

Vantage 6.2

Post Producer Developer's Guide

Note: This guide is the 6.2 version. The 6.3 version will be available for download from Telestream.net in September 2014.

Copyrights and Trademark Notices

Copyright © 2014 Telestream, Inc. All rights reserved worldwide. No part of this publication may be reproduced, transmitted, transcribed, altered, or translated into any languages without the written permission of Telestream. Information and specifications in this document are subject to change without notice and do not represent a commitment on the part of Telestream.

Telestream. Telestream, Episode, Flip4Mac, FlipFactory, Flip Player, Lightspeed, ScreenFlow, Vantage, Wirecast, GraphicsFactory, MetaFlip, MotionResolve, and Split-and-Stitch are registered trademarks and Pipeline, Launch, and Videocue are trademarks of Telestream, Inc. All other trademarks are the property of their respective owners.

Adobe. Adobe® HTTP Dynamic Streaming Copyright © 2014 of Adobe Systems All right reserved.

Apple. QuickTime, MacOS X, and Safari are trademarks of Apple, Inc. Bonjour, the Bonjour logo, and the Bonjour symbol are trademarks of Apple, Inc.

Avid. Portions of this product Copyright 2012 Avid Technology, Inc.

Dolby. Dolby and the double-D symbol are registered trademarks of Dolby Laboratories.

Fraunhofer IIS and Thomson Multimedia. MPEG Layer-3 audio coding technology licensed from Fraunhofer IIS and Thomson Multimedia.

Google. VP6 and VP8 Copyright Google Inc. 2014 All rights Reserved.

MainConcept. MainConcept is a registered trademark of MainConcept LLC and MainConcept AG. Copyright 2004 MainConcept Multimedia Technologies.

Manzanita. Manzanita is a registered trademark of Manzanita Systems, Inc.

MCW. HEVC Decoding software licensed from MCW.

MediaInfo. Copyright © 2002-2013 MediaArea.net SARL. All rights reserved.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Microsoft. Microsoft, Windows NT|2000|XP|XP Professional|Server 2003|Server 2008|Server 2012, Windows 7, Windows 8, Media Player, Media Encoder, .Net, Internet Explorer, SQL Server 2005|2008|Server 2012, and Windows Media Technologies are trademarks of Microsoft Corporation.

SharpSSH2. SharpSSH2 Copyright (c) 2008, Ryan Faircloth. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Diversified Sales and Service, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Telerik. RadControls for ASP.NET AJAX Copyright Telerik All rights reserved.

VoiceAge. This product is manufactured by Telestream under license from VoiceAge Corporation.

x264 LLC. The product is manufactured by Telestream under license from x264 LLC.

Xceed. The Software is Copyright ©1994-2012 Xceed Software Inc., all rights reserved.

ZLIB. Copyright (C) 1995-2013 Jean-loup Gailly and Mark Adler.



Authorized Developer
And DPA®



Other brands, product names, and company names are trademarks of their respective holders, and are used for identification purpose only.

MPEG Disclaimers

MPEGLA MPEG2 Patent

ANY USE OF THIS PRODUCT IN ANY MANNER OTHER THAN PERSONAL USE THAT COMPLIES WITH THE MPEG-2 STANDARD FOR ENCODING VIDEO INFORMATION FOR PACKAGED MEDIA IS EXPRESSLY PROHIBITED WITHOUT A LICENSE UNDER APPLICABLE PATENTS IN THE MPEG-2 PATENT PORTFOLIO, WHICH LICENSE IS AVAILABLE FROM MPEG LA, LLC, 4600 S. Ulster Street, Suite 400, Denver, Colorado 80237 U.S.A.

MPEGLA MPEG4 VISUAL

THIS PRODUCT IS LICENSED UNDER THE MPEG-4 VISUAL PATENT PORTFOLIO LICENSE FOR THE PERSONAL AND NON-COMMERCIAL USE OF A CONSUMER FOR (i) ENCODING VIDEO IN COMPLIANCE WITH THE MPEG-4 VISUAL STANDARD ("MPEG-4 VIDEO") AND/OR (ii) DECODING MPEG-4 VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL AND NON-COMMERCIAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION INCLUDING THAT RELATING TO PROMOTIONAL, INTERNAL AND COMMERCIAL USES AND LICENSING MAY BE OBTAINED FROM MPEG LA, LLC. SEE [HTTP://WWW.MPEGLA.COM](http://www.mpegla.com).

MPEGLA AVC

THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE [HTTP://WWW.MPEGLA.COM](http://www.mpegla.com).

MPEG4 SYSTEMS

THIS PRODUCT IS LICENSED UNDER THE MPEG-4 SYSTEMS PATENT PORTFOLIO LICENSE FOR ENCODING IN COMPLIANCE WITH THE MPEG-4 SYSTEMS STANDARD, EXCEPT THAT AN ADDITIONAL LICENSE AND PAYMENT OF ROYALTIES ARE NECESSARY FOR ENCODING IN CONNECTION WITH (i) DATA STORED OR REPLICATED IN PHYSICAL MEDIA WHICH IS PAID FOR ON A TITLE BY TITLE BASIS AND/OR (ii) DATA WHICH IS PAID FOR ON A TITLE BY TITLE BASIS AND IS TRANSMITTED TO AN END USER FOR PERMANENT STORAGE AND/OR USE. SUCH ADDITIONAL LICENSE MAY BE OBTAINED FROM MPEG LA, LLC. SEE [HTTP://WWW.MPEGLA.COM](http://www.mpegla.com) FOR ADDITIONAL DETAILS.

Limited Warranty and Disclaimers

Telestream, Inc. (the Company) warrants to the original registered end user that the product will perform as stated below for a period of one (1) year from the date of shipment from factory:

Hardware and Media—The Product hardware components, if any, including equipment supplied but not manufactured by the Company but NOT including any third party equipment that has been substituted by the Distributor for such equipment (the “Hardware”), will be free from defects in materials and workmanship under normal operating conditions and use.

Software—If software is supplied as part of the product, the Company does not warrant that the software will operate uninterrupted or error-free, will meet your requirements, or that software errors will be corrected.

Warranty Remedies

Your sole remedies under this limited warranty are as follows:

Hardware and Software Media—The Company will either repair or replace (at its option) any defective Hardware component or part, or Software Media, with new or like new Hardware components or Software Media. Components may not be necessarily the same, but will be of equivalent operation and quality.

Software Updates

Except as may be provided in a separate agreement between Telestream and You, if any, Telestream is under no obligation to maintain or support the Software and Telestream has no obligation to furnish you with any further assistance, technical support, documentation, software, update, upgrades, or information of any nature or kind.

Restrictions and Conditions of Limited Warranty

This Limited Warranty will be void and of no force and effect if (i) Product Hardware or Software Media, or any part thereof, is damaged due to abuse, misuse, alteration, neglect, or shipping, or as a result of service or modification by a party other than the Company, or (ii) Software is modified without the written consent of the Company.

Limitations of Warranties

THE EXPRESS WARRANTIES SET FORTH IN THIS AGREEMENT ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. No oral or written information or advice given by the Company, its distributors, dealers or agents, shall increase the scope of this Limited Warranty or create any new warranties.

Geographical Limitation of Warranty—This limited warranty is valid only within the country in which the Product is purchased/licensed.

Limitations on Remedies—YOUR EXCLUSIVE REMEDIES, AND THE ENTIRE LIABILITY OF TELESTREAM, INC. WITH RESPECT TO THE PRODUCT, SHALL BE AS STATED IN THIS LIMITED WARRANTY. Your sole and exclusive remedy for any and all breaches of any Limited Warranty by the Company shall be the recovery of reasonable damages which, in the aggregate, shall not exceed the total amount of the combined license fee and purchase price paid by you for the Product.

Damages

TELESTREAM, INC. SHALL NOT BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF YOUR USE OR INABILITY TO USE THE PRODUCT, OR THE BREACH OF ANY EXPRESS OR IMPLIED WARRANTY, EVEN IF THE COMPANY HAS BEEN ADVISED OF THE POSSIBILITY OF THOSE DAMAGES, OR ANY REMEDY PROVIDED FAILS OF ITS ESSENTIAL PURPOSE.

Further information regarding this limited warranty may be obtained by writing:

Telestream, Inc.

848 Gold Flat Road

Nevada City, CA 95959 USA

You can call Telestream, Inc. via telephone at (530) 470-1300.

Part number: 121275

Contents

Preface 15

Post Producer Overview 17

- Introduction 18
- How Post Producer Works 20
 - Creating a Composition 20
 - Updating the Composition 21
 - Generating Media from a Composition in Vantage 21

Using Post Producer in Vantage 23

- Post Producer Actions 24
 - Conform Action 24
 - Chronicle action 25
 - Compose action 25
- Typical Post Producer Conforming Workflow 27
- Submitting CML Processing Jobs 28

Composition Overview 29

- Composition Hierarchy 30
- Sequences 31
- Segments 32
- Material 33
 - Visual (and Auditory) Effects 33

Creating and Working with Compositions 35

- Introduction 36
- Approaches to Composition Development 37
 - Skills and Tools Required for Programming Approach 37
 - Skills and Tools Required for Manual Approach 37
- Developer Resources 38

Media Characteristics Affect Composition Design	39
Composition File Naming Conventions	40
Specifying Unit Designators in Numeric Attributes	41
The Basic Format of a Composition	43
Basic Composition with Source and Sequence Elements	43
Validating a Composition	44
Resolving Errors in CML Processing Jobs	45
Using Nickname, Expressions, and Variables in Compositions	46
Using Nicknames in Compositions	46
Nickname Examples	47
Expressions	47
Expression Examples	47
Variables	48
Variable Examples	48
Numeric Variables	48
Channel Number Variables	49
String Variables	49
Commenting CML Files	50

Working with Material 51

Introduction to Material	52
Material Attributes	52
Sample Story Board	52
Sample CML	53
Adding Material to a Composition	54
Specifying Files for Video and Audio Material	54
Video Example	55
Audio Example	56
Specifying Files for Image Material	57
Specifying Files for Advisory Material	58
Extracting Clips from Master Media	59
Overview	59
Heads, Tails, and Bodies	59
Example	61
Specifying Mark Times	62
Using the Mode Attribute	62
Using the Time Attribute	63
Time and Timecode Examples	63
Cropping and Masking	66
Calculating the New Frame Size	66
Cropping Details	66
Masking Details	68
Example	69
Applying Effects to Material	70
Controlling Material on the Timeline	71
Specifying the Duration of Material on the Timeline	71
Offsetting Material on the Timeline (offset)	72

Aligning Material on the Timeline (align)	72
Tail Alignment Example	72
Both Alignment Example	73
Adjusting Material on the Timeline	73
Example	73
Modifying the Display of Material on the Timeline (fill)	74

Composition Cookbook 75

Adding Titles	76
Adding a Basic, Static Title	76
Sliding a Title On or Off Screen	76
Image Scaling and Placement	78
Mixing Audio	80
Creating a Mixed Stereo Pair	80
Creating Dolby E 5.1 from Compressed Dolby E	80
Routing Audio Without Decoding	82
Passing Dolby E to Output	82
Passing Pro Tools Audio to Output	82
Adding a Content Advisory	83
Example	84
Creating a Squeezeback	86
Example	86
Downmixing 5.1 Dolby E to 2 Channel Stereo	88
Fade to Black	89
Example	89

Post Producer Prototype Workflows 91

Simple Conforming Workflow	92
Compose and Conform Workflow	93
BlackArrow SCTE-35 Ad Insertion Workflow	94

Composition Markup Language Reference 95

CML Hierarchy Map	96
Advisory	97
Enabling the Advisory Filter in Workflow Designer	97
Example	98
Elements (CML Hierarchy Map)	98
Attributes	98
Area	105
Example	105
Attributes	105
Audio	107
Example	107
Elements (CML Hierarchy Map)	107
Attributes	108

Body	111
Elements (CML Hierarchy Map)	111
Canvas	112
Example	112
Elements (CML Hierarchy Map)	112
Attributes	112
Composition	116
Example	116
Elements (CML Hierarchy Map)	116
Attributes	116
Crop	117
Examples	117
Center Cutting an HD Frame for HD Frame with Pillars	117
Center Cutting an HD Frame for SD Frame	117
Summary	118
Attributes	119
DolbyE	121
DolbyE Program Channel Configurations	121
Example	122
Elements (CML Hierarchy Map)	122
Attributes	122
Edit	123
Examples	123
Attributes	124
Fade	125
Example	125
Attributes	126
File	128
Example	128
Attributes	128
Head	129
Example	129
Elements (CML Hierarchy Map)	130
Image	131
Example	131
Implementing Image Sequences	132
Example Image Sequence with Wildcard	132
Example	133
Elements (CML Hierarchy Map)	134
Attributes	134
Mask	140
Example	140
Attributes	140
Mix	143
Example	143
Attributes	143
Opacity	145
Example	145

Attributes	146
Processor	147
Example	147
Attributes	147
Rotation	148
Example	148
Attributes	149
Route	150
Example	150
Attributes	150
Scaling	152
Example	152
Attributes	153
Segment	154
Example	154
Elements (CML Hierarchy Map)	154
Attributes	155
Sequence	156
Example	156
Elements (CML Hierarchy Map)	156
Attributes	157
Shadow	158
Example	158
Attributes	158
Source	160
Example	160
Elements (CML Hierarchy Map)	160
Attributes	160
Subtitle	162
Example	162
Timecode Example	162
Elements (CML Hierarchy Map)	163
Attributes	164
Tail	165
Example	165
Elements (CML Hierarchy Map)	165
Target	167
Example	167
Attributes	167
Title	169
Example	169
Elements (CML Hierarchy Map)	169
Attributes	170
Translation	176
Example	176
Attributes	177
Video	178
Examples	178

Elements (CML Hierarchy Map)	179
Attributes	179
Volume	184
Example	184
Attributes	185

Post Producer Formats 187

Input Video Formats	188
Input Audio Formats	189
Input Image Formats	190
Output Formats	190

Preface

To obtain product information, technical support, or provide comments on this guide, contact us using our Web site, email, or phone number as listed below.

Table 1. Telestream Contact Information

Resource	Contact Information
Vantage Technical Support	<p>Web Site: http://www.telestream.net/telestream-support/Vantage/support.htm</p> <p>Support Email: support@telestream.net</p> <p>Terms and times of support services vary, per the terms of your current service contract with Telestream.</p>
Vantage Information, Assistance, FAQs, Forums, & Upgrades	<p>Web Site: http://www.telestream.net/telestream-support/Vantage/support.htm</p> <p>Support Email: support@telestream.net</p> <p>Enterprise Product Support: (877) 257-6245</p>
Telestream, Inc.	<p>Web Site: www.telestream.net</p> <p>Sales and Marketing Email: info@telestream.net</p> <p>Telestream, Inc. 848 Gold Flat Road, Suite 1 Nevada City, CA USA 95959</p>
International Distributor Support	<p>Web Site: www.telestream.net</p> <p>See the Telestream Web site for your regional authorized Telestream distributor.</p>
Telestream Technical Writers	<p>Email: techwriter@telestream.net</p> <p>If you have comments or suggestions about improving this document, or other Telestream documents—or if you've discovered an error or omission, please email us.</p>

Post Producer Overview

The purpose of this guide is to help producers, editors, operators, and others involved in promotion production learn how to use Post Producer in Vantage to automate repetitive editing and encoding of assets.

Note: This guide is written for video professionals who are familiar with Vantage. In order to use Post Producer in Vantage, you should know how to create workflows and submit jobs. If you are not familiar with Vantage, we suggest that you review the Vantage User's Guide. This guide also assumes that you also understand XML.

Learning to use Post Producer to effectively design, develop, and implement compositions to produce media involves skills including XML, Composition Markup Language, and Vantage. The language of Post Producer—Composition Markup Language—is powerful and comprehensive. Telestream recommends that you read the first four chapters in sequence before you begin working with compositions. The remaining chapters are reference material you can refer to when you need information.

In this chapter, you'll learn what Post Producer is designed to do, and how it works.

- [Introduction](#)
- [How Post Producer Works](#)
- [Creating a Composition](#)
- [Updating the Composition](#)
- [Generating Media from a Composition in Vantage](#)

Note: If you have comments or suggestions about improving this guide—or if you've discovered an error or omission, please email us at techwriter@telestream.net.

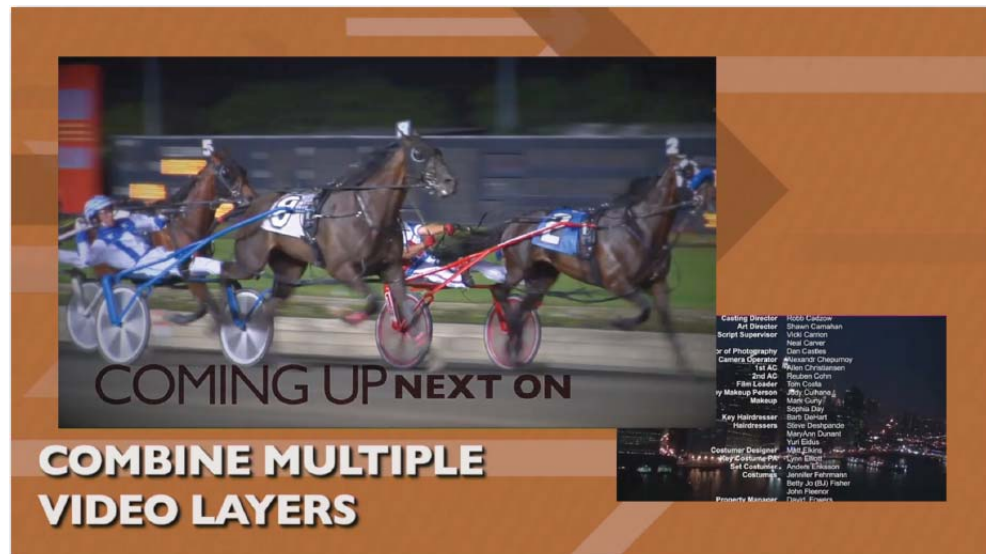
Introduction

Post Producer™ is software that automates repetitive media production processes. It accomplishes this by assembling multilayered video from content sources and masters—based on a common design template and varying sets of input media.

Post Producer reduces promo production costs by eliminating operational, or technical editing in the edit bay. Post Producer dramatically reduces labor by using automated workflows to perform repetitive tasks such as adding graphics, audio tracks, lower-thirds tag lines, etc.

Automated promo production also increases efficiency—allowing you to scale up your distribution capacity. This is accomplished by employing Vantage’s comprehensive workflow design and execution capabilities, job management, and encoding scalability, including accelerated encoding on GPU-accelerated Lightspeed servers.

Figure 1. Typical Post Producer Composition



Typical applications include:

- Cable and IPTV VOD production
- Automated news and weather report assembly
- International localization
- Promo assembly and insertion.

Post Producer is implemented as an optional, licensed feature integrated directly in Vantage—which enables you to build and execute Post Producer workflows.

Post Producer allows you to automatically construct multi-layered video, mixing different resolutions, formats, and frame rates, compositing video, graphics captions, images, audio, and subtitles. Post Producer renders output from multiple source media

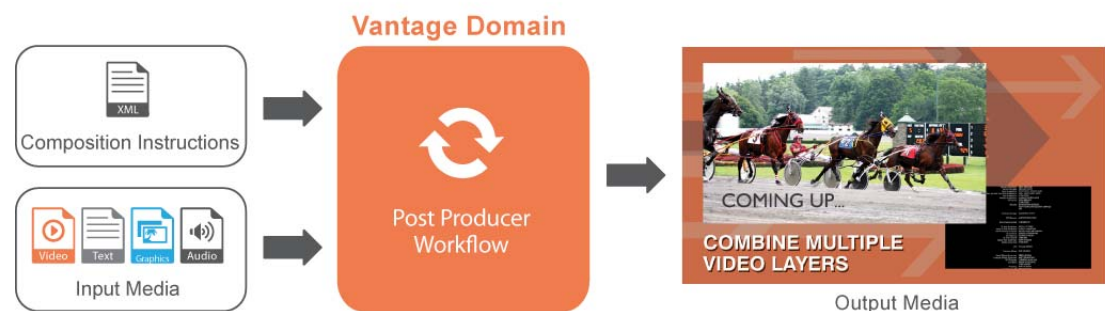
file sets into one or more output files, producing unlimited variations of a promo in any supported format/container and media format you require

How Post Producer Works

Post Producer generates a media file based on a design template—a comprehensive set of media composition instructions (commonly called a *composition*) created by you—using the media resources you specify, as input.

During the process, Post Producer applies display properties to the input media. It composes the video (including audio), with graphics and text overlays in exactly the proper arrangement (both spatially and temporally), as specified in the composition. Finally, it renders a media file from your input media that conforms to the instructions.

Figure 2. Post Producer Media Conforming and Generation Process



Creating a media file using Post Producer involves these steps:

1. Create a general purpose composition file (CML) to describe the media you want to create.
2. Update the CML file with the resources to use for the media you want to generate.
3. Generate the media by processing the CML file in a Post Producer workflow.

Creating a Composition

The composition (the design template) you create must be written in a specific schema—Composition Markup Language (*CML* for short). CML is designed to make describing media (and thus providing the composition instructions) as easy, flexible, and comprehensive as possible. Each composition is saved as an XML file.

Post Producer is intended for repetitively producing media files from different sets of input media (hourly weather updates, for example). Thus, the composition must be specific enough to comprehensively describe the output media characteristics, but general enough to generate the media from any given set of input media provided.

When designing the composition, you won't typically know in advance the file names of the input media resources. All you'll know is that you need a baseline video, and you want a logo, plus lower-thirds text, for example. So you have to leave these elements undefined—or, just identify a sample set of test input media for development.

You typically create CML files using a computer program that you have developed, or you can create them manually in an XML or text editor.

Note: Before you can create CML files (or write programs that create CML files), you must have a solid understanding of its elements and topology. For details, see [Creating and Working with Compositions](#).

You can create CML files in the following ways:

- Develop a program using the Post Producer Class Library in the Vantage SDK
- Using an XML or text editor
- In Vantage, use a Compose action workflow to convert a media descriptor file in another format to CML format.

Each of these methods are described in detail later in the guide.

Note: While creating Composition files manually is feasible, it is not usually the preferred practice. The best practice is to generate compositions programmatically.

Updating the Composition

Each time you submit a job (for testing or for production), you modify the composition file to identify the path to each input media file you require—creating a ready-to-run composition—and then you can submit it to a Vantage workflow to process the job and generate the output.

Note: Before you submit a composition file to generate a new output file, update it with the correct input files. While you can update it manually, you can also automate this process in various ways in production.

Generating Media from a Composition in Vantage

In Vantage, you first create a Post Producer workflow—one which contains a Conform action. This workflow assembles and encodes the input media files—the video, audio, text, and still images you’ve identified in the ready-to-run composition—into an output file.

To generate your media, you submit your ready-to-run composition file to the workflow. Details are described in the next section. When the workflow is done, Post Producer has automatically created your promo—based on the common design template—from the source media you identified.

Using Post Producer in Vantage

This chapter describes how Post Producer is implemented in Vantage. In this chapter you'll learn about the Post Producer actions, review a prototypical Post Producer workflow, and review how to submit a Post Producer job to generate media.

- [Post Producer Actions](#)
- [Typical Post Producer Conforming Workflow](#)
- [Submitting CML Processing Jobs](#)

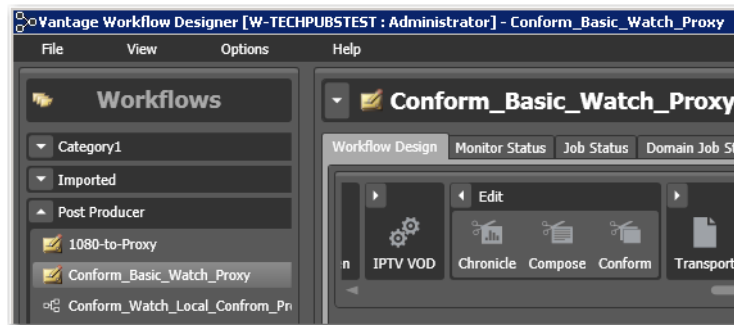
Post Producer Actions

Post Producer is implemented in Vantage via actions you can use to build Post Producer workflows. To produce promos and other media from source media and a composition, you process it in a Post Producer workflow that contains a Conform action. Two other actions—Compose and Chronicle—are also Post Producer actions, which provide utility features, and are described later in this chapter.

All Post Producer actions are executed by the Vantage Edit Service, and are displayed in Workflow Designer under the Edit action category.

For higher performance, the Conform action (as well as Compose and Chronicle) can run on a Lightspeed server.

Figure 3. Post Producer Actions: Conform, Chronicle, and Compose



Note: Like all Vantage actions, you must configure them for use in a specific workflow before you can run jobs through it. For configuration details on each action, see the man page for the selected action in Workflow Designer.

In addition to these actions, you will usually have many other actions in the workflow, designed to support your objective. For example, the workflow probably has a Watch action so that jobs can be started automatically by dropping a CML file into a folder. The workflow may also include Flip actions, and file operation actions such as Copy, Move, or Deploy.

- [Conform Action](#)
- [Chronicle action](#)
- [Compose action](#)

Conform Action

The *Conform action* is the primary Post Producer action. It assembles (*conforms*) source video, audio, and still image files, based on specifications contained in the CML file that started the job, and then transcodes (or direct-converts) into an output file, based on the configuration you have specified in the Conform action's inspector.

