Motion XML File Format

Apple Applications > Motion

2008-04-14

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, 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 OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Figures, Tables, and Listings

About This Document

This document describes components of the Motion XML file format. In particular, it provides information about XML elements of the Motion scene graph. (A Motion scene graph specifies the objects in a Motion project, their attributes, the media files they incorporate, and their dependencies on other Motion objects.)

You can use the information in this document to directly edit the XML specification of a Motion scene graph and thus modify a Motion project offline. Possible modifications include changing scene object parameters, replacing media files, and editing text objects. You might also develop a utility that lets you automate modifications to Motion projects.

Topics

This document describes aspects of version 3 of the Motion XML file format. In particular it discusses:

- the Motion scene graph.
- the Motion scene objects.
- related parameter settings.

It does *not* describe:

- filter parameters.
- behavior parameters.
- canvas and viewer details.
- Motion UI state.
- curve interpolation formulae.
- scene object and parameter flags.

This document assumes that you are familiar with general XML conventions and the Motion application.

Organization

This document contains the following chapters:

- "Motion XML Overview" (page 9)
- "Elements, Subelements, and Attributes" (page 15)
- "The Properties Parameter" (page 27)

- "The Object Parameter" (page 31)
- "Channel Folders and Related Elements" (page 41)
- "Customizing a Motion XML Project File " (page 55)

Typographical Conventions

Elements appear in computer voice and initial lower case: `scenenode`, `layer`, and so on.

Names of attributes of elements appear in computer voice and initial lower case, and are followed by the = symbol: `uuid=`, `factoryID=`, and so on.

Values for attributes appear in computer voice and initial uppercase: `Properties`, `Position`, and so on.

An asterisk (*) following an element indicates there may be 0 or more instances of the element.

A plus sign (+) following an element indicates there must be at least 1 or more instances of the element.

An ellipses (. . .) in a code listing indicates that some lines of code are not displayed.

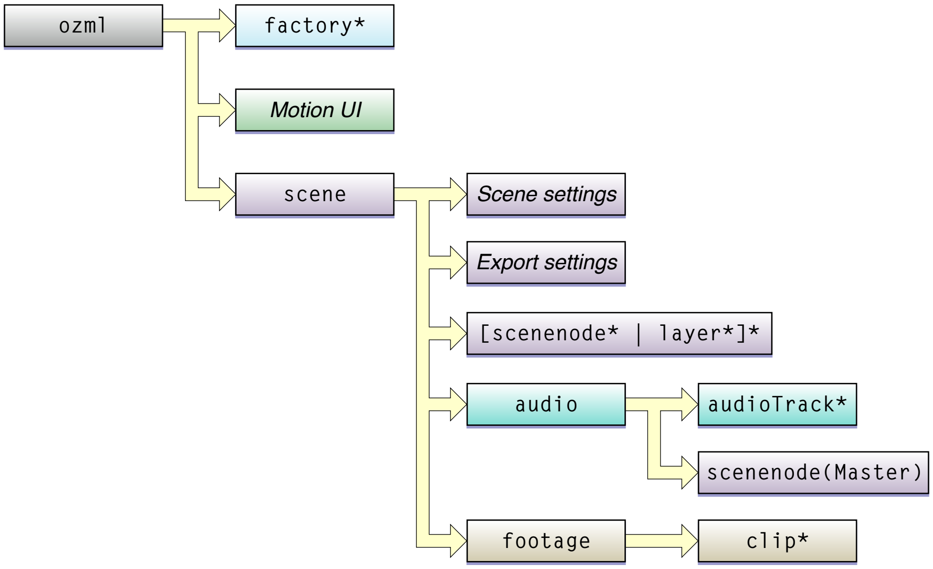
Motion XML Overview

A Motion XML file is stored on disk as plain text document that you can view, parse, and edit on any platform. There are two types of Motion XML files: the Motion project file and the Motion library object file.

Motion Project File

A Motion project file has these major components:

Figure 2-1 The Motion project file



- A declaration of the Motion XML version used in the file. A file using version 3.0 of Motion XML begins with the declaration: <ozml version="3.0">.
- A list of Motion factories used in the project. Factories are encoded with a factory element. They have a direct relation with objects in the scene graph. See "Factories" (page 12).
- Information about Motion UI state—the state of windows, tabs, and so on—at the time a project was saved.
- The top-level scene element that encodes scene settings, export settings, and scene graph information.
- The Motion scene graph encoded with elements such as scenenode, layer, audio, footage, and so on.

The Motion Scene Graph

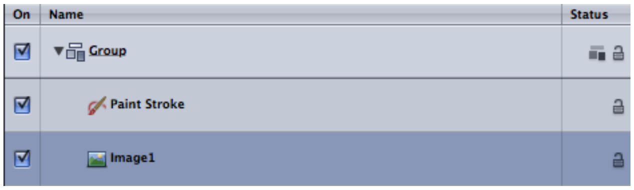
The XML for a Motion scene graph has a hierarchical structure that replicates the hierarchical structure of a Motion project displayed in the Layer tab of the Project Pane in the Motion UI.

Listing 2-1 A Scene Graph Example

```
<layer name="Group" id="10000">
   <scenenode name="Paint Stroke" id="10013" factoryID="7" version="3">
   ...
   </scenenode>
   <scenenode name="Image1" id="10011" factoryID="5" version="3">
   ...
   </scenenode>
   ...
   <aspectRatio>1</aspectRatio>
   ...
</layer>
```

In Listing 2-1, the `layer` element contains two attributes: `name=` and `id=`. There are three subelements of `layer`: a `scenenode` element representing a paint stroke, a second `scenenode` element representing an image, and an `aspectRatio` element representing a floating-point number. Note that the `aspectRatio` element is a subelement of the `layer` element and encodes a characteristic of the Motion layer, not the `Paint Stroke` or `Image1` elements. These have their own `aspectRatio` subelements (not shown here).

In the Motion UI, the group that generated the XML in Listing 2-1 (page 10), would appear as follows:



On	Name	Status
☑	▼ Group	
☑	Paint Stroke	
☑	Image1	

The group contains two elements, just as the `layer` XML element contains the Paint Stroke `scenenode` and the Image `scenenode`. Note that the hierarchy and ordering in the Layers tab matches that of the XML file. Note also that a `layer` element can be a subelement of a parent `layer` element. In other words, `layer` elements can be nested.

Generally, a Motion XML project file is verbose. All subelements of a scene object or parameter are written to the file, regardless of their actual use in the project. For instance, a file containing a movie clip contains the "`Retime Value`" parameter even if the movie clip is set to play at a constant speed.

Channels and Parameters

A *channel* is an element that encodes a specific value that determines some aspect of the appearance or behavior of a scene object. For example, an element such as:

```
<parameter name="X" id="1" flags="16" value="-166.0831249"/>
```

encodes a value for a channel called X.

A *channel folder* is a collection of related channels that determine a particular aspect of appearance or behavior. For example:

```
<parameter name="Position" id="101" flags="4112">
  . . .
  <parameter name="X" id="1" flags="16" value="-166.0831249"/>
  <parameter name="Y" id="2" flags="16" value="-83.97980993"/>
  <parameter name="Z" id="3" flags="16" value="357.2331479"/>
</parameter>
```

The `Position` channel folder contains three channels (X, Y, Z) that specify the position of a scene object.

A reference to a channel folder in this document may be flagged with a "–>" symbol. For example: `Position`–>, `Brush Color`–>, or `Scale`–>.

Note: The data types for values encoded by channels are listed in the table: Table 2-1 (page 14)

Other parameters group related channels and channel folders together. For example:

```
<parameter name="Transform" id="100" flags="4112">
  <parameter name="Position" id="101" flags="4112">
    . . .
    <parameter name="X" id="1" flags="16" value="-166.0831249"/>
    <parameter name="Y" id="2" flags="16" value="-83.97980993"/>
    <parameter name="Z" id="3" flags="16" value="357.2331479"/>
  </parameter>
  <parameter name="Rotation" id="109" flags="4112">
  . . .
  <parameter name="Scale" id="105" flags="4112">
  . . .
  <parameter name="Shear" id="106" flags="4112">
  . . .
  <!-- and so on -->
</parameter>
```

The `Transform` parameter groups together several channel folders related to the transformation of a scene object: `Position`, `Rotation`, `Scale`, `Shear`, and so on.

At a higher level of the XML hierarchy, the `Properties` and `Object` parameters provide the complete specifications for the properties and definitions of a scene object. See "The Properties Parameter" (page 27) and "The Object Parameter" (page 31).

Constant and Variable Channel Values

You can specify the value of a channel as a constant, or use a `curve` element to make it vary over time.

Listing 2-2 Constant and Variable Channel Values Example

```
<parameter name="Properties" id="1" flags="4112">
  <parameter name="Transform" id="100" flags="4112">
    <parameter name="Position" id="101" flags="4112">
      <foldFlags>15</foldFlags>
      <parameter name="X" id="1" flags="16" value="22.5" />
      <parameter name="Y" id="1" flags="16">
       <curve type="1" round="0">
```

```
        <numberOfKeypoints>2</numberOfKeypoints>
        <keypoint flags="32">
          . . .
          <time>0</time>
          <value>20</value>
          . . .
          </keypoint>
          <keypoint flags="32">
          . . .
          <time>30</time>
          <value>100</value>
          . . .
        </keypoint>
       </curve>
      </parameter>
      <parameter name="Z" id="1" flags="16" value="10" />
    </parameter> ["Position"]
   . . .
  </parameter> ["Transform"]
 . . .
</parameter> ["Properties"]
```

The X channel uses the value= attribute to specify the X position of the scene object as a constant (22.5). In a similar fashion, the Z channel specifies the Z position as a constant (10).

The Y channel represents the Y position as a curve with values that change over time. It has two keypoints, one at time 0 and the second at time 30. Note that in this case the channel does not have a value= attribute.

Generally, a single-value channel is equivalent to a scene object in the Motion project that contains no keyframes. A parameter that contains a curve subelement is equivalent to a object in a Motion project that has one or more keyframes.

Factories

Motion factories are used to type Motion scene objects and parameters. They are listed at the top of the XML project file.

Scene objects and parameters refer to a factory by referencing its id= attribute in their own factoryID= attribute. Associating a scene object or parameter with a factory in this way tells Motion the type of subcomponents to expect during internal processing.

Note: In this document, to indicate that an element is typed with a factory, the name of the factory appears in parentheses after name of the element. For example: scenenode(Text), scenenode(Image), and so on.

Listing 2-3 Factory Example

```
<ozml version="3.0">
  <factory id="1" uuid="66fc0d6af6a911d6a7a7000393670732">
    <description>Image</description>
    <manufacturer>Apple</manufacturer>
    <version>1</version>
  </factory>CodeLine
  . . .
```

```
<layer name="Group" id="10000">
  <scenenode name="Movie1" id="10420" factoryID="1" version="3">
  . . .
  </scenenode>
  . . .
</layer>
```

In Listing 2-3, the XML file begins with a specification of an image `factory`. Then a `scenenode` is specified that references this `factory`. Specifically, the value of 1 for the `factoryID=` attribute in the `scenenode` references the `id=` attribute value of 1 in the `factory`. The `scenenode` is typed as an image. (An image can be a movie clip, an image, an image sequence, a PDF file, or other media.)

Because the name of a Motion object in the Layers list can be changed arbitrarily, you should use the `factoryID=` attribute to reference a `factory` element.

Here is a formal description of the `factory` element:

factory

Description A Motion factory used to type Motion scene objects in the Motion XML file.

Attributes `id=` : the factory's unique ID in the project file. (This value may differ in a different project file.)

`uuid=` : the factory's UUID. You should not modify this value.

Subelements `description` : a string describing the type of Motion object that this factory represents.

`manufacturer` : a string specifying the originating manufacturer of the factory.

`version` : an integer encoding the version of this Motion factory.

Motion Library Object File

A Motion library object file ends with the `.molo` extension and may contain several Motion objects such as custom presets, shapes, and text styles. The objects in this file share the same XML structure as those in a Motion project file with the addition of two unique elements:

- `primaryObjects` : an element specifying a Motion library object. It has a single `id` subelement.
- `primaryFactories` : an element specifying the factories associated with the Motion objects in the library file. It has a single `uuid` subelement.

