduce) #REQUIRED>

<!ELEMENT adjust-matchEQ (data)>

<!ELEMENT adjust-volume (param*)>
<!ATTLIST adjust-volume amount CDATA "0dB">

<!ELEMENT adjust-panner (param*)>
<!ATTLIST adjust-panner mode CDATA #IMPLIED>
<!ATTLIST adjust-panner amount CDATA "0">
<!ATTLIST adjust-panner original_decoded_mix CDATA #IMPLIED>
<!ATTLIST adjust-panner ambient_direct_mix CDATA #IMPLIED>
<!ATTLIST adjust-panner surround_width CDATA #IMPLIED>
<!ATTLIST adjust-panner left_right_mix CDATA #IMPLIED>
<!ATTLIST adjust-panner front_back_mix CDATA #IMPLIED>
```

```

<!ATTLIST adjust-panner LFE_balance CDATA #IMPLIED>
<!ATTLIST adjust-panner rotation CDATA #IMPLIED>
<!ATTLIST adjust-panner stereo_spread CDATA #IMPLIED>
<!ATTLIST adjust-panner attenuate_collapse_mix CDATA #IMPLIED>
<!ATTLIST adjust-panner center_balance CDATA #IMPLIED>

<!-- The 'intrinsic-params' entities declare intrinsic video and audio adjustments.
-->
<!ENTITY % intrinsic-params-video "(adjust-crop?, adjust-corners?, adjust-conform?,
adjust-transform?, adjust-blend?, adjust-stabilization?, adjust-rollingShutter?)">
<!ENTITY % intrinsic-params-audio "(adjust-loudness?, adjust-noiseReduction?,
adjust-humReduction?, (adjust-EQ | adjust-matchEQ)?, adjust-volume?,
adjust-panner?)">
<!ENTITY % intrinsic-params "(%intrinsic-params-video;, %intrinsic-params-audio;)">

<!-- The 'timing-params' entity declare rate conform and time mapping adjustments.
-->
<!ENTITY % timing-params "(conform-rate?, timeMap?)">

<!-- The 'anchor_item' entity declares the valid anchorable story elements. -->
<!-- When present, anchored items must have a non-zero 'lane' value. -->
<!ENTITY % anchor_item "audio | video | clip | title | mc-clip | ref-clip | audition
| spine">

<!-- The 'clip_item' entity declares the primary story elements that may appear
inside a clip. -->
<!ENTITY % clip_item "audio | video | clip | title | mc-clip | ref-clip | audition
| gap">

<!ENTITY % marker_item "(marker | chapter-marker | rating | keyword |
analysis-marker)">

<!-- The 'audio_comp_items' entity declares the component-level audio adjustments
for a clip. -->
<!ENTITY % audio_comp_items "(audio-source*, audio-aux-source*)">

<!ENTITY % video_filter_item "(filter-video | filter-video-mask)">

```

```

<!-- An 'audio-source' element adjusts playback settings for a single audio component
in a clip's primary audio layout -->
<!-- The primary audio layout is comprised of all audio from elements in the primary
(lane 0) storyline. -->
<!ELEMENT audio-source (%intrinsic-params-audio;; filter-audio*, mute*)>
<!ATTLIST audio-source name CDATA #IMPLIED>
<!ATTLIST audio-source srcCh CDATA #REQUIRED>          <!-- source audio channels
(comma separated, 1-based index) -->
<!ATTLIST audio-source outCh CDATA #IMPLIED>          <!-- output audio channels
(comma separated, from: L,R,C,LFE,Ls,Rs,X) %outputChannel -->
<!ATTLIST audio-source role CDATA #IMPLIED>          <!-- output role assignment
-->
<!ATTLIST audio-source enabled (0 | 1) '1'>