The process of creating media from a set of input media files using CML-base instructions is referred to as *conforming*. The input files and instructions for conforming the output file are all defined in the Composition XML file.

The Conform action can process source video and audio in certain formats. If your input video and audio is not in a supported format, you can transcode it into one of the supported formats prior to processing it in the Conform action. You can create a separate workflow to perform this task, or flip it right in the same workflow.

Note: For a list of supported formats, see [Input Video Formats](#), [Input Audio Formats](#), and [Input Image Formats](#). For supported output formats, see [Output Formats](#).

The Conform action can encode media in various essences, into these three formats: MP4, QuickTime MOV, and TIFO. The Conform action can also apply filters as specified by you, along with the media and container to generate during transcoding. If you require output media in formats other than those directly created by the Conform action, you can pass the media on to Flip, IPTV VOD, or Multiscreen Flip actions in the main workflow or in other workflows for processing.

Note: You can optionally add Video Processing Library (VPL) templates that can be applied to specific media files. VPL templates can be configured to process input media with specific decoding requirements, such as frame rate conversion, Telecine or inverse Telecine, or complex audio mapping. Templates are created by adding a Processor, configuring, and naming it. The template can then be referenced in the source configuration section of a composition, and used to perform decoding as specified when the composition is processed in a workflow with a Conform action.

Chronicle action

The *Chronicle action* transforms *As Run CML files* that have been produced from a Conform action to other formats for other downstream processes. In Vantage Version 6.0, you can use the Chronicle action to generate Manzanita-specific SCTE35 messages in a file, for digital program insertion applications.

Note: An *As Run CML file* is one that has been produced by a Conform action after the file has been conformed (that is, the media file has been produced), and any unresolved timecode values and variables have been fully resolved. To generate an *As Run CML file*, check the Generate Composition Chronicle checkbox in the Conform action.

The SCTE35 file is provided a nickname, for further use in downstream actions.

Compose action

The *Compose action* converts media descriptor files in other formats into a Telestream Composition XML file. Formats that you can convert to compositions include Anystream (AN2) files and Telestream TSEDL files.

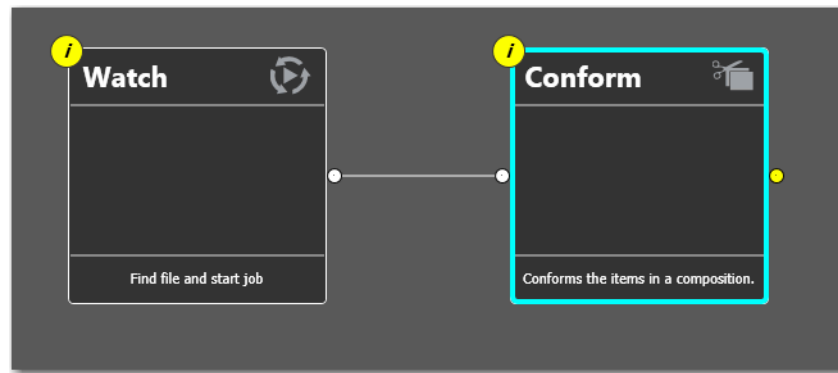
Once a file has been converted to CML file (composition), it can be submitted to the Conform action to compose media. You could create one workflow with both a Compose and a Conform action, or you could create two workflows—one to convert the file (Compose), and the second to conform it.

Typical Post Producer Conforming Workflow

The type of workflow you'll typically use repeatedly to produce your promos with Post Producer is a conforming workflow.

Conforming workflows can be very simple: the basic workflow involves a Watch, and a Conform—to composite the media as described in the CML, and encode it to a variety of formats: DNxHD, JPEG2000, ProRes, Uncompressed, or H264, with AAC, AC-3 or PCM audio.

Figure 4. Basic Post Producer Conforming Workflow



Of course, if these formats are not suitable for your production requirements, you can add Flip, Multi-screen or IPTV-VOD encoding to the workflow. In short—a Post Producer conforming workflow is no different than any other workflow in terms of flexibility or application—except that it uses a Conform action to produce your primary video—which you can use as the starting point for your video production.

Note: For a comprehensive review of all major Post Producer workflows, see [Post Producer Prototype Workflows](#).

Submitting CML Processing Jobs

You can submit a CML file to a Post Producer conforming workflow just like you submit media files to other workflows. Here are the ways that you can submit CML files for processing:

- Dropping a composition file into a folder monitored by a Watch action
- Manually submitting a composition file to a Post Producer workflow in Workflow Designer or other Vantage client program.
- Submitting a composition file to a Vantage workflow via a computer program using the Vantage SDK.

Composition Overview

The purpose of this chapter is to introduce you to the concept of a composition. It helps you understand fundamentally what a composition is. This chapter describes (conceptually) the major components of a composition, and their relationship with each other. Of course, at this introductory level some details are left out, and no mention is made of Composition Markup Language (CML)—the XML schema used to implement a composition. CML is covered in detail in succeeding chapters.

To understand how to describe the behavior and the look and feel of your intended media in a composition, it is important to understand its building blocks. Whether you plan to write a program to auto-generate compositions or manually create a composition in a text editor, this conceptual understanding is vital to your success as a composition creator.

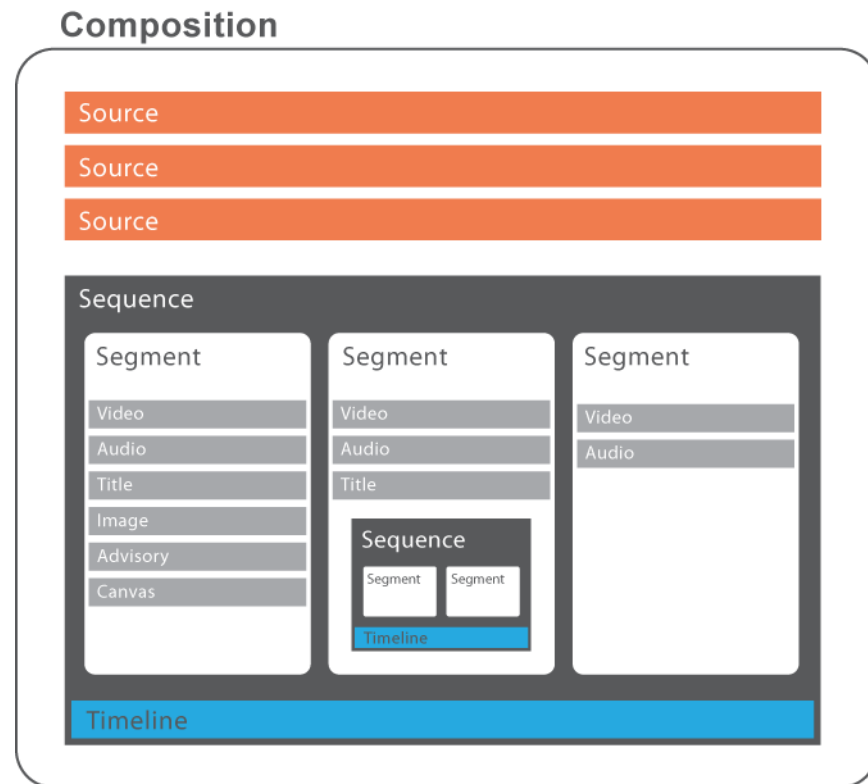
- [Composition Hierarchy](#)
- [Sequences](#)
- [Segments](#)
- [Material](#)

Composition Hierarchy

As illustrated below, a composition is a collection of multi-level objects, working together to create a hierarchical, temporal, and spatial arrangement of media.

A composition also contains sources, which identify the source files of the media that are utilized in the composition.

Figure 5. Composition Hierarchy, Simplified



Referring to the illustration above, note that each composition as a minimum, has these components:

Sources—These are simply pointers to input media files—nothing more than fully-qualified paths to the media files.

Sequences—An independent timeline; a collection of one or more segments, played out sequentially.

Segments—A layered collection of material (and optionally, other sequences) that is played simultaneously.

Material—The media—video, audio, titles, content advisories, canvases, and images—each with unique properties and effects, and spatial and temporal arrangements.

Sequences, segments, and material are described in detail in the following topics.

Sequences

A sequence defines a set of segments, which play sequentially, one after the other, to completion. At least one sequence is required, in order to hold segments. The total length of a sequence is the sum of the length of its segments. Sequences can be placed in the composition, or inside other segments.

Sequences contains segments, where the temporal position (start time) of each segment in the sequence is determined by the accumulated duration of the preceding segments in that sequence. Segments can contain other sequences, as well.

All sequences in a composition begin playing concurrently. Likewise, sequences in a segment begin at the same moment—when that segment begins.

You change the moment when a sequence plays by offsetting it with a time value. You use the layer property of a sequence to control visual layering (compositing): which sequence can obscure the visual material in other sequences. Thus, visual material in a sequence with a lower layer value may be obscured by visual material in sequences with a higher layer value.

The material in a given sequence is played back independently of material in all other sequences. Thus, you should add an additional sequence when you want an additional, independent timeline upon which to present material.

The duration of a sequence is equal to the accumulated duration of all the segments it contains. The longest sequence in the composition determines the duration of the composition.

Segments

Segments define a set of material that is played back concurrently. At least one segment is required in each sequence. Multiple segments play consecutively. Segments are dynamic, bounded time periods—the ordinal (ordered) building blocks of a sequence. Each segment has a duration—and the duration of each segment in the sequence determines the total duration of the sequence.

Segments create time boundaries on the enclosing sequence’s timeline; the material within a segment can not cross the segment’s boundary. You add additional segments to a sequence when you want to increase the timeline of the sequence.

The start time of a given segment is determined by the accumulated duration of the preceding segments. Like a sequence, you can change the moment when material begins playing by offsetting the item with a time value.

Each segment also includes a reference to the source media file, and may include mark-in and mark-out -point to specify the portion of the media to play back.

Material

Material items are the building blocks of your output media.

You can provide the following categories of material in your segments:

- *Video*—instances of the video or video clips referenced by sources
- *Audio*—instances of audio in a file, ignoring video (if present)
- *Images*—instances of raster images, or image sequences
- *Content Advisories*—instances of advisory graphics and metadata
- *Titles*—instances of text, burned in on the frame
- *Canvases*—instances of black rectangles, used for fades

Each type of material has certain format requirements, and a unique set of properties, based on its functional requirements. Like sequences, all items in the same segment play concurrently—at a specific temporal position in the timeline of the enclosing sequence—beginning at the segment time. To alter the moment when material is presented, you provide a time-based offset property.

Like sequences, items within a segment are also layered from back to front visually. Thus, visual items in a lower layer within the segment may be obscured by visual items in a higher layer. Within a segment, every visual item can be layered to establish visual precedence. You can also optionally set opacity, to control obscuring of lower-layered items.

Visual (and Auditory) Effects

Visual effects (such as rotation, scaling, opacity, etc.) can be implemented by simultaneously changing the display attributes of multiple visual material elements. (Volume adjustment is also an effect, which can be applied to any media that has audio.) Using these effects effectively requires a solid understanding of each effect, as well as how to identify various temporal portions of the media.

Note: The application of material in your composition is really where the rubber hits the road. The ability to implement the material in the manner you desire is the key to creating functional compositions. For a comprehensive, detailed discussion of working with material, see Chapter 5, [Working with Material](#).

Creating and Working with Compositions

The previous chapters introduce you to the purpose and features Post Producer, how it is implemented in Vantage, and a conceptual model of a composition. The purpose of this chapter is to provide referential information—practical, detailed guidelines for various aspects of creating and working with composition files.

Note: You may be a programmer; you may not be. As a programmer, you may be writing a program to generate (or otherwise utilize) a composition. Regardless of your background, skills, or method of generating a composition, these topics should be of general value to you in your composition development endeavors.

- [Introduction](#)
- [Approaches to Composition Development](#)
- [Developer Resources](#)
- [Media Characteristics Affect Composition Design](#)
- [Composition File Naming Conventions](#)
- [Specifying Unit Designators in Numeric Attributes](#)
- [The Basic Format of a Composition](#)
- [Validating a Composition](#)
- [Resolving Errors in CML Processing Jobs](#)
- [Using Nickname, Expressions, and Variables in Compositions](#)
- [Commenting CML Files](#)

Note: If you have comments or suggestions about improving this guide—or if you've discovered an error or omission, please email us at techwriter@telestream.net.

Introduction

Composition Markup Language (CML) is an XML schema that provides the vocabulary to describe media elements and their temporal and spatial relationships on a video timeline, in a manner suitable for conforming the specified input files into an output media file that adheres to the general description.

A CML file (or optionally, a blob, in a computer program) is submitted to a Vantage workflow that contains a Conform action to generate output media from the source files provided, according to the instructions in the CML.

Approaches to Composition Development

There are two different approaches to creating or developing compositions. Telestream recommends generally, that you use the Playlist .Net Class Library and develop your compositions in supported .Net languages. The benefit of this approach is that you're relieved of the requirement to hand-craft valid, complex XML. Even after years of working with XML, it is too easy to make XML coding mistakes. With a program, you're assured of producing a valid XML file.

However, you can also write a composition manually, using an XML editor or text editor. The benefit with this approach is that you don't have to be a programmer or learn how to use the Playlist Class Library.

Note: In both cases, there is a common requirement: you need to know how to properly construct a composition to meet your media generation requirements. If you don't understand Composition Markup Language, you'll have a difficult time creating the composition you intend without a lot of trial and error.

The approach you take depends on several factors:

- Technical skills—that you have, or are available to you
- Scale of effort—do you need a few compositions, or lots of them
- Intended use—whether this a testing/learning experience or a production task.

Skills and Tools Required for Programming Approach

If you are planning to write a program to generate a composition, you'll need the following skills and tools:

- Composition Markup Language
- C# or other .Net CLI language
- Playlist Class Library
- Microsoft Visual Studio (or other .Net development tool).

Skills and Tools Required for Manual Approach

If you are planning to write a composition manually, you'll need these skills and tools:

- Composition Markup Language
- XML
- XML or text editor.

Developer Resources

In order to learn about CML and apply it to your video processing requirements, refer to the following products and documents.

Vantage SDK—The Vantage SDK is an indispensable part of CML development. It contains the .Net class libraries, C# code examples, and documents that you need to create compositions, to develop programs for submitting files to Vantage, and also for developing programs to generate Post Producer compositions.

Registered users can download the Vantage SDK by logging into the [Registered Vantage Customer](#) section on the Telestream Web site and following the steps.

Composition Class Library Help—A CHM file in the SDK for programmers who are creating a composition or utilizing compositions for other purposes. This reference describes the Vantage Composition class library (referred to as the Playlist library), which implements the Composition Markup Language.

Post Producer Developer's Guide—The guide you're reading now.

Composition XSD—The specification for a valid composition. You can use this file to validate your compositions before submitting them. It is helpful when you are testing a composition generator program during development, and especially when you are hand-coding a composition. You can also review the XSD in a text editor to determine the exact rules for each element and attribute.

Post Producer Example Files—This collection of files forms the basis for all cookbook examples. They make excellent tutorials, and templates for starting your own projects, or improving them.

Media Characteristics Affect Composition Design

Compositions can't be created in a vacuum. For any composition to be practical and useful, some assumptions must be made.

For any composition, the user should know:

- The intended length of output (60 seconds, 3 minutes, etc.)
- Input and output video formats (SD / HD 720, 1080, etc.)
- Input and output audio track layout: 5.1, stereo, Dolby E, track count and layout

These assumptions directly affect source selection and modification, audio channel mapping, and workflow configuration to generate the required output.

CML assumptions example:

```
<?xml version="1.0" encoding="utf-8"?>
<Composition version="1.0" xmlns="Telestream.Soa.Facility.Playlist">
<!-- Up Next Promo DUR=1 min, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
<!-- All PNG files: 600 x 420 -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conforming Workflow: 1080 VOD -->
```

Telestream recommends that you document your assumptions (or requirements) as comments directly in the CML, so that you and others are cognizant of them as the composition is used or modified. Optionally, provide other design or runtime (Conform action configuration requirements) details including the author, and a change history.

Composition File Naming Conventions

It is common to name XML files with the .xml suffix. By convention, Telestream often names composition files with the .cml file suffix, to identify it as a Composition Markup Language file. The fact is, you can name a composition file with any suffix you want, and submit it to a Post Producer workflow. The suffix is immaterial in this context.

Specifying Unit Designators in Numeric Attributes

Whenever you specify a number in an attribute (for example, `<Crop top="30px" left="0" bottom="-2px" right="100%" />`), you can optionally identify the logical unit designator that applies to it.

Note: Per XML syntax rules, all attribute values must be enclosed in double quotes (" and "). This applies to all value types: strings, integers and other numbers, and keywords. All values must be straight quotes, not smart left and right quotes. Use of smart quotes will cause the workflow to fail. For example, `<Crop top="30px" ... />` If you submit a job with quotes other than straight double quote, it will fail.

If you specify a value without a unit designator, it is *always* treated as a ratio. The ratio may be expressed as an integer, a decimal number, or a rational number {5/10} which must be presented as an expression, in braces. The only functional difference between a ratio and a percent is the location of the decimal point.

The table below describes each unit designator supported in CML

Table 2. Unit Designators You Can Use with Numeric Values

Unit Designator	Value	Description
Percent	%	Use percent (%) to define the ratio of the attribute value compared to another base value, expressed as a percent. The base value being related varies by attribute, and is usually obvious. You can use percent to describe most value types—for example, an increase in volume, or the width of an output frame. You can specify an integer or real number when specifying a percentage. In some attributes, the value may be negative. For other attributes, it must be in the range 0-100%. Or, the value may be over 100%, as in the case of Rotation: 200% is equal to 720 degrees, or two complete rotations.
Decibels	dB	Use Decibels (dB) to define audio volume values.
Degrees	°	Use degrees (°) to specify the amount of rotation. In some attributes, it may be in the range 0 to 360; in others (such as Rotate), you may specify an unlimited amount of rotations—for example, 720 for 2 turns, or 3600 (for 10 turns). You might also be able to specify a negative degree: <code>phase="-90°"</code> , for example.

Table 2. Unit Designators You Can Use with Numeric Values

Unit Designator	Value	Description
Pixels	px	Use pixels to define values relating to dimensions or points in raster images, input and output frames. In some cases, you may use negative pixels. Pixels are always expressed as integer values.
Points	pt	Use points (1/72 of an inch) to define values relating to the size of text, as in a Title element. You can specify an integer or real number when specifying a point value.
Ratio		There is no value for a ratio. Use ratio in the same manner as percent: to define the ratio of the attribute value to another value. Ratio is always a percentage value, with the decimal moved 2 places left. For example, 100% = 1.00. In some attributes, the range is 0 to 1 (or fractions, expressed as a decimal) thereof. In other cases, the value may be greater than 1.

The unit designator you use should immediately follow the value, with no intervening space. For example, `top="30px"`.

The Basic Format of a Composition

Now that you conceptually understand the major elements of a composition and their relationship to each other, let's turn to the task of implementing them in Composition Markup Language—by creating a CML file.

A composition is comprised of two types of elements:

- Optionally, one or more Source elements
- One or more Sequence elements

Basic Composition with Source and Sequence Elements

In the example below, a very simple composition is presented. Note that all attribute values are in straight double quotes—an XML requirement. Also, note the UNC path in the File element—the most effective way to access files in a Vantage domain.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition version="1.0" xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share/path/my_promo.mov" />
  </Source>
  <Sequence layer="1">
    <Segment>
      <Video align="head" adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
  </Sequence>
</Composition>
```

This example illustrates the proper format and organization of the basic composition elements.

A composition always starts with an XML element, as required per XML standards, and has one Composition element. This composition has the following elements:

- Inside the Composition element is a Source element (shown in bold), to identify the source file for the Video. You may or may not need Source elements—you can optionally add one or more as necessary, and you must place them before any Sequence elements.
- A Sequence element (in bold). As with sources, you can have one or more sequences.
- A Segment element (in bold) in the sequence, where the video is utilized.

This example composition will produce an output clip in the format specified in the Conform action. This is the foundation you'll build on to create complex, high-quality highlights, promos, and other clips.

We suggest you copy and paste this into a file, save it as XML, and change the location attribute on the File element to point to a suitable file that your Vantage server has access to. Then, submit the XML file to a Conform workflow and view the output.

Validating a Composition

To validate a composition, you can use a variety of tools (Notepad++, Microsoft Visual Studio, Oxygen, etc.) You can also validate your composition using Web sites provided by the Worldwide Web Consortium at (<http://www.utilities-online.info/xsdvalidation/>), and others.

The validation file to use is named *Composition.xsd*, located in the Vantage SDK.

Note: You should always validate your compositions as you are developing them—regardless of how you generate them. If you submit a composition to a Post Producer workflow with XML or CML errors, it will fail.

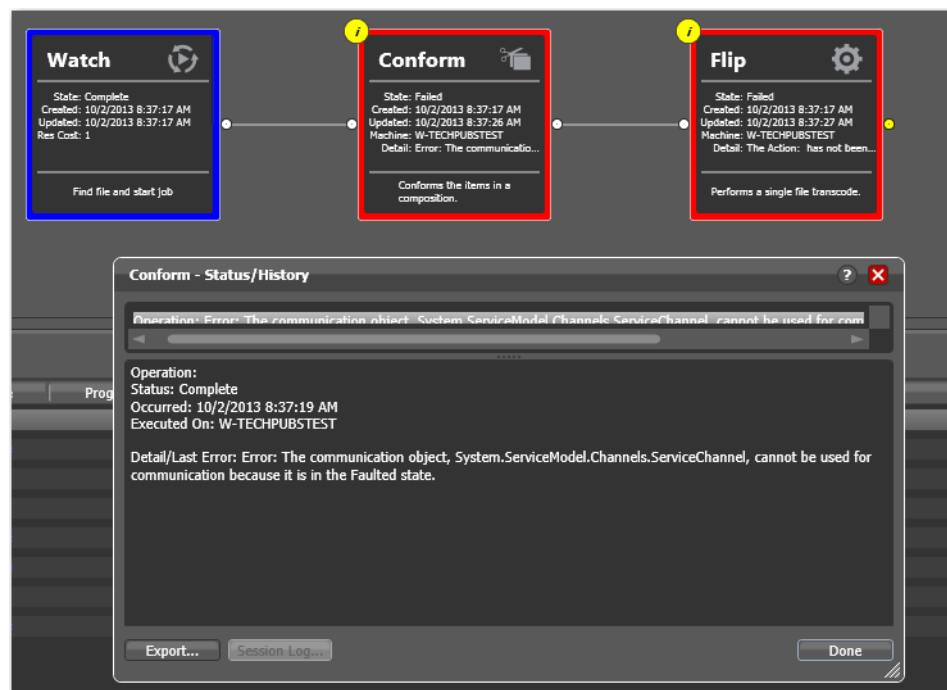
Resolving Errors in CML Processing Jobs

When you submit a CML file to a Post Producer conforming workflow, it may fail due to the following reasons:

- Invalid XML—you may have created an invalid XML file.
- Invalid CML—you may have created an XML file that does not conform to Composition Markup Language.
- Errors in values—you may have a value in an attribute that is not correct.

When a composition file can not be processed by the Conform action, you typically see the following error in Workflow Designer's Job Status tab:

Figure 6. Workflow which has Failed



To resolve the issue, you should validate your composition (see [Validating a Composition](#)). If the composition is valid, verify that the values you supply are correct.

Note: If you provide an attribute that is not a valid attribute (for example, you add *duration*, but its in the wrong element, or you misspell *align* as *allign*), the attribute is ignored. If you provide a value that is of the wrong type (for example, providing a string where an int is required), the action will fail. Or, if you provide a keyword that is not permitted, the action will fail.

Using Nickname, Expressions, and Variables in Compositions

In any attribute or element value in CML (those values surrounded by double quotes), you can utilize nicknames, expressions, and variables, which you place in braces { }.

By applying nicknames, expressions, and variables in your compositions, you can make them more broadly applicable and thus, more practical—you'll usually end up creating fewer, more flexible compositions.

The evaluation of the expressions and variables is performed by the Conform action during execution to resolve expressions and variables to the final string value before the composition is processed. All values are ultimately presented as strings, of course.

Table 3. Nickname, Expression, and Variable Format

Type	Format
Nickname	{#Nickname}
Expression	as shown—see Expression Examples
Variable	\${VariableName}

Using Nicknames in Compositions

- [Using Nicknames in Compositions](#)
- [Expressions](#)
- [Expression Examples](#)
- [Variables](#)
- [Variable Examples](#)

Using Nicknames in Compositions

Vantage allows you to associate a name with a file, to make it easier to design and implement workflows. For example, you can use the term *Original* as the nickname for the file ingested by a Watch action, or *Proxy* for a Flip action that produces a QuickTime, low-res file. Vantage supplies several; you can also add your own and use them anytime.

A nickname can also be used in place of file names in a composition. During processing, the runtime value of the nickname is substituted. In order to use nicknames, an upstream action must have either ingested the file or produced it, with the associated nickname.

A nickname in a composition always takes the form {#Nickname}.

The use of nicknames creates more flexibility in the design of compositions. For example, you could have one composition that works in three different workflows, each processing video for a different output size, or language. In each workflow, you could use an Associate action to identify the correct file for the workflow.

Nickname Examples

In the example below, the File element references a file by nickname: `{#promo_mov}`.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition version="1.0" xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="{#promo_mov}" />
  </Source>
  <Sequence layer="1">
    <Segment>
      <Video align="head" adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
  </Sequence>
</Composition>
```

In the workflow that processes this composition, you'd have to include an associate (or other) action to ingest or produce the file, and assign it the *promo_mov* nickname.