Here is an example of a Motion library object file representing a saved shape style:

Listing 2-4 Motion library object example

```
<ozml version="3.0">
 <factory id="1" uuid="0e8d443513b611d89395000a95af9f7e">
   <description>Channel</description>
   <manufacturer>Apple</manufacturer>
   <version>1</version>
 </factory>
 . . .
 <primaryObjects>
   <id>10055</id>
 </primaryObjects>
 <primaryFactories>
   <uuid>9f0e69d364f44d58b1f4502de48df903</uuid>
 </primaryFactories>
 <scenenode name="Paint Stroke" id="10055" factoryID="6" version="3">
 ...
 </scenenode>
 <clip name="Image" id="10093">
 ...
 </clip>
</ozml>
```

In Listing 2-4, a shape style saved in the library object file uses an image as its outline source. This is encoded in the scenenode element and clip elements. The primaryObjects element specifies an id subelement whose value (10055) references the id= of the Paint Stroke scenenode. The primaryFactories element contains a uuid subelement that specifies the primary Motion object in the file.

Data Types

Data values in Motion XML include the following data types:

Table 2-1 Data Types

Channel type	Valid values
float	Any floating-point number
integer	Any integer
unsigned integer	Any positive integer
id	An unsigned integer corresponding to the id attribute of some Motion object
Boolean	0 or 1
enumerated	A valid integer; see the specific channel description for valid integers
percent	Any floating-point number between 0.0 and 1.0, inclusive
angle	Any floating-point number, in radians or degrees as specified by the channel description
frame	A floating-point number, typically constrained to the number of frames in the scene node or project the channel is associated with unless otherwise specified

Elements, Subelements, and Attributes

This chapter provides a general description of the elements, subelements, and attributes that encode scene objects. It includes the following sections:

Higher-Level Elements

> **Important:** All of the elements listed here include some or all of the subelements described in the section "Common Subelements" (page 19). To determine the exact subelements for an element, examine the element's XML coding in the Motion project file.
>
> The attributes for these elements are described in the section "Common Attributes" (page 24).
>
> The name in parentheses after the name of an element refers to the type of factory the element is associated with. For example, **scenenode(Camera)** means the `factoryID=` attribute of the `scenenode` references the `id=` attribute of a Camera factory in the XML file. See Listing 2-3 (page 12).

layer

Description	A Motion layer (also known in the UI as a group).
Attributes	`name=`, `id=`.
Subelements	`[scenenode*	layer*]*` : layers or scenenodes a layer contains. The order in which `layer` and `scenenode` elements appear follows the Motion scene graph hierarchy. `mask*` : shape or image masks in the layer.

mask(Shape)

Description	A Motion shape mask.
Attributes	`name=`, `id=`, `factoryID=`
Subelements	`scenenode(Replicator)`

`isShapeStyle` : Motion uses this element internally. You should not modify it.

mask(Image)

Description	A Motion image mask.
Attributes	`name=`, `id=`, `factoryID=`
Subelements	See "Common Subelements" (page 19).

scenenode(Camera)

Description	A Motion camera.
Attributes	`name=`, `id=`, `factoryID=`, `version=`
Subelements	This element does not contain `validTracks` or `filter` subelements.

scenenode(Light)

Description	A Motion light.
Attributes	`name=`, `id=`, `factoryID=`, `version=`
Subelements	This element does not contain `validTracks` or `filter` subelements.

scenenode(Image)

Description	A Motion image object representing a media file such as an image, an image sequence, a movie clip, an Adobe PDF or Photoshop file.
Attributes	`name=`, `id=`, `factoryID=`, `version=`
Subelements	`numberOfPages` (unsigned interger) : the number of pages in a PDF file. Present only if a PDF file has more than one page. `linkedObjects` (id): the id of the audioTrack element this movie clip is linked to, if the movie clip contains audio (located after flags). Present only if an audio track is linked to this `scenenode`.

scenenode(Text)

Description	A Motion text object.
Attributes	`name=`, `id=`, `factoryID=`, `version=`
Subelements	`style+`: styles a text layout incorporates. `styleRun` : style runs a text layout incorporates. `text` (string) : the text in a text layout. `object*`: describes a glyph in a text layout. There must be one object element for every glyph in a text layout.

scenenode(Shape)

Description	A Motion shape such as a rectangle, circle, curve, line, or paint stroke.
Attributes	`name=`, `id=`, `factoryID=`, `version=`
Subelements	`scenenode(Replicator)` `isShapeStyle` : an element used internally by Motion that you should not modify.

scenenode(Generator)

Description A Motion generator.

Attributes `name=`, `id=`, `factoryID=`, `version=`

`pluginUUID=` (UUID) : internal plugin UUID for this generator.

`pluginVersion` (float) : internal plugin version of this generator.

`pluginName` (string) : internal name of this generator.

Subelements See "Common Subelements" (page 19).

scenenode(Emitter)

Description A Motion emitter. Motion masks, shapes, and paint strokes contain a `scenenode` element whose `name=` attribute is `"Emitter"` and whose `factoryID=` attribute corresponds to a Motion replicator. These are not Motion emitters but rather a special class of Motion replicator used to enhance masks, shapes, and paint strokes. See `scenenode` (replicator).

Attributes `name=`, `id=`, `factoryID=`, `version=`

Subelements `scenenode(Particle Cell)*` : particle cell sources that this emitter incorporates.

scenenode(Replicator)

Description A Motion replicator. May also represent a Motion mask or shape If not specifying a Motion replicator, the `name=` attribute is `"Emitter"` and the `factoryID=` attribute corresponds to a replicator.

Attributes `name=`, `id=`, `factoryID=`, `version=`

Subelements `scenenode(Replicator Cell)*` : replicator cell sources that this replicator incorporates.

`Behavior ("Pen Pressure")` Note: present only if this replicator represents a shape or paint stroke and incorporates a shape style.

`Behavior ("Pen Speed")` Note: present only if this replicator represents a shape or paint stroke and incorporates a shape style

scenenode(Particle Cell)

Description An individual cell from a Motion emitter.

Attributes `name=`, `id=`, `factoryID=`, `version=`

Subelements See "Common Subelements" (page 19).

scenenode(Replicator Cell)

Description An individual cell from a Motion replicator. May also specify a Motion mask or shape cell.

Attributes `name=`, `id=`, `factoryID=`, `version=`

Subelements See "Common Subelements" (page 19).

footage

Description A top-level element that serves as a folder for a project's `clip` elements.

Attributes `id=`, `name=`. The `name=` attribute is set to `"Media Layer"` and should not be changed.

Subelements `clip*`

clip

Description A media file, such as a movie file, image, or audio file, found in the Media tab of the Project pane.

Attributes `id=`, `name=`.

Subelements `pathURL` (URL): absolute URL to the file on disk that this clip represents.

`relativeURL` (URL) : the relative URL to the file with respect to the location of the Motion project file. Present only if the source file is not in the same folder as the Motion project file.

`missingWidth` (integer): the width of this file. Used as a placeholder when the file is missing from disk.

`missingHeight` (integer): the height of this file. Used as a placeholder when the file is missing from disk

`missingDuration` (float) : the duration of this file in seconds. Equal to the reciprocal of the project framerate if the clip represents an image. Used as a placeholder when the file is missing from disk.

`creationDuration` (integer) : duration of this element in frames with respect to the project framerate. Used as a placeholder for the time line when the file is missing from disk

`mediaID` (UUID): UUID of clip if present in file. Empty if no UUID is present.

audio

Description A top-level element that serves as a folder for a project's `audiotrack` elements.

Attributes `id=`, `name=`. The `name=` attribute is set to `"Audio Layer"` and should not be changed.

Subelements `audioTrack*`

`scenenode(Master)`

scenenode(Master)

Description A top-level node that represents Motion's master audio settings.

Attributes `name=`, `id=`, `factoryID=`, `version=`

Subelements See "Common Subelements" (page 19).

audioTrack

Description An audio track in a Motion project, as found in the Audio tab of the Project pane.

Attributes `id=`, `name=`

Subelements `linkedObjects` (id): the id of the `scenenode` element that this `audioTrack` is linked to. Present only if the audio track is linked to a `scenenode` element

filter

Description A Motion filter.

Attributes `name=`, `id=`, `factoryID=`

`pluginUUID=` (UUID) : internal plugin UUID for this filter.

`pluginVersion` (float) : internal plugin version of this filter.

`pluginName` (string) : internal name of this filter.

Subelements Filter-specific parameters. A complete description of filter parameters is beyond the scope of this document. Refer to the *Motion User Manual* for details about each filter.

behavior

Description A Motion behavior.

Attributes `name=`, `id=`, `factoryID=`

Subelements Behavior-specific parameters. A complete description of behavior parameters is beyond the scope of this document. Refer to the *Motion User Manual* for details about each behavior.

Common Subelements

Many scene objects and parameters contain similar subelements describing general characteristics of the scene object.

CommonReserved Subelements

Motion uses these subelements for internal processing. You should not change them.

override

Description An unsigned integer Motion uses for backward compatibility purposes. You should not modify this element.

ignoreBehaviorsBeforeID

Description An integer that Motion uses internally. You should not modify this element.

flags

Description An unsigned integer that Motion uses for internal processing. You should not modify this element.

foldFlags

Description An unsigned integer that Motion uses for internal processing. You should not modify this element.

baseFlags

Description An unsigned integer that Motion uses for internal processing. You should not modify this element.

Common Timing Subelements

timing

Description An element encoding a scene object's position on the timeline.

Attributes

`in=` (frame) : the in point of a scene object in the timeline.

`out=` (frame) : the out point of a scene object in the timeline.

`offset=` (frame) : the frame where the media's first frame begins. This is applicable to Motion image objects referencing movie clips.

Notes

As an example, if a 20-frame movie clip is dropped into the project and its object track slipped so that it begins at frame 30, the `timing` element then appears as: `<timing in="30" out="49" offset="30" />`.

If the left edge of the object track is moved 5 frames to the right, effectively setting the In point to 35 and the duration to 15 frames, the `timing` element is then: `<timing in="35" out="49" offset="30" />`.

timemarkerset

Description

The set of time markers for a scene object.

Subelement

`timemarker*`

timemarker

Description

Specifies a time marker for a Motion scene object as it appears in the timeline.

Subelements

`inpoint` (frame): the frame where this time marker appears.

`outpoint` (frame): the frame where this time marker ends. This element appears only if the time marker's duration is greater than 1

`label` (string): the name of this time marker. This element appears if the time marker has an assigned name.

`comment` (string): the comment for this time marker. This element appears if the time marker has an assigned comment.

`color` (enumerated): 0: red; 1: green; 2: blue; 3: purple; 4: orange; 5: yellow; 6: gray; 7: cyan.

`type` (enumerated): 0: standard; 1: DVD menu loop; 2: DVD alpha transition.

Common Parameter Subelements

Properties

Description

A `parameter` element with this value for the `name=` attribute specifies a scene object's unique properties. See "The Properties Parameter" (page 27).

Object

Description

A `parameter` element with this value for the `name=` attribute specifies a scene object's unique settings. See "The Object Parameter" (page 31).

Common Filter Subelement

filter*

Description

A filter that this scene object incorporates. To see the XML details of a particular filter, you can include it in a Motion project and then examine the XML project file.

Common Behavior Subelement

behavior*

Description

A behavior that this scene object incorporates. To see the XML details of a particular behavior, you can include it in a Motion project and then examine the XML project file.

Other Common Subelements

aspectRatio

Description

A floating point number that sets the aspect ratio of a scene object

enabled

Description

A Boolean value. Only appears in the XML file if the value is 0 and object is disabled.

locked

Description

A Boolean value. Only appears in the XML file if the value is 1 and object is locked.

Curve Subelements

curve

Description

A curve for a channel. It represents a channel's value across time based on keypoints and their interpolation types. In some cases, a `curve` may actually represent a constant value. In this case, its `type=` attribute is not present, but a `value=` attribute is.

Attributes

`type=` (enumerated): the type of value the curve represents: not present: constant 1: spline 3: used for backward-compatibility purposes.

`parametric=` (Boolean): 1 if a curve is parametric; 0 otherwise (and the attribute is not visible). Used internally by Motion and should no be modified.

`round=` (Boolean): 1 if the curve values are rounded; 0 otherwise. Used internally by Motion and should no be modified.

`retimingExtrapolation=` (Boolean): 1 if a curve is used for retiming footage and has constant timing; 0 otherwise (and not visible).

`value=` (dependent on the channel). Visible only if the value is a constant.

Subelements

`numberOfKeyPoints` (integer): the number of keypoints in this curve

`closed` (Boolean): 1 if the value of the channel this curve represents is interpolated after the last keyframe to values of the first keyframe; 0 otherwise (and not visible). This element is generally used only for curves representing a text object's path position parameters.

`preExtrapolation` (enumerated): the curve extrapolation method for values before the first keyframe: not visible: constant; 1: linear; 2: Ping-Pong; 3: repeat; 4: progressive repeat.