<!-- An 'audio-aux-source' element adjusts playback settings for a single audio
component in a clip's auxiliary audio layout. -->
<!-- The auxiliary audio layout is comprised of all audio from elements outside
(i.e. anchored to) the primary storyline. -->
<!ELEMENT audio-aux-source (%intrinsic-params-audio;; filter-audio*, mute*)>
<!ATTLIST audio-aux-source name CDATA #IMPLIED>
<!ATTLIST audio-aux-source srcCh CDATA #REQUIRED>          <!-- source audio channels
(comma separated, 1-based index) -->
<!ATTLIST audio-aux-source outCh CDATA #IMPLIED>          <!-- output audio channels
(comma separated, from: L,R,C,LFE,Ls,Rs,X) %outputChannel -->
<!ATTLIST audio-aux-source role CDATA #IMPLIED>          <!-- output role
assignment -->
<!ATTLIST audio-aux-source enabled (0 | 1) '1'>

<!-- An 'audition' is a container with one active story element followed by
alternative story elements. -->
<!ELEMENT audition (audio | video | title | ref-clip | clip)+ >
<!ATTLIST audition %ao_attrs;>

<!-- A 'spine' is a container for elements ordered serially in time. -->
<!-- Only one story element is active at a given time, except when a transition
is present. -->
<!ELEMENT spine (%clip_item; | transition)* >
<!ATTLIST spine      %ao_attrs;>

```

```

<!ATTLIST spine name CDATA #IMPLIED>
<!ATTLIST spine format IDREF #IMPLIED>                                <!-- default is same as
parent -->

<!-- A 'sequence' is a container for a spine of story elements in a sequence
project. -->
<!ELEMENT sequence (note?, spine, %audio_comp_items;, metadata?)>
<!ATTLIST sequence %media_attrs;>
<!ATTLIST sequence audioLayout (mono | stereo | surround) #IMPLIED>
<!ATTLIST sequence audioRate %audioHz; #IMPLIED>
<!ATTLIST sequence renderFormat CDATA #IMPLIED>
<!ATTLIST sequence keywords CDATA #IMPLIED>

<!-- A 'multicam' is a container for multiple "angles" of related content. -->
<!ELEMENT multicam (mc-angle*, metadata?)>
<!ATTLIST multicam %media_attrs;>
<!ATTLIST multicam renderFormat CDATA #IMPLIED>

<!-- An 'mc-angle' is a container for elements ordered serially in time for one
angle of a multicam clip.-->
<!-- Only one story element is active at a given time, except when a transition
is present. -->
<!ELEMENT mc-angle ((%clip_item; | transition)*, %audio_comp_items;) >
<!ATTLIST mc-angle name CDATA #IMPLIED>
<!ATTLIST mc-angle angleID CDATA #REQUIRED>

<!-- An 'mc-clip' element defines an edited range of a/v data from a source
'multicam' media. -->
<!ELEMENT mc-clip (note?, %timing-params;, %intrinsic-params-audio;, mc-source*,
(%anchor_item;)*, (%marker_item;)*, filter-audio*, metadata?)>
<!ATTLIST mc-clip ref IDREF #REQUIRED>                                <!-- 'media' ID -->
<!ATTLIST mc-clip %clip_attrs;>
<!ATTLIST mc-clip srcEnable (all | audio | video) "all">
<!ATTLIST mc-clip audioStart %time; #IMPLIED>
<!ATTLIST mc-clip audioDuration %time; #IMPLIED>

```

```

<!-- An 'mc-source' element defines custom settings and filters to apply to an
angle of a multicam clip or edit. -->
<!ELEMENT mc-source (%audio_comp_items;; %intrinsic-params-video;;
(%video_filter_item;)*>
<!ATTLIST mc-source angleID CDATA #REQUIRED>
<!ATTLIST mc-source srcEnable (all | audio | video | none) "all">

<!-- A 'clip' is a container for other story elements. -->
<!-- Clips have only one primary item, and zero or more anchored items. -->
<!-- Use 'audioStart' and 'audioDuration' to define J/L cuts (i.e., split edits)
on composite A/V clips. -->
<!ELEMENT clip (note?, %timing-params;; %intrinsic-params;; (spine | (%clip_item;))*
(%marker_item;)*, %audio_comp_items;; (%video_filter_item;)*, filter-audio*,
metadata?>
<!ATTLIST clip %clip_attrs;>
<!ATTLIST clip format IDREF #IMPLIED>                <!-- default is same as parent
-->
<!ATTLIST clip audioStart %time; #IMPLIED>
<!ATTLIST clip audioDuration %time; #IMPLIED>
<!ATTLIST clip tcStart %time; #IMPLIED>                <!-- clip timecode origin
-->
<!ATTLIST clip tcFormat (DF | NDF) #IMPLIED>          <!-- timecode display format
(DF=drop frame; NDF=non-drop frame) -->