Expressions

You can use arithmetic expressions to provide an attribute or element string value. (Using expressions alone—without variables—may not be very practical—though it does present the expression's logic.)

Supported operations are:

- Addition (+)
- Subtraction (-)
- Division (/)
- Multiplication (*).

Expressions can include parentheses to support precedence.

Expressions that use only integers return an integer presented as a string. Expressions that involve real numbers return a real number presented as a string, even if the decimal portion is zero.

Expression Examples

Here are some expression examples:

```
<... "{4 * (1+3)}" /> is evaluated as <... "16" />
<... "{4 * 4.0}" /> is evaluated as <... "16.0" />
<... "{4 / 3}" /> is evaluated as <... "1.33333333" />
<Translation x="{40+40}%" /> is evaluated as <Translation x="80%" />.
```

Variables

You can also use variables in attributes and elements, to provide a value during processing, rather than at design time. The variables may be of any type, as appropriate: integers, strings, timecodes, etc.

Note: Variables must be signified by pre-pending them with a dollar sign(\$). For example, the variable *Rating* is represented as *\$Rating* in the CML.

If the value of the variable will be computed or derived from analysis, the value must be specified in a previous action in the workflow. If a static value will be used for the workflow, then the variable simply needs to be declared with a default value at design time, either in the Conform action (right-click on the action and select Add Variable) or by declaring the variable and the default value directly in the Vantage Management Console.

Variable Examples

Here are some examples of using various types of variables in compositions.

Numeric Variables

In this example, you are supplying an SD source file which contains NTSC IMX video (720 pixels by 512 lines), and you want to display it in an HD target aspect ratio. You use the Crop element to remove vertical blanking, and the Mask element to convert the source to the 16:9 target aspect ratio by adding a 2:9 (0.222) curtain to each side of the image.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition version="1.0" xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share/path/Source_video.mxf" />
    <Crop top="{ $VBI_Top }px" left="0" bottom="-{ $VBI_Bottom }px"
right="100%" />
    <Mask top="0" left="{ $2_9_Curtain }" bottom="100%" right="-
{ $2_9_Curtain }" />
  </Source>
  <Sequence layer="1">
    <Segment>
      <Video align="head" adjust="edge" fill="none" source="1" layer="0"/>
    </Segment>
  </Sequence>
</Composition>
```

In your workflow, in an appropriate action upstream of the Conform action, you'll add three variables: *VBI_Top* and *VBI_Bottom*, and *2_9_Curtain*, and then assign the values: 30, 2, and 0.222222, respectively.



Channel Number Variables

In this example, channel numbers are identified by variables, whose values are set at runtime in the workflow.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition version="1.0" xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="{#Original}" />
    <Mix source="{ $AudioChannelLeft}" target="1" level=".9"/>
    <Mix source="{ $AudioChannelRight}" target="2" level=".9"/>
  </Source>
  <Sequence layer="0">
    <Segment>
      <Video align="head" adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
  </Sequence>
</Composition>
```

String Variables

You can also provide text variables, and optionally, concatenate strings. For example:

```
<Title align="head" adjust="edge"... >Watch { $Show_Name} tonight
at { $Show_Time}</Title>
```

In your workflow, in an appropriate action upstream of the Conform action, you'll add two variables: *Show_Name* and *Show_Time*, and assign the string values "What's Up, Doc?" and "7:00 PM Eastern", respectively.

Note: If the action does not explicitly provide the variable you require, you can always right-click the action and select Add Variables to provide any variable you want.

Commenting CML Files

XML permits comment lines. Telestream encourages you to comment your compositions as appropriate.

Here's an example code snippet, with comments in bold.

```
...
</Source>
<!-- Stereo audio in video source and 10 VOs map source mono -->
<!-- track to a track in the output file. Channel map must be -->
<!-- set in the Compose action in the Vantage workflow. -->
  <Sequence layer="1">
    <Segment>
      <Video align="head" adjust="edge" fill="none" source="1" layer="0">
...

```

Using comments can make understanding, modifying, and maintaining your compositions much easier.

Working with Material

The purpose of this chapter is to help you learn the best approach to working with material in the context of a composition: accessing source files, creating clips, composing media, applying visual effect, and aligning it on the timeline, for example.

- [Introduction to Material](#)
- [Adding Material to a Composition](#)
- [Extracting Clips from Master Media](#)
- [Cropping and Masking](#)
- [Applying Effects to Material](#)
- [Controlling Material on the Timeline](#)

Note: All of the examples in this chapter are complete compositions. To use them, copy and paste the example into a text editor. Change the references to the media files you are supplying (note the design specs as comments at the top of each composition), and save the file as a text file.

Then, submit it to a Post Producer conforming workflow ([Simple Conforming Workflow](#)) in Vantage, using Workflow Designer.

Introduction to Material

Post Producer supports six types of material that you can use in a composition; each defined by a specific element:

- *Video*—instances of temporal, frame-based video
- *Audio*—instances of audio, ignoring the video stream (if present)
- *Image*—instances of raster images, or image sequences
- *Advisory*—instances of advisory graphics and metadata
- *Title*—instances of text, rendered on the frame
- *Canvas*—instances of black rectangles, used for fades.

Material of course, is arranged in sequences and segments within the timeline, along with formatting instructions: montage ordering, composition layering, audio mixing and mapping, etc.

Material Attributes

Each element of material includes attributes that describe it in relation to the other material, and also describe its properties. There are fundamentally two types of attributes in CML:

Spatial Attributes—these are attributes that describe characteristics of the material and its relationship to other material.

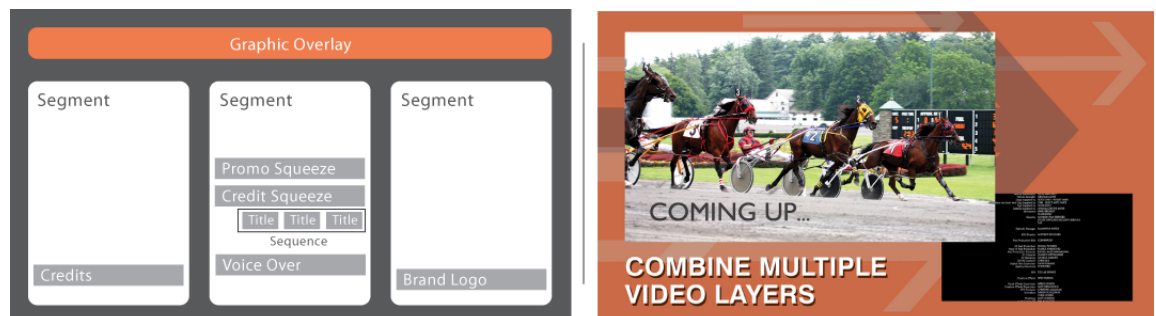
Transformational Attributes—attributes that describe the behavior and timing of the material: size, location, opacity, rotation. and audio volume.

The application of material elements is illustrated in the following graphic.

Sample Story Board

It's often helpful to draw out your composition graphically, to help you organize the material and determine specifically what you are trying to achieve.

Figure 7. Typical Composition Story Board and Sample Frame



On the left, is a designer's story board of each element in the composition, arranged temporally, with three individual segments. Each segment is implemented when you want to divide the composition into distinct temporal segments, in which you can independently present media. On the right is a sample frame of the resulting video. Notice that this story board does not illustrate spatial arrangement; only temporal arrangement of material.

Story boarding helps you identify each segment and sequence, and each instance of material in every segment. Once you know these details, you can create your composition and begin adding attributes to control how each item of material should be presented.

Sample CML

The story board (or other form of design) must be implemented in CML (converted to a composition), and processed in a Conform workflow to generate the output. Here is a very simple composition, with one instance of video, in one segment, within one sequence.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
<!-- Basic CML Sample DUR=NA, AR=16x9, res. 1920x1080 -->
<!-- Input: 1080i Apple ProRes 25 FPS 16b 48k stereo -->
<!-- Output: 1080i x264 -->
<!-- Conform Workflow: Basic Conform 1920x1080 x264 AAC + Flip Proxy -->
  <Source identifier="1">
    <File location="//share/path/source_video.mov" />
  </Source>
  <Sequence layer="1">
    <Segment>
      <Video adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
  </Sequence>
</Composition>
```

This is about as basic as it gets, and it illustrates how all these elements fit together.

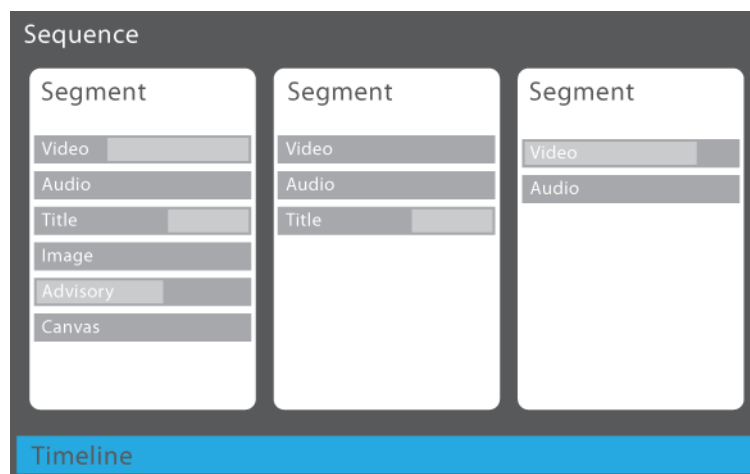
Adding Material to a Composition

To add material (or a portion of it) to your composition, you add its element to the [Segment](#) where you want it to play, as you saw in the CML example above.

Note: For specifications of each element and its attributes, click on the link to view its reference topic in the [Composition Markup Language Reference](#) chapter.

Recall that each segment identifies a specific portion of the composition's timeline, based on its ordinal position in the sequence:

Figure 8. Segments Play Sequentially in their Order in the Composition



Also recall that each sequence creates an independently-functioning timeline within a composition—the point of a sequence is to provide you a new, independent timeline upon which you can place segments of material and play them independently of any other material in the composition.

Certain material is stored in files: Video, audio, and images. Post Producer also supplies Advisories with USTV images. For other rating systems, you must supply your own.

Canvas and Title material (which creates formatted titles from text) does not use files.

- [Specifying Files for Video and Audio Material](#)
- [Specifying Files for Image Material](#)

Specifying Files for Video and Audio Material

For each instance of video and audio that you plan to use in a composition, you need to add a [Video](#) or an [Audio](#) element to the correct segment. You can add as many video and/or elements as you need, and any number of them can create new instances of media from the same file. You can also create clip instances, by setting in and out points.

You also need a corresponding [Source](#) (a child of *Composition*), to identify the source file. By default, if a video source contains audio, it also plays when the video is played.

For convenient access, place *Source* elements as the first elements in the composition.

Post Producer allows you to combine source files with differing frame rates and telecine cadences, plus SD, HD, television and 23.98 sources and re-encode them as specified in your output media specifications.

All video and audio files must be in a supported format that the Conform action can process. A file may contain the media essence, or it may reference additional files containing media essences (for example, a QuickTime reference movie).

For a list of supported input formats, see [Post Producer Formats](#). If your video and audio files can't be encoded by the Conform action, you can add a Flip action to the workflow upstream of the Conform action (or create a separate workflow) to transcode the source file into a supported format that the Conform action can encode.

Note: The Vantage Edit service (there may be more than one in your domain) that executes the Conform action in the job must have access to each file. Because files are typically not stored on the same server as the Vantage domain, Telestream recommends using shares to reference files. For details on dealing with file access by Vantage services, see the Domain Management Guide.

Here are the steps:

1. Add a [Video](#) or [Audio](#) to a [Segment](#).
2. Add a [Source](#) to the [Composition](#) (usually at the top, where it's easy to access).
3. Add an identifier attribute to the video or audio element with a unique integer value in the composition.
4. Add a [File](#) to the Source element, supplying a location attribute with a fully-qualified file path and the file name.
5. Finally, add a source attribute to the video or audio element, using the same integer value as the Source element's identifier attribute.

Video Example

In this example, one source file is specified, and one instance of it is used in a segment.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
<!-- Basic CML Sample DUR=NA, AR=16x9, res. 1920x1080 -->
<!-- Input: 1080i Apple ProRes 25 FPS 16b 48k stereo -->
<!-- Output: 1080i x264 -->
<!-- Conform Workflow: Basic Conform 1920x1080 x264 AAC + Flip Proxy -->
  <Source identifier="1">
    <File location="//share/path/source_video.mov" />
  </Source>
  <Sequence layer="1">
    <Segment>
```

```

        <Video adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
</Sequence>
</Composition>

```

Why not just reference the file directly in the Video or Audio element? Because it is common to use multiple segments of material from a given source (for example, a master file). Also, in a source, you can perform basic modifications in one place: cropping and masking, audio mixing and routing, etc.

Audio Example

In this example, four sources files are specified; one for video and four for audio.

```

<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
<!-- Audio DUR=NA, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conform Workflow: Basic Conform 1920x1080 x264 AAC + Flip Proxy -->
    <Source identifier="1">
        <File location="//share/path/Amsterdam_promotion.mp4"/>
    </Source>
    <Source identifier="2">
        <File location="D:\media\Audio_Clips\english.m4a" />
        <Mix source="1" target="3" level=".9"/>
    </Source>
    <Source identifier="3">
        <File location="D:\media\Audio_Clips\French.wav"/>
        <Mix source="1" target="4" level=".9%"/>
    </Source>
    <Source identifier="4">
        <File location="D:\media\Audio_Clips\italian.wav" />
        <Mix source="1" target="5" level=".9%"/>
    </Source>
    <Sequence layer="1">
        <Segment>
            <Video align="head" adjust="edge" fill="none" source="1"
layer="0">
                ...
            </Video>
            <Audio align="head" source="2" offset="00:00:00.50" />
            <Audio align="head" source="3" offset="00:00:00.50" />
            <Audio align="head" source="4" offset="00:00:00.50" />
        </Segment>
    </Sequence>
</Composition>

```

This audio will be mixed or routed, as appropriate, so that viewers of different languages can hear the audio channel of their native language.

Specifying Files for Image Material

When you add an *Image* element (either a single image, or an image sequence) to a segment, the element directly references its file or files via the *location* attribute, as illustrated below (in bold).

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <!-- Image Example DUR=NA, AR=16x9, res. 1920x1080 -->
  <!-- All PNG files: 1920 x 1080 -->
  <!-- Input: NA -->
  <!-- Output: 1080i x264 -->
  <!-- Conform Workflow: Basic Conform 1920x1080 x264 AAC + Flip Proxy -->
    <Sequence layer="1">
      <Segment>
        <Image align="tail" adjust="body" fill="none" layer="2"
location="\share\path\1C_Arrow_Title-PNGs\frame-000508.png"
frames="72" layout="none">
          <Head>
            <Fade duration="00:00:01.000" />
          </Head>
          <Tail>
            <Fade duration="00:00:01.000" />
          </Tail>
        </Image>
      </Segment>
    </Sequence>
  </Composition>
```

When you add a single image, you need to adjust its properties to play in a certain manner for a given duration. Likewise, you can supply an image sequence by referencing to the first file of a given pattern in a folder of sequential images. For full details, see *Image*.

1. Add an *Image* to a *Segment*.
2. Add a location attribute to the Image element, which references the image (or first image in the sequence).

For complete details, see *Image*.

Specifying Files for Advisory Material

For each instance of an advisory that you plan to use in a composition, you insert an [Advisory](#) in the correct segment.

If you are using the USTV rating system, default images (the larger ones) are supplied for each combination of ratings and reasons, so you don't need to reference a file. For other rating systems, you need to supply your own image, and reference it in the element, as shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
<!-- Advisory Example DUR=10s., AR=16x9, res. 1920x1080 -->
<!-- All PNG files: large USTV Ratings Bugs -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conform Workflow: Basic Conform 1920x1080 x264 AAC + Flip Proxy -->
  <Sequence>
    <Source identifier="1">
      <File location="//share/path/source_video.mov" />
    </Source>
    <Sequence layer="1">
      <Segment>
        <Video adjust="edge" fill="none" source="1" layer="0" />
        <Advisory align="head" adjust="body" fill="none" layer="1"
duration="00:00:05.000" type="USTV" rating="PG-13" />
      </Segment>
    </Sequence>
  </Composition>
```

Extracting Clips from Master Media

The purpose of this topic is to explain how to perform frame-accurate editing—extracting clips from master files at run-time—directly in your compositions.

Much of the video that you use as source material in Post Producer applications is longer than you require for a given promo. For example, you may want a 30-second clip of a movie that's about to air. Or, you may want to create a squeeze-back video and run the credits of the last movie that aired, in an Up Next promo. Thus, you need a way to create clips.

There are two practical ways to create video and audio clips for use in a composition:

- *Create clips externally, then reference each clip for use in a composition*
- *Create clips dynamically from master media directly in a composition*

To use clips that have been created externally, you simply add a Source element, and then reference the media as appropriate ([Adding Material to a Composition](#)).

However, you may have a master file—a full-length movie—from which you want to identify and extract several smaller instances for your highlights composition.

As an alternative, Composition Markup Language also enables you to dynamically create clips from master media—producing the clips at run time by extracting the clip with mark-in and mark-out points—using [Head](#) and [Tail](#) elements.

Overview

To extract a clip (create an instance of media) from a master file, you need to:

- Identify the source (using a Source element) that you want to extract clips from.
- Set mark in and mark out points (by adding [Head](#) and [Tail](#) elements with an [Edit](#) to the [Audio](#) or [Video](#) element) on the media that define the portion you want to use.

Here are the CML elements that you'll use when defining a clip:

- Heads and Tails—the point in the media which defines the mark-in ([Head](#)) and mark-out ([Head](#)) point.
- Edit—when used in a Head, the mark-in point of the clip. When used in a Tail, the mark-out point of the clip.

Together, the Head and Tail elements with an Edit thus define the clip of material to be extracted from the source file.

Note: It is important to note that Head Edits are inclusive; Tail Edits are exclusive. That is, the first frame after the specified time in the Head Edit is *included* in the clip. However, the first frame after the specified time in the Tail Edit is *excluded*.

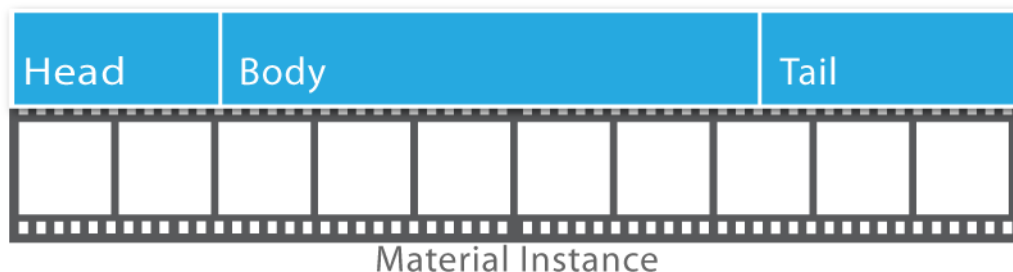
Heads, Tails, and Bodies

To crop media from a master file to create a clip, or to apply effects to your material, you need to understand how to use and implement heads, tails, and bodies. (If you don't

have effects and you don't need to clip the instance, there's no point in having a Head, Tail, or Body.)

First, let's examine a graphic illustration of these components of an instance of media (a Video or Audio element).

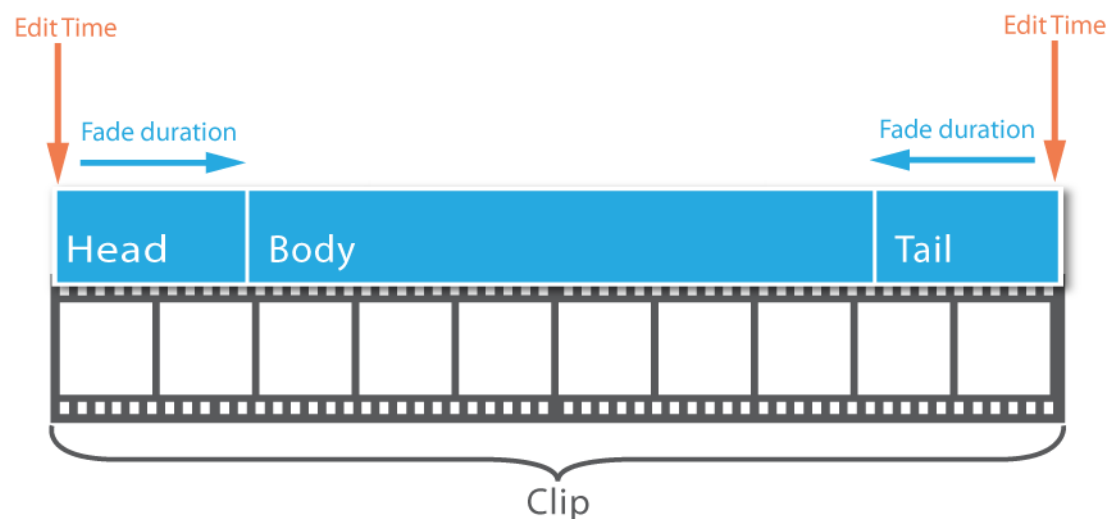
Figure 9. Heads, Body, and Tail are Points on Material



The Head, Body, and Tail are technically temporal locations (mark-in and mark-out points) on the timeline of the material instance—but they're not depicted that way here. Instead, they are depicted as having a duration, which is usually the easiest way to think of using them, because in most cases, Edits are implemented with Fades, which are fundamental to applying effects.

The mark-in point of the Head is the left edge—by default the beginning time or timecode of the material. By adding an Edit element to a Head, you can set a mark-in point to trim the beginning of the clip. Likewise, you can set a mark-out point in the Tail, to trim it from the end of the clip. (The mark-out point of the tail is of course the right edge—the mirror opposite of the Head.) Without an Edit, the mark-out point of the tail is the end of the clip. Edit elements can only be added to a Head and a Tail.

Figure 10. Heads and Tails can have Edits—Head, Bodies, and Tails can have Fades



By definition, a Head can not begin (have a time) before the beginning of the instance. It can be later (with an Edit), and it can have an effect duration (Fade), extending to the right (forward) along the instance. Similarly, a Tail can not be later than the end of the instance, but it can be earlier with an Edit, and it can also have a Fade, extending in this instance backward along the instance.

The Body's point (and duration) is not explicit—it's implicit—controlled entirely by how you set up the Head and Tail with Edit and Fade elements.

Note: You can also add a Fade element, to control each effect's duration. The Fade element is what gives a Head, Body, and Tail its length, so to speak. Using a Fade is describe in detail in [Applying Effects to Material](#).