`postExtrapolation` (enumerated): the curve extrapolation method for values after the last keyframe: not visible: constant;1: linear; 2: Ping-Pong; 3: repeat; 4: progressive repeat.

`keyPoint*`: a keypoint (also known as a keyframe for parameters) for this curve. The number of `keyPoint` elements must equal the value of the `numberOfKeyPoints` element.

keyPoint

Description A keypoint for a curve. (Also known as a keyframe for parameters.)

Attributes `interpolation=` (enumerated): the interpolation method for this keypoint. Not present: bezier; 0: constant; 1: linear; 6: continuous; 7: ease In; 8: ease Out; 13: exponential; 14: logarithmic.

`flags=`

Subelements `time` (frame): the frame where this keypoint appears on the timeline. For curve elements representing text object paths, the `time` subelement is the index of the keypoint on the text path, starting at 0.

`value` (dependent on the channel): the value of the channel at this keypoint's time.

`bias` (percent): the bias for the keypoint. Used by `keyPoint` elements for B-spline curves. Visible in the XML file only if the bias is not 1.0

`normal` (float): the normal value of this keypoint, if applicable. For curve elements representing text object paths, the `normal` subelement is used to determine the change in the normal vector of the path at this keypoint.

`inputTangentTime` (float): independent variable for the input handle. Visible in the XML file only if the keypoint is not the first in a curve.

`inputTangentValue` (float): dependent variable for the input handle; Visible in the XML file only if the keypoint is not the first in a curve.

`outputTangentTime` (float): independent variable for the output handle. Visible in the XML file only if the keypoint is not the last in a curve.

`outputTangentValue` (float): dependent variable for the output handle. Visible in the XML file only if the keypoint is not the last in a curve.

curve_X curve_Y

Description A container for a series of vertices pertaining to a Motion shape. The element `curve_X` describes the X positions of the vertices of a shape. The element `curve_Y` describes the Y positions of the vertices.

Subelements `vertex*`: a vertex for this shape in the specified dimension. There must be as many `vertex` elements as points in the shape.

vertex

Description A vertex for a Motion shape.

Attributes `index=` (integer): the index of a vertex in the shape. The first vertex is 0.

`flags=`. See "Motion Project File" (page 9).

Subelements `vertex_folder`: a container for elements describing this vertex.

vertex_folder

Description A container of elements describing a vertex. If the positions of a vertex are keyframed, the parameters in `vertex_folder` are curves; otherwise, they are constant values.

Attributes `name=`(defaults to "Vertex"), `id=`, `flags=`. See "Motion Project File" (page 9)

Subelements

Several parameter subelements specify a vertex_folder element.

Enabled (Boolean): 1 if this point is enabled, 0 otherwise.

Value(float): the value of the vertex that this vertex folder represents.

Bias(percent): the point bias for this vertex. Used when this vertex represents a Motion B-Spline curve.

Input Tangent (float): the value of the input tangent handle for this vertex. Used when this vertex represents a Motion Bezier curve.

Output Tangent (float): the value of the output tangent handle for this vertex. Used when this vertex represents a Motion Bezier curve.

Text Subelements

font

Description

A Motion font name.

Attributes

type= (Boolean): 0 if representing a system font; 1 if a LiveFont.

Notes

The font name is specified by a string that provides the PostScript or LiveFont name of a font. System fonts are named as they appear in the Font Book application. LiveFonts are named as they appear in the Family drop-down box of the Format pane in the Inspector. They are prefaced by the font collection. (For example, Skywriter is named "Pro Series/Skywriter," Chance is named "Collectors Edition/Chance").

style

Description

Encodes the features of a text style. (The style element does not contain information regarding the portions of text it affects. See the styleRun element.)

Attributes

name=, id= , factoryID= , version=. Note: the name= attribute is set according to the following convention: for a text layout object containing n styles, the first style will be named Style; the second style will be named Style 1, and so forth until Style (n - 1).

Subelements

font

copyFlags (unsigned integer) : you should not modify this element that Motion uses internally.

The subelements for style include the following channel folders:

Size (float): the size in points of the font referenced in this style.

Tracking (float): the tracking amount in pixels for this style.

Kerning (float): the kerning amount for this style, in pixels.

Baseline (float): the baseline amount for this style, in pixels.

Slant (float): the slant amount for this style, in pixels.

Offset-

Rotation->

Rotation: a parameter used for backward compatibility that should not be modified.

Monospace (Boolean): 1 if enabled; 0 if disabled.

All Caps (Boolean): 1 if enabled; 0 if disabled.

All Caps Size (percent): the size of the capital letters relative to the size of the font.

Style Preset : a parameter used internally by Motion that should not be modified.

Face–>: a container for the parameters for the style's face.

Drop Shadow–>: a container for the parameters for the style's drop shadow.

Outline–>: a container for the parameters for the style's outline.

Glow–>: a container for the parameters for the style's glow.

LiveFont Timing–>: a container describing the parameters for LiveFont timing.

styleRun

Description An element describing the range of characters a particular style affects within a text layout scenenode.

Attributes style= (id): the id of the style that this style run incorporates.

offset= (unsigned integer): the index of the character where this style run begins, where the index of the first character is 0.

length= (unsigned integer): the number of consecutive characters this style run affects.

text

Description An element containing the literal string for a text layout object.

Notes The text for this text layout object; empty if the text layout object contains no text.

object

Description Encodes information for a glyph in a text layout object.

Attributes value= (unsigned integer): an decimal ASCII character code.

Subelement Kerning (float): the kerning value for this glyph in pixels.

Common Attributes

In general, Motion uses the attributes listed here for internal processing. The common attributes include: name=, id=, factoryID=, and version=.

name=

Description A string encoding a name for a scene object in the Layers list or parameter. The name= attribute of a scene object generally matches the name found in the Layers list of the Project Pane. The name= attribute of a parameter generally matches the name found in the Inspector or other relevant UI element.

For instance, a name= attribute with the value Position in a parameter element references the position controls found in the Transform section of the Properties tab in the Inspector. This parameter has subelements for X, Y, and Z , also found when the Position control in the Inspector is collapsed.

Generally, you should not modify the name= attribute of a parameter element. On the other hand, you can change name= attributes of elements referring to Motion scene objects, such as layer and scenenode.

id=

Description

An unsigned integer identifying an object or parameter. Many elements in the Motion XML file format contain an `id=` attribute. Generally, this attribute is unique to the element and is used internally by Motion to identify its role in the project. You should not modify an `id=` attribute, particularly for `parameter` elements.

factoryID=

Description

An unsigned integer identifying a factory. Motion uses this attribute to identify the internal structures of an object by associating the object with a factory. (See "Factories" (page 12)). You should not modified this attribute.

version=

Description

An unsigned integer specifying Motion version information. Some objects contain a `version=` attribute that references the major version of Motion used to create this object. Motion uses this attribute for internal processing. You should not modify it.

More About the id= Attribute

The `id=` attributes for `scenenode`, `layer`, `footage`, `audioTrack`, and other elements that specify scene objects may be interrelated.

For example, if a movie clip scene object appears in the scene graph, its Media parameter will refer to a clip element's `id=` attribute to describe the source media of the movie clip. Additionally, if the movie clip scene object references an audio track, the movie clip scene object will refer to the audio track's `id=` attribute and vice versa:

Listing 3-1 id= Example

```
<scenenode name="Movie1" id="10395" factoryID="1" version="3">
  . . .
  <linkedobjects>10397<linkedobjects>
  . . .
  <parameter name="Object">
  . . .
    <parameter name="Media" id="300" flags="65552" value="10393" />
  </parameter>
</scenenode>
<audio name="Audio Layer" id="10398">
  <audioTrack name="Movie1" id="10397">
  . . .
    <linkedobjects>10395<linkedobjects>
    . . .
    <parameter name="Properties" id="1" flags="4112">
      <parameter name="Media" id="104" flags="65552" value="10393" />
      . . .
    </parameter>
    . . .
  </audioTrack>
  . . .
</audio>
<footage name="Media Layer" id="10394">
  <clip name="Movie1" id="10393">
  . . .
  </clip>
  . . .
</footage>
```

Since many scene objects may use the same media file as their source, it may be the case that the media parameters for many `scenenode` elements reference the same `clip` element.

As the code sample above shows, the `id=` attribute (10393) for a `clip` element may be referenced by other `scenenode` or `parameter` elements. When modifying the `id=` attribute for a `scenenode` element, you may need to modify the related `scenenode` or `parameter` elements accordingly.

The `id=` attribute of a `parameter` element is used to identify a channel. You should not modify it.

The Properties Parameter

The `Properties` parameter described in this chapter corresponds to the Properties tab in the Inspector. It encodes basic properties such as an object's transform, blending and drop shadow.

The exact components of a `Properties` parameter vary depending on the scene object the parameter is describing.

Note: Many subelements of a `Properties` parameter are channel folders that encode a set of related channels (such as X, Y, Z values.) In this chapter, these parameters are flagged with a "–>" symbol. They are described in the chapter "Channel Folders and Related Elements" (page 41).

Structurally Identical Properties Parameters

The following scene objects have a structurally identical `Properties` parameter:

- `mask(Shape).`
- `scenenode(Emitter).`
- `scenenode(Generator).`
- `scenenode(Replicator).`
- `scenenode(Shape Mask).`
- `scenenode(Text).`

Other `Properties` parameters for other scene objects are described below. (See "Other Properties Parameters" (page 29).)

Transform

Description	A scene object's transformation parameters.
Attributes	`id=, flags=`
Subelements	`foldFlags` `Position`–>: a scene object's position in world space. `Rotation`–>: a scene object's rotation. `Scale`–>: a scene object's scale properties. `Shear`–>: a scene object's shearing properties. `Anchor Point`–> : a scene object's anchor point position. `Rotation`: a legacy parameter used for backwards compatibility purposes that should not be modified.
Note	For details about these subelements, see "Transform Channels" (page 41).

Blending

Description A scene object’s blending parameters.

Attributes `id=, flags=`

Subelements `foldFlags`

`Opacity` (percent): the opacity of a scene object.

`Blend Mode` (enumerated): the type of blending used for a scene object. Valid values are: 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.

`Preserve Opacity` (Boolean): 0 if disabled; 1 if enabled.

Lighting

Description A scene object’s lighting parameters. Present only if a Motion light is in the scene.

Attributes `id=, flags=`

Subelements `foldFlags`

`Shading` (enumerated) : 0: inherited; 1: on; 2: off.

`Highlights`: a parameter used internally by Motion that you should not modify.

Drop Shadow

Description A scene object’s drop shadow parameters.

Attributes `id=, flags=`

Subelements `foldFlags`

`Color->`

`Opacity` (percent): the opacity value for this scene object’s drop shadow.

`Blur` (float): the amount of blur to apply to a scene object’s drop shadow.

`Distance` (float): the offset to apply to this scene object’s drop shadow.

`Angle` (angle): the angle at which to apply this scene object's drop shadow, in radians.

`Fixed Source` (Boolean): 0 if disabled, 1 if enable.

Four Corner

Description A scene object’s four-corner transform parameters.

Attributes `id=, flags=`

Subelements `foldFlags`

`Bottom Left->`

`Bottom Right->`

`Top Right->`

`Top Left->`

Crop

Description Encodes a scene object's cropping parameters.

Attributes `id=`, `flags=`

Subelements `foldFlags`

`Left` (float): the left cropping position.

`Right` (float): the right cropping position.

`Bottom` (float): the bottom cropping position.

`Top` (float): the top cropping position.

Timing

Description A parameter used internally by Motion that you should not modify.

Other Properties Parameters

The structure of the `Properties` parameter for other scene object varies according to the object.

scenenode(Camera/Light)

`Properties` `Transform` parameters only.

scenenode(Image)

`Properties` All parameters listed in "Structurally Identical Properties Parameters" (page 27) plus the following:

`Media` (id): the id of the footage node this scene node represents. Keyframes not supported.

`Page Number` (unsigned integer): the page number of the PDF file this scene node represents, if applicable; 0 otherwise.

`Background Color->`

`Speed` (float): the speed for this movie clip when Constant Speed retiming is used. Keyframes not supported.

`Reverse` (Boolean): whether or not to play this movie clip in reverse. 1 = true; 0 = false.

`Time Remap` (enumerated). 0: Constant speed retiming; 1: Variable speed retiming.

`Retime Value` (frame): the frame of the movie clip that will be played with respect to the time at each keyframe. Must be represented as a curve. This parameter defaults to a curve with two keypoints, whose `time` elements correspond to the first and last frame of the movie clip (0-based) and whose `value` elements correspond to the first and last frame of the movie clip (1-based).