<!-- A 'ref-clip' is a clip that references (rather than contains) other story
elements. -->
<!-- Clips have a media reference and zero or more anchored items. -->
<!-- Use 'audioStart' and 'audioDuration' to define J/L cuts (i.e., split edits)
on composite A/V clips. -->
<!ELEMENT ref-clip (note?, %timing-params;; %intrinsic-params;; (%anchor_item;)*,
(%marker_item;)*, %audio_comp_items;; (%video_filter_item;)*, filter-audio*,
metadata?>
<!ATTLIST ref-clip ref IDREF #REQUIRED>                <!-- 'media' ID -->
<!ATTLIST ref-clip %clip_attrs;>
<!ATTLIST ref-clip srcEnable (all | audio | video) "all">
<!ATTLIST ref-clip audioStart %time; #IMPLIED>
<!ATTLIST ref-clip audioDuration %time; #IMPLIED>

```



```

<!-- An 'audio' element defines a range of audio data in a source asset. -->
<!ELEMENT audio (note?, %timing-params;, %intrinsic-params-audio;, (%anchor_item;)*,
(%marker_item;)*, filter-audio*)>
<!ATTLIST audio ref IDREF #REQUIRED>                                <!-- 'asset' or 'effect' ID
-->
<!ATTLIST audio %clip_attrs;>
<!ATTLIST audio srcID CDATA #IMPLIED>                                <!-- source/track identifier
in asset (if not '1') -->
<!ATTLIST audio role CDATA #IMPLIED>
<!ATTLIST audio srcCh CDATA #IMPLIED>                                <!-- source audio channels
in asset (comma separated, 1-based index) -->
<!ATTLIST audio outCh CDATA #IMPLIED>                                <!-- output audio channels
(comma separated, from: L,R,C,LFE,Ls,Rs,X) %outputChannel -->

<!-- A 'video' element defines a range of video data in a source asset. -->
<!ELEMENT video (param*, note?, %timing-params;, %intrinsic-params-video;,
(%anchor_item;)*, (%marker_item;)*, (%video_filter_item;)*, reserved?)>
<!ATTLIST video ref IDREF #REQUIRED>                                <!-- 'asset' or 'effect' ID
-->
<!ATTLIST video %clip_attrs;>
<!ATTLIST video srcID CDATA #IMPLIED>                                <!-- source/track identifier
in asset (if not '1') -->
<!ATTLIST video role CDATA #IMPLIED>                                <!-- default is 'video' -->

<!-- A 'gap' element defines a placeholder with no associated media. -->
<!-- Gaps cannot be anchored to other objects. -->
<!ELEMENT gap (note?, (%anchor_item;)*, (%marker_item;)*, metadata?)>
<!ATTLIST gap name CDATA #IMPLIED>
<!ATTLIST gap offset %time; #IMPLIED>
<!ATTLIST gap start %time; #IMPLIED>
<!ATTLIST gap duration %time; #REQUIRED>
<!ATTLIST gap enabled (0 | 1) "1">

<!-- A 'title' element contains one or more 'text' elements that customize a
referenced effect. -->
<!ELEMENT title (param*, text*, text-style-def*, note?, %intrinsic-params-video;,
(%anchor_item;)*, (%marker_item;)*, (%video_filter_item;)*, metadata?)>
<!ATTLIST title ref IDREF #REQUIRED>                                <!-- 'effect' ID for a Motion
template -->

```

```

<!ATTLIST title %clip_attrs;>
<!ATTLIST title role CDATA #IMPLIED>

<!-- A 'text' element defines an unformatted text string for a 'title' element.
-->
<!ELEMENT text (#PCDATA | text-style)*>