Example

Here is a simple example, with bold text identifying the operative elements:

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <!-- Up Next Promo DUR=1 min, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
  <!-- All PNG files: 600 x 420 -->
  <!-- Input: 1080i -->
  <!-- Output: 1080i x264 -->
  <!-- Conforming Workflow: 1080 VOD -->
    <Source identifier="1">
      <File location="\share\path\clip1.mov" />
    </Source>
    <Source identifier="2">
      <File location="\share\path\master.mov" />
    </Source>
    <Sequence layer="0">
      <Segment>
        <Video source="1" layer="0" />
        <Video source="2" layer="1" align="head" adjust="edge"
fill="none" >
          <Head>
            <Edit time="00:00:04.000" />
          </Head>
          <Tail>
            <Edit time="00:00:22.200" />
          </Tail>
        </Video>
      </Segment>
    </Sequence>
  </Composition>
```

The Source elements in this composition define the media files to be used in the composition. The *identifier* value in each source is what you use in the Video element's *source* attribute to reference the correct file.

The first Source points to *clip1.mov*, a clip which has been clipped and saved as a file outside of Post Producer, for use in this composition as-is. The second Source points to *master.mov*, a master file from which we want to extract a clip dynamically.

Here's how the second Video element is modified to create a clip dynamically:

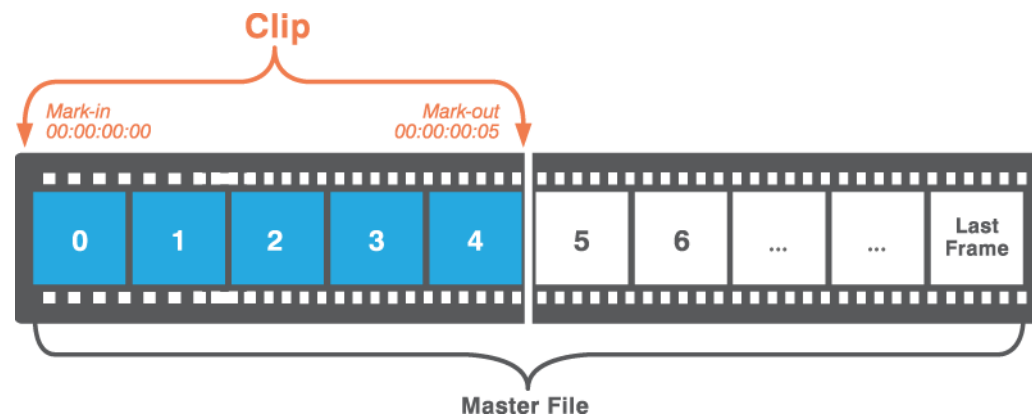
1. Add a Head element to define the mark-in point on your master media file.
2. Add an Edit to the Head and specify the mark-in time.
3. Add a Tail element to define the mark-out point on your master media file.
4. Add an Edit to the Tail and specify the mark-out time.

Specifying Mark Times

In both the Head and the Tail, you specify the time by adding an Edit element and configuring it. The Edit element has two attributes—*time* and *mode*. Together, these two attributes allow you to comprehensively describe the mark-in and mark-out point in various situations.

Note: The time of the mark-in point on the master file is *inclusive*; the mark-out point is *exclusive*—it references the point in time immediately following the last frame to be included—the first frame to be excluded. For example, you specify mark-in 00:00:00:00 (the first frame in the clip) and mark-out 00:00:00:05 (the 6th frame in the clip), to create a clip with 5 frames (0, 1, 2, 3, and 4), as illustrated in the figure below.

Figure 11. Typical Content Advisory Graphic Overlay



Using the Mode Attribute

The Mode attribute has 3 keyword values:

- *absolute* (default)—used when a timecode is present in the source; identifies the edit point relative to the timecode. If you supply a timecode but the source lacks a timecode, it is converted to a time value and utilized.
- *relative*—specifies an edit point measured from the beginning (in a Head) or ending (in a Tail) of the source, irrespective of the timecode (if present).

- *duration*— specifies an edit point measured as a length of time, from the end (in a Head) or the beginning (in a Tail) of the source. (You can't use duration in both Head and Tail on the same instance of media.)

Using the Time Attribute

You can specify the *time* value using time or timecode, as appropriate:

- *Time Notation*—always specified with seconds as a decimal number, in the pattern HH:MM:SS.sss, with microseconds (fractional second in 3 digits).
- *Time Code Notation*— specified in the patterns (HH:MM:SS:FF | HH:MM:SS:FF), optionally using a drop timecode indicator (;) for US timecodes.

Note: You should always use non-drop-frame time code, unless the time code in the source is drop-frame time code. Then you should specify your time code as drop-frame time code (using a semicolon between the second and frame value) as well.

A time is not valid, when mode is set to *absolute*. A timecode is not valid when mode is set to *duration*.

When you specify a time, Post Producer ignores the timecode in the source material, even if it exists. The time always represents the temporal distance from the beginning of the media, whether there is a time code or not in the source media.

When you specify a time code, Post Producer uses the timecode in the source material, if it exists—the time represented by your time code is relative to the timecode in the file. Thus, it must be consistent with the timecode in the source. For example, if your media has a timecode of 01:00:00:00, your timecode must be specified relative to that timecode.

Time and Timecode Examples

The following example illustrate how to use time and mode attributes in Edit elements.

Head Edit in Absolute Mode

```
<Head>
  <Edit mode="absolute" time="01:00:02:12"/>
</Head>
```

This material must have a timecode, or this will fail. Assuming that the timecode begins at 01:00:00, the mark-in for this video is set to 2 seconds, 12 frames.

```
<Head>
  <Edit mode="absolute" time="00:00:02.500"/>
</Head>
```

This material must also have a timecode. The mark-in is set to the point in the source at 2.5 seconds, regardless of the start time on the timecode.

Head Edit in Relative Mode

```
<Head>
```

```
<Edit mode="relative" time="00:00:02:12"/>
</Head>
```

This material may or may not have a timecode; it is ignored if present. The mark-in for this video is set to 2 seconds, 12 frames from the beginning of the source.

```
<Head>
  <Edit mode="relative" time="00:00:02.500"/>
</Head>
```

This material may or may not have a timecode; it is ignored if present. The mark-in for this video is set to 2.5 seconds from the beginning of the source.

Head Edit in Duration Mode

```
<Head>
  <Edit mode="duration" time="00:00:54.200"/>
</Head>
```

In *duration* mode, you must specify time only—you can't use a time code value. The Head's mark-in point is set at 54.200 seconds back from the end of the source.

Tail Edit in Absolute Mode

```
<Tail>
  <Edit mode="absolute" time="01:00:30"/>
</Tail>
```

This material must have a timecode, or this will fail. Assuming that the timecode begins at 01:00:00, the mark-out for this video is set to 30 seconds from the beginning.

```
<Tail>
  <Edit mode="absolute" time="00:00:30.00"/>
</Tail>
```

This material must also have a timecode. The mark-out is set to the point in the source at 30 seconds, regardless of the start time on the timecode.

Tail Edit with Relative Mode

```
<Tail>
  <Edit mode="relative" time="00:00:02:12"/>
</Tail>
```

In a Tail element, when specifying a time code, Edit sets the mark out point at 2 sec, 12 frames from the end of the source.

```
<Tail>
  <Edit mode="relative" time="00:00:02.50"/>
</Tail>
```

In a Tail element, when specifying a time, Edit sets the mark out point at 2.5 seconds from the end of the source.

Tail Edit with Duration Mode

```
<Tail>
  <Edit mode="duration" time="00:00:30.000"/>
</Tail>
```

In *duration* mode, you must specify time only—you can't use a time code value. The point is specified from the opposite end. For example, the mark-out point in this example is set at 30 seconds from the beginning of the source or the Head element, if specified. Thus, the instance of material will run for (have a duration of) 30 seconds from the Head.

Cropping and Masking

Sometimes, you need to crop or mask your video in a segment. For example, you might want to use SD material in an HD promo, and thus, need to crop and mask it.

You can perform cropping and masking directly in a composition. To do so, you add a [Crop](#) and/or [Mask](#) to the [Source](#) that identifies your source video file, and configure the Crop to suit your input frame size and the mask your output target aspect ratio and/or output frame size requirements.

Calculating the New Frame Size

Both elements have top, left, right, and bottom attributes. For maximum flexibility, you can supply integer or real values, identified as a fraction (no designator), pixels (px) or percent (%). If you do not supply a unit designator (% or px), the value is a ratio (for example, 0.222 = 222/1000 or 22.2%) of the frame.

The origin X and Y coordinates are the top left corner—0%, 0% or 0,0 pixels. The bottom right corner is indicated by 100%, 100% or X,Y pixels, where X and Y are the input frame's pixel dimensions.

Cropping Details

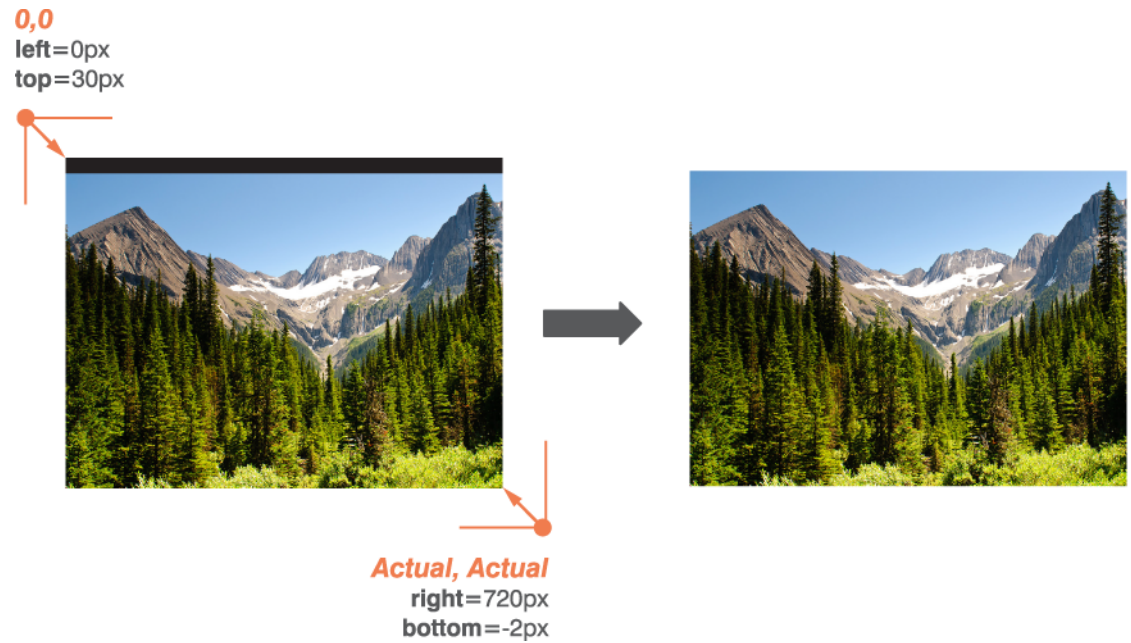
You perform cropping (making the frame smaller) to remove material from your input frame *before* creating an output frame with a mask. Cropping is performed on the input frame, relative to the input media frame size.

Top and left adjustments values are in relation to the origin; bottom and right from the bottom right corner. Positive values are downward or to the right; negative values are upward or to the left.

If you specify a pixel dimension, the left and top edge origins are at 0, 0 pixels; the right and bottom edge origins are at the width and height of the source frame.

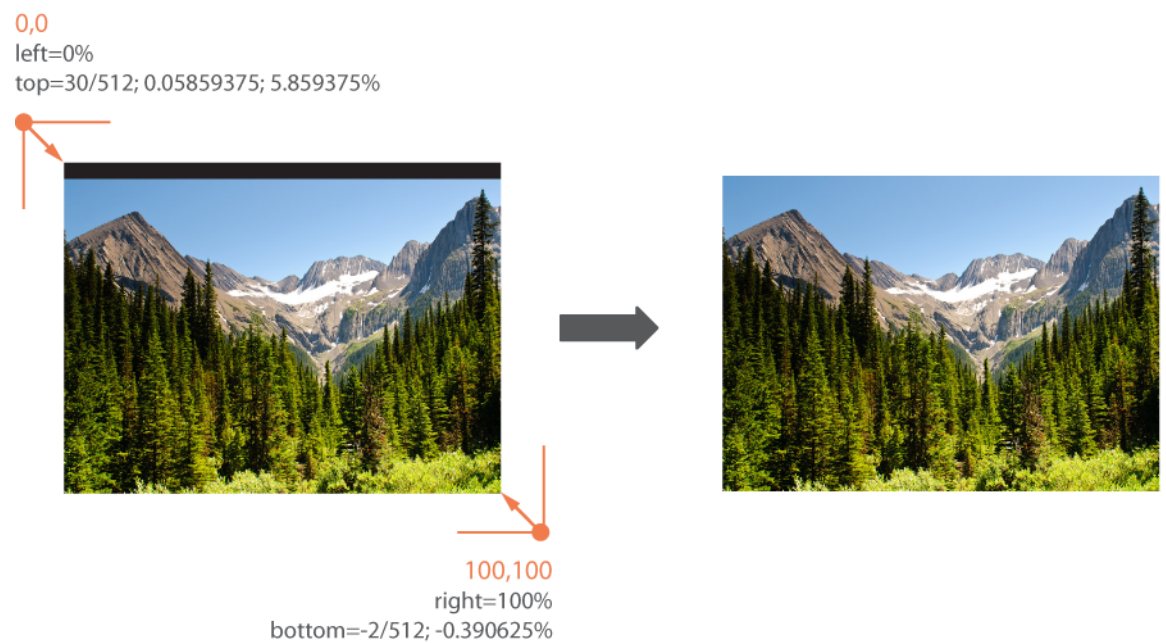


Figure 12. Specifying Crop Dimensions in Pixels



If you specify a percent, the left and top edge origins are at 0%; the right and bottom edge origins are at 100%.

Figure 13. Specifying Crop Dimensions in Percent, Fractions, or Ratios



Base your crop values on the increase downward and to the right from 0 for top and left values, and the decrease to the left and upward from 100% or N (dimension) pixels for

bottom and right. (The bottom and right dimensions obviously may change from job to job, based on your input media.)

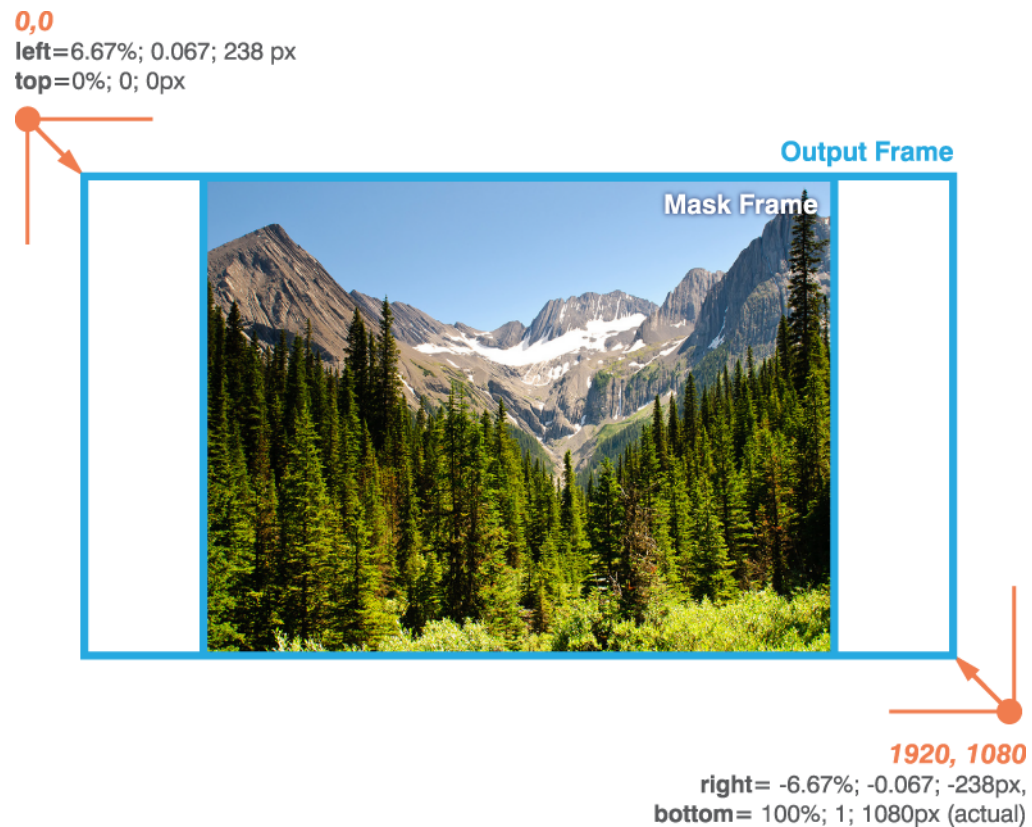
Masking Details

Masking on the other hand, is performed to implement an active rectangle inside the output frame, relative to the output frame size—often to preserve aspect ratio.

Note: Masking is usually applied only when you have mixed media—converting input media of one aspect ratio to output media of another aspect ratio.

When scaling occurs (for example, from SD to HD), the input frame (after optional cropping) will be scaled into the masked (active) region (if specified) and display correctly.

Figure 14. Typical Masking Application



You also base your mask values on the difference from 0 for top and left, and from 100% or N (dimension) pixels for right and bottom dimensions of the output frame.

Note: By using percent as your unit of measurement on a mask, you can make various HD-aspect ratio output files, without changing the mask specification.

When masking, the default fill color for either pillars or bars is black. To change the color, use the *color* attribute (as shown below in the example). Keyword values are *black* | *white* | *transparent* | *ARGB value*.

To replace black as the fill color, specify an ARGB value indicated by the # sign, followed by 4 pairs of hex values: Alpha, Red, Green, and Blue. (Alpha at FF is fully-opaque; 00 is fully transparent.)

Example

This example converts an SD source file to an HD target aspect ratio. In this example, the source contains an NTSC IMX video track (720 pixels by 512 lines). Use the Crop element to remove vertical blanking, and then use the Mask element to convert the source to a 1080 output frame with a 16:9 target aspect ratio by adding a 2:9 (0.222) curtain to each side of the image.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition version="1.0" xmlns="Telestream.Soa.Facility.Playlist">
<!-- Name: Crop and Mask Demo Author: Jim D 10/2013 -->
<!-- Input: -->
<!-- Output: 1080i x264 w/DolbyE -->
<!-- Conforming Workflow: -->
  <Source identifier="1">
    <File
location="\sun\techpubs2\PostProducer\Collage_Comp\collage\1_A2_Show_ONL
Y_NO-GFX2.mov" />
    <Crop top="30px" left="0" bottom="-2px" right="100%" />
    <Mask top="0" left="0.222222" bottom="100%" right="-0.222222"
color="#FFA2A4A6" />
  </Source>
  <Sequence layer="1">
    <Segment>
      <Video adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
  </Sequence>
</Composition>
```

The Crop element removes 30 lines (pixels) from the top of the source frame, none from the left edge, 2 pixels from the bottom, and no material from the right.

This mask leaves the top edge intact; while removing 22/100 (22%) of the pixels from the left edge. When encoding 1080 HD, this creates a mask boundary 238 pixels in from the left, filling the left pillar with the fill color. The bottom edge is also left intact. The mask also removes 22% of the pixels from the right edge, creates another mask boundary 238 pixels in from the right as well. The region bounded by this rectangle is now a 4:3 ratio, so that the SD image can be scaled proportionally into the new HD frame and presently accurately, at a larger size.

Applying Effects to Material

One of the most powerful (and flexible) features of Post Producer is its ability to manipulate media as it is being played out. You can apply the following effects to material ([Advisory](#) / [Canvas](#) / [Image](#) / [Title](#) / [Video](#) / [Audio](#)) elements:

- *Opacity*—when you want to change the relative opaqueness material, you add an Opacity element to it. For implementation details, see [Opacity](#).
- *Scaling*—when you want to change the display dimensions of material, you add a Scaling element to it. For implementation details, see [Scaling](#).
- *Translation*—when you want to move the material around in the frame, you add a Translation element to it. For implementation details, see [Translation](#).
- *Rotation*—when you want to rotate material on the Z axis, you add a Rotation element to it. For implementation details, see [Rotation](#).
- *Volume*—For audio material (or video that has audio in it), you can adjust the volume. For implementation details, see [Volume](#).

Each effect has its own element. However, you don't apply these effects directly to a material element (a Video, for example). Instead, you apply it (as appropriate), to portions of the material, to achieve animation effects and smooth changes over time.

In Post Producer, these portions are referred to as Head, Body, and Tail. By identifying each temporal part of the media (and specifying its mark-in or mark-out point and duration)—and applying effects as appropriate to each portion, you can apply effects in almost limitless ways, such as fades and animation.

Controlling Material on the Timeline

The functions we discuss in this topic are all about time: they all relate to the temporal location and playout of material (Advisory | Canvas | Image | Video | Audio | Title) in a segment. Each function is specified by a given attribute on the material element, and they are all inter-related.

These functions *only come into play* when the duration of the material is less than the duration of the segment.

These functions provide a flexible method to define the behavior of overlay material so that it is consistently applied, regardless of the duration of the underlying content. For example, you can consistently display a branding bug (or an advisory bug) on underlying video of arbitrary duration. Or, you can squeeze back a credit roll, and cause them to always end 2 seconds before the end of the promo, regardless of their length.

Note: In practice, most of these situations involve visual material; audio is treated similarly of course.

There are several aspects that come into play when modifying material temporally. Each aspect is identified by a specific attribute you can add to the element:

- *Specifying the Duration of Material on the Timeline*—How long to play the material
- *Offsetting Material on the Timeline (offset)*—When to start (or stop) the playout of the material
- *Aligning Material on the Timeline (align)*—Which point to anchor so the other end floats depending on duration
- *Adjusting Material on the Timeline*—What to adjust—the body or the edge (using *adjust*)
- *Modifying the Display of Material on the Timeline (fill)*—What portion of the material to replay

Specifying the Duration of Material on the Timeline

The duration of temporal material (video and audio) is specified by the length of the material in the instance. Thus, the duration attribute is only used for material that does not have a native length, or duration: Image, Advisory, Canvas, and Title.

Duration specifies the amount of time to display the material, and the default value is 0.

The time can be specified using a time, or a timecode, with these patterns: *HH:MM:SS* | *HH:MM:SS.sss* | *HH:MM:SS:FF* | *HH:MM:SS:FF@FPS*.

Time is measured on a 24-hour clock. Time may include milliseconds (*sss*); frames (*:|:FF*) as standard (*:*) or drop-frame (*:*), and optional frame rate (*@FPS*), as an integer, decimal number, or rational (for example, 30000/1001).

Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.

Timecode references are applied to the timecode track specified in the associated Source element's file.

A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.

Examples:

```
<Advisory duration = "00:00:05.500"... />
<Advisory duration = "00:00:05;14@29.97"... /
```

Offsetting Material on the Timeline (offset)

By default, all material (Image, Advisory, Canvas, Title, Video, and Audio) instances in a segment begin at the same time. To change the time when a given instance begins to play, add an offset attribute, and supply a time value.

For example: `offset="00:00:10.000"` will cause this material to start playing 10 seconds after the segment begins, provided that the align is set to Head.

Note: You can use a time or timecode value, based on the media and your requirements. For implementation details, see the offset attribute description in any material element.

Aligning Material on the Timeline (align)

When the duration of the material is less than the duration of the segment, you can add an align attribute to your material element, and assign the appropriate keyword to pin (anchor) the material to the timeline of the segment. When you pin or anchor a point, you prevent it from being altered, when the length of the segment changes from job to job. By default, align = head.

head (default)—Anchors the beginning of the material to the beginning of the segment.

tail—Anchors the end of the material to the end of the segment.

both—Anchors the center of the material at the center of the segment, so that the Head and Tail are equidistant from the beginning and end of the segment.

Note: If an offset is used in this situation, it is also applied to both ends.

Tail Alignment Example

In this example, you are creating a one minute What's Next promo on a movie channel, and you want to show a preview which may run up to 30 seconds. You always want the preview to end two seconds before the end of the promo, leaving the beginning of the clip (regardless of its length) unspecified:

Example: `<Video align="tail" offset="00:10:02.000"... />`

Using this code, the video is aligned at the end (the tail) of the clip, and it is offset from the total segment duration by 2 seconds from the end. The beginning timecode is not specified: it is calculated by the Conform action when the job runs, based on the exact length of the clip.

Both Alignment Example

Continuing the What's Next promo example, you need to add an Advisory to the promo. You always want advisory bug to start two seconds after the start, and 2 seconds before the end of the promo.

Example: `<Advisory align="both" offset="00:10:02.000"... />`

Using this code, the advisory bug is aligned at both ends of the clip, and it is offset from the total segment duration by 2 seconds on each the end.

Adjusting Material on the Timeline

Use the `adjust` attribute to specify which portion of the material should be extended, using the keywords: *edge* | *body*. For example, using the *adjust* attribute, you can consistently display a branding bug on underlying video of arbitrary duration.

edge (default)—The material is adjusted by extending the edge that is *not* identified by the *align* property and is applied to the duration of the head and tail of the material.

body—The material is adjusted by extending the body of the material. It is applied to the end of the body that is *not* identified by the *align* property.

When you adjust an item, you indicate whether you want to extend the duration of the body, or of the edge. When you adjust the body, the head and tail duration remain fixed; when you adjust the edge, the edge that is not anchored (see Align) is the one that is extended.

Example

In this example, 4 PNG files are played out one after the other due to the offset attribute. As a PNG (with no intrinsic duration), you specify how long to play the image (in this case, 9 seconds), and what part of the material to adjust—the body.

```
<Segment>
  <Image align="head" adjust="body" fill="loop" layer="2"
location="//share/path/SAV_1700_1_0001_curCon.png" frames="1"
duration="00:00:09.000" layout="stretch" />
  <Image align="head" adjust="body" fill="loop" layer="3"
location="//share/path/SAV_1700_2_0001_curCon.png" frames="1"
duration="00:00:09.000" layout="stretch" offset="00:00:10.000"/>
  <Image align="head" adjust="body" fill="loop" layer="4"
location="//share/path/SAV_1700_3_0001_iconmap.png" frames="1"
duration="00:00:09.000" layout="stretch" offset="00:00:20.000"/>
  <Image align="head" adjust="body" fill="loop" layer="5"
location="//share/path/SAV_1700_4_0001_fiveday.png" frames="1"
duration="00:00:09.000" layout="stretch" offset="00:00:30.000"/>
```

</Segment>

Modifying the Display of Material on the Timeline (fill)

By default, material is only played once per instance in a segment. When the length of the material is less than the duration of the segment, the use of the *fill* attribute determines if and how material continues to play until the end of the segment.

none (default)—do not extend the material. (In theory, it actually plays to the end, as empty: no visual or auditory content.)

hold—continue playing the first or last frame to play the material the entire duration of the segment. If *align*="head", the last (or only) frame is repeated. If *align*="tail", the first (or only) frame is repeated.

loop—play the media repeatedly for the duration of the segment.

Of course, for temporal (multi-frame) media (video and audio), this will play out differently than for non-temporal (single-frame) media (Image, Advisory, Title, and Canvas).

Composition Cookbook

The purpose of this chapter is to illustrate practical ways to approach and implement many aspects of compositions.

Note: If you have comments or suggestions about improving this guide—or if you've discovered an error or omission, please email us at techwriter@telestream.net.

- [Adding Titles](#)
- [Image Scaling and Placement](#)
- [Mixing Audio](#)
- [Routing Audio Without Decoding](#)
- [Adding a Content Advisory](#)
- [Creating a Squeezeback](#)
- [Downmixing 5.1 Dolby E to 2 Channel Stereo](#)
- [Fade to Black](#)

Note: All of the examples in this chapter are complete compositions. To test them, you can copy and paste the example into a text editor. Change the references to the media files you are supplying, and save the file as XML. Then, submit it to a Post Producer conforming workflow ([Simple Conforming Workflow](#)) in Vantage, using Workflow Designer.