`Retime Value Cache` (frame): the previously used curve for movie clip retiming. Must be represented as a curve. See `Retime Value`.

`Frame Blending` (enumerated). 0: none; 1: blending; 2: motion-blur blending; 3: optical flow.

`End Condition` (enumerated). 0: none; 1: loop; 2: ping-pong; 3: hold.

`End Duration` (frame): the number of frames to repeat when the `End Condition` is set to Loop. Keyframes not supported.

`Duration Cache` (frame): the previously held value for `End Duration`. Keyframes not supported.

`Layer` (unsigned integer): the index of the layer of the Photoshop file this scene node represents. 0 if not applicable

audioTrack

`Properties`

`Media` (id): the id of the audio element this audio track represents. No support for keyframing.

`Timing` : a parameter used internally by Motion that you should not modify.

`Retime Value` (frame): the frame of the audio track that will be played with respect to the time at each keyframe. Must be encoded as a `curve`. This parameter defaults to a `curve` with two keypoints whose `time` elements correspond to the first and last frame of the audio track (0-based) and whose `value` elements correspond to the first and last frame of the audio track (1-based).

`Retime Value Cache` (frame): the previously used curve for audio track retiming. Must be encoded as a `curve`. See `Retime Value`.

`Speed` (float): the speed for this audio track when Constant Speed retiming is used. Does not support keyframes.

`Time Remap` (enumerated). 0: Constant speed retiming; 1: Variable speed retiming.

The following scene objects have a Properties parameter with no subelements:

- `clip`
- `footage`
- `scenenode(Image Mask)`
- `scenenode(Master)`
- `scenenode(Particle Cell)`
- `scenenode(Replicator Cell)`

The Object Parameter

All scenenode and layer elements contain an Object parameter that encodes an object's unique characteristics. These parameters typically appear in the scene object's tab in the Inspector and usually labeled "Image," "Text," "Behavior," and so on.

The components of an Object parameter are based on a scene object's factoryID= attribute, if present. Thus, you can check the components of an Object parameter by using the object's factoryID= to associate the object with a factory. Then make sure the components match the components specifed in this chapter.

Note: Many subelements of a Object parameter are channel folders that encode a set of related channels (such as X, Y, Z values.) In this chapter, these parameters are flagged with a "–>" symbol. They are described in the chapter "Channel Folders and Related Elements" (page 41).

Scene Objects

The scene objects and their object parameters listed in this chapter include:

- "layer" (page 32)
- "mask(Image Mask)" (page 32)
- "scenenode(Camera)" (page 32)
- "scenenode(Light)" (page 33)
- "scenenode(Clone Layer)" (page 33)
- "scenenode(Image Object)" (page 33)
- "scenenode(Text)" (page 33)
- "scenenode(Shape)" (page 34)
- "scenenode(Generator)" (page 34)
- "scenenode(Behavior)" (page 34)
- "scenenode(Filter)" (page 34)
- "scenenode(Emitter or Replicator)" (page 34)
- "scenenode(Particle Cell or Replicator Cell)" (page 35)
- "footage" (page 38)
- "clip" (page 38)
- "audio" (page 38)
- "audioTrack" (page 38)
- "scenenode(Master)" (page 39)

layer

`Object`

`Type` (Boolean): 0 if this layer is 2D; 1 if 3D.

`Fixed Resolution` (Boolean): 1 if this layer's size is specified in fixed resolution; 0 if not.

`Fixed Width` (unsigned integer): the width of this layer. Keyframes not supported.

`Fixed Height` (unsigned integer): the height of this layer. Keyframes not supported.

`Flatten` (Boolean): 1 if flattened; 0 if not.

`Layer Order` (Boolean): 1 if objects in this layer are sorted by their order in the Layers list; 0 if not.

`Aperture Width` : this parameter is used for internal processing and should not be modified.

`Aperture Height` : this parameter is used for internal processing and should not be modified.

mask(Image Mask)

`Object`

`Mask->`

`Source Channel` (enumerated). 0: red; 1: green; 2: blue; 3: alpha; 4: luminance.

`Mask Blend Mode` (enumerated). 0: add; 1: subtract; 2: replace; 3: intersect.

`Invert Mask` (Boolean). 1 if enabled; 0 if disabled.

`Stencil` (Boolean). 1 if enabled; 0 if disabled.

`Stretch` (Boolean). 1 if enabled; 0 if disabled.

mask(Shape Mask)

`Object`

`Shape Animation->`.

`Shape Type` (enumerated). 0: linear;1: bezier; 2: B-spline.

`Closed` (Boolean): 1 if this mask represents a closed curve; 0 if not.

`Initial Position`: a parameter used internally by Motion that should not be modified.

`Fill->`

`Outline->`

`Feather` (float) : the number of pixels to feather this mask.

`Mask Blend Mode` (enumerated). 0: add; 1: subtract; 2: replace; 3: intersect.

`Invert Mask` (Boolean): 0 if disabled; 1 if enabled

`Initial Scale X` : a parameter used internally by Motion that should not be modified.

`Initial Scale Y` : a parameter used internally by Motion that should not be modified.

`Use Old B-Spline Model` : a parameter used internally by Motion that should not be modified.

scenenode(Camera)

`Object`

`Camera Type` (enumerated). 0: view point; 1: framing.

`Angle of View` (angle): angle of view, in degrees.

`Near Plane` (float): the position of the near plane for this Camera.

`Far Plane` (float): the position of the far plane for this Camera relative to the Near Plane.

`Near Fade` (float): the offset from the near plane at which fading begins.

`Far Fade` (float): the offset from the far plane at which fading begins.

scenenode(Light)

`Object`

`Light Type` (enumerated). 0: ambient; 1: directional; 2: point; 3: spot.

`Color->`

`Intensity` (float): the intensity of this light.

`Falloff Start` (float): the distance from the center of the light where intensity begins to fall off.

`Falloff` (float): amount of change in intensity.

`Spot Options->`

scenenode(Clone Layer)

`Object`

`Clone Layer->`

scenenode(Image Object)

`Object`

`Drop Zone` (Boolean): 1 if this scene node serves as a drop zone; 0 if not.

`Fit` (enumerated). 0: fit; 1: center; 2: stretch.

`Replaced`: a parameter used internally by Motion that should not be modified.

`Fit Factor`: a parameter used internally by Motion that should not be modified.

`Clear`: a parameter used internally by Motion that should not be modified.

`Width`: a parameter used internally by Motion that should not be modified.

`Height`: a parameter used internally by Motion that should not be modified.

scenenode(Text)

`Object`

`Tracking` (float): the tracking value for this text object.

`Flatten` (Boolean): 1 if flattened; 0 if not.

`Face Camera` (Boolean): 1 if glyphs always face the camera, 0 if not.

`Layout Method` (enumerated). 0: type; 1: paragraph; 2: path.

`Direction`: a parameter used internally by Motion that should not be modified.

`Alignment` (enumerated). 0: left; 1: center; 2: right.

`Justification` (enumerated). 0: none; 1: partial; 2: full.

`Line Spacing` (float): the line spacing for this text object.

`Type On` –>: Type On settings for this text object.

`Left Margin` (float): the position of the left margin relative to the Position of the text layout.

`Right Margin` (float): the position of the right margin relative to the Position of the text layout.

`Top Margin` (float): the position of the top margin, relative to the Position of the text layout.

`Bottom Margin` (float): the position of the bottom margin, relative to the Position of the text layout.

`Sequence->`: used internally by Motion and should not be modified.

`Format->`: used internally by Motion and should not be modified.

`Face->`

`Outline->`

`Glow->`

`Drop Shadow->`

`Path Options->`

`Publish to FCP` (Boolean): 0 if disabled, 1 if enabled.

scenenode(Shape)

`Object`

`Shape Animation->`

`Shape Type` (enumerated). 0: linear; 1: bezier. 2: B-spline.

`Closed` (Boolean): 1 if this shape represents a closed curve; 0 if not.

`Initial Position`: a parameter used internally by Motion that should not be modified.

`Fill->`

`Outline->`

`Feather->`

`Mask Blend Mode` (enumerated). 0: add. 1: subtract. 2: replace. 3: intersect.

`Invert Mask` (Boolean): 1 if masking area is inverted; 0 if not.

`Initial Scale X`: a parameter used internally by Motion that should not be modified.

`Initial Scale Y`: a parameter used internally by Motion that should not be modified.

`Use Old B-Spline Model`: a parameter used internally by Motion that should not be modified.

scenenode(Generator)

`Object`

Components of the `object` parameter depend on the particular generator selected. Refer to the Motion UI for details.

scenenode(Behavior)

`Object`

Components of the `object` parameter depend on the particular behavior selected. Refer to the Motion UI for details.

scenenode(Filter)

`Object`

Components of the `object` parameter depend on the particular filter selected. Refer to the Motion UI for details.

scenenode(Emitter or Replicator)

`Object`

`Shape Parameters->`

`Emission Angle` (angle): the emission angle, in radians; present only for 2D.

`Emission Latitude` (angle): the emission latitude, in radians; present only for 3D.

`Emission Longitude` (angle): the emission longitude, in radians.

`Emission Range` (angle): the emission range, in radians.

`Master Controls->`

`Tint Color`: a parameter used internally by Motion that should not be modified.

`Tint Amount`: a parameter used internally by Motion that should not be modified.

`Render Order` (enumerated): used for emitter scenenodes only. 0: oldest first; 1: oldest last.

`Reverse Stacking` (Boolean): 0 if disabled; 1 if enabled. Used for replicator scenenodes only.

`Interleave Particles` (Boolean): 0 if disabled; 1 if enabled. Used for emitter scenenodes only.

`Fill Points`: a parameter used internally by Motion that should not be modified.

`Emitter Seed` (unsigned integer): random seed for Shuffle Order parameter. Present only for emitter `scenenodes`.

`Replicate Seed` (unsigned integer): random seed for Shuffle Order parameter. Present only for replicator `scenenodes`.

`Uniform Distribution`: this parameter is used internally by Motion and should not be changed.

`Preview Position`: a parameter used internally by Motion that should not be modified.

`3D` (Boolean): 1 if emitter cells are to be distributed in 3D; 0 if not.

`Render Particles` (enumerated). 0: in global 3D; 1: in local 3D.

`Face Camera` (Boolean): 1 if all cells should face the camera, 0 if not.

`Depth Ordered` (Boolean): 1 if all cells should be drawn from back to front, 0 if drawn in order of emission.

scenenode(Particle Cell or Replicator Cell)

Note

A particle cell or a replicator cell can represent an individual cell for these scene objects: `Replicator`, `Emitter`, `Paint Stroke`, `Shape`.

`Object`

`Birth Rate` (float): the birth rate for this cell. Used when the parent `scenenode` is an emitter.

`Birth Rate Randomness` (float): random variant for the Birth Rate parameter.

`Initial Number` (float): the initial number of particle cells; used when the parent `scenenode` is an emitter.

`Life` (float): the life span of a Particle Cell, in seconds.

`Life Randomness` (float): random variant for the Life parameter.

`Speed` (float): the speed for this cell.

`Speed Randomness` (float): random variant for the Speed parameter.

`Align Angle` (Boolean): 1 if cells should rotate based on their position on the shape of the replicator; 0 if not. Present only when the parent `scenenode` is a Motion replicator or emitter.

Brush Align Angle (Boolean): 1 if cells should rotate based on their position on the length of a paint stroke or shape edge; 0 if not. Present only when the parent scenenode is a Motion paint stroke or shape.

Angle–>: present only when the parent scenenode is a Motion replicator or emitter.

Brush Angle–>: present only when parent scenenode is a Motion paint stroke or shape.

Angle End: this parameter is used for backwards compatibility purposes and should not be modified.

Angle Over Stroke (angle): describes the angle modulation over the length of a paint stroke. Used when the grandparent scenenode represents a Motion paint stroke or shape. Must be represented as a curve.

Angle Randomness–>: present only when the parent scenenode is a Motion replicator or emitter.

Brush Angle Randomness–>: present only when the parent scenenode is a Motion paint stroke or shape.

Spin (angle): amount to spin each cell, in radians.

Spin Randomness (angle): random variant for the Spin parameter.

Color Mode (enumerated). 0: original; 1: colorize; 2: over life; 3: pick from color range; 4: take image color.

Color–>: used only when Color Mode is Colorize.

Opacity Over Life–>: used when Color Mode is Original or Colorize.

Color Gradient–>: present when Color Mode is Over Pattern and the parent scenenode is a replicator.

Opacity Gradient–>: present when Color Mode is Original or Colorize and the parent scenenode is a replicator.

Color Over Life–>: present when Color Mode is Over Life.