<!ELEMENT text-style-def (text-style)>
<!ATTLIST text-style-def id ID #REQUIRED>
<!ATTLIST text-style-def name CDATA #IMPLIED>

<!-- A 'text-style-def' element defines the style for a formatted text string in
a 'text' element. -->
<!ELEMENT text-style (#PCDATA | param)*>
<!ATTLIST text-style ref IDREF #IMPLIED>
<!ATTLIST text-style font CDATA #IMPLIED>
<!ATTLIST text-style fontSize CDATA #IMPLIED>
<!ATTLIST text-style fontFace CDATA #IMPLIED>
<!ATTLIST text-style fontColor CDATA #IMPLIED>
<!ATTLIST text-style bold (0 | 1) #IMPLIED>
<!ATTLIST text-style italic (0 | 1) #IMPLIED>
<!ATTLIST text-style strokeColor CDATA #IMPLIED>
<!ATTLIST text-style strokeWidth CDATA #IMPLIED>
<!ATTLIST text-style baseline CDATA #IMPLIED>
<!ATTLIST text-style shadowColor CDATA #IMPLIED>
<!ATTLIST text-style shadowOffset CDATA #IMPLIED>
<!ATTLIST text-style shadowBlurRadius CDATA #IMPLIED>
<!ATTLIST text-style kerning CDATA #IMPLIED>
<!ATTLIST text-style alignment (left | center | right | justified) #IMPLIED>
<!ATTLIST text-style lineSpacing CDATA #IMPLIED>
<!ATTLIST text-style tabStops CDATA #IMPLIED>

<!-- A 'transition' element defines an effect that overlaps two adjacent story
elements. -->
<!-- For example,

```

```

<video ref="1" duration="5s"/>
<transition duration="2s"/>
<video ref="3" duration="5s"/>

```

Here, the transition element overlaps the last 2 seconds of the previous video (ref="1") and the first 2 seconds of the next video (ref="3"). -->

```

<!ELEMENT transition (filter-video?, filter-audio?, (%marker_item;)*, metadata?,
reserved?)>

```

```

<!ATTLIST transition name CDATA #IMPLIED>

```

```

<!ATTLIST transition offset %time; #IMPLIED>

```

```

<!ATTLIST transition duration %time; #REQUIRED>

```

<!-- A 'filter-video' defines a video effect including intrinsic color filter effect that's applied to its parent element. -->

<!-- Video filters are concatenated in the order in which they appear. -->

<!-- There may be a comment in the form of 'info-asc-cdl' element that describes the correction, if it's the Final Cut Pro X's intrinsic color correction filter. -->

```

<!ELEMENT filter-video (info-asc-cdl?, data?, param*)>

```

```

<!ATTLIST filter-video ref IDREF #REQUIRED>
<!-- 'effect' ID -->

```

```

<!ATTLIST filter-video name CDATA #IMPLIED>

```

```

<!ATTLIST filter-video enabled (0 | 1) "1">

```

```

<!ENTITY % mask_item "(mask-shape | mask-isolation)">

```

<!-- A 'filter-video-mask' defines a video effect with a mask applied to its parent element. -->

<!-- The mask may consist of multiple shapes or color isolation. -->

<!-- For a color correction filter, in addition to the correction applied inside the mask, -->

<!-- There may be another set of correction parameters applied outside the mask. -->

<!-- For example,

```

<filter-video-mask>

```

```

  <mask-shape name="Shape Mask" blendMode="add">

```

```

    <param name="Radius" key="160" value="750 150"/>

```

```

    <param name="Curvature" key="159" value="0.7"/>

```

```

    <param name="Feather" key="102" value="170"/>

```

```

    <param name="Transforms" key="200">

```

```

        <param name="Position" key="201" value="10 110"/>
        <param name="Rotation" key="202" value="14"/>
    </param>
</mask-shape>
<filter-video ref="100"/>
</filter-video-mask>

```

This represents a masked video filter with a single shape mask. -->

<!-- Another example,