Adding Titles

Titles are a frequent element of many promos. These examples show you some typical applications of titles you can adapt to your own requirements.

For implementation details, see [Title](#), [Area](#), and [Shadow](#).

Adding a Basic, Static Title

This example illustrates how to set the area exactly where you want it displayed, using the Area element, set off the left edge 10%, slightly taller than the font for good measure. For this example, the background rectangle of the text (not the Area—just the text rendering rectangle) is set to blue so it stands out.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <!-- Basic Title DUR=NA, AR=16x9, res. 1920x1080 -->
  <!-- Input: 1080i -->
  <!-- Output: 1080i x264 -->
  <!-- Conforming Workflow: 1080 + proxy -->
  <Source identifier="1">
    <File location="//share/path/1_A2_Show_ONLY_NO-GFX2.mov" />
  </Source>
  <Sequence layer="0">
    <Segment>
      <Title align="head" adjust="edge" fill="none" layer="1"
duration="00:00:006.00" font="Helvetica" size="50pt" style="italic"
weight="bold" foreground-color="bisque" background-color="blue"
wrap="false" horizontal-align="center" vertical-align="middle"
overflow="overflow" layout="none">
        <Area top="10%" left="20%" bottom="60px" right="80%" />
        <Shadow color="gold" softness="10%" vertical-offset="6%"
horizontal-offset="6%" />
      </Title>
      <Video align="head" adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
  </Sequence>
</Composition>
```

Sliding a Title On or Off Screen

This example illustrates how to set the title to an offscreen value (using a negative percent value), and then move the title onscreen, using a head/body/tail Translation. Also note the use of the motion-blur to smooth the motion.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <!-- Up Next Promo DUR=1 min, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
  <!-- All PNG files: 600 x 420 -->
  <!-- Input: 1080i -->
```




```
<!-- Output: 1080i x264 -->
<Source identifier="1">
  <File location="//share/path/1_A2_Show_ONLY_NO-GFX2.mov" />
</Source>
<Sequence layer="0">
  <Segment>
    <Title align="head" adjust="edge" fill="none" layer="2"
duration="00:00:006.00" font="Helvetica" size="50pt" style="italic"
weight="bold" foreground-color="bisque" background-color="blue"
wrap="false" horizontal-align="left" vertical-align="middle"
overflow="overflow" layout="stretch">
      <Area top="0%" left="20%" bottom="90%" right="80%" />
      <Shadow color="gold" softness="10%" vertical-offset="6%" horizontal-
offset="6%" />
    </Title>
    Post Producer Titles
  <Head>
    <Opacity level="100%" />
    <Translation x="00%" y="-20%" />
    <Fade duration="00:00:02.000" motion-blur="3" />
  </Head>
  <Body>
    <Translation x="0%" y="50%" />
  </Body>
</Title>
  <Video align="head" adjust="edge" fill="none" source="1" layer="0">
    <Head>
      <Edit time="00:00:00.000" />
    </Head>
    <Tail>
      <Edit time="00:00:07.000" />
    </Tail>
  </Video>
</Segment>
</Sequence>
</Composition>
```

Image Scaling and Placement

This image example scales and places an image in the lower right-hand portion of the frame, and aligns it in time with an underlying video sequence.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share\path\Mystic River Seg 1.mov" />
  </Source>
  <Source identifier="2">
    <File location="//share\path\vod\FTD_Start_at_16.21.mov" />
    <Mask left="12.5%" right="87.5%" top="0%" bottom="100%" />
  </Source>
  <Sequence layer="0">
    <Segment>
      <Image align="both" adjust="body" fill="loop" layer="1"
location="//share\path\vod\Vantage-logo_small.png" frames="1"
layout="none" offset="00:00:02.000">
        <Head>
          <Scaling x="30%" y="30%" />
          <Translation x="80%" y="70%" />
          <Opacity level="0%" />
          <Fade duration="00:00:01.000" />
        </Head>
        <Body>
          <Scaling x="30%" y="30%" />
          <Translation x="80%" y="70%" />
          <Opacity level="100%" />
        </Body>
        <Tail>
          <Fade duration="00:00:01.000" />
          <Scaling x="30%" y="30%" />
          <Translation x="80%" y="70%" />
          <Opacity level="0%" />
        </Tail>
      </Image>
    </Segment>
    <Segment>
      <Video source="1" >
        <Tail>
          <Edit time="00:00:8.933" />
        </Tail>
      </Video>
    </Segment>
    <Segment>
      <Video source="2" >
        <Tail>
          <Edit time="00:00:10.330" />
        </Tail>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

```
        </Sequence>  
      </Segment>  
    </Sequence>  
  </Composition>
```

Mixing Audio

Post Producer enables you to perform audio mixing—mapping source to output channels—directly in a [Source](#), where you identify your input media files. You use the [Mix](#) for mapping audio tracks from the input file in the associated [File](#) to the output. During mixing, Vantage decodes and re-encodes the audio.

You can approach mixing in two different ways: adding a Mix for every input or output channel, or by combining common channels, using the *source* and *target* attributes.

Creating a Mixed Stereo Pair

Both of these compositions combine a stereo pair with a mono track to make a mixed stereo pair. Here are two CML samples that perform the same task:

```
<!-- Up Next Promo DUR=1 min, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
<!-- All PNG files: 600 x 420 -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conforming Workflow: 1080 VOD -->
<Source identifier="2"> <!-- English -->
  File location="//share/path/Master Oh_Susanna.mov"/>
  <Mix source="1 2" target="1 2" level=".707%"/>
  <Mix source="3" target="1" level=".9"/>
  <Mix source="3" target="2" level=".9"/>
</Source>
```

OR

```
<Source identifier="2"> <!-- English -->
  File location="//share/path/Master Oh_Susanna.mov"/>
  <Mix source="1 2" target="1 2" level=".707%"/>
  <Mix source="3" target="1 2" level=".9"/>
</Source>
```

In the first example, source 3 is mixed to targets 1 and 2 using two Mix elements. In the second example, the mix is performed with a single Mix, by specifying targets 1 and 2 in combination.

Creating Dolby E 5.1 from Compressed Dolby E

This example illustrates how you can map Dolby E 5.1 into your output from a compressed dolby E stereo pair, to make 6 surround sound channels.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition version="1.0" xmlns="net.telestream.composition">
<!-- Create Dolby E 5.1 DUR=1m AR=16x9 res. 1920x1080 DolbyE 5.1+2 -->
<!-- All PNG files: 600 x 420 -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conforming Workflow: 1080 VOD -->
  <Source identifier="1">
```

```
File location="\\share\path\Canon_in_D_DolbyE_Stereo.mov"/>
<Mix source="1 2" target="1 2" level="0" phase="0°" />
<DolbyE channels="1 2" program="5.1">
  <Mix source="1" target="1" level="100%" phase="0°" />
  <Mix source="2" target="2" level="100%" phase="0°" />
  <Mix source="3" target="1 2" level="0.707107" phase="0°" />
  <Mix source="4" target="1 2" level="0.707107" phase="0°" />
  <Mix source="5" target="1" level="0.87178" phase="-90°" />
  <Mix source="5" target="2" level="0.489898" phase="-90°" />
  <Mix source="6" target="1" level="0.489898" phase="90°" />
  <Mix source="6" target="2" level="0.87178" phase="90°" />
</DolbyE>
</Source>
<Sequence
```

Routing Audio Without Decoding

Routing (via the [Route](#)) allows you to move audio from tracks in the source to other tracks in the output, without decoding the audio. It simply passes it through. You can not mix audio channels, or change audio levels using the Route element.

Note: Mixing channels or adjusting audio levels requires decoding—use the [Mix](#) instead of the Route element.

Passing Dolby E to Output

A common use of the Route element is to pass compressed Dolby E stereo to the output without decoding it, as shown in this example.

```
<!-- Up Next Promo DUR=1 min, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
<!-- All PNG files: 600 x 420 -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conforming Workflow: 1080 VOD -->
<Source identifier="2"> <!-- English -->
  <File location="//share/path/Master_DolbyE_Stereo.mov"/>
  <Route source="1 2" target="1 2"/>
</Source>
```

Passing Pro Tools Audio to Output

If audio is fully conformed in Pro Tools and it should be left untouched, you can use Route to populate the output.

Adding a Content Advisory

You add a content advisory (an Advisory element) to a segment to provide USTV, MPAA, CELR, or CFLR V-chip encoding to a segment, and optionally provide a graphic on your video, as shown below.

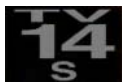
Note: For complete details on implementing and configuring a content advisory, see [Advisory](#) in [Composition Markup Language Reference](#).

Figure 15. Typical Content Advisory Graphic Overlay



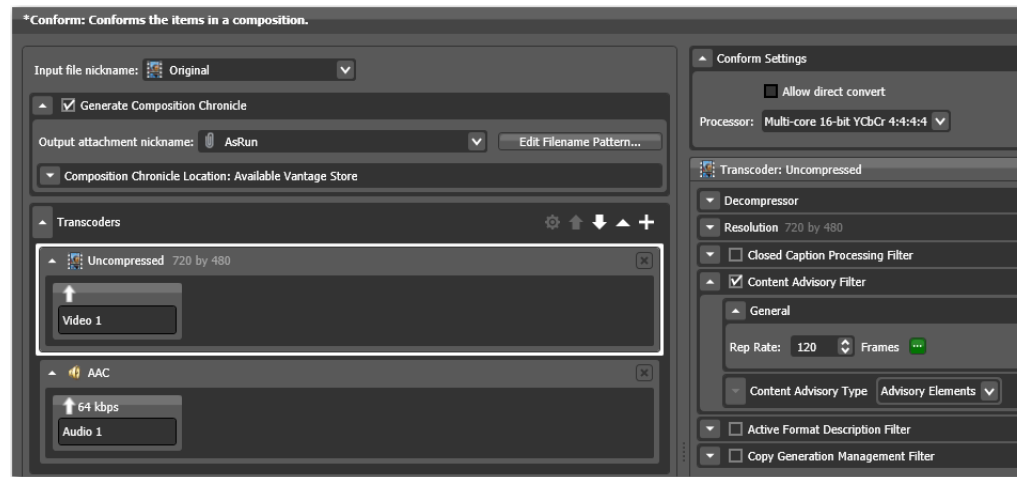
When using the USTV rating system, Post Producer allows you to use default icons, as shown below. The icon displayed is determined by the rating you specify using the *rating* attribute. You can override the default icons by supplying your own, with a *location* attribute. When using other rating systems, you must supply your own icons.

Figure 16. Typical Advisory Image for USTV



When you add a content advisory, you also need to enable the Content Advisory filter in the Transcoder section of the Conform action, so that the rating information you've configured in the Advisory element is inserted properly into the VANC.

Figure 17. Enabling the Content Advisory Filter in the Conform Action



Example

This example, creates a promo for a movie with a USTV rating of TV-14, FV. It uses the default icon for the rating supplied, and fades in and displays it in the upper left corner for 5 seconds.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
<!-- Up Next Promo DUR=1 min, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
<!-- All PNG files: 600 x 420 -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conforming Workflow: 1080 VOD -->
  <Source identifier="1">
    <File location="\media\My_TV_14_S_Show_Promo.mov" />
  </Source>
  <Sequence layer="1">
    <Segment>
      <Advisory align="head" adjust="body" fill="none" layer="1"
duration="00:00:05.000" type="USTV" rating="TV-14 FV">
        <Head>
          <Fade duration="00:00:02.000" />
        </Head>
        <Body>
          <Translation x="5%" y="5%" />
        </Body>
      </Advisory>
    </Segment>
  </Sequence>
</Composition>
```

Because the default *layer* is 0, a layer attribute is added to the Advisory. To display the icon on top of all other visual material, it should be the highest numbered of all the

material in the segment. For the same reason, you need to add a *duration*, so that the advisory logo displays for the time you specify, in the format HH:MM:SS.SSS.

You can supply a single rating value, or multiple rating values, by separating each value in the string with a space. For example, "TV-14 FV" represents *Parent Strongly Cautioned + Fantasy Violence*. Post Producer displays the proper icon for your rating. If it is not a logical combination, no icon is displayed.

Creating a Squeezeback

Here's how to create a squeezeback from two videos—where one video fades in, while another fades out. To create the squeezeback, you create 3 segments (which play sequentially, within the sequence), and specify playback attributes to create this clever squeezeback, regardless of the length of either clip.

Example

In this example, the first segment plays video 1 and 2 concurrently: video 1 is aligned at the head (by default), and fades out at the end for 1 second (due to the Tail Fade duration of 1 second).

Video 2 on the other hand, is aligned at the tail (at the end of the longest running segment in the sequence), and the mark out point (Tail Edit time) is 1 second—meaning, that for this instance of use, the file is effectively 1 second long. So, it begins its fade 1 second prior to the end of the segment and runs for 1 second.

Finally, in the second segment, video 2 is played again. This time, it is aligned at the head by default, and the mark in point (Head Edit time) is also 1 second—to start immediately where it left off in the first segment, and it plays to the end of its length.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition version="1.0" xmlns="Telestream.Soa.Facility.Playlist">
<!-- Up Next Promo DUR=1 min, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
<!-- All PNG files: 600 x 420 -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conforming Workflow: 1080 VOD -->
  <Source identifier="1">
    <File location="D:\PP_Media\Source_video_1.mov" />
  <Source identifier="2">
    <File location="D:\PP_Media\Source_video_2.mov" />
  </Source>
</Source>
<Sequence layer="1">
  <Segment>
\\ video 1 plays to the end and fades out for 1 second.
    <Video source="1" layer="0">
      <Tail>
        <Fade duration="00:01:00">
      </Tail>
    </Video>
\\ video 2 is aligned at the tail - it fades in and plays for 1 second.
    <Video source="2" layer="1" align="tail">
      <Head>
        <Fade duration="00:01:00">
      </Head>
      <Tail>
        <Edit time="00:00:01:00">
      </Tail>
```

```
        </Video>
      </Segment>
    <Segment>
\\ video 2 now plays from the 1 second mark in to the end.
      <Video source="2">
        <Head>
          <Edit time="00:00:01:00">
        </Head>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

Downmixing 5.1 Dolby E to 2 Channel Stereo

The composition shown below presents a single source media file containing a Dolby E encoded surround sound program. The source element identifies the location of the media file. In this example the source contains four PCM audio channels (1-4). The dolby element indicates that channels 1-2 carry a 5.1 surround sound program.

The first mix element mutes the raw Dolby E encoded channels (1-2). The remaining elements down mix the surround program (source channels 1-6) to stereo (target channels 1-2) using the standard Dolby Pro Logic matrix.

```
<?xml version="1.0" encoding="utf-8"?>
<Playlist version="1.0" xmlns="net.telestream.playlist">
  <Source identifier="1">
    <File location="Source.mov" />
    <Mix source="1 2" target="1 2" level="0" phase="0°" />
    <DolbyE channels="1 2" program="5.1">
      <Mix source="3" target="1 2" level="0.707107" phase="0°" />
      <Mix source="4" target="1 2" level="0.707107" phase="0°" />
      <Mix source="5" target="1" level="0.87178" phase="-90°" />
      <Mix source="5" target="2" level="0.489898" phase="-90°" />
      <Mix source="6" target="1" level="0.489898" phase="90°" />
      <Mix source="6" target="2" level="0.87178" phase="90°" />
    </DolbyE>
  </Source>
```

Fade to Black

This example shows you how to fade to black.

Example

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
<!-- Fade to Black DUR=1 min, AR=16x9, res. 1920x1080, DolbyE 5.1+2 -->
<!-- All PNG files: 600 x 420 -->
<!-- Input: 1080i -->
<!-- Output: 1080i x264 -->
<!-- Conforming Workflow: 1080 VOD -->
  <Source identifier="1">
    <File location="//share\<folder>\fade_to_black.mov" />
  </Source>
  <Sequence layer="0">
    <Segment>
      <Canvas align="head" adjust="body" duration="00:00:10.00" foreground-
color="black" background-color="black" layer="1" />
      <Video source="1" layer="2" >
        <Head>
          <Opacity level="0%" />
          <Fade duration="00:00:02.00" />
          <Edit time="00:00:00.00"/>
        </Head>
        <Body>
          <Opacity level="100%" />
        </Body>
        <Tail>
          <Opacity level="0%" />
          <Fade duration="00:00:02.00" />
          <Edit time="00:00:10.00"/>
        </Tail>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```


Post Producer Prototype Workflows

This chapter provides examples of workflows that are typically used in Post Producer applications.

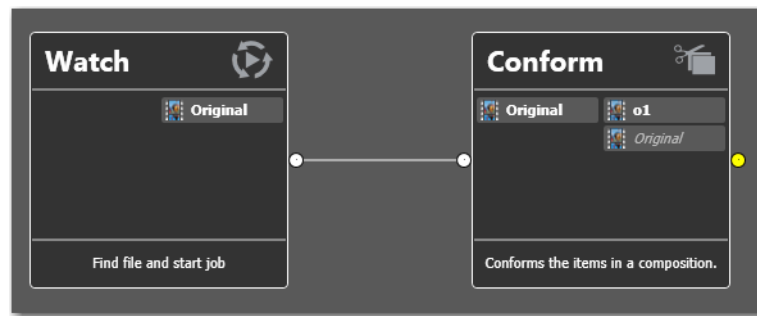
- [Simple Conforming Workflow](#)
- [Compose and Conform Workflow](#)
- [BlackArrow SCTE-35 Ad Insertion Workflow](#)

Simple Conforming Workflow

This workflow is a basic Post Producer conform workflow, where you submit a composition file for encoding.

You can use this basic workflow as the basis of all other conforming workflows; you add and configure other actions to create a workflow that meets your operational requirements.

Figure 18. Basic Post Producer Conforming Workflow



This workflow utilizes the following actions to perform these tasks:

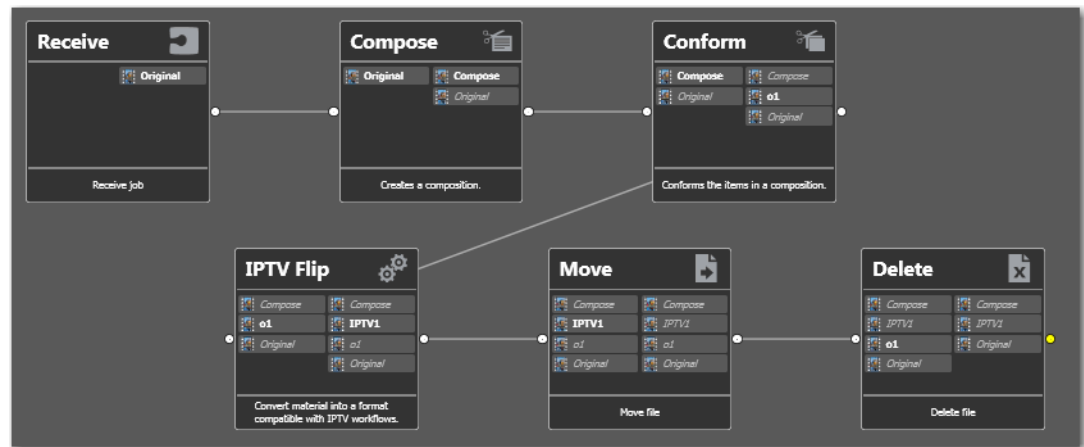
Watch—The Watch action starts jobs when you submit a composition file for processing. The Watch action can be configured to watch a wide variety of systems for files; commonly a conforming workflow would monitor a Windows computer or share for CML files. Under the File Match Pattern, you could specify *.cml | *.xml to only pick up files with these extensions. You can replace Watch with a Receive action, so that you can submit jobs from other workflows, programs, or systems using the Vantage SDK.

Conform—The [Conform Action](#) processes the CML file, generating an MPEG, QuickTime, or TIFO file from six different supported video formats and 3 audio formats (See [Post Producer Prototype Workflows](#)). If you require your output in other formats, just add a Flip, IPTV VOD, or Multiscreen Flip action after the Conform action to encode the output into the required format.

Compose and Conform Workflow

This prototypical workflow ingests an Anystream AN2 file and converts it to CML, which is which is processed by the [Conform Action](#). The output is transcoded into Manzanita format, and moved to a production server.

Figure 19. Compose and Conform Workflow Diagram



This workflow utilizes the following actions to perform these tasks:

Receive—The Receive action starts jobs when you submit an AN2 file for processing. The Receive action is used so that you can submit jobs from other workflows, programs, or systems using the Vantage SDK. If this type of submission is not required, you can replace this with a Watch action.

Compose—The [Compose action](#) automatically converts the AN2 file (or other media descriptor files, such as TSEDL) into a CML file for processing by the Conform action. File names are unchanged, so the Edit service that processes the CML file must have access to the input media files specified in the AN2 file.

Conform—The [Conform Action](#) processes the auto-generated CML file, from the Compose action, generating an MPEG4 file from the MPEG video with AAC audio.

IPTV Flip—The IPTV Flip action wraps the MPEG4 media in a Manzanita wrapper.

Move—The Move action moves the file to an appropriate production server.

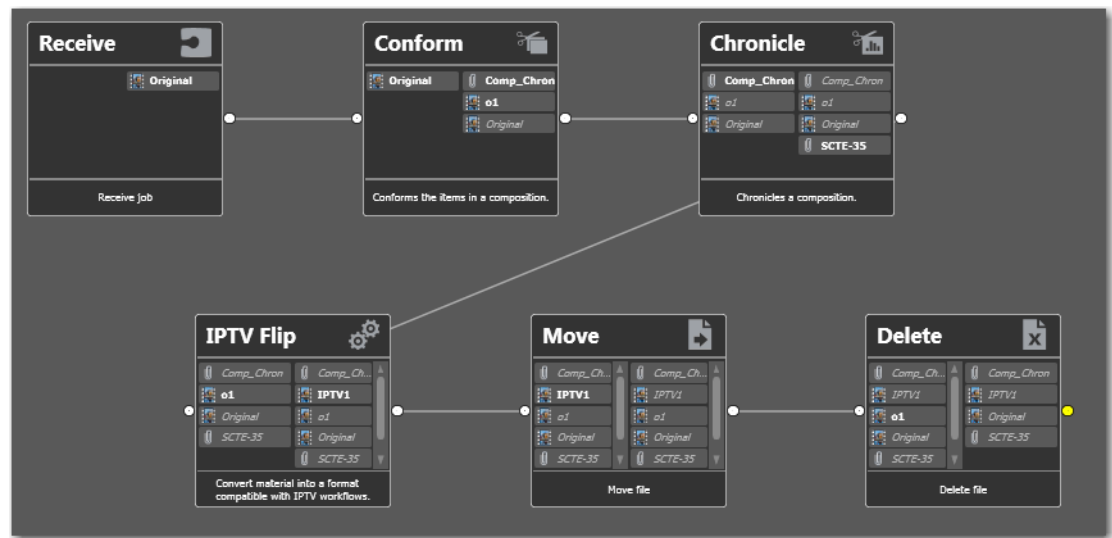
Delete—After the Move action completes, the Delete action deletes the mezzanine MPEG4 file generated by the Conform action.

BlackArrow SCTE-35 Ad Insertion Workflow

This prototypical workflow ingests a Composition file, which is which is processed by the Conform action configured to generate an As Run CML file. The As Run CML is processed by a Chronicle action to create an SCTE-35 compatible file, for delivery to a BlackArrow ad insertion system.

The output is transcoded into Manzanita format, and moved to a production server.

Figure 20. BlackArrow SCTE-35 Ad Insertion Workflow



This workflow implements the following actions to perform these tasks:

Receive—The Receive action starts jobs when you submit a composition file for processing. The Receive action is used so that you can submit jobs from other workflows, programs, or systems using the Vantage SDK. If this type of submission is not required, you can replace this with a Watch action.

Conform—The [Conform Action](#) processes the ingested CML file, generating an MPEG4 file from the MPEG video with AAC audio. It is configured to create an As Run CML file, for use in Black Arrow SCTE-35 ad insertion systems.

Chronicle—The [Chronicle action](#) uses the incoming As Run CML file to create an SCTE-35 file. The action is configured to write the file to a Black Arrow folder for pickup by the Black Arrow system.

IPTV Flip—The IPTV Flip action wraps the MPEG4 media in a Manzanita wrapper,

Move—The Move action moves the file to an appropriate production server.

Delete—After the Move action completes, the Delete action deletes the mezzanine MPEG4 file generated by the Conform action.

Composition Markup Language Reference

The purpose of this chapter is to describe each element in the Composition Markup Language, and its attributes. The elements are listed below in alphabetic order. On the next page is a [CML Hierarchy Map](#), showing each element in relation to other elements.