Color Range–>: present when Color Mode is Pick From Color Range.

Random Color: this parameter is used for backward compatibility purposes and should not be modified.

Color Repetitions (float) : the number of repetitions to choose color. Used only when Color Mode is Over Life.

Take Image Color: this parameter is used for backward compatibility purposes and should not be modified.

Scale–>

Scale End–>

Scale Randomness–>

Premultiplied: a parameter used internally by Motion that should not be modified.

Additive Blend (Boolean): 0 if disabled; 1 if enabled.

Play Frames (Boolean): 0 if disabled; 1 if enabled.

Random Start Frame (Boolean): 0 if disabled; 1 if enabled.

Source Start Frame (frame): the frame at which the source media begins. Used when the source is a video clip; not used if the Random Start Frame parameter is enabled.

Source Frame Offset (frame): the offset to apply to the Source Start Frame parameter to each replicator cell. Used when the source is a video clip and the grandparent scenenode is a replicator.

`Source Start Frame Over Stroke` (frame): the frame to use from the source footage over the length of a paint stroke. Applicable when the grandparent `scenenode` is a paint stroke. Must be represented as a curve.

`Hold Frames` (unsigned integer): the number of frames to hold each frame of source media. Used when the source is a video clip; not used if the Play Frames parameter is disabled.

`Hold Frames Randomness` (unsigned integer): random variant for the Hold Frames parameter.

`Attach To Emitter` (Boolean): 0 if disabled; 1 if enabled.

`Show Particles As` (enumerated). 0: points; 1: lines; 2: wireframe; 3: image.

`Random Seed` (unsigned integer): random seed.

`Point Size` (float): the point size. Used when the Show Particles As parameter is set to points.

`Dynamics`: a parameter used internally by Motion that should not be modified.

`Spacing` (float): the spacing across particles. Used when the grandparent `scenenode` is a paint stroke.

`Spacing Over Stroke` (float): the spacing curve across the length of a paint stroke. Used only when grandparent `scenenode` is a paint stroke. Must be represented as a curve.

`Width Over Stroke` (float): the width curve across the length of a paint stroke or shape edge. Used only when the grandparent `scenenode` is a Motion paint stroke or shape. Must be represented as a curve.

`Fixed Brush Dabs` (Boolean): 0 if disabled; 1 if enabled. Only used by Paint Strokes.

`Anchor Dabs To` (enumerated): used when parameter Fixed Brush Dabs is disabled. 0: start; 1: start and end.

`Pos Offset`: a parameter used internally by Motion that should not be modified.

`Jitter`–>: the jitter amount to apply to each cell. Used when the grandparent `scenenode` is a paint stroke or shape.

`Jitter Over Stroke`–>: the jitter amount to apply to each cell over the length of a paint stroke or shape edge. Used when the grandparent `scenenode` is a Motion paint stroke or shape.

`Particle Source` (id): the id of the media this cell incorporates as its source. Present only when the parent `scenenode` is a Motion emitter or replicator.

`Brush Source` (id): the id of the media this cell incorporates as its source. Present only when the grandparent `scenenode` is a Motion paint stroke or shape.

`Object Source` (id): the id of the media that this cell incorporates as its source. Present only when the parent `scenenode` is a Motion replicator or emitter.

`Hidden channel`: a parameter used internally by Motion that should not be modified.

`Obsolete Rotation`: a parameter used for backward compatibility that should not be modified.

`Obsolete Rotation End`: a parameter used for backward compatibility that should not be modified.

`Obsolete Rotation Variance`: a parameter used for backward compatibility that should not be modified.

Brush Profile–>: present only when the grandparent scenenode is a Motion paint stroke or shape.

Hidden Opacity Over Stroke: this parameter is used internally by Motion and should not be modified. Present only when the grandparent scenenode is a Motion paint stroke or shape.

footage

Object The Object parameter for the footage element has no subelements.

clip

Object Alpha Type–>

Invert Alpha (Boolean): 1 if enabled; 0 if disabled.

Pixel Aspect Ratio (float): the pixel aspect ratio for this video clip. No keyframing support.

Field Order (enumerated). 0: none; 1: Upper (odd); 2: Lower (even).

3:2 Pulldown (enumerated). 0: off; 1: SSWWW; 2: WSSWW; 3: WWSSW; 4: WWWSS; 5: SWWWS.

Frame Rate (float): the frame rate for this clip. No keyframing support.

Fixed Resolution (Boolean): 1 if the PDF file is displayed at fixed resolution; 0 if not.

Fixed Width (unsigned integer): the width of this PDF file. No keyframing support.

Fixed Height (unsigned integer): the height of this PDF file. No keyframing support.

Use Background Color (Boolean): 0 if disabled; 1 if enabled.

Background Color–>

Crop–>

Replace Media File: a parameter used internally by Motion that should not be modified.

OpenEXR–>

Gamma (float): the gamma value for this clip.

audio

Object The Object parameter for the audio element has no subelements.

audioTrack

Object Amplitude Mix: a parameter used internally by Motion that should not be modified.

Amplitude Left: a parameter used internally by Motion that should not be modified.

Amplitude Right: a parameter used internally by Motion that should not be modified.

Peak Mix: a parameter used internally by Motion that should not be modified.

Peak Left: a parameter used internally by Motion that should not be modified.

Peak Right: a parameter used internally by Motion that should not be modified.

Track ID: this value is used internally by Motion and should not be modified.

channel ID: this value is used internally by Motion and should not be modified.

channels Layout (enumerated). 0: mixed; 1: discrete; 2: single channel

`Mute` (Boolean): 1 if muted; 0 if not.

`Solo` (Boolean): 1 if this track is soloed; 0 if not.

`Level` (float): the level of this audio track, adjustable between 0.0 and 2.0, inclusive.

`Pan` (float): the pan position of this audio track.

`Output Bus` (enumerated). 0: stereo; 1: left; 2: right; 3: center; 4: LFE; 5: left surround; 6: right surround.

scenenode(Master)

`Object`

`Amplitude Mix`: a parameter used internally by Motion that should not be modified.

`Amplitude Left`: a parameter used internally by Motion that should not be modified.

`Amplitude Right`: a parameter used internally by Motion that should not be modified.

`Peak Mix`: a parameter used internally by Motion that should not be modified.

`Peak Left`: a parameter used internally by Motion that should not be modified.

`Peak Right`: a parameter used internally by Motion that should not be modified.

`Mute` (Boolean): 1 if muted; 0 if not.

`Level` (float): the level of this audio track, adjustable between 0.0 and 2.0, inclusive.

`Pan` (float): the pan position of this audio track.

`Output Layout` (enumerated). 0: stereo; 1: 5.1 Surround.

Channel Folders and Related Elements

This chapter provides details about channel folders and related elements. It includes the following sections:

- "Transform Channels"
- "Color Channels" (page 42)
- "Text Channels" (page 44)
- "Shape, Emitter, Replicator Channels" (page 47)
- "Footage and Audio Channels" (page 52)
- "Other Channels" (page 53)

Transform Channels

The channel folders and elements in this section include:

- "Position"
- "Rotation" (page 41)
- "Scale" (page 42)
- "Shear" (page 42)
- "Anchor Point" (page 42)
- "Bottom Left/Bottom Right/Top Right/Top Left" (page 42)

Position

Description Encodes a scene object's position.

Channels `X` (float): the X position of the object.

Term `Y` (float): the Y position of the object.

Term `Z` (float): the Z position of the object.

Rotation

Description Encodes an object's rotation in ZYX order. A rotation parameter with an empty XML tag is maintained for backward compatibility purposes and should not be modified.

Channels `X` (float): the rotation about the X axis, in radians.

`Y` (float): the rotation about the Y axis, in radians.

`Z` (float): the rotation about the Z axis, in radians.

`Animate` (enumerated): 0: use rotation; 1: use orientation.

Scale

Description Encodes an object's scale.

Channels `X` (float): the scale factor of the object about the X axis.

`Y` (float): the scale factor of the object about the Y axis.

`Z` (float): the scale factor of the object about the X axis.

Shear

Description Encodes an object's shearing properties.

Channels `X` (float): the shearing factor of the object about the X axis.

`Y` (float): the shearing factor of the object about the Y axis.

Anchor Point

Description Encodes an object's anchor point.

Channels `X` (float): the X position of the object's anchor point.

`Y` (float): the Y position of the object's anchor point.

`Z` (float): the Z position of the object's anchor point.

Bottom Left/Bottom Right/Top Right/Top Left

Description Encodes each corner of the Four-Corner warping parameter of a scene node.

Channels `X` (float): the X position of this corner.

`Y` (float): the Y position of this corner.

Color Channels

The channel folders and elements in this section include:

- "Background Color" (page 42)
- "RGB" (page 43)
- "RGBn" (page 43)
- "Alpha" (page 43)
- "Alphan" (page 43)
- "Color" (page 44)
- "Gradient" (page 44)
- "End" (page 44)
- "Start" (page 44)

Background Color

Description Encodes background color for various elements.

Channels `Red` (float): the value of the Red channel for this background color.

`Green` (float): the value of the Green channel for this background color.

`Blue` (float): the value of the Blue channel for this background color.

RGB

Description

Encodes a series of gradient color tags.

Attributes

`id=`, `flags=`, `factoryID=`

Subelements

`flags`

`RGBn*->`

RGBn

Description

Encodes color information specific to a gradient color tag. The `name=` attribute for this parameter is set according to the following convention: for a gradient tag containing m gradient color tags, the first parameter's name attribute is RGB1; the second is RGB2, and so on, until RGBn, where n is equal to m.

Channels

`flags`

`Offset` (percent): the absolute location of this color tag on a gradient line, represented as a constant curve when not keyframed. This parameter contains a `factoryID=` attribute referring to a channel. Must be represented as a curve.

`Middle` (percent): the relative location of this color tag's middle point between this color tag and the next, represented as a constant curve when not keyframed. This parameter contains a `factoryID=` attribute referring to a channel. Must be represented as a curve.

`Interpolation` (enumerated): the interpolation method for this color, represented as a constant curve. This parameter contains a factoryID attribute referring to a channel. 0: constant; 1: linear; 2: continuous.

`Color->`: the color at the offset of this color tag. This color parameter's subelements, which represent red, green and blue color channels, must be represented as curves (constant curves when not keyframed). This parameter contains a `factoryID=` attribute referring to a channel.

Alpha

Description

Encodes a series of gradient opacity tags.

Attributes

`id=`, `flags=`, `factoryID=`

Subelements

`flags`

`Alphan*->`

Alphan

Description

Encodes color information specific to a gradient opacity tag. For a gradient tag containing m gradient opacity tags, the first parameter's name attribute is Alpha1; the second is Alpha2, and so on until Alphan, where n is equal to m.

Subelement

`flags`

Channels

`Offset` (percent): the absolute location of this opacity tag on a gradient line, represented as a constant curve when not keyframed. This parameter contains a `factoryID=` attribute referring to a channel.

Middle (percent): the relative location of this opacity tag's middle point between this color tag and the next, represented as a constant curve when not keyframed. This parameter contains a factoryID= attribute referring to a channel.

Interpolation (enumerated): the interpolation method for this opacity tag, represented as a constant curve. This parameter contains a factoryID= attribute referring to a channel. 0: constant; 1: linear; 2: continuous.

Alpha (float): the value of the alpha channel at this point, represented as a constant curve when not keyframed. This parameter contains a factoryID= attribute referring to a channel.

Color

Description
Encodes color in Motion. Can be used to represent individual colors or colors for gradient tags.

Attributes
id=, flags=, factoryID=. The factoryID= for this channel folder; present only for color channel folders that represent colors for gradient tags.

Channels
Red (percent): the amount of red color to contribute to the final output color.

Green (percent): the amount of green color to contribute to the final output color.

Blue (percent): the amount of blue color to contribute to the final output color.

Gradient

Description
Encodes information pertaining to a gradient for various Motion objects.

Subelements
RGB->

Alpha->

Type (enumerated): 0: linear; 1: radial.

Start->: the starting point for this gradient.

End->: the ending point for this gradient.

End

Description
The ending point for a gradient.

Channels
X (float): the X position of this point.

Y (float): the Y position of this point.

Start

Description
The starting point for a gradient.

Channels
X (float): the X position of this point.

Y (float): the Y position of this point.

Text Channels

The channel folders and elements in this section include:

- "LiveFont Timing"

- "Type On" (page 45)
- "Path Options" (page 45)
- "Path" (page 46)
- "Face" (page 46)
- "Outline" (page 46)
- "Glow" (page 47)

LiveFont Timing

Description Encodes the parameters for LiveFont timing.