```

<filter-video-mask>
    <mask-shape name="Shape Mask" blendMode="add">
        <param name="Curvature" key="159" value="0.7"/>
        <param name="Transforms" key="200">
            <param name="Position" key="201" value="260 -30"/>
        </param>
    </mask-shape>
    <filter-video ref="1000">
        <param name="color_global" value="0.2 0.8"/>
    </filter-video>
    <filter-video ref="1000"/>
        <param name="color_highlight" value="0.3 0.6"/>
    </filter-video>
</filter-video-mask>

```

This represents a color correction filter with a single shape mask and different correction parameters applied inside and outside of the mask.

Note the first 'filter-video' element represents the inner correction and the second corresponds to the outer correction. -->

<!-- Video filters are concatenated in the order in which they appear. -->

<!ELEMENT filter-video-mask ((%mask\_item;)+, (filter-video, filter-video?))>

<!-- second filter-video is for outer color correction only -->

<!ATTLIST filter-video-mask enabled (0 | 1) "1">

<!ATTLIST filter-video-mask inverted (0 | 1) "0">

<!-- The 'mask-shape' element describes a shape mask used for a mask filter. -->

<!ELEMENT mask-shape (param\*)>

<!ATTLIST mask-shape name CDATA #IMPLIED>

```

<!ATTLIST mask-shape enabled (0 | 1) "1">
<!ATTLIST mask-shape blendMode (add | subtract | multiply) "add">

<!-- The 'mask-isolation' element describes a color mask used for mask filter. -->
<!ELEMENT mask-isolation (data, param*)>
<!ATTLIST mask-isolation name CDATA #IMPLIED>
<!ATTLIST mask-isolation enabled (0 | 1) "1">
<!ATTLIST mask-isolation blendMode (add | subtract | multiply) "multiply">

<!-- A 'filter-audio' defines an audio effect that's applied to its parent element.
-->
<!-- Audio filters are concatenated in the order in which they appear. -->
<!ELEMENT filter-audio (data?, param*)>
<!ATTLIST filter-audio ref IDREF #REQUIRED <!-- 'effect' ID -->
<!ATTLIST filter-audio name CDATA #IMPLIED>
<!ATTLIST filter-audio enabled (0 | 1) "1">
<!ATTLIST filter-audio presetID CDATA #IMPLIED>

<!-- A 'conform-rate' defines how the clip's frame rate should be conformed to the
sequence frame rate -->
<!ELEMENT conform-rate EMPTY>
<!ATTLIST conform-rate scaleEnabled (0 | 1) "1">
<!ATTLIST conform-rate srcFrameRate (23.98 | 24 | 25 | 29.97 | 30 | 60 | 47.95 |
48 | 50 | 59.94) #IMPLIED>
<!ATTLIST conform-rate frameSampling (floor | nearest-neighbor | frame-blending |
optical-flow-classic | optical-flow) "floor">

<!-- A 'timeMap' is a container for 'timept' elements that change the output speed
of the clip's local timeline. -->
<!-- When present, a 'timeMap' defines a new adjusted time range for the clip using
the first and last 'timept' elements. -->
<!-- All other time values are interpolated from the specified 'timept' elements.
-->
<!ELEMENT timeMap (timept)*>
<!ATTLIST timeMap frameSampling (floor | frame-blending | optical-flow-classic |
optical-flow) "floor">
<!ATTLIST timeMap preservesPitch (0 | 1) "1">

```

<!-- A 'timept' defines the re-mapped time values for a 'timeMap'. -->

<!-- For example,