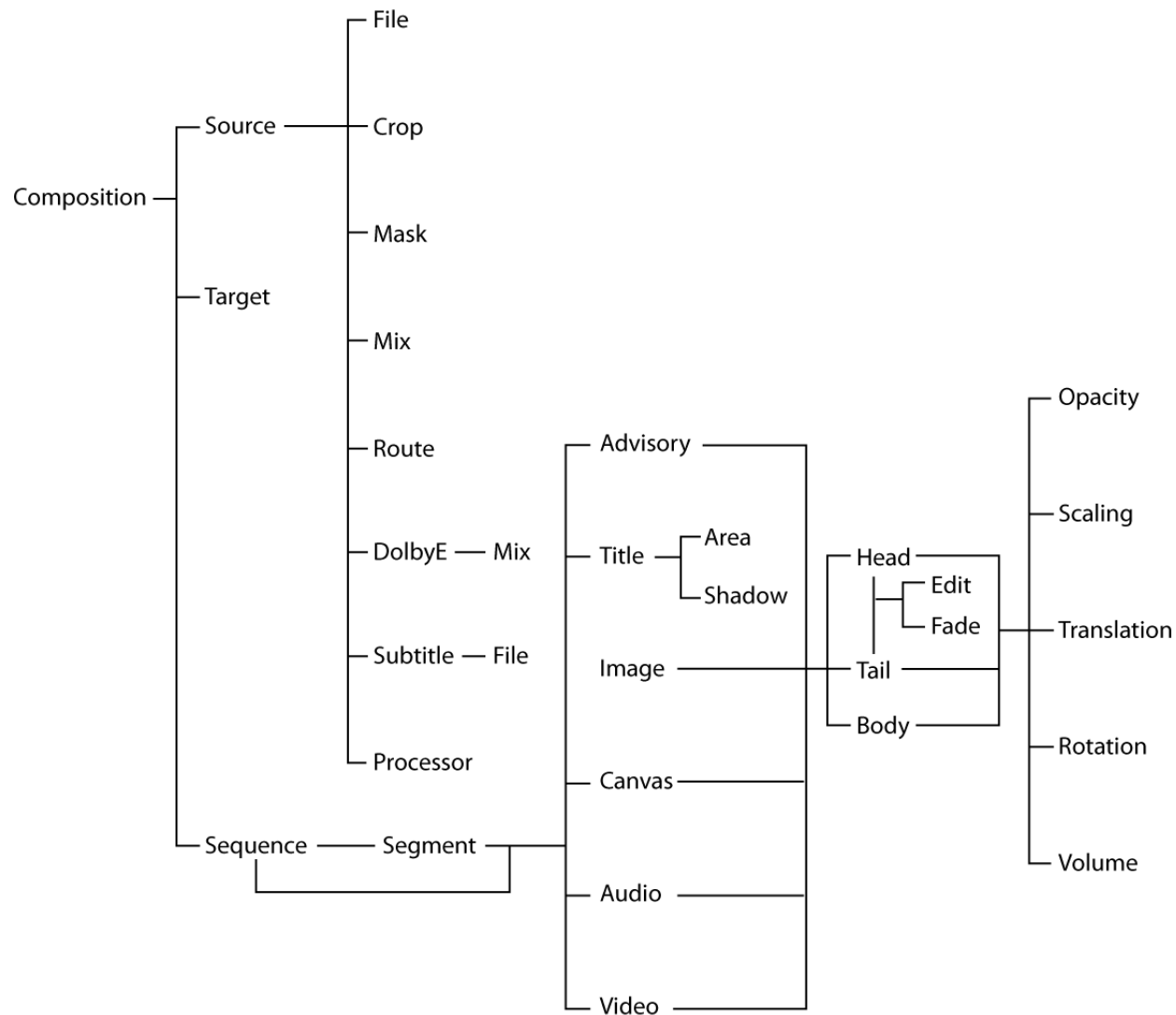
Note: The example CML code snippets in this reference are not intended for execution; they are intended for study and to illustrate proper syntax.

In general, dimensions should be designated as pixels, percent, or a ratio. Volumes should be specified in decibels; rotations in degrees, text sizes—points. CML is designed to be rational and flexible. Pixels are best with for specific input and output sizes; percent (or ratio) are helpful for processing media of different sizes.

File references are always shown as a UNC path example: `\\share\path\filename.ext`; to reinforce the use of shares to access files in a Vantage domain.

- | | |
|-------------------------------|-------------------------------|
| ■ Advisory | ■ Processor |
| ■ Area | ■ Rotation |
| ■ Audio | ■ Route |
| ■ Body | ■ Scaling |
| ■ Canvas | ■ Segment |
| ■ Composition | ■ Sequence |
| ■ Crop | ■ Shadow |
| ■ DolbyE | ■ Source |
| ■ Edit | ■ Subtitle |
| ■ Fade | ■ Tail |
| ■ File | ■ Target |
| ■ Head | ■ Title |
| ■ Image | ■ Translation |
| ■ Mask | ■ Video |
| ■ Mix | ■ Volume |
| ■ Opacity | |

CML Hierarchy Map



Advisory

To display an advisory, you add an optional Advisory element to a [Segment](#). The Advisory element defines the parental advisory material for the segment. The content advisory value is encoded into the video and is interpreted by the V-chip in the television or set top box. The advisory graphic must be placed and scaled on each segment where you want it to display. (Typically, the first segment displays a larger advisory bug than the others.)

The Advisory element adds two types of information:

- A graphic overlay that displays the content rating. To display the graphic, specify the time in the *duration* attribute. The size, position and presentation of the graphic may be controlled using Effect elements.

For USTV only, system-supplied images (the larger set) are displayed, based on the ratings you supply. These may be replaced (or supplied when using other rating systems) using the *location* attribute.

- Metadata that identifies the rating system, content rating, and parental advisories. It is typically transmitted in the vertical blanking interval (using XDS protocol).

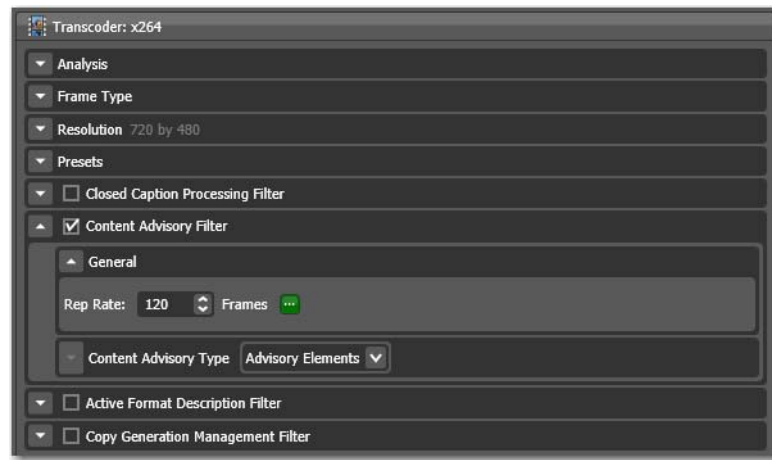
The Advisory element sets the V-chip encoding value for the entire composition, unless additional Advisory elements are added in later segments. Each succeeding Advisory sets the V-chip encoding for the remainder of the composition.

Note: To correctly insert Advisory metadata in the video VBI or VANC, you must enable the Content Advisory Filter in the Transcoder section of the Conform action, and configure the repeat rate. The value you put in the advisory element will be repeated at that rate for the entire file unless another advisory element is encountered in a segment, in which case the value that is repeated will be changed to the new value.

Enabling the Advisory Filter in Workflow Designer

To display the Content Advisory Filter editor panel, open the Conform action's inspector. Add a video transcoder if you haven't already done so, and click on the transcoder panel to display its controls on the right.

Figure 21. Align and Adjust Attributes Work Together on Material Expansion



Click to open the Content Advisory Filter editor panel and set the repeat rate, as shown.

Note: For a practical example of implementing an advisory, see [Adding a Content Advisory](#).

Example

This snippet illustrates a typical Advisory element in a segment:

```
<Segment>
  <Advisory align="head" adjust="body" fill="none" layer="1"
duration="00:00:05.000" type="USTV" rating="TV-14">
</Segment>
```

Elements (CML Hierarchy Map)

One each of these optional elements may be added to an Advisory element:

- [Head](#)
- [Body](#)
- [Tail](#)

Attributes

These attributes may be applied to an Advisory element. The *rating* and *advisory* attributes are required.

Note: The *adjust*, *align*, and *fill* attributes are closely related. Together, they specify how the material is arranged along the timeline in the segment.

Table 4. Advisory Attributes

Name	Description and Values
adjust (optional)	<p>Applies to the graphic—specifies which portion of the advisory graphic may be extended within the segment when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>edge</i> <i>body</i></p> <p>Example: <code><Advisory adjust = "body"... /></code></p> <p><i>edge</i> (default)—The material is adjusted by extending the edge that is <i>not</i> identified by the <i>align</i> property and is applied to the duration of the head and tail of the material.</p> <p><i>body</i>—The material is adjusted by extending the body of the material. It is applied to the end of the body that is <i>not</i> identified by the <i>align</i> property.</p>
align (optional)	<p>Specifies which end (or both) to (anchor) the material to the segment, when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>head</i> <i>tail</i> <i>both</i>.</p> <p>Example: <code><Advisory align = "tail"... /></code></p> <p><i>head</i> (default)—Anchors the beginning of the material to the beginning of the segment.</p> <p><i>tail</i>—Anchors the end of the material to the end of the segment.</p> <p><i>both</i>—Anchors the center of the material at the center of the segment, so that the Head and Tail are equi-distant from the beginning and end of the segment.</p> <p>Note: If an offset is used in this situation, it is also applied to both ends.</p>

Table 4. Advisory Attributes

Name	Description and Values
duration (optional)	<p>The amount of time to display the graphic. By default, duration is 0.</p> <p>Time values: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS:FF@FPS</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i>;<i>FF</i>) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source. e</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Examples:</p> <pre><Advisory duration = "00:00:05.500"... /> <Advisory duration = "00:00:05;14@29.97"... /</pre>
fill (optional)	<p>When the length of the material is less than the length of the segment, the fill attribute determines if and how the item continues to play:</p> <p>Keywords: <i>none</i> <i>hold</i> <i>loop</i></p> <p>Example: <code><Advisory fill = "loop"... /></code></p> <p>There are 3 things to consider when modifying the material:</p> <ol style="list-style-type: none"> 1. What to adjust—the body or the edge (using <i>adjust</i>) 2. Which end to adjust from (based on <i>align</i> attribute) 3. How to adjust—hold, loop, none (based on <i>fill</i> attribute) <p><i>none</i> (default)—extend with empty media (no visual or auditory material). From a practical perspective, no extension.</p> <p><i>hold</i>—continue playing the first or last frame the entire duration of the segment. If <i>align="head"</i>, the last frame is repeated. If <i>align="tail"</i>, the first frame is repeated.</p> <p><i>loop</i>—play the media repeatedly for the duration of the segment.</p>

Table 4. Advisory Attributes

Name	Description and Values
layer (optional)	<p>An integer value describing the ordinal position in the composite of all material in this segment. Minimum value: 0, default: 0.</p> <p>Example: <code><Advisory layer = "10"... /></code></p> <p>Material with a higher layer value obscures material with a lower layer value in the same segment.</p>
location (optional)	<p>Specifies a fully-qualified path to the advisory image, including file name. The location may be specified as:</p> <ul style="list-style-type: none">• <i>Windows</i>—D:\pathname\filename.ext• <i>UNC</i>—\\share\path\filename.ext• <i>URL</i>—http://SierraStudio.com/media/Vane_logo.png <p>Example: <code><Advisory location = "\\Pluto\media\RatingSystems\USTV\PG13.png" /></code></p> <p>The location must be accessible to the Vantage Playlist service which is executing the Conform action.</p> <p>For USTV ratings, a default image is supplied, based on the rating(s) you specify. For all other rating systems (or to override the USTV rating system graphics), supply the fully-qualified path to the image.</p> <p>The image may be in PNG, BMP, or JPEG format.</p>

Table 4. Advisory Attributes

Name	Description and Values
offset (optional)	<p>Specifies a time value which changes the location of the beginning or end of the material, relative to the segment.</p> <p>The offset is applied to the end of the content that is specified by the align property. If align="both" the offset is applied at the defined value to both the head and tail of the material.</p> <p>Default: 0.</p> <p>Time Value: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS;FF</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i>FF) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Example: <code><Advisory duration = "00:00:05"... /></code></p>
rating (required)	<p>Specifies the rating, plus the reason or reasons.</p> <p>Keywords: One or more keywords from Table 7, Rating Reasons, separated by spaces.</p> <p>Example:</p> <pre><Advisory type="USTV" rating="PG-13 V" /></pre> <p>The content rating. You may specify multiple ratings and reasons. If the combination is not logical, no file is displayed.</p>
type (required)	<p>Specifies the rating body.</p> <p>Keyword Values. One keyword from Table 5, Rating System Keywords (for type attribute). Default: MPAA. Optionally, add one or more rating reasons (Table 7, Rating Reasons). Default: Not Rated—specify as an empty string.</p> <p>Example:</p> <pre><Advisory type="USTV" rating="PG-13 V" /></pre>

This table provides the string value to use in the *type* attribute to identify the rating system you are using.

Table 5. Rating System Keywords (for *type* attribute)

Rating System Keyword	Description
CELR	Canada English Language Rating
CFLR	Canada French Language Rating
MPAA	Motion Picture Association of America
USTV	U.S. TV Parental Guidelines

Note: The graphics are supplied automatically for USTV by Advisory material. Graphics for other rating systems will be supplied by the content producer and must be applied using the Image.

This table describes the ratings from various rating systems.

Table 6. Ratings Keywords (for *rating* attribute)

Rating Keyword	Description.
"" (Empty String)	Not rated.
G	General audiences (MPAA, CELR, CFLR)
PG	Parental guidance suggested (MPAA, CELR)
PG-13	Parental strongly cautioned (MPAA)
R	Restricted (MPAA)
NC-17	No one under 17 admitted (MPAA)
X	Explicit (MPAA)
TV-Y	Directed to all children (USTV)
TV-Y7	Directed to children age 7 and up (USTV)
TV-G	General audiences (USTV)
TV-PG	Parental guidance suggested (USTV)
TV-14	Parent strongly cautioned (USTV)
TV-MA	Mature audience only (USTV)

This table describes the rating reasons (which may only apply to one or more ratings). You may use one, or multiple keywords from this list with the Rating value to form a logical rating. For example, *rating = "PG-13 L V"*.

Table 7. Rating Reasons

Keyword	Not rated.
V	Fantasy violence (USTV)
V	Violence (USTV)
L	Adult language (USTV)
S	Sexual situations (USTV)
D	Suggestive dialog (USTV)
E	Exempt (CELR, CFLR)
C	Children (CELR)
C8+	Children age 8 and up (CELR)
8+	Age 8 and up (CFLR)
13+	Age 13 and up (CFLR)
14+	Age 8 and up (CFLR)
16+	Age 8 and up (CFLR)
18+	Age 18 and up (CFLR)

Area

One Area element should be added to a [Title](#) to define the location and size of the rectangle in which to render the title text. Without it, the default area is the same as the output frame, and the title text is rendered in the vertical center of the frame, on the left edge. Though the Area element is optional, in most cases an Area element is a practical necessity.

You can not define an Area off-screen (outside the output frame). To place it offscreen for a slide, add a Head, Body, and Tail and use Translate to animate it.

There are no elements in an Area.

Note: For a practical example of using titles including an Area, see [Adding Titles](#).

See also [Title](#), [Shadow](#).

Example

Here is an example with every attribute, including an Area and Shadow element. The area element is indicated in bold.

```
<Segment>
  <Video adjust="edge" fill="none" source="1" layer="0" />
  <Title align="head" adjust="edge" fill="none" layer="2"
duration="00:00:04.00" offset="00:00:01.000" font="Helvetica"
size="50pt" style="italic" weight="bold" foreground-color="bisque"
background-color="transparent" wrap="false" horizontal-
align="left" vertical-align="middle" overflow="overflow">
    <Area top="70%" left="20%" bottom="90%" right="80%" />
    <Shadow color="gold" softness="10%" vertical-offset="6%"
horizontal-offset="6%" />This is My Story...</Title>
</Segment>
```

Attributes

These attributes may be applied to an Area element. These attributes are relative to origin of output frame at 0,0—the upper, left corner.

The effect of attributes is easier to predict by also locating the title text in the Area at the top left corner using the vertical-align and horizontal-align settings.

You can specify a unit designator of pixels (px) or a ratio (no designator) or percent (%). For example, specifying *bottom=90%* places the bottom edge 10% above the bottom of the frame.

Note: If you use pixels, be aware that the area definition is directly correlated to the output frame—which is defined in the Conform action, beyond the scope of the Composition. Thus, if you design for HD output, but submit the composition to an SD

Conform workflow, the area will not be located correctly. Use percent or ratio to specify the definition relative to the output frame regardless of its size.

If you specify a value without a unit designator, it is treated as a ratio. (The only functional difference between a ratio and a percent is the location of the decimal point.) The ratio may be expressed as an integer, a decimal number, or a rational number {5/10} and must be presented as an expression, in braces. The ratio is the relationship of the edge to the output frame in the appropriate dimension.

Positive (on-screen) and negative (off-screen) values may be used to specify the Area.

Table 8. Area Attributes

Name	Description and Values
bottom (optional)	<p>Specifying the distance from the top of the frame to the bottom edge of the area, as pixels (px), percent (%), or ratio (no unit designator).</p> <p>Default: 1. (The bottom of the output frame.)</p> <p>Example: <code><Area bottom="480px" ... /></code> specifies the bottom edge of the Area as 480 pixels below the top edge of the output frame.</p>
left (optional)	<p>Specifies the distance from the left edge of the frame to the left edge of the area, as pixels (px), percent (%), or ratio (no unit designator).</p> <p>Default: 0. (The left edge of the frame.)</p> <p>Example: <code><Area left="33.3%" ... /></code> specifies the left edge of the Area as 1/3 of the way across the frame from the left edge of the output frame.</p>
right (optional)	<p>Specifies the distance from the left edge of the frame to the right edge of the area, as pixels (px), percent (%), or ratio (no unit designator).</p> <p>Default: 1. (The right edge of the frame.)</p> <p>Example: <code><Area right="100%" ... /></code> specifies the ridge edge of the Area identical to the right edge of the output frame.</p>
top (optional)	<p>Specifying the distance from the top of the frame to the top edge of the area, as pixels (px), percent (%), or ratio (no unit designator).</p> <p>Default: 0. (The top of the frame.)</p> <p>Example: <code><Area top="0" ... /></code> specifies the top edge of the Area identical to the top edge of the output frame.</p>

Audio

You can add one or more Audio elements to a segment. Each Audio element adds audio material from a file identified by a [Source](#).

Example

This is a typical Audio element in a segment.

```
<Audio align="head" adjust="none" fill="none" source="2"
offset="00:00:00.50" >
  <Head>
    <Edit time="00:00:10.000"/>
    <Volume level="0%"/>
    <Fade duration="00:00:01.000"/>
  </Head>
  <Body>
    <Volume level="100%"/>
  </Body>
  <Tail>
    <Volume level="0%"/>
  </Tail>
</Audio>
```

Elements ([CML Hierarchy Map](#))

One each of these optional elements may be added to an Audio element:

- [Head](#)
- [Body](#)
- [Tail](#)

Attributes

These attributes may be applied to an Audio element.

Table 9. Audio Attributes

Name	Description and Values
adjust (optional)	<p>Specifies which portion of the material may be extended within the segment when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>edge</i> <i>body</i></p> <p>Example: <code><Audio adjust = "body"... /></code></p> <p><i>edge</i> (default)—The material is adjusted by extending the edge that is <i>not</i> identified by the <i>align</i> property and is applied to the duration of the head and tail of the material.</p> <p><i>body</i>—The material is adjusted by extending the body of the material. It is applied to the end of the body that is <i>not</i> identified by the <i>align</i> property.</p>
align (optional)	<p>Specifies which end (or both) to (anchor) the material to the segment, when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>head</i> <i>tail</i> <i>both</i>.</p> <p>Example: <code><Audio align = "tail"... /></code></p> <p><i>head</i> (default)—Anchors the beginning of the material to the beginning of the segment.</p> <p><i>tail</i>—Anchors the end of the material to the end of the segment.</p> <p><i>both</i>—Anchors the center of the material at the center of the segment, so that the Head and Tail are equi-distant from the beginning and end of the segment.</p> <p>Note: If an offset is used in this situation, it is also applied to both ends.</p>

Table 9. Audio Attributes

Name	Description and Values
compose (required)	<p>Specifies that compositing of images should be performed at either the source or output resolution.</p> <p>Keywords: <i>output-size</i> <i>source-size</i> <i>null</i></p> <p>Example: <code><Audio compose="source-size"... /></code></p> <p><i>output-size</i>—Renders the image in the output frame at the output frame size.</p> <p><i>source-size</i>—Renders the image in the output frame at the input frame size.</p> <p><i>null</i> (" ")—By default, Post Producer will use the optimal method of compositing. If you do not want to specify a size, specify an empty string: <code><Audio compose = "... /></code>.</p> <p>Note: An As-run CML reports this attribute to specify how a given image sequence was rendered.</p>
fill (optional)	<p>When the length of the material is less than the length of the segment, the fill attribute determines if and how the item continues to play:</p> <p>Keywords: <i>none</i> <i>hold</i> <i>loop</i></p> <p>Example: <code><Audio fill = "loop"... /></code></p> <p><i>none</i> (default)—extend with empty media (no visual or auditory material).</p> <p><i>hold</i>—continue playing the first or last frame to play the material the entire duration of the segment. If <i>align</i>="head", the last frame is repeated. If <i>align</i>="tail", the first frame is repeated.</p> <p><i>loop</i>—play the media repeatedly for the duration of the segment.</p>

Table 9. Audio Attributes

Name	Description and Values
offset (optional)	<p>The amount of time to wait from the beginning of the segment to play the material. Default: 0.</p> <p>Time Value: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS;FF</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i> <i>FF</i>) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Example: <code><Audio offset = "00:00:05"... /></code></p>
source (required)	<p>Integer value corresponding to the Source element's <i>identifier</i> value, to identify the file to be used in this instance.</p> <p>Example: <code><Audio source = "6"... /></code></p>

Body

The Body element is the portion of a material item identified by the beginning of the material (or the head, if defined, and the end of the material (or the tail, if defined).

The Body (like the [Head](#) and the [Tail](#)) may contain effects elements (opacity, scaling translation, rotation and volume) that control the audio and video composition. Post Producer interpolates between the values set in the body and those in the Head and Tail to create transition and video effects.

See Chapter 5, [Working with Material](#) for a description of the relationship between Head, Body and Tail parameters and timing.

There are no attributes in a Body element.

See also [Head](#), [Tail](#).

Elements ([CML Hierarchy Map](#))

One each of these effects elements may be optionally added to a Head, Body, or Tail element:

- [Opacity](#)
- [Scaling](#)
- [Translation](#)
- [Rotation](#)
- [Volume](#)

Canvas

The optional Canvas element adds a visual rectangle at the same size as the output frame to a [Segment](#). The Canvas element is intended as a filler, and is rendered in black.

The Canvas element is used as content fill or as a background layer when fading in and out or dipping to black.

Example

This snippet illustrates the use of a Canvas element in a Segment (shown in bold).

```
<Segment>
  <Canvas adjust="body" duration="00:00:10.00" layer="1" />
```

Elements ([CML Hierarchy Map](#))

One each of these optional elements may be added:

- [Head](#)
- [Body](#)
- [Tail](#)

Attributes

These attributes may be applied to a Canvas element.

For a visual representation of the effect of adjust and align attributes when using a Fade element, see the illustration below.

Figure 22. Align and Adjust Attributes Work Together on Material Expansion

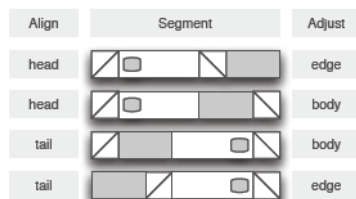


Table 10. Canvas Attributes

Name	Description and Values
adjust (optional)	<p>Specifies which portion of the material may be extended within the segment when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>edge</i> <i>body</i></p> <p>Example: <code><Canvas adjust = "body"... /></code></p> <p><i>edge</i> (default)—The material is adjusted by extending the edge that is <i>not</i> identified by the <i>align</i> property and is applied to the duration of the head and tail of the material.</p> <p><i>body</i>—The material is adjusted by extending the body of the material. It is applied to the end of the body that is <i>not</i> identified by the <i>align</i> property.</p>
align (optional)	<p>Specifies which end (or both) to (anchor) the material to the segment, when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>head</i> <i>tail</i> <i>both</i>.</p> <p>Example: <code><Canvas align = "tail"... /></code></p> <p><i>head</i> (default)—Anchors the beginning of the material to the beginning of the segment.</p> <p><i>tail</i>—Anchors the end of the material to the end of the segment.</p> <p><i>both</i>—Anchors the center of the material at the center of the segment, so that the Head and Tail are equi-distant from the beginning and end of the segment.</p> <p>Note: If an offset is used in this situation, it is also applied to both ends.</p>

Table 10. Canvas Attributes

Name	Description and Values
compose (required)	<p>Specifies that compositing of images should be performed at either the source or output resolution.</p> <p>Keywords: <i>output-size</i> <i>source-size</i> <i>null</i></p> <p>Example: <code><Canvas compose="source-size"... /></code></p> <p><i>output-size</i>—Renders the image in the output frame at the output frame size.</p> <p><i>source-size</i>—Renders the image in the output frame at the input frame size.</p> <p><i>null</i> (" ") —By default, Post Producer will use the optimal method of compositing. If you do not want to specify a size, specify an empty string: <code><Canvas compose=""... /></code>.</p> <p>Note: An As-run CML reports this attribute to specify how a given image sequence was rendered.</p>
duration (optional)	<p>The amount of time to display the canvas. Default: 0.</p> <p>Time values: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS:FF@FPS</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i>;<i>FF</i>) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Examples:</p> <p><code><Canvas duration = "00:00:05.500"... /></code></p> <p><code><Canvas duration = "00:00:05;14@29.97"... /</code></p>

Table 10. Canvas Attributes

Name	Description and Values
fill (optional)	<p>When the length of the material is less than the length of the segment, the fill attribute determines if and how the item continues to play:</p> <p>Keywords: <i>none</i> <i>hold</i> <i>loop</i>.</p> <p>Example: <code><Canvas fill = "loop"... /></code></p> <p><i>none</i> (default)—extend with empty media (no visual or auditory material).</p> <p><i>hold</i>—continue playing the first or last frame to play the material the entire duration of the segment. If <i>align="head"</i>, the last frame is repeated. If <i>align="tail"</i>, the first frame is repeated.</p> <p><i>loop</i>—play the media repeatedly for the duration of the segment.</p>
layer (optional)	<p>An integer value describing the ordinal position in the composite of all material in this segment. Default: 0.</p> <p>Example: <code><Canvas layer = "10"... /></code></p> <p>Material with a lower layer value may be obscured by material with a higher layer value in the same segment.</p> <p>When used in Canvas, the value may be higher or lower than the video (or other material) you are fading with, because in either case one becomes fully transparent.</p>
offset (optional)	<p>The amount of time to wait from the beginning of the segment, to display the material. Default: 0.</p> <p>Time Value: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS;FF</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i>;FF) as standard (<i>:</i>) or drop-frame (<i>;</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Example: <code><Canvas duration = "00:00:05"... /></code></p>

Composition

The Composition element is the root element or container of every composition. Each composition (whether as a file or an object in a program) must have only one Composition element, at the highest level in the XML.