Channels `Random` (float): a random variant for the order in which LiveFont movies are played. No keyframing supported.

`Random Seed` (unsigned integer): random seed for the Random parameter. No keyframing supported.

`Sequence` (percent): the percentage of a LiveFont movie to play for a character before the subsequent character's LiveFont movie plays.

`Direction` (enumerated): 0: From Left; 1: From Right; 2: Ping-Pong.

`Speed` (float): the speed at which to play each LiveFont movie in this sequence, where 1.0 is the default speed. No keyframing supported.

`Play` (enumerated): 0: Forward; 1: Backward; 2: Ping-Pong.

`Loop` (unsigned integer): the number of times to loop through each LiveFont movie. No keyframing supported.

`To End` (Boolean).

`Hold First` (positive float): the number of seconds to delay the beginning of each LiveFont movie. No keyframing supported.

`Hold Last` (positive float): the number of seconds to hold the last frame of this font's LiveFont movie. No keyframing supported.

Type On

Description Encodes the Type On characteristics for a text object.

Channels `Start` (percent): denotes the first letter to draw, as a percentage of the number of letters in the text object.

`End` (percent): denotes the last letter to draw, as a percentage of the number of letters in the text object.

`Fade In` (Boolean): 1 if letters fade in as they are brought on-screen; 0 if not.

Path Options

Description Encodes path options for a text object.

Channels `Path`–>: a container for the curve describing an open spline for a text object.

`Path`–>: a container for the curve describing a closed spline for a text object.

`Path Shape` (enumerated): 0: open spline; 1: closed spline; 2: circle; 3: rectangle; 4: wave; 5: geometry.

`Path Type` (enumerated): affects open and closed spline shapes only. 0: bezier; 1: B-spline.

`Radius->`: affects circle path shape only.

`Size->`: affects rectangle path shape only.

`Start Point->`: affects wave path shape only.

`End Point->`: affects wave path shape only.

`Amplitude` (float): the amplitude for a wave path. Affects wave path shape only.

`Frequency` (float): the frequency of a wave path. Affects wave path shape only.

`Phase` (angle): the phase of the wave path, in radians. Affects wave path shape only.

`Damping` (float): the damping of the wave path. Affects wave path shape only.

`Shape Source` (id): the id of the geometry source. Affects geometry path shape only.

`Attach To Shape` (Boolean): 1 if enabled; 0 if disabled. Affects wave path shape only.

`Path Offset` (float): amount to offset from beginning of path, in 1/100s of a percent.

`Wrap Around` (Boolean): 1 if enabled; 0 if disabled.

`Inside Path` (Boolean): 1 if enabled; 0 if disabled.

`Align to Path` (Boolean): 1 if enabled; 0 if disabled.

`Align to Text` (Boolean): 1 if enabled; 0 if disabled.

Path

Description

Encodes the keypoints in a path for a text object. The subelements in this channel folder are X, Y, and Z position parameters, represented as curves. Each keypoint in the curve element representing the X, Y, and Z parameters represents the position of a control point in the text path.

Subelement

`foldFlags`

Channels

`X` (float): the X value of the position of each key point in the text path, as a parametric curve.

`Y` (float): the Y value of the position of each key point in the text path, as a parametric curve.

`Z` (float): the Z value of the position of each key point in the text path, as a parametric curve.

Face

Description

Encodes the Face parameter for text styles.

Channels

`Fill With` (enumerated): 0: color; 1: gradient; 2: texture.

`Color->`

`Gradient->`

`Texture->`

`Opacity` (percent): the opacity of the glyph face for this text style.

`Blur` (float): the radius of the blur factor, in pixels.

Outline

Description

Encodes the Outline parameter for text styles.

Channels

`Fill With` (enumerated): 0: color; 1: gradient; 2: texture.

`Color->`

`Gradient->`

`Texture->`

`Opacity` (percent): the opacity of the outline for this text style.

`Blur` (float): the radius of the blur factor, in pixels.

`Width` (float): the width of the outline.

`Layer Order` (enumerated): 0: under face; 1: over face.

Glow

Description: Encodes the Glow values for text styles.

Channels: `Fill With` (enumerated): 0: color; 1: gradient; 2: texture.

`Color->`

`Gradient->`

`Texture->`

`Opacity` (percent): the opacity of the glow for this text style.

`Blur` (float): the radius of the blur factor, in pixels.

`Radius` (float): the radius of the glow.

`Scale->`: the scaling factor of the glow for this text style.

`Offset->`: the offset of the glow for this text style.

`Layer Order` (enumerated): 0: under face; 1: over face.

Shape, Emitter, Replicator Channels

The channel folders and elements in this section include:

- "Shape Animation"
- "Start Point" (page 48)
- "End Point" (page 48)
- "Angle" (page 48)
- "Angle Randomness" (page 48)
- "Opacity Over Life" (page 49)
- "Brush Profile" (page 49)
- "Scale" (page 49)
- "Scale End" (page 49)
- "Shape Parameters" (page 49)
- "Size" (page 50)
- "Outline" (page 51)
- "Alpha Type" (page 51)

- "Alpha Color" (page 51)

Shape Animation

Description	Encodes the X and Y positions of the keypoints in a Motion shape.
Subelements	`curve_X`: specifies the X positions of the keypoints.
	`curve_Y`: specifies the Y positions of the keypoints.

Start Point

Description	Encodes the starting point for line or wave replicators, or text paths.
Channels	`X` (float): the X coordinate of the starting point.
	`Y` (float): the Y coordinate of the starting point.
	`Z` (float): the Z coordinate of the starting point.

End Point

Description	Encodes the ending point for line or wave replicators, or text paths.
Channels	`X` (float): the X coordinate of the ending point.
	`Y` (float): the Y coordinate of the ending point.
	`Z` (float): the Z coordinate of the ending point.

Angle

Description	Encodes angle properties for particle cells or replicator cells, in radians.
Subelement	`foldFlags`
Channels	`X` (float): the angle about the X axis to rotate each cell.
	`Y` (float): the angle about the Y axis to rotate each cell.
	`Z` (float); the angle about the Z axis to rotate each cell. Present for replicator cells in 3D mode only.
	`Angle` (angle): the angle about the Z axis to rotate each cell; present in 2D mode only, always present for particle cells.
	`Animate` (enumerated): 0: use rotation; 1: use orientation.

Angle Randomness

Description	Encodes the angle randomness properties for particle cells or replicator cells.
Subelement	`foldFlags`
Channels	`X` (float): the variance to apply to the X axis rotation of each cell.
	`Y` (float): the variance to apply to the Y axis rotation of each cell.
	`Z` (float): the variance to apply to the Z axis rotation of each cell. Present for replicator cells in 3D mode only.
	`Angle Randomness` (angle): the variance to apply to the Z axis rotation of each cell; present in 2D mode only, always present for particle cells.
	`Animate` (enumerated): 0: use rotation; 1: use orientation.

Opacity Over Life

Description	Encodes Opacity Over Life values for various Motion objects, such as particle cells and replicator cells.
Subelement	`foldFlags`
Channel	`RGB->`
	`Alpha->`

Brush Profile

Description	Encodes Brush Profile values for various Motion objects, such as replicator cells for Motion paint strokes and shapes.
Subelement	`foldFlags`
Channel	`RGB->`
	`Alpha->`

Scale

Description	Encodes scaling factors for various parameters.
Subelement	`foldFlags`
Channel	`X` (float): the horizontal scale factor.
	`Y` (float): the vertical scale factor.

Scale End

Description	Encodes scaling factor for replicator cells sitting at the edges of a replicator.
Subelement	`foldFlags`
Channels	`X` (float): the horizontal scale factor.
	`Y` (float): the vertical scale factor.

Shape Parameters

Description	Encodes the characteristics of a replicator cell. Many parameters are used by emitters, replicators, shapes and paint strokes. Other parameters are used only by some of these objects. See the Motion Inspector tab for details.
Channels	`Shape` (enumerated): the shape for the replicator. 0: line; 1: rectangle; 2: circle; 3: burst; 4: spiral; 5: wave; 6: geometry; 7: image; 8: box (3D only); 9: sphere (3D only).
	`Object Animation`: a parameter used internally by Motion that should not be modified.
	`Emit At Points` (Boolean): 0 if disabled, 1 for enabled. Used by emitters.
	`Arrangement` (enumerated): the arrangement for rectangle replicators. 0: outline; 1: tile fill; 2: random fill.
	`Points` (unsigned integer): the number of cells for the replicator.
	`Columns` (unsigned integer): the number of columns for rectangle replicators.
	`Rows` (unsigned integer); the number of rows for rectangle replicators.
	`Ranks` (unsigned integer); the number of ranks for 3D rectangle replicators.
	`Spacing` (float): the spacing between individual cells.

`Spacing Over Stroke`: this parameter is used for backwards compatibility and should not be modified.

`Width` (float): the width of each individual cell.

`Width Over Stroke`: this parameter is used for backwards compatibility and should not be modified.

`Tile Offset` (float): the offset at which to begin placement of rows of cells on this tile-arranged replicator.

`Origin` (enumerated): the build origin for rectangle or image replicators in fill arrangements. 0: upper left; 1: upper right; 2: lower left; 3: lower right; 4: center; 5: left; 6: right; 7: top; 8: bottom.

`Build Style` (enumerated): the rectangle build style in tile arrangement and corner-based origin. 0: across; 1: by row; 2: by column; 3: by rank (valid only for replicators).

`Hidden`: a parameter used internally by Motion that should not be modified.

`Origin` (enumerated): the build origin for line and wave replicators. Not present for emitters. 0: start point; 1: end point; 2: center.

`Build Style` (enumerated): the build style for geometric replicators or rectangle replicators in an outline arrangement; not present for emitters. 0: counter clockwise; 1: clockwise.

`Origin` (enumerated): the build origin for circle, burst, and spiral replicators. 0: center; 1: edge.

`Shuffle Order` (Boolean): 1 if enabled; 0 if disabled; used by replicators.

`Offset` (float): the percentage of the replicator shape at which to begin placing cells.

`Radius` (float): the radius of this spiral or circle, outline-arranged replicator.

`Start Point->`

`End Point->`

Size: this parameter is used for backward compatibility purposes and should not be modified.

`Size->`

`Number of Arms` (unsigned integer): the number of arms for this spiral replicator.

`Points Per Arm` (unsigned integer): the number of cells per arm for this spiral replicator.

`Twists` (float): the number of twists for this spiral replicator.

`Amplitude` (float): the amplitude for this wave replicator.

`Frequency` (positive float): the frequency for this wave replicator.

`Phase` (angle): the phase angle for this wave replicator, in radians.

`Damping` (float): the damping factor for this wave replicator.

`Emit At Alpha` (Boolean): 1 if enabled; 0 if disabled; used for emitters.

`Emission Alpha Cutoff` (percent): the alpha cutoff point; used for emitters.

`Shape Source` (id): the id of the geometry to use for this geometry replicator.

`Image Source`(id): the id of the image to use for the image color mode.

Size

Description	Encodes the size of the rectangle or box defining a replicator's shape.
Channels	`Width` (float): the width of the rectangle or box.

`Height` (float): the height of the rectangle or box.

`Depth` (float): the depth of the box; used only for 3D replicators.

Outline

Description: Encodes the outline for Motion shapes.

Channels: `Brush Type` (enumerated): 0: solid; 1: airbrush; 2: image.

`Brush Color->`: the color of the outline.

`Brush Opacity` (percent): the opacity for the outline.

`Width` (float): the width of the outline.

`Preserve Width` (Boolean): 0 if disabled; 1 if enabled.

`Joint` (enumerated): 0: square; 1: round; 2: bevel.

`Start Cap` (enumerated): 0: square; 1: round; 2: bevel; 3: arrow.

`End Cap` (enumerated): 0: square; 1: round; 2: bevel; 3: arrow.

`Arrow Length` (float): the length of the arrow end cap.

`Arrow Width` (float): the width of the arrow end cap.

`Order` (enumerated): 0: under fill; 1: over fill.

`First Point Offset` (percent): the point at which the outline should begin along the shape.

`Last Point Offset` (percent): the point at which the outline should end along the shape.

`Hidden Stroke Completion`: a parameter used internally by Motion that should not be modified.

`Use Natural Stroke`: a parameter used internally by Motion that should not be modified.

`Offset`: a parameter used internally by Motion that should not be modified.

`Pressure Over Stroke`: a parameter used internally by Motion that should not be modified.

`Tilt Over Stroke`: a parameter used internally by Motion that should not be modified.

`Pen Speed Over Stroke`: a parameter used internally by Motion that should not be modified.