```
<timeMap>
  <timept time="0s" value="0s" interp="linear"/>
  <timept time="10s" value="5s" interp="linear"/>
  <timept time="20s" value="0s" interp="linear"/>
</timeMap>
```

Here, when applied to a clip whose original timeline was 0-5s, the 'timeMap' will adjust the clip's timeline to 0-20s

and play the original content at 50% speed, followed by -50% speed. -->

<!-- ELEMENT timept EMPTY -->

<!-- ATTLIST timept time %time; #REQUIRED --> <!-- new adjusted clip time -->

<!-- ATTLIST timept value CDATA #REQUIRED --> <!-- original clip time -->

<!-- ATTLIST timept interp (smooth2 | linear | smooth) "smooth2" --> <!-- interpolation type for point. smooth has been deprecated -->

<!-- ATTLIST timept inTime %time; #IMPLIED --> <!-- transition in-time for point (used only with smooth interpolations) -->

<!-- ATTLIST timept outTime %time; #IMPLIED --> <!-- transition out-time for point (used only with smooth interpolations) -->

<!-- KEYWORDS, MARKERS, NOTES -->

<!-- If 'completed' is specified, this marker becomes a to-do item. -->

<!-- ELEMENT marker EMPTY -->

<!-- ATTLIST marker start %time; #REQUIRED -->

<!-- ATTLIST marker duration %time; #IMPLIED -->

<!-- ATTLIST marker value CDATA #REQUIRED -->

<!-- ATTLIST marker completed CDATA #IMPLIED --> <!-- (0=not completed, 1=completed) -->

<!-- ATTLIST marker note CDATA #IMPLIED -->

<!-- ELEMENT rating EMPTY -->

<!-- ATTLIST rating name CDATA #IMPLIED -->

<!-- ATTLIST rating start %time; #IMPLIED -->

<!-- ATTLIST rating duration %time; #IMPLIED -->

```

<!ATTLIST rating value (favorite | reject) #REQUIRED>
<!ATTLIST rating note CDATA #IMPLIED>

<!ELEMENT keyword EMPTY>
<!ATTLIST keyword start %time; #IMPLIED>
<!ATTLIST keyword duration %time; #IMPLIED>
<!ATTLIST keyword value CDATA #REQUIRED>          <!-- comma-separated list of
keywords -->
<!ATTLIST keyword note CDATA #IMPLIED>

<!ELEMENT analysis-marker (shot-type | stabilization-type)+>
<!ATTLIST analysis-marker start %time; #IMPLIED>
<!ATTLIST analysis-marker duration %time; #IMPLIED>

<!ELEMENT keyword-collection EMPTY>
<!ATTLIST keyword-collection name CDATA #REQUIRED>

<!ELEMENT collection-folder (%collection_item;)*>
<!ATTLIST collection-folder name CDATA #REQUIRED>

<!-- SMART COLLECTIONS -->
<!ELEMENT smart-collection ((match-text | match-ratings | match-media | match-clip
| match-stabilization | match-keywords | match-shot | match-property | match-time
| match-timeRange | match-roles)*)>
<!ATTLIST smart-collection name CDATA #REQUIRED>
<!ATTLIST smart-collection match (any | all) #REQUIRED>

<!ELEMENT match-text EMPTY>
<!ATTLIST match-text enabled (0 | 1) "1">
<!ATTLIST match-text rule (includes | doesNotInclude | is | isNot) "includes">
<!ATTLIST match-text value CDATA #REQUIRED>

<!ELEMENT match-ratings EMPTY>
<!ATTLIST match-ratings enabled (0 | 1) "1">
<!ATTLIST match-ratings value (favorites | rejected) #REQUIRED>

```

```
<!ELEMENT match-media EMPTY>
<!ATTLIST match-media enabled (0 | 1) "1">
<!ATTLIST match-media rule (is | isNot) "is">
<!ATTLIST match-media type (videoWithAudio | videoOnly | audioOnly | stills)
#REQUIRED>

<!ELEMENT match-clip EMPTY>
<!ATTLIST match-clip enabled (0 | 1) "1">
<!ATTLIST match-clip rule (is | isNot) "is">
<!ATTLIST match-clip type (audition | synchronized | compound | multicam |
layeredGraphic | project) #REQUIRED>

<!ELEMENT match-stabilization (stabilization-type*)>
<!ATTLIST match-stabilization enabled (0 | 1) "1">
<!ATTLIST match-stabilization rule (includesAny | includesAll | doesNotIncludeAny
| doesNotIncludeAll) "includesAny">

<!ELEMENT match-keywords (keyword-name*)>
<!ATTLIST match-keywords enabled (0 | 1) "1">
<!ATTLIST match-keywords rule (includesAny | includesAll | doesNotIncludeAny |
doesNotIncludeAll) "includesAny">