A composition may contain multiple [Sources](#) and [Sequences](#), and may also contain one [Target](#).

Example

This basic composition creates one sequence and encodes it. The output is the same as the input. Not very practical, but it demonstrates the basic functionality of a composition.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share/path/My_TV_Show_Promo.mov" />
  </Source>
  <Sequence layer="1">
    <Segment>
      <Video adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
  </Sequence>
</Composition>
```

Elements (CML Hierarchy Map)

These elements may be added to a Composition element:

- [Source](#) (one or more)
- [Sequence](#) (one or more)
- [Target](#) (one only).

Attributes

These attributes may be applied to a Composition element.

Table 11. Composition Attributes

Name	Description and Values
created (optional)	optional string, providing a creation or modification date.
version (optional)	optional string, providing a version number.

Crop

To crop your source video for use in a segment, you can add a Crop element to the [Source](#) element that identifies the file. (The Crop element is optional; one may be added per Source.) All Video instances that reference a Source with a Crop element inherit the media as specified. That is, cropped.

The Crop element creates a new rectangle of the input frame that is smaller than the original. All pixels that lie outside the crop rectangle are discarded the source as specified. For example, to remove VBI lines or for aspect ratio adjustment. The crop frame is defined relative to the dimensions of the source frame. The area may be specified in pixels, as a ratio, or as a percent. Use positive values for top and left values; use negative values for bottom and right values.

The cropped input frame is then resized to the output frame size (or mask, if specified) before encoding. Cropping is frequently used in conjunction with masking (see [Mask](#)). With a mask, you'll end up with pillars or bars, depending on how the mask ratio differs from the frame ratio.

There are no elements in a Crop element.

See also [Mask](#).

Examples

Here are some examples of using a Crop element.

Center Cutting an HD Frame for HD Frame with Pillars

In this first example, you want to apply a center cut to 16x9, 1920x1080 input to produce 4x3, 1440x1080 output by removing 480 pixels from the width of the image. (25% total; 12.5% from each side).

Input HD Frame: 1920x1080

Output HD Frame: 1440x1080

To remove the 480 pixels, you can configure Crop by percent, ratio, or by pixels:

```
<Crop left="12.5%" right="-12.5%" top="0%" bottom="100%"/>
<Crop left="0.125%" right="-0.125%" top="0%" bottom="100%"/>
<Crop left="240px" right="-240px" top="0px" bottom="0px"/>
```

Center Cutting an HD Frame for SD Frame

In the 2nd example, you want to center cut a 16x9 1920x1080 input to down-convert to 4x3 720x540 output, also by removing 480 pixels from the width of the image.

Input HD frame: 1920x1080

Output SD frame: 720x540, specified in the encoder.

To remove the 480 pixels, you can configure Crop in the same manner as the first example. In both cases, it is necessary to remove 480 pixels from source, 240 from the left and right of the original frame. In the first case, no scaling is performed on the output after cropping. In the second case, the image is scaled to 1/2 in the x and y directions, by virtue of the frame size set in the encoder.

This composition illustrates this example.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share/path/1920x1080.mov"/>
    <Crop left="240px" right="-240px" top="0px" bottom="0px"/>
    <!-- center cut 4x3 cropping for any resolution output-->
  </Source>
  <Sequence layer="0">
    <Segment>
      <Video source="1" >
        <Head>
          <Edit time="00:00:00.000"/>
        </Head>
        <Tail>
          <Edit time="00:00:05.000"/>
        </Tail>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

Summary

Cropping parameters are applied to the output frame and the input resolution is placed within the output frame.

In both cases, it is necessary to remove 480 pixels from source, 240 from the left and right of the original frame. In the first case, no scaling is performed on the output after cropping, in the second case, the image is scaled to 1/2 in the x and y directions.

Therefore, specifying the cropping values in pixels produces exactly the same result in both cases. Crop values specified as a percent are relative to the output resolution. 240 pixels is 16.17% of 1440 and 33.34% of 720.

Attributes

Each of these values may be specified by a units designator: pixels (px) or percent (%). If you do not specify a designator, the value is treated as a percent of the frame dimension. For example, *bottom*="0.222" is the same as *bottom*="22.2%".

Table 12. Crop Attributes

Name	Description and Values
bottom (optional)	<p>Specifies the amount of material to remove upward from the bottom edge of the input frame, as pixels (px), percent (%), or ratio (no unit designator).</p> <p>Example: <code><crop bottom = "-2px" ... /></code></p> <p>Default: 0.</p> <p>Pixels (px) are relative to the bottom edge of the frame. When specifying pixels, the value must be negative.</p> <p>Percentage (%) or ratio values are relative to the origin (0,0—top left corner). Therefore:</p> <p>A bottom crop value of 75% will set the lower edge of the image at 75% of the output resolution, removing 25% of the lines from bottom of the image.</p> <p>A bottom crop value of -75% will crop 75% of the lines from the bottom of the image. In other words, it retains only 25% of the image relative to the top.</p>
left (optional)	<p>Specifies the amount of material to remove from the left edge of the input frame, as pixels (px), percent (%), or ratio (no unit designator).</p> <p>Example: <code><crop left = "0" ... /></code></p> <p>Default: 0.</p> <p>Pixels (px) are relative to the left edge of the frame. When specifying pixels, the value must be positive.</p> <p>Percentage (%) or ratio values are relative to the origin (0,0—top left corner). Therefore:</p> <p>A left crop value of 25% will set the left edge of the image at 25% of the output resolution, removing 25% of the lines from left of the image.</p>

Table 12. Crop Attributes

Name	Description and Values
right (optional)	<p>Specifies the amount of material to remove from the right edge of the input frame, as pixels (px), percent (%), or ratio (no unit designator).</p> <p>Example: <code><crop right = "100%"... /></code></p> <p>Default: <i>100</i>.</p> <p>Pixels (px) are relative to the right edge of the frame. When specifying pixels, the value must be negative.</p> <p>Percentage (%) or ratio values are relative to the origin (0,0—top left corner). Therefore:</p> <p>A right crop value of 75% will adjust the left edge of the image at 75% of the output resolution, removing 25% of the lines from right side.</p> <p>A right crop value of -25% will set the right edge 25% into the image, relative to the right edge of the output image, at the same point as the previous description.</p>
top (optional)	<p>Specifies the amount of material to remove from the top edge of the input frame, as pixels (px), percent (%), or ratio (no unit designator).</p> <p>Example: <code><crop top = "30px"... /></code></p> <p>Default: <i>0</i>.</p> <p>Pixels (px) are relative to the top of the frame. When specifying pixels, the value must be positive.</p> <p>Percentage (%) or ratio values are relative to the origin (0,0—top left corner). Therefore:</p> <p>A top value of 25% will set the upper edge of the image at 25% of the output resolution, removing 25% of the lines from top of the image.</p>

DolbyE

The DolbyE element specifies the audio track layout of a source file that contains compressed, Dolby E audio channels that are to be decoded and remapped or remixed to the output. You add a DolbyE element to a [Source](#) when you want to re-arrange the channels in the output media.

The [Mix](#) element should be added to the DolbyE element to decode and remap the audio channels. The [Route](#) element is used to perform Dolby E pass through.

Dolby E source must have one of the following program layouts:

- 5.1+2 Surround plus one stereo pair.
- 2+2+2+2 Four stereo pairs.
- 5.1 5.1 Surround.
- 7.1 7.1 Surround.

When using Dolby E audio, it is necessary to:

- Silence the WAV pair that contains the Dolby E audio. This is accomplished by mixing it to channels 1 and 2 at zero volume. Failure to do this will result in output with static.
- Specify the Dolby E program configuration that is contained in the source.
- Map and mix the desired tracks that will be contained in the output.

DolbyE Program Channel Configurations

Each of these attributes are required in a DolbyE element.

In the table below, each Dolby E program is identified in the column on the left. On the right, each channel (from 1 through 8) is identified by its logical segment of the program. (-1 indicates the channel is unused in this program configuration.)

For example, if your input is a 4+2+2 DolbyE program, then channels 1,2,5, and 6 comprise the 4 segment; channels 3 and 7 comprise the first 2 segment, and channels 4 and 8 comprise the second or last 2 segment.

Table 13. Dolby E Program Channel Indexes

Program	Index by Channel
5.1 + 2	0, 0, 0, 1, 0, 0, 0, 1
2 + 2 + 2 + 2	0, 1, 2, 3, 0, 1, 2, 3
5.1	0, 0, 0, 0, 0, 0, -1, -1
7.1	0, 0, 0, 0, 0, 0, 0, 0

Example

Here is a typical DolbyE element, used to mix channels in the output:

```
<Composition version="1.0" xmlns="net.telestream.composition">
  <Source identifier="1">
    <File location="Source.mov" />
    <Mix source="1 2" target="1 2" level="0" phase="0°" />
    <DolbyE channels="1 2" program="5.1">
      <Mix source="1" target="1" level="100%" phase="0°" />
      <Mix source="2" target="2" level="100%" phase="0°" />
      <Mix source="3" target="1 2" level="0.707107" phase="0°" />
      <Mix source="4" target="1 2" level="0.707107" phase="0°" />
      <Mix source="5" target="1" level="0.87178" phase="-90°" />
      <Mix source="5" target="2" level="0.489898" phase="-90°" />
      <Mix source="6" target="1" level="0.489898" phase="90°" />
      <Mix source="6" target="2" level="0.87178" phase="90°" />
    </DolbyE>
  </Source>
</Composition>
```

In this example, we are taking a 5.1 Dolby E stereo pair source and mapping it to a stereo pair in the output file.

Elements (CML Hierarchy Map)

One or more [Mix](#) or [Route](#) elements may be added to a DolbyE element.

Attributes

Each of these attributes are required in a DolbyE element.

Table 14. DolbyE Attributes

Name	Description and Values
channels (required)	<p>Specifies the PCM WAV stereo pair that contains Dolby E audio, in the form N N.</p> <p>Keywords: <i>none</i> 1 through 32 <i>all</i></p> <p>Example: <DolbyE channels = "1 2"... /></p> <p><i>none</i>—use to specify no channels.</p> <p><i>1...32</i>—The channel pair, separated by spaces.</p> <p><i>all</i>—use to specify all channels.</p> <p>You identify the 2 channels that make the stereo pair by separating them with a space.</p>
program (required)	<p>Specifies the channel layout of the compressed Dolby E audio, by keyword.</p> <p>Keywords: <i>5.1+2</i> <i>2+2+2+2</i> <i>5.1</i> <i>7.1</i></p> <p>Example: <DolbyE program = "5.1"... /></p>

Edit

The optional Edit element can be used in the [Head](#) and the [Tail](#) of material: Advisory, Image, Title, Canvas, or Media. The purpose of adding an Edit element is to create a clip from a longer source—by specifying a mark-in or mark-out point—for this instance of the material. Used in the Head, it is the mark-in point; in the Tail, the mark-out point.

By default, the mark-in point of material is the beginning of the file, the mark-out point is the end of the file. Effectively, the Edit element moves the beginning or ending of the clip from its default position (the beginning or ending of the file) to the point specified.

You can provide an Edit in either the Head or the Tail, or both. Providing an Edit only in the Head results in a clip running from the Head's Edit point to the end of the file. Providing an Edit only in the Tail results in a clip running from the beginning of the clip to the Edit in the Tail. In *both*, you have a clip trimmed at both the beginning and ending of the file.

Note: It is important to note that Head Edits are inclusive; Tail Edits are exclusive. That is, the first frame after the specified time in the Head Edit is *included* in the clip. However, the first frame after the specified time in the Tail Edit is *excluded*.

Post Producer supports back-timing edits from either end. For example, you may want to roll movie credits in a What's Next promo. You align the credits video at the tail, so that it ends at the same time as the segment. Next, create a Head Edit and place a mark-in at the credits cut using *mode = duration*, and set the time equal to the length of the credit roll.

The combination of the two attributes *mode* and *time* enable you to specify the points in any manner you choose, depending on your media and your requirements.

There are no elements in an Edit.

Note: For a detailed explanation of using an Edit element to create clips from a master file, see [Extracting Clips from Master Media](#).

Examples

These samples illustrate all possible combinations of mode and time attribute settings in Heads and Tails.

Attributes

These two attributes may be applied to an Edit element.

Table 15. Edit Attributes

Attribute Name	Description and Values
mode (required)	<p>Specifies how the time or timecode (for <i>absolute</i> and <i>relative</i> only) is applied to the head or tail of the material.</p> <p>Keywords: <i>absolute</i> <i>relative</i> <i>duration</i>.</p> <p>Example: <code><Edit mode = "relative"... /></code></p> <p><i>absolute</i> (default)—used when a timecode is present in the source; identifies the edit point relative to the timecode. If you supply a timecode but the source lacks a timecode, it is converted to a time value and utilized.</p> <p><i>relative</i>—specifies an edit point measured from the beginning (in a Head) or ending (in a Tail) of the source, irrespective of the timecode (if present).</p> <p><i>duration</i>— specifies an edit point measured as a length of time, from the end (in a Head) or the beginning (in a Tail) of the source.</p> <p>Note: You can't use <i>duration</i> in both Head and Tail in the same instance of material.</p>
time (optional)	<p>Specifies (by time or timecode) the location of the edit point. Drop frame, non-drop frame and time references are valid.</p> <p>A time is not valid, when mode is set to <i>absolute</i>. A timecode is not valid when mode is set to <i>duration</i>.</p> <p>Time is always specified with seconds as a decimal number, in the pattern HH:MM:SS.sss, with microseconds (fractional second in 3 digits).</p> <p>Timecode is specified in the patterns (HH:MM:SS:FF HH:MM:SS:FF), optionally using a drop timecode indicator (;) for US timecodes.</p> <p>Examples:</p> <pre><Edit mode = "absolute" time = "01:00:10" /> <Edit mode = "absolute" time = "01:27;10" /> <Edit mode = "relative" mode = "00:00:10.000" /> <Edit mode = "duration" mode = "00:00:10.600" /></pre>

Fade

The Fade element adds a duration, and smoothly interpolates between the effects (Opacity, Scaling, Transition, Rotation, Volume) elements specified for the [Head](#) and [Tail](#) elements and those in the Body. The interpolation occurs over the specified duration.

It is common to think of the term fade as diminishing video or audio material; in the context of CML, the term is used generally, to increase or decrease *any effect*, blending it gradually and harmoniously with the properties of the Body, based on the duration and shape specified.

The optional Fade element may be used in the [Head](#) and [Tail](#).

If you are using Fade for effects other than Volume or Opacity (that is, you do not want to fade these), you must supply a volume or opacity level explicitly, because in Head and Tail, the default level for both is 0%.

There are no elements in a Fade.

See also:

- [Opacity](#)
- [Scaling](#)
- [Translation](#)
- [Rotation](#)
- [Volume](#)

Example

This example illustrates the use of Fade to adjust the opacity and loudness of the video and audio of clip over a 1/2 second interval.

```
<Segment>
  <Video source="1" layer="0">
    <Head>
      <Edit time="00:00:57.300" />
      <Opacity level="0%" />
      <Volume level="0%" />
      <Fade duration="00:00:00.500" motion-blur="3" />
    </Head>
    <Body>
      <Opacity level="100%" />
      <Volume level="100%" />
    </Body>
    <Tail>
      <Edit time="00:01:09.00" />
      <Opacity level="0%" />
      <Fade duration="00:00:00.500"/>
    </Tail>
  </Video>
</Segment>
```

Attributes

These attributes may be applied to a Fade element.

Table 16. Fade Attributes

Name	Description and Values
duration (optional)	<p>The amount of time to perform the fade. Default: 0.</p> <p>Time values: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS:FF@FPS</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i>;FF) as standard (<i>:</i>) or drop-frame (<i>;</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Examples:</p> <pre><Fade duration = "00:00:05.500"... /> <Fade duration = "00:00:05;14@29.97"... /</pre>
motion-blur (optional)	<p>A real or integer value (range 0-10), indicating the number of intermediate steps between each frame used to generate the blur. The faster you move material using the visual effects elements (Rotate, Translate, Scale, Opacity), the larger the number should be, to smooth out DVE motion effects.</p> <p>For instance, if the image moves 20 pixels per frame, and you set a value of 10 (the maximum) then the blur will be generated by averaging 10 frames, each 2 pixels apart.</p> <p>Note: The higher the value, the longer transcoding will take.</p>

Table 16. Fade Attributes

Name	Description and Values
shape (required)	<p>Specifies the shape of the interpolation.</p> <p>Keywords: <i>linear</i> <i>scurve</i> <i>positive3db</i> <i>negative3db</i></p> <p>Default: <i>linear</i>.</p> <p>Example: <code><Fade shape = "scurve"... /></code></p> <p><i>linear</i> (default)—Maintains a constant rate of change over the length of the fade.</p> <p><i>scurve</i>—Eases in and out of the fade with the midpoint at 0 dB.</p> <p><i>positive3db</i>—Starts quickly at and then slowly tapers off toward the end.</p> <p><i>negative3db</i>—Starts slowly and then quickly tapers off toward the end.</p>

File

The File element adds a fully-qualified path to a file identified by a *Source* or *Subtitle*. The path identifies a file to be utilized in a *Video* or *Audio* element.

The file must be in a valid Post Producer format (*Post Producer Formats*).

The connection between the two elements is that the source value in Video or Audio element, matches the identifier value in the Source element.

The File element has no child elements.

Example

Here is a Example illustrating the use of a Source element.

```
<Source identifier="1">
  <File location="//share\path\source.mov" />
</Source>
<Sequence layer="0">
  <Segment>
    <Video source="1" layer="2" >
```

Note that the Segment’s video clip is referenced as *source 1*. It points to the Source element with an identifier attribute of value 1. The source 1 attribute in the Segment ties the clip to the file specified in the Source element with identifier 1.

Attributes

These attributes may be applied to a File element.

Table 17. File Attributes

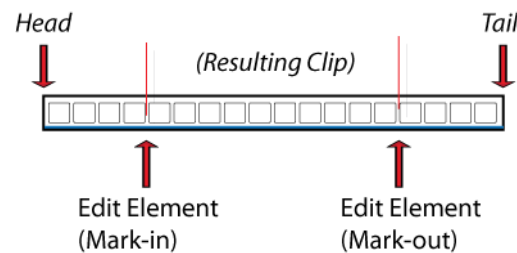
Name	Description and Values
location (optional)	<p>Specifies the fully-qualified path to a file, including file name. The location may be specified as:</p> <ul style="list-style-type: none">• <i>Windows</i>—D:\pathname\filename.ext• <i>UNC</i>—\\share\path\filename.ext• <i>URL</i>—http://SierraStudio.com/media/Vane_logo.png <p>The location must be accessible to the Vantage Playlist service which is executing the Conform action.</p>

Head

The Head element is the point (the first frame) on a material item (an Advisory, Image, Title, Canvas, or a Audio or Video element) at the beginning (or mark-in point, when using an *Edit*) of the material.

The Head (like the *Body* and the *Tail*) may contain effects elements (opacity, scaling translation, rotation and volume) that control the audio and video composition. If you do not supply a Fade with a duration, the effects change instantaneously with the time transition from the Head to the Body—the Head effects will not be applied to any video, because the Head and Body begin at the same moment.

Figure 23. The Head is the Beginning of the File or the Head's Edit Element Value



You define a Head when you want to add effects on the beginning of the material.

There are no attributes in a Head element.

See also *Body*, *Tail*, *Edit*.

Example

In this example, the Head (highlighted in bold, below), had an edit time of 2 seconds; shortening the clip by 8 seconds from the start of the video file. In conjunct

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share\path\Seg1.mov" />
  </Source>
  <Sequence layer="0">
    <Segment >
      <Video source="1" >
        <Head>
          <Edit time="00:00:02:00"/>
        </Head>
        <Tail>
          <Edit time="00:00:08.03"/>
        </Tail>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

In conjunction with the tail, this clip is 10 seconds, 3 frames shorter than the master file.

Elements ([CML Hierarchy Map](#))

One each of these elements may be optionally added to a Head or Tail element:

- *Edit*
- *Fade*

One each of these effects elements may be optionally added to a Head, Body, or Tail element:

- *Opacity*
- *Scaling*
- *Translation*
- *Rotation*
- *Volume*

Image

The Image element adds a raster graphic image or image sequence to a [Segment](#). Unlike Video elements, Image elements reference source file(s) directly, rather than using a Source > File element.

You can add a single image, or you can add an image sequence—a series of pictures that constitutes a video clip. For example, you may have a 3-second, cartooned series of graphics to portray partly cloudy weather in a weather update promo.

Images are typically used in two ways:

- As an overlay (branding bug, rendered title, etc., applied to an upper layer; typically smaller (or resized) so that the underlying layers are visible. Or, the image may contain an alpha channel to achieve the same goal. Settings for opacity, scaling, translation, and rotation can all be applied to images. Images also have a layout setting to control how the image fills the frame.
- Used to produce full frame video at the same resolution as the output frame.

Note: Each material item in a segment may logically be divided into three parts: the beginning ([Head](#)), the middle ([Body](#)) and the end ([Tail](#)). You can change the presentation of the material by adding an [Edit](#), a [Fade](#), and various effect elements.

Example

Here is a comprehensive example of using images in a composition.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="\\share\path\Mystic River Seg 1.mov" />
  </Source>
  <Source identifier="2">
    <File location="\\share\path\3_FTD_Start at_16.21.mov" />
    <Mask left="12.5%" right="87.5%" top="0%" bottom="100%" />
  </Source>
  <Sequence layer="0">
    <Segment>
      <Image align="both" adjust="body" fill="loop" layer="1"
location="\\share\path\logo.png" frames="1" layout="none"
offset="00:00:02.000">
        <Head>
          <Scaling x="30%" y="30%" />
          <Translation x="80%" y="70%" />
          <Opacity level="0%" />
          <Fade duration="00:00:01.000" />
        </Head>
        <Body>
          <Scaling x="30%" y="30%" />
          <Translation x="80%" y="70%" />
        </Body>
      </Image>
    </Segment>
  </Sequence>
</Composition>
```

```

        <Opacity level="100%" />
    </Body>
    <Tail>
        <Fade duration="00:00:01.000" />
        <Scaling x="30%" y="30%" />
        <Translation x="80%" y="70%" />
        <Opacity level="0%" />
    </Tail>
</Image>
<Video source="1" >
    <Tail>
        <Edit time="00:00:8.933" />
    </Tail>
</Video>
</Segment>
</Sequence>
</Composition>

```

Implementing Image Sequences

To implement an image sequence (create video), you should add all of the images to a given directory, each named with a numerically increasing pattern. In the Image element, you only specify the file name of the image (*PartlyCloudy0001.png*, for example) that starts the sequence. Based on its base name at the start, Post Producer will attempt to determine the rest of the files in the sequence. Post Producer (without the presence of a wild card character) assumes all files in the sequence start with the same base name, and are sequentially numbered—you can't have any gaps: *PartlyCloudy0002*, *PartlyCloudy0003*, etc.

Or, you can use the asterisk (*) as a wildcard. If you supply the file name with a wildcard, Post Producer will perform an alpha-numeric sort on all the files in the folder and use them in the sequence, even if there is a gap in sequence.

The wildcard is used to specify image sequence file names that precede a fixed name, where the files may be produced by systems and workflows with naming conventions that embed the incremental component at other places in the file name. For example:

```

000filename.png
001filename.png
002filename.png

```

In this case, the wildcard would be used in this manner: **filename.png*.

Example Image Sequence with Wildcard

In this example, a wild card is used in the Image element's file name reference (bold):

```

<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
    <Source identifier="0">
        <File location="\share\path\Radars.wav" />
    </Source>

```




```
<Sequence>
  <Segment>
    <Audio fill="hold" source="0">
      <Head>
        <Edit mode="absolute" time="00:00:00.000" />
      </Head>
      <Tail>
        <Edit mode="absolute" time="00:00:10.000" />
      </Tail>
    </Audio>
    <Image fill="loop"
location="//share/path/Wx_Image_Sequence/*_iah.jpg" frames="35"
duration="00:00:03.917" />
  </Segment>
</Sequence>
</Composition>
```

The settings for implementing image sequences:

1. The first image of the sequence.
2. The total number of images in the sequence.
3. The length in time of the sequence. This may be specified or you can use the default of frame count * the output frame rate.