Alpha Type

Description: Encodes the alpha channel to use for clips.

Subelement: Note: this element does not contain a `foldFlags subelement..`

Channels: `Alpha Type` (enumerated): 0: none/ignore; 1: straight; 2: premultiplied black; 3: premultiplied white; 4: premultiplied color.

`Alpha Color->`

Alpha Color

Description: Encodes the color to use for the alpha channel of a clip.

Channels: `Red` (percent): the amount of red to contribute to the alpha color. Keyframes unsupported.

`Green` (percent): the amount of green to contribute to the alpha color. Keyframes unsupported.

`Blue` (percent): the amount of blue to contribute to the alpha color. Keyframes unsupported.

Footage and Audio Channels

The channel folders and elements in this section include:

- "OpenEXR" (page 52)
- "Crop" (page 52)
- "Master Controls" (page 52)

OpenEXR

Description — Encodes properties related to OpenEXR clips.

Channels — `Exposure` (float): the exposure value for this clip.

`Defog` (float): the defog value for this clip.

`KneeLow` (float): the KneeLow value for this clip.

`KneeHigh` (float): the KneeHigh value for this clip.

Crop

Description — Encodes cropping information for an image or movie in the Media list.

Channels — `Left` (float): the number of pixels to crop from the left.

`Right` (float): the number of pixels to crop from the right.

`Bottom` (float): the number of pixels to crop from the bottom.

`Top` (float): the number of pixels to crop from the top.

Master Controls

Description — Encodes information for emitters containing more than one particle cell.

Channels — `Birth Rate` (float): the birth rate for this emitter.

`Hidden`: a parameter used internally by Motion that should not be modified.

`Initial Number` (float): the initial number of cells.

`Life` (float): the lifespan of emitter cells.

`Scale–>`

`Speed` (float): the speed of emitter cells.

`Spin` (float): the spin rate of emitter cells.

Other Channels

The channel folders and elements in this section include:

- "Spot Options" (page 53)
- "Clone Layer" (page 53)
- "Texture" (page 53)
- "Offset" (page 53)
- "Mask" (page 53)

Spot Options

Description	Encodes options for a light scene object.
Channels	`Cone Angle` (float): the angle of the cone.
	`Soft Edge` (float): the angle of the soft edge.

Clone Layer

Description	Encodes a clone layer's source information.
Subelement	`Source` (id): the id of the layer that this clone Layer uses as its source.

Texture

Description	Encodes information pertaining to texture for various Motion objects.
Channels	`Image` (id): the id of the media to use as the source for this texture.
	`Frame` (frame): the frame of this footage to use as the image for this texture; 0 for images.
	`Hold Frame` (Boolean): 0 if disabled; 1 if enabled.
	`Offset->`
	`Wrap Mode` (enumerated) 0: none; 1: repeat; 2: mirror.

Offset

Description	Encodes the offset for various parameters.
Channels	`X` (float): the offset along the X direction.
	`Y` (float): the offset along the Y direction.
	`Z` (float): the offset along the Z direction. May not appear for parameters with two-dimensional offsets.

Mask

Description	Encodes the properties of an image mask.
Channels	`Mask Source` (id): the id corresponding to the source for this mask's shape.
	`Frame` (frame): the frame at which to derive the source image for this mask; 0 for image files.
	`Hold Frame` (Boolean): 0 if disabled; 1 if enabled.

`Offset->`

`Wrap Mode` (enumerated): 0: none; 1: repeat; 2: mirror.

Customizing a Motion XML Project File

This chapter provides three examples of editing a Motion XML project file. You can use these examples as a starting point to develop your own routines or tools to automate Motion offline editing or management tasks.

The examples in this chapter are:

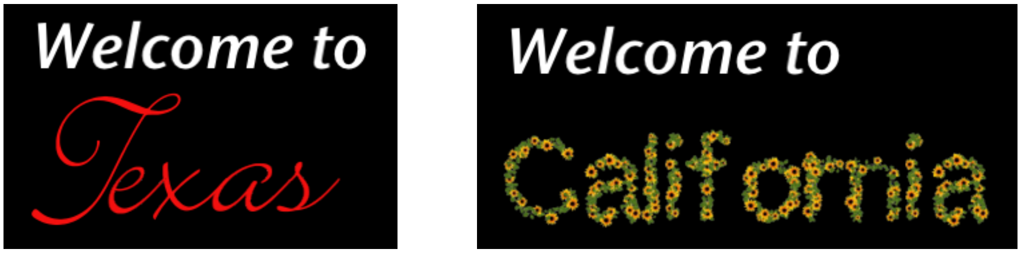
- "Modifying Text"
- "Importing Camera Data" (page 58)
- "Swap in New Media" (page 62)

Modifying Text

Suppose a Motion project contains a Motion text object whose text is set to "Welcome to Texas". This text is written with two different fonts: Stone Sans Sem OS ITCTT (a system font) for "Welcome To," and Script (a LiveFont) for "Texas." Also, there is carriage return after the word "to."

In this example, the goal is to change the word "Texas" to "California" and the font for "California" to Sunflower.

Figure 7-1 Modifying Text



Before **After**

A Motion text object primarily relies on the following elements to specify its characteristics:

- `text`
- `object`
- `styleRun`
- `style`

In general, Motion text objects are subelements of a scenenode. If a Motion project contains multiple Motion text objects, each scenenode element with a Motion text object contains its own style, styleRun, text and object elements. To modify a text object, you need to first accurately identify its scenenode and then change the subelements.

Modifying the text Element

First, change the text element. Depending on your XML editor, it appears as:

```
<text>Welcome to
Texas</text>
```

Change this to:

```
<text>Welcome to
California</text>
```

Modifying the object Elements

Motion contains one object element for every glyph in the associated text object:

Listing 7-1 Initial Glyphs in the Text Object

```
<object value="87"><! letter "W" -->
  <parameter name="Kerning" id="1" flags="16" value="0" />
</object>
<object value="101"><! letter "e" -->
  <parameter name="Kerning" id="2" flags="16" value="0" />
</object>
<object value="108"><! letter "l" -->
  <parameter name="Kerning" id="3" flags="16" value="0" />
</object>
. . .
<object value="84"><! letter "T" -->
  <parameter name="Kerning" id="12" flags="16" value="0" />
</object>
<object value="101"><! letter "e" -->
  <parameter name="Kerning" id="13" flags="16" value="0" />
</object>
<object value="120"><! letter "x" -->
  <parameter name="Kerning" id="14" flags="16" value="0" />
</object>
. . .
```

Note the following:

- The value= attribute of each object element represents the decimal ASCII value of the character for each glyph. The ASCII value should match the associated letter in the text element.
- The id= of the Kerning parameter is equal to the index of each glyph, starting with 1.
- For “Welcome to Texas,” there are 16 object elements, one for every character in the text element. As well, newline, tab, space and other whitespace characters require an object element.

Modifying the glyphs for "Welcome to California," results in these object elements (partially shown):

Listing 7-2 Changed Glyphs in the Text Object

```
<object value="87"><! letter "W" -->
  <parameter name="Kerning" id="1" flags="16" value="0" />
</object>
<object value="101"><! letter "e" -->
  <parameter name="Kerning" id="2" flags="16" value="0" />
</object>
<object value="108"><! letter "l" -->
  <parameter name="Kerning" id="3" flags="16" value="0" />
</object>
. . .
<object value="67"><! letter "C" -->
  <parameter name="Kerning" id="12" flags="16" value="0" />
</object>
<object value="97"><! letter "a" -->
  <parameter name="Kerning" id="13" flags="16" value="0" />
</object>
<object value="108"><! letter "l" -->
  <parameter name="Kerning" id="14" flags="16" value="0" />
</object>
. . .
```

Modifying the styleRun Element

Motion uses styleRun elements to specify the range that text style elements refer to. Without properly specified styleRun elements, style elements, which encode font information, do not have any effect.

In the original project file, the styleRun elements appear as follows:

```
<styleRun style="10032" offset="0" length="11" />
<styleRun style="10107" offset="11" length="5" />
```

The style= attribute refers to the id= attribute of a style element. The length= attribute indictes the span of the styleRun.

The first styleRun element in this listing refers the style element that whose font is Stone Sans OS ITCTT. The first 11 characters snap the text "Welcome to " (including the newline character).

The second styleRun element begins on the 12th character and spans 5 characters: "Texas".

In the modified project file, the length of the string changes (since "Texas" changes to "California"). The styleRun elements then change as follows:

```
<styleRun style="10032" offset="0" length="11" />
<styleRun style="10107" offset="11" length="10" />
```

Note the following:

- The offset= attribute of the first styleRun element, which refers to the first character the indicated style affects, is 0.
- When the lengths of the text strings change, it is important to modify the appropriate styleRun element to ensure that the appropriate text style is applied to the new characters.

- Every character in a text object must be associated with a styleRun. It is important to check for consistency across all styleRun elements and to ensure there are no overlaps and no gaps in the range of characters affected.

Modifying the style Elements

You can change the font applied to "Texas" from Script to Sunflower by modifying the appropriate style element. In the original project file, there are two style elements for the text object.

```
<style name="Style" id="10032" factoryID="1">
  <font type="0">StoneSansOSITCTT-SemiIta</font>
...
</style>
<style name="Style1" id="10107" factoryID="1">
  <font type="1">Pro Series/Script</font>
...
</style>
```

Since each font has a unique name and a different type, you can use the font element's type= attribute to identify the font element you want to change. In this case, it is in the second style element with an id= 10107. (If the two style elements have similar characteristics, you can use the id= attribute and the associated styleRun element to identify the correct font element to change.)

The second font element, whose type attribute refers to a LiveFont and whose contents is the system name of the Script font, pertains to the style element that is associated with the styleRun element spanning the characters in "Texas." It is then changed to the following:

```
<style name="Style" id="10032" factoryID="1">
  <font type="0">StoneSansOSITCTT-SemiIta</font>
...
</style>
<style name="Style1" id="10107" factoryID="1">
  <font type="1">Pro Series/Sunflower</font>
...
</style>
```

The id= of the second style element is 10107 and corresponds to the style= attribute of the styleRun element. This element applies to the word "California," which now appears in the Sunflower font.

Note: You can change a font element in order to change the font used by a style. However, take care to represent the font name and type accurately. (See "font" (page 23)).

Importing Camera Data

You can import camera data from a third-party application into Motion by editing Position, Rotation, and other relevant parameters for a Motion camera. The goal is to generate position and rotation information that conforms to Motion's channel XML structure.

Suppose the camera data you want to import contains position and rotation information for each frame.

Motion curve and keyframe Elements

Since Motion supports manual keyframing of parameters, the `Position` parameter of the Camera is represented as a `curve` with m `keypoint` elements. M is the number of frames for which camera data is available. The interpolation for each `keypoint` in the curve would is constant.

If the camera data has 300 frames, the XML for the `Position` parameter's `X` channel would look like this:

```
<parameter name="X" id="1" flags="16">
  <curve type="1" round="0">
    <numberOfKeypoints>300</numberOfKeypoints>
    <keypoint interpolation="1" flags="32"
      <time>0</time>
      <value>20</time>
    </keypoint>
    <keypoint interpolation=”1” flags=”32”>
      <time>1</time>
      <value>20.5</time>
    </keypoint>
    <keypoint interpolation=”1” flags=”32”>
      <time>2</time>
      <value>21.5</time>
    </keypoint>
    ...
    <keypoint interpolation=”1” flags=”32”>
      <time>299</time>
      <value>90</time>
    </keypoint>
  </curve>
</parameter>
```

Note the following points:

- The `X` channel does not contain a `value=` attribute. Keyframed parameters do not have a constant value and the `value=` attribute is not relevant. The `curve` subelement specifies all the position and time information for the `X` channel. See "curve" (page 21) for more details.
- The `curve` element contains a `numberOfKeypoints` subelement specifying the number of data points being imported (300, in this case). The `curve` element then has 300 `keypoint` elements.
- Each `keypoint` element contains `time` and `value` subelements that represent the frame at which the data was recorded and the X position for that frame, respectively. The `interpolation=` attribute is set to 1, indicating constant interpolation across each keypoint.
- The `keypoint` elements do not contain the usual subelements specifying the input and output tangent times and values. Since the `interpolation=` attribute is set to a constant, tangent information is not necessary.
- The `time` value for the first `keypoint` is 0. Internally, Motion refers to the first frame of any project as frame 0.

To finish importing the camera data, you would need to encode information for the Y and Z channels of the camera's `Position` parameter, as well as the X, Y and Z channels for the `Rotation` parameter.