<!ELEMENT keyword-name EMPTY>
<!ATTLIST keyword-name value CDATA #REQUIRED>

<!ELEMENT match-shot (shot-type*)>
<!ATTLIST match-shot enabled (0 | 1) "1">
<!ATTLIST match-shot rule (includesAny | includesAll | doesNotIncludeAny |
doesNotIncludeAll) "includesAny">

<!ELEMENT shot-type EMPTY>
<!ATTLIST shot-type value (onePerson | twoPersons | group | closeUp | mediumShot
| wideShot) #REQUIRED>

<!ELEMENT stabilization-type EMPTY>
<!ATTLIST stabilization-type value (excessiveShake) #REQUIRED>
```



```

<!ELEMENT match-property EMPTY>
<!ATTLIST match-property enabled (0 | 1) "1">
<!ATTLIST match-property key (reel | scene | take | audioOutputChannels | frameSize
| videoFrameRate | audioSampleRate | cameraName | cameraAngle) #REQUIRED>
<!ATTLIST match-property rule (includes | doesNotInclude | is | isNot) "includes">
<!ATTLIST match-property value CDATA #REQUIRED>

```

```

<!ELEMENT match-time EMPTY>
<!ATTLIST match-time enabled (0 | 1) "1">
<!ATTLIST match-time type (contentCreated | dateImported) #REQUIRED>
<!ATTLIST match-time rule (is | isBefore | isAfter) #REQUIRED>
<!ATTLIST match-time value CDATA #REQUIRED>

```

```

<!ELEMENT match-timeRange EMPTY>
<!ATTLIST match-timeRange enabled (0 | 1) "1">
<!ATTLIST match-timeRange type (contentCreated | dateImported) #REQUIRED>
<!ATTLIST match-timeRange rule (isInLast | isNotInLast) #REQUIRED>
<!ATTLIST match-timeRange value CDATA #REQUIRED>
<!ATTLIST match-timeRange units (hour | day | week | month | year) #IMPLIED>

```

```

<!ELEMENT match-roles (role*)>
<!ATTLIST match-roles enabled (0 | 1) "1">
<!ATTLIST match-roles rule (includesAny | includesAll | doesNotIncludeAny |
doesNotIncludeAll) "includesAny">

```

```

<!ELEMENT role EMPTY>
<!ATTLIST role name CDATA #REQUIRED>

```

```

<!ELEMENT chapter-marker EMPTY>
<!ATTLIST chapter-marker start %time; #REQUIRED>
<!ATTLIST chapter-marker duration %time; #IMPLIED>
<!ATTLIST chapter-marker value CDATA #REQUIRED>
<!ATTLIST chapter-marker note CDATA #IMPLIED>
<!ATTLIST chapter-marker posterOffset %time; #IMPLIED>

```

<!ELEMENT note (#PCDATA)>

<!ELEMENT bookmark (#PCDATA)>

<!ELEMENT reserved (#PCDATA)>

<!ELEMENT array (string\*)>

<!ELEMENT string (#PCDATA)>

# Document Revision History

This table describes the changes to *Final Cut Pro X XML Format*.

Date	Notes
2015-04-13	Updated for Final Cut Pro X XML v1.5.
2014-06-30	Updated for Final Cut Pro X XML v1.4.
2013-12-18	Updated for Final Cut Pro X XML v1.3.
2012-10-23	Updated for Final Cut Pro X XML v1.2.
2012-01-30	Updated for Final Cut Pro X XML v1.1.
2011-09-19	New document that describes the XML format used by Final Cut Pro X.



Apple Inc.  
Copyright © 2015 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer or device for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-branded products.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Final Cut, Final Cut Pro, Mac, Sand, Shake, and Spotlight are trademarks of Apple Inc., registered in the U.S. and other countries.

Adobe, Acrobat, and PostScript are trademarks or registered trademarks of Adobe Systems Incorporated in the U.S. and/or other countries.

Helvetica is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

**APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT, ERROR OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

Some jurisdictions do not allow the exclusion of implied warranties or liability, so the above exclusion may not apply to you.