Replacing Media Files

The Motion XML project file encodes a variety of links between scene objects and the media files they incorporate. If you understand these links, you can develop tools to automatically replace media files in a project. You can then easily create multiple copies of a project with different footage for each copy.

Top-level Media Elements

Here is an XML example that displays top-level elements that are relevant to media and that are not described elsewhere in this document:

Listing 7-3 Top-level Media Elements

```
<ozml version="3.0">
  ...
  <scene>
    <sceneSettings>
    ...
    <frameRate>30</frameRate>
    <NTSC>1</NTSC>
    ...
   </sceneSettings>
   ...
   <timeRange offset="0" duration="4172" />
   <playRange offset="0" duration="4172" />
   ...
 </scene>
</ozml>
```

- scene: encodes scene settings, export settings, and the Motion scene graph. This element is a subelement of the ozml declaration element.
- sceneSettings: encodes Motion scene settings. This element is a subelement of scene.
- frameRate (unsigned integer): encodes the framerate of a Motion project. This element is a subelement of sceneSettings. Note that the framerate is saved as an integer rate. The true framerate for the project is determined by the NTSC element.
- NTSC (Boolean): indicates if the project is an NTSC project. This element is a subelement of sceneSettings. When the NTSC element is set to TRUE, the value in frameRate is adjusted internally to conform with NTSC standards. For example, 30 fps is adjusted to 29.97 fps.
- timeRange: the visible time range on the Motion Canvas. This element is a subelement of scene and contains the following attributes: offset= (frame): the frame at which the visible time range begins; duration= (unsigned integer): the number of frames visible on the time range.
- playRange: the playable time range on the Motion timeline. This element is a subelement of scene and contains the following attributes: offset= (frame): the frame at which playback begins; duration= (unsigned integer): the number of frames to play.

Steps for the Task

Suppose there is a Motion project with a media file named “oldMovie.mov” and that you want to replace it with a different file called “newMovie.mov.” (You will use the new footage at its defalt frame rate without retiming.)

The steps to follow are:

- Identify the `scenenode` referring to the relevant footage.
- Replace the old footage with new footage.
- Modify the accompanying audio data if necessary.
- Modify the scenenode referring to the relevant footage if necessary.

Identify the scenenode

The following XML example displays the `scenenode` with the relevant information:

Listing 7-4 Finding the scenenode

```
<scenenode name="My movie object" id="1056774" factoryID="1" version="3">
...
  <linkedobjects>1056776</linkedobjects>
  <timing in="0" out="4171" offset="0" />
  ...
  <parameter name="Properties" id="1" flags="4112">
  ...
  <parameter name=”Media” id=”300” flags=”65552” value=”1056770” />
   ...
 </parameter>
 ...
</scenenode>
...
<footage name="Media Layer" id="1056771">
  <clip name="oldMovie" id="1056770">
   <pathURL>file://localhost/Users/steve/Movies/oldMovie.mov</pathURL>
   ...
</footage>
```

Note these points:

- This `scenenode` element contains a `Properties` parameter that, in turn, contains a `Media` parameter. The `value=` attribute of this `Media` parameter (1056770) references the `id=` attribute of a `clip` element. This clip element specifies the media that you want to change.
- The `linkedObjects` element references an `audioTrack` element that contains audio information related to the `clip` element.

Swap in New Media

In general, a media file in Motion is specified by a `clip` element that is a subelement of a `footage` element. (See "clip" (page 18).)

Here is the XML specifying the current media file ("oldMovie"):

Listing 7-5 Current Media File

```
<footage name="Media Layer" id="1056771">
  <clip name="oldMovie" id="1056770">
  <pathURL>file://localhost/Users/steve/Movies/oldMovie.mov</pathURL>
  <missingWidth>1920</missingWidth>
  <missingHeight>816</missingHeight>
  <missingDuration>139.1808475</missingDuration>
  <creationDuration>4172</creationDuration>
  ...
  <timing in="0" out="4171" offset="0" />
  ...
  <parameter name=”Object” id=”2” flags=”4176”>
  ...
  <parameter name="Frame Rate" id="107" flags="64" value="23.976" />
   ...
   <parameter name="Fixed Width" id="114" flags="66" value="1920" />
   <parameter name="Fixed Height" id="115" flags="66" value="816" />
   ...
 </parameter>
</footage>
```

To swap in the new media file:

- `name=`: As appropriate, change the `name=` attribute of the `clip` element to reflect the name of the new clip.
- `pathURL`: Set the value of this element to a URL pointing to the new media file.
- `relativeURL`: If applicable, set the value of this element to a relative URL pointing to the new media file.
- `missingWidth`: Set the value of this element to the image width of the new media file.
- `missingHeight`: Set the value of this element to the image height of the new media file.
- `missingDuration`: Set the value of this element to the duration of the new media file, in seconds. If the current media file is a still image, the value should be the reciprocal of the project frame rate.
- `creationDuration`: Set the value of this element to the number of frames this media file represents in the Motion project. This value is equal to `missingDuration` multiplied by the project's frame rate, rounded up to the nearest integer.

  The project's frame rate is set in the `frameRate` subelement of the `sceneSettings` element, which itself is a subelement of the `scene` element. (See Listing 7-3 (page 60).)

  For NTSC projects, the frame rate must be adjusted to match standard NTSC framerates (for example, a 30 fps project will be 29.976 fps). The `NTSC` subelement of the `sceneSettings` element is set to 1 if the project is an NTSC project.

- `timing`: Set the `in=` and `out=` attributes to the desired beginning and end points for the clip. If the length of the new media file differs from the media file it replaces, you should modify this element to ensure that the entire media file plays back.
- `Frame Rate`: Set this parameter to match the frame rate of the media file. A value differing from the media file’s natural frame rate may introduce timing discrepancies during playback or export.
- `Fixed Width`, `Fixed Height`: Set the value of these parameters to the media file’s width and height, respectively.

The current clip (“oldMovie.mov”) is a 23.976 fps HD movie 139.18 seconds in length. The new clip (“newMovie.mov”) is a 29.976 fps HD movie that is 111.14 seconds long.. Making all the appropriate changes in the `clip` element to swap in the new clip results in the following XML:

Listing 7-6 New Media File

```
<footage name="Media Layer" id="1056771">
 <clip name="newMovie" id="1056770">
  <pathURL>file://localhost/Users/steve/Movies/newMovie.mov</pathURL>
  <missingWidth>1920</missingWidth>
  <missingHeight>816</missingHeight>
  <missingDuration>111.14</missingDuration>
  <creationDuration>3332</creationDuration>
  ...
  <timing in="0" out="3331" offset="0" />
  ...
  <parameter name="Object" id="2" flags="4176">
   ...
   <parameter name="Frame Rate" id="107" flags="64" value="29.976" />
   ...
   <parameter name="Fixed Width" id="114" flags="66" value="1920" />
   <parameter name="Fixed Height" id="115" flags="66" value="816" />
   ...
  </parameter>
</footage>
```

Modify the Audio Track

Media files that contain audio may require modifying the `audiotrack` element associated with the `clip` element.

Listing 7-7 Current Audio

```
<audio name="Audio Layer" id="1056777">
 <audioTrack name="oldMovie" id="1056776">
 ...
  <linkedObjects>1056774</linkedObjects>
  <timing in="0" out="4171" offset="0" />
  ...
  <parameter name="Properties" id="1" flags="4112">
   <parameter name="Media" id="104" flags="65552" value="1056770" />
  ...
  <parameter name="Retime Value" id="109" flags="131090">
    <curve type="1" round="0" retimingExtrapolation="1">
    <numberOfKeypoints>2</numberOfKeypoints>
    <keypoint interpolation="1" flags="32">
     <time>0</time>
     <value>1</value>
      ...
    </keypoint>
     <keypoint interpolation="1" flags="32">
     <time>4172</time>
     <value>4173</value>
      ...
     </keypoint>
    </curve>
   </parameter>
   <parameter name="Retime Value Cache" id="109" flags="131090">
    <curve type="1" round="0" retimingExtrapolation="1">
     <numberOfKeypoints>2</numberOfKeypoints>
    <keypoint interpolation="1" flags="32">
      <time>0</time>
      <value>1</value>
      ...
     </keypoint>
     <keypoint interpolation="1" flags="32">
      <time>4172</time>
      <value>4173</value>
      ...
     </keypoint>
    </curve>
   </parameter>
   ...
  </parameter>
  ...
 </audioTrack>
</audio>
```

Note these points:

- As appropriate, you can modify the `name=` attribute of the `audioTrack` element.
- Because you are replacing the contents of a `clip` element, the `value=` attribute of the `Media` parameter should equal to the `id=` attribute of the `clip` element, and not require modification.
- Additionally, the `audiotrack` element contains a `linkedObjects` element that specifies the associated `scenenode`, and does not require modification.
- If the duration of the new movie file differs from the old movie file, you will need to change the timing information in the `audioTrack` element.
- The `timing` element contains the timing information playing audio in constant time. In this example, the audio track is the same length as the new media file.
- The `Retime Value` and `Retime Value Cache` parameters are not used when an audio track is played in constant time. But they may be modified to maintain consistency when variable timing is selected in the Inspector.
- Both the `Retime Value` and `Retime Value Cache` parameters are represented as curves with two keypoint subelements corresponding to the first and last frames in the file, in that order.
- The `time` and `value` subelements of the second `keypoint` element need to match the last frame in the new media file. Note that time is 0-based, while value is 1-based. That is, if the new media file is 3000 frames long, the value specified by `time` will be 2999 and `value` will be 3000.

- Because the audio track is not retimed, normal, inputTangentTime, inputTangentValue, outputTangentTime, and outputTangentValue are not used in this example and can be left as is.

The revised `audioTrack` element appears as follows:

Listing 7-8 New Audio

```
<audio name="Audio Layer" id="1056777">
 <audioTrack name="newMovie" id="1056776">
  ...
  <linkedObjects>1056774</linkedObjects>
  <timing in="0" out="3331" offset="0" />
  ...
  <parameter name="Properties" id="1" flags="4112">
   <parameter name="Media" id="104" flags="65552" value="1056770" />
   ...
   <parameter name="Retime Value" id="109" flags="131090">
    <curve type="1" round="0" retimingExtrapolation="1"
     <numberOfKeypoints>2</numberOfKeypoints
     <keypoint interpolation="1" flags="32">
     <time>0</time>
     <value>1</value>
      ...
    </keypoint>
<keypoint interpolation="1" flags="32">
     <time>3331</time>
      <value>3332</value
      ...
    </keypoint>
   </curve>
  </parameter>
   <parameter name="Retime Value Cache" id="109" flags="131090">
    <curve type="1" round="0" retimingExtrapolation="1">
    <numberOfKeypoints>2</numberOfKeypoints>
    <keypoint interpolation="1" flags="32">
    <time>0</time>
     <value>1</value>
     ...
    </keypoint>
    <keypoint interpolation="1" flags="32">
      <time>3331</time>
      <value>3332</value>
      ...
     </keypoint>
    </curve>
   </parameter>
   ...
  </parameter>
  ...
 </audioTrack>
</audio>
```

Modify the scenenode Object

Because the `id=` attribute of the clip referencing the new media file is unchanged, there is no need to modify the `Media` parameter or the `linkedObjects` subelement of the `scenenode` element. However, if the media length differs, you may need to change the `timing` subelement in the same way you may need to change the `timing` element of the `clip` to ensure that the entire movie file is played back.

The `Retime Value` and `Retime Value Cache` parameters are not used when a movie clip is played in constant time. But you may need modify them to maintain consistency if variable timing is selected in the Inspector.

As well, you may need to modify `layer` elements above this `scenenode` in the XML hierarchy, including the `timeRange` and `playRange` subelements of the `scene` element. (See Listing 7-3 (page 60).)

Add Media Files

Rather than replacing media files in a Motion project, you may wish to add a media file to a project. Here are some relevant points :

- To add a new `clip` element, you must set the value of its `id=` attribute to a unique unsigned integer not used by any other scene object, media file, or audio track in the project.
- To add a new `audiotrack` element, you must set the `id=` attribute to unique unsigned integer not used by any other scene object, media file, or audio track in the project. Additionally, you must set the `Media` parameter to the `id=` of the new `clip` element, and the value of the `linkedObjects` element must be an unsigned integer matching the `id=` of the relevant `scenenode` element.
- To modify the relevant `scenenode` element, you must set the `Media` parameter to equal the `id=` attribute of the new `clip` element. As well, you must set the value of the `linkedObjects` subelement to the `id=` attribute of the new `audioTrack` element.

Document Revision History

This table describes the changes to *Motion XML File Format*.

Date	Notes
2008-04-14	New document describing aspects of the Motion XML file format.