For example, you can provide 40 frames and then specify a 4-second duration—Post Producer will play each file the proportionate number of frames for 4 seconds, to play it out correctly. If you don't specify a duration, the frames are played out at the output frame rate specified in the encoder settings in the Conform action.

Note: For a practical example of an Image sequence, see [Image Scaling and Placement](#).

Example

This example illustrates the use of 2 image sequences to create a lower third title.

Figure 24. Align and Adjust Attributes Work Together on Material Expansion



The image elements are highlighted in bold, below.

The first file of the first image sequence (`\\share\path\Lower3rd-BG_PNGs\frame-000000.png`) is identified with a count of the total image sequence (using the *frames* attribute). The *duration* of the entire sequence can be set to a desired value. Specify 00:00:00:00 to output one frame per image, at the frame rate of the output video.

The *fill* attribute may be used to extend the duration of the image sequence by looping the entire sequence or replaying the final frame.

Adjusting the *Edge* will prevent the application of any effects (opacity, scale, translation and rotation). If this is desired, then adjust the *Body*.

Elements (CML Hierarchy Map)

One each of these elements may be optionally added to an Image element:

- *Head*
- *Body*
- *Tail*

Attributes

These attributes (some required, some optional) may be applied to an Image element. The rating and advisory attributes are required.

For a visual representation of the effect of adjust and align attributes when using a Fade element, see the illustration below.

Figure 25. Align and Adjust Attributes Work Together on Material Expansion

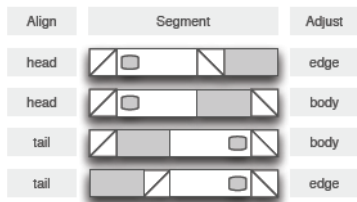


Table 18. Image Attributes

Name	Description and Values
adjust (optional)	<p>Specifies which portion of the material may be extended within the segment when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>edge</i> <i>body</i></p> <p>Example: <code><Image adjust = "body"... /></code></p> <p><i>edge</i> (default)—The material is adjusted by extending the edge that is <i>not</i> identified by the <i>align</i> property and is applied to the duration of the head and tail of the material.</p> <p><i>body</i>—The material is adjusted by extending the body of the material. It is applied to the end of the body that is <i>not</i> identified by the <i>align</i> property.</p>
align (optional)	<p>Specifies which end (or both) to (anchor) the material to the segment, when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>head</i> <i>tail</i> <i>both</i>.</p> <p>Example: <code><Image align = "tail"... /></code></p> <p><i>head</i> (default)—Anchors the beginning of the material to the beginning of the segment.</p> <p><i>tail</i>—Anchors the end of the material to the end of the segment.</p> <p><i>both</i>—Anchors the center of the material at the center of the segment, so that the Head and Tail are equidistant from the beginning and end of the segment.</p> <p>Note: If an offset is used in this situation, it is also applied to both ends.</p>

Table 18. Image Attributes

Name	Description and Values
compose (required)	<p>Specifies that compositing of images should be performed at either the source or output resolution.</p> <p>Keywords: <i>output-size</i> <i>source-size</i> <i>null</i></p> <p>Example: <code><Image compose="source-size"... /></code></p> <p><i>output-size</i>—Renders the image in the output frame at the output frame size.</p> <p><i>source-size</i>—Renders the image in the output frame at the input frame size.</p> <p><i>null</i> (" ")—By default, Post Producer will use the optimal method of compositing. If you do not want to specify a size, specify an empty string: <code><Audio compose=""... /></code>.</p> <p>Note: An As-run CML reports this attribute to specify how a given image sequence was rendered.</p>
duration (optional)	<p>The amount of time (specified as time or timecode) to display the entire sequence of material. By default, <i>duration</i> = "0". Duration of 0 causes the images to be used one per frame, at the output frame rate. (The duration is equal to the number of frames times the output frame rate.)</p> <p>Time values: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SExampleS:FF@FPS</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i>FF) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Examples:</p> <p><code><Image duration = "00:00:05"... /></code></p> <p><code><Image duration = "00:00:07.5"... /</code></p>

Table 18. Image Attributes

Name	Description and Values
fill (optional)	<p>When the length of the material is less than the length of the segment, the fill attribute determines if and how the item continues to play:</p> <p>Keywords: <i>none</i> <i>hold</i> <i>loop</i>.</p> <p>Example: <code><Image fill = "loop"... /></code></p> <p><i>none</i> (default)—extend with empty media (no visual or auditory material).</p> <p><i>hold</i>—continue playing the first or last frame to play the material the entire duration of the segment. If <i>align="head"</i>, the last frame is repeated. If <i>align="tail"</i>, the first frame is repeated.</p> <p><i>loop</i>—play the media repeatedly for the duration of the segment.</p>
frames (optional)	<p>An integer (default=1), which specifies the number of frames in the image sequence.</p> <p>Example: <code><Image frames = "60"... /></code></p> <p>If there are more images in the folder than you want to use in a given instance (Image), you can specify the first frame you want to use (by file name), and then limit the number to use in the sequence with this attribute.</p>
layer (optional)	<p>An integer value describing the ordinal position in the composite of all material in this segment. Default: 0.</p> <p>Example: <code><Image layer = "10"... /></code></p> <p>Material with a lower layer value may be obscured by material with a higher layer value in the same segment.</p> <p>You should supply a layer value higher than the video layer—by default, layer is 0; thus, the video will obscure the graphic.</p>

Table 18. Image Attributes

Name	Description and Values
layout (optional)	<p>Specifies the method used to alter the image to the target dimensions prior to applying any <i>Scaling</i> effects.</p> <p>Keywords: <i>null</i> <i>center</i> <i>tile</i> <i>zoom</i> <i>fill</i> <i>stretch</i>.</p> <p>Example: <code><Image layer = "zoom"... /></code></p> <p><i>null</i> (" " =empty string)—The image is top-left aligned to the target area. Any unused target area is masked.</p> <p><i>center</i>—The image is centered within the target area while maintaining the source size and aspect ratio. Any unused target area is masked.</p> <p><i>tile</i>—The image is tiled across the target area maintaining the source size and aspect ratio. Any unused source area is cropped.</p> <p><i>zoom</i>—The image is resized to fill one dimension of the target area while maintaining the source aspect ratio. Any unused target area is masked.</p> <p><i>fill</i>—The image is resized to fill both dimensions of the target area while maintaining the source aspect ratio. Any unused source area is cropped.</p> <p><i>stretch</i>—The image is resized to completely fill the target area.</p>
location (optional)	<p>Specifies the fully-qualified path to the image (or first image) file, including file name. The location may be specified as:</p> <ul style="list-style-type: none"> • <i>Windows</i>—D:\pathname\filename.ext • <i>UNC</i>—\\share\path\filename.ext • <i>URL</i>—http://SierraStudio.com/media/Vane_logo.png <p>The location must be accessible to the Vantage Playlist service which is executing the Conform action.</p> <p>Example: <code><Image location = "\\share\path\PartlyCloudy0001.png"... /></code></p> <p>The image(s) may be in png, jpeg, or bmp format.</p>

Table 18. Image Attributes

Name	Description and Values
offset (optional)	<p>The amount of time to wait from the beginning of the segment, to display the material. Default: 0.</p> <p>Time Value: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS;FF</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i> <i>FF</i>) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Example: <code><Image offset = "00:00:05.000"... /></code></p>

Mask

To specify a new rectangle in the output frame in which to present material, add a Mask element to the [Source](#) element that identifies the file. (The Mask element is optional; one may be added per Source.)

The Mask Element resizes the source to fill the Mask frame. To that extent, either pillars or bars are filled with the fill color specified in the regions outside the mask region. Mask is commonly used for filling space when converting content of 4x3 aspect ratio to output with a 16x9 aspect ratio. The Mask element is commonly used in conjunction with a [Crop](#) element.

The Mask dimensions are specified relative to the top, bottom, left and right edges of the target output frame. The Mask values can be specified as a ratio (no designator) percent (%), or pixels (px). Pixels are useful if the output resolution will be constant for all output. Percent (or ratio) is useful if the aspect ratio remains constant, but the resolution is likely to change.

There are no elements in a Mask element.

See also [Crop](#).

Example

This sample CML snippet specifies a 640x480 (4x3) source, placed into a 960x540 (16x9) output using a Mask Element applied to preserve the aspect ratio of the source in the output. All segments in which this source is used will have masked video.

```
<Source identifier="1">  
  <File location="//share/path/butterfly.mov" />  
  <Mask left="120px" right="-120px" top="0px" bottom="100px"  
color="#FFA2A4A6"/>  
</Source>
```

This Mask element assumes a 16x9 output of 960x540 with a 4:3 center cut 540 lines high. Therefore, a width of 720 pixels is required to preserve the 4x3 aspect ratio. The 640x480 source is stretched to fill the output.

Attributes

These attributes may be applied to a Mask element. The X and Y attributes refer to the output frame dimension, with 0,0 being the upper left corner.

All of these attribute values may be followed by a units designator: pixels (px) or percent (%). If you do not specify a designator, you are specifying a ratio. For example, *bottom="0.222"* is the same as specifying *bottom="22.2%"*.

Positive values are relative to the origin (the upper left corner) of the output frame; negative values are relative to the associated edge or side. For example, if you specify *right="-25%"*, you are intending for the right edge of the area to be located 25 percent of the distance of the frame width inward from the right edge of the frame.

Edges are specified relative to the edge of the output frame. The allowable values assume that all settings will locate the mask within the frame.

Percent:

When using%, positive values between 0% and 50% will move the edge toward the center of the frame. Positive values greater than 50% will be relative to the 0,0 origin of the top left hand corner of the output. Values that refer to off screen locations will result in errors.

Pixels (px):

- Positive values move the left edge right toward the center of the frame.
- Negative values move the right edge left toward the center of the frame.
- Positive values move the top edge downward to the center of the frame.
- Negative values move the bottom edge toward the center of the frame.

Percent (%) or Ratio:

- Positive values move the edge on-screen from the left; off-screen on the right.
- Negative values move the edge off-screen from the left; on-screen from the right.
- Positive values move the edge on-screen from the top; off-screen from bottom.
- Negative values move the edge off-screen from top; on-screen from bottom.

Table 19. Mask Attributes

Name	Description and Values
bottom (optional)	An integer or decimal number specifying the bottom edge, as pixels (px), percent (%), or ratio (no unit designator). Example: <code><Mask bottom="480px" ... /></code>
color (optional)	Keywords: <i>transparent</i> <i>black</i> <i>white</i> <i>ARGB</i> value The default keyword is <i>black</i> . Example: <code><Mask color="#FFA2A4A6" ... /></code> The supported colors are defined in .NET. For details, see the .NET Color Table . <i>ARGB</i> —The RGB + Alpha color, in 0xAARRGGBB format, where the high-order byte (AA) contains the alpha channel and the remaining bytes contain color components for red, green and blue, respectively. The alpha channel specifies the opacity of the pixel, where a value of 0x00 represents a pixel that is fully transparent and a value of 0xFF represents a fully opaque pixel.
left (optional)	A integer or decimal number specifying the left edge, as pixels (px), percent (%), or ratio (no unit designator). Example: <code><Mask left="0" ... /></code>

Table 19. Mask Attributes

Name	Description and Values
right (optional)	A integer or decimal number specifying the right edge, as pixels (px), percent (%), or ratio (no unit designator). Example: <Mask right="100%" ... />
top (optional)	A integer or decimal number specifying the top edge, as pixels (px), percent (%), or ratio (no unit designator). Example: <Mask top="0px" ... />

Mix

You can apply a Mix element to a [Source](#) to map and mix audio channels in the source file. The Mix element defines how audio is used in each instance. For example, a single master file with 10 stereo language tracks might be used in 10 separate source definitions in the composition. Each language-specific Source instance would contain the same video, with a different set of audio channels.

Both mixing and routing require that you set up and configure the Channel Map filter in the Conform action.

There are no child elements in a Mix element.

Note: For a practical example of typical audio mixing applications, see [Mixing Audio](#).

See also [Route](#).

Example

This source instance combines a mono channel on source channel 3 with each channel of a stereo pair on source channels 1 and 2 to produce a stereo track on channels 1 and 2 of the Source instance.

It is necessary to map the channels to the output for wrapping of the final encoded output. Metadata labels for channels (Left/Right), language (French, German) are defined in the Channel map filter and the stream settings of the Conform action.

```
<Source identifier="2"> <!-- English -->
  File location="//share/path/Master Oh_Susanna.mov"/>
  <Mix source="1 2" target="1 2" level=".707%"/>
  <Mix source="3" target="1" level=".9"/>
  <Mix source="3" target="2" level=".9"/>
</Source>
```

Attributes

These attributes may be applied to a Mix element.

Table 20. Mix Attributes

Name	Description and Values
level (optional)	A number value (default: 1), specifying the level in unit gain, in Percent (%) (default) or Decibels (dB). Examples: <pre><Mix level=".707%"... /> <Mix level="3dB"... /></pre>

Table 20. Mix Attributes

Name	Description and Values
phase (optional)	<p>A number value specifying the phase shift in degrees. (The degree symbol is not required.) A negative number rotates clockwise, and a positive number rotates counter-clockwise. The default phase value is 0.0 degrees.</p> <p>Example: <code><Mix level="-90"... /></code></p> <p>This property may be used to implement surround sound matrix encoding (+/-90°) or to correct phase inversions (+/-180°). No other values are supported.</p>
source (required)	<p>Specifies the channel number (as an integer) of the input audio track.</p> <p>Keywords: <code>null</code> <code>1</code> through <code>32</code> <i>all</i></p> <p>Example: <code><Mix source="1 2"... /></code></p> <p>You may combine multiple channels by separating them with a space. You can also specify none ("") or <i>all</i>, as defined below.</p> <p><code>null(""-empty string)</code>—use to specify no channels.</p> <p><code>1...32</code>—the channel number.</p> <p><i>all</i> (default)—use to specify all channels.</p>
target (required)	<p>Specifies the channel number (as an integer) of the output audio track.</p> <p>Keywords: <code>null</code> <code>1</code> through <code>32</code> <i>all</i></p> <p>Example: <code><Mix source="1 2"... /></code></p> <p>You may combine multiple channels by separating them with a space. You can also specify none ("") or <i>all</i>, as defined below.</p> <p><code>null(""-empty string)</code>—use to specify no channels.</p> <p><code>1...32</code>—the channel number.</p> <p><i>all</i> (default)—use to specify all channels.</p>

Opacity

One Opacity element may optionally be added to a [Body](#), [Head](#), or [Tail](#).

The Opacity element specifies the relative degree to which items below it can be viewed—its transparency.

It is important to note that the default value in a Body element is 100%, while in a Head or Tail, it is 0%. For this effect to be of practical value, the Fade element must be present, and its duration must be greater than 0. If you are using Fade for effects other than opacity (that is, you do not want to fade the opacity), you must supply an opacity level explicitly.

There are no elements in an Opacity element.

See also [Volume](#), [Scaling](#), [Translation](#), [Rotation](#).

Example

In this example, a Vantage logo is placed over a video, and using opacity and fade in the head, body, and tail, fades the logo in and out over the course of the video.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="\\share\path\Mystic_River_Seg_1.mov" />
  </Source>
  <Sequence layer="0">
    <Segment>
      <Image align="both" adjust="body" fill="loop" layer="1"
location="\\share\path\Vantage-logo_small.png" frames="1"
layout="none" offset="00:00:02.000">
        <Head>
          <Opacity level="0%"/>
          <Fade duration="00:00:01.000"/>
        </Head>
        <Body>
          <Opacity level="100%"/>
        </Body>
        <Tail>
          <Fade duration="00:00:01.000"/>
          <Opacity level="0%"/>
        </Tail>
      </Image>
      <Video source="1" >
        <Tail>
          <Edit time="00:00:8.933"/>
        </Tail>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

Attributes

This attribute must be applied to an Opacity element.

Table 21. Opacity Attributes

Name	Description and Values
level (required)	<p>An integer or decimal number representing the relative opaqueness (where 0 is fully transparent and 100 is fully opaque), expressed as a ratio (no designator) or percent (%) of the original material.</p> <p>Example: <Opacity level="50%" /></p> <p>Default (in Head and Tail elements): 0%.</p> <p>Default (in Body elements): 100%.</p>

Processor

The Processor element is a reference to a separate set of transcoder settings, which are created and maintained directly in the transcode actions in a workflow, in Workflow Designer.

The preprocessing settings of any Vantage transcode action (IPTV/VOD, Flip, Multiscreen, and Conform) that use the 16-bit Multi-core or Lightspeed processor can be saved in that action and then implemented in the Conform action. This feature enables you to utilize settings that may not be displayed in the Conform Action itself.

After configuring the settings in a given action, to save the template right-click on it and select Create Template. Then, name the template and click OK to save it in the domain database. Now, you can specify the template either in the composition directly (using the Processor element) or the Conform action in your workflow, to apply the desired transcoder settings.

VPL templates can be configured to process input media that has specific decoding requirements for audio and video, such as frame rate conversion, Telecine or inverse Telecine, or any other Vantage pre-processing settings.

You name each template, and then specify Source elements with the file and processor template you want to use, as shown in the composition example below.

There are no elements in a Processor element.

Example

This example illustrates the use of a Processor element, which specifies how the file should be decoded in the Conform action, by specifying *ProcessorTemplate1*, which contains the decoding requirements.

```
<Source identifier="1">  
  <File location="//share/path/speed_racer.mov" />  
  <Processor template="ProcessorTemplate1" />  
</Source>
```

Attributes

These attributes may be applied to a Processor element.

Table 22. Processor Attributes

Name	Description and Values
template (required)	The name of the processor template (as defined in the target Conform action) to use for decoding this file. Example: <pre><Processor template="ProcessorTemplate1" /></pre>

Rotation

One Rotation element may optionally be added to a [Body](#), [Head](#), or [Tail](#). The image rotates material on the Z axis, from its upper left corner.

For this effect to be of practical value in a Head or Tail element, the Fade element must be present, and its duration must be greater than 0.

You can use this element to spin the material on the Z axis for the effect duration. If you want a transition where the material rotates twice counter clockwise you would specify -720. If you want a transition where it rotates once clockwise you would specify 360.

There are no elements in a Rotation element.

See also [Volume](#), [Opacity](#), [Scaling](#), [Translation](#).

Example

This example illustrates the application of a typical set of effects on a video clip, including scaling, translation, and opacity.

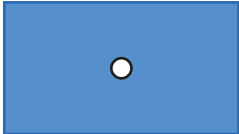




```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="D:\media\10seconds.mov"/>
  </Source>
  <Sequence layer="0">
    <Segment>
      <Video align="head" adjust="edge" fill="none" source="1"
layer="1" >
        <Head>
          <Scaling x="40%" y="40%" />
          <Translation x="50%" y="50%" />
          <Rotation z="0°" center="top-left"/>
          <Opacity level="100%" />
          <Fade duration="00:00:05.000" />
        </Head>
        <Body>
          <Translation x="50%" y="50%" />
          <Scaling x="40%" y="40%" />
          <Rotation z="360°" center="top-left"/>
        </Body>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```



Attributes

These attributes may be applied to a Rotation element.

Table 23. Rotation Attributes

Name	Description and Values
z (optional)	<p>The amount of rotation on the Z axis. Positive values rotate clockwise; negative values rotate counter-clockwise.</p> <p>Default: 0 degrees.</p> <p>Example: <code><Rotation z="360"... /></code></p>
center (optional)	<p>Keyword: the point on the image that is the center of rotation; the point about which the image will move when the rotation occurs.</p> <p>center bottom-left bottom-right top-right top-left top-right.</p> <p>Example: <code><Rotation center="bottom-left"... /></code></p> <p>center:</p>  <p>bottom-left:</p>  <p>bottom-right</p>  <p>top-right</p>  <p>top-left</p> 

Route

You add a Route element to a [Source](#) to map audio tracks from the input file in the associated [File](#) to other tracks in the output file. Unlike the Mix element, the Route element does not decode the audio.

Both mixing and routing require that you set up and configure the Channel Map filter in the Conform action.

The *source* and *target* attributes are required in a Route element.

There are no child elements in a Route element.

Note: For a practical example of typical audio routing applications, see [Routing Audio Without Decoding](#).

See also [Mix](#).

Example

This sample ensures that audio in track 1 is placed in output track 1 (and same for track 2), without decoding the audio.

```
<Source identifier="2"> <!-- English -->
  <File location="//share/path/Master_DolbyE_Stereo.mov"/>
  <Route source="1 2" target="1 2"/>
</Source>
```

Attributes

These attributes may be applied to a Route element.

Table 24. Route Attributes

Name	Description and Values
source (required)	<p>Specifies the channel number (as an integer) of the input audio track. Default: "".</p> <p>Keywords: null 1 through 32 all</p> <p>Example: <Route source="1 2"... /></p> <p>You may combine multiple channels by separating them with a space. You can also specify "" (empty string) or all, as defined below.</p> <p>null (" "-empty string) (default)—use to specify no channels.</p> <p>1...32—the channel number itself. May be combined, separated by spaces.</p> <p>all—use to specify all channels.</p>

Table 24. Route Attributes

Name	Description and Values
target (required)	<p>Specifies the channel number (as an integer) of the output audio track. Default: "".</p> <p>Keywords: <i>none</i> 1 through 32 <i>all</i></p> <p>Example: <code><Route target = "1"... /></code></p> <p>You may combine multiple channels by separating them with a space. You can also specify "" (empty string) or <i>all</i>, as defined below.</p> <p><i>none</i>—use to specify no channels.</p> <p><i>1...32</i>—the channel number itself. May be combined, separated by spaces.</p> <p><i>all</i>—use to specify all channels.</p>
order (optional)	<p>If multiple Route elements are present, specifies (by an integer) the order of this route relative to other routes targeting the same channels. Default: 0.</p> <p>Example: <code><Route order = "4"... /></code></p> <p>This value determines the order in which routes to the same target channel are resolved. The route with the lowest number takes precedence.</p>

Scaling

One Scaling element may optionally be added to a [Body](#), [Head](#), or [Tail](#).

Add a Scaling element when you want to change the display dimensions of visual material. You can adjust the vertical and horizontal dimensions independently, by ratio, percent or pixels. Positive values increase the size of the image, negative values reduce the size of the image.

If you don't supply a Scaling element, the image is displayed in its actual pixel dimensions, regardless of the output frame size or ratio.

For this effect to be of practical value in a Head or Tail element, the Fade element must be present, and its duration must be greater than 0.

The Scaling element is often used in conjunction with the [Translation](#).

There are no elements in a Scaling element.

See also [Volume](#), [Opacity](#), [Translation](#), [Rotation](#).

Example

In this code snippet, an [Advisory](#) image (the default image—thus, not specified) is scaled to twice its original size. A percent is used here; but a ratio (of 2) could also be used. The result is the same in both cases.

```
<Advisory layer="1" rating="TV-14 V S D L" type="USTV"
duration="00:00:30.00" fill="none" adjust="edge" align="head">
  <Body>
    <Scaling x="2" y="2"/>
  </Body>
</Advisory>
```

If you were to use a pixel designator (*px*), you'd have less flexibility in uniformly applying this attribute to images of varying sizes: Adding 10 pixels to a 100 pixel image would increase its size by 10 percent, but adding 10 pixels to a 200 pixel image would only increase it by 5 percent.

Note also, that when using a ratio or a percent, the use of the same but negative value does not have the directly opposite effect. For example: To double the size of an image, you would increase it by a ratio of 2 or 200%; but to halve the size of an image, you would decrease it by a ratio of 0.5, or 50%.

Attributes

These attributes may be applied to a Scaling element. Although both x and y are optional, if neither were supplied, the Scaling element as defined would be of no practical value.

Table 25. Scaling Attributes

Name	Description and Values
x (optional)	Specifies the change in the horizontal dimension of the display rectangle of the visual material, as pixels (px), percent (%), or ratio (no unit designator). Example: <code><Scaling x="200%" y="200%"/></code>
y (optional)	Specifies the change in the vertical dimension of the display rectangle of the visual material, as pixels (px), percent (%), or ratio (no unit designator). Example: <code><Scaling x="2" y="2"/></code>

Segment

One or more Segment elements may be added to a [Sequence](#). A segment is comprised of material items (video, audio, titles, graphics, etc.). A segment may also contain one or more sequences, which allows a single material item (for example, a Title) to span a sequence of other material items.

The duration of a segment is determined by the duration of the longest item (including any offset applied to that item).

Example

Here is a typical segment in a composition.

```
<Segment>
  <Image align="both" adjust="body" fill="loop" layer="1"
location="//share/path/Vantage-logo_small.png" frames="1"
duration="00:00:15.000" offset="00:00:02.000">
    <Head>
      <Scaling x="30%" y="30%" />
      <Translation x="80%" y="70%" />
      <Opacity level="0%" />
      <Fade duration="00:00:01.000" />
    </Head>
    <Body>
      <Scaling x="30%" y="30%" />
      <Translation x="80%" y="70%" />
      <Opacity level="100%" />
    </Body>
    <Tail>
      <Scaling x="30%" y="30%" />
      <Translation x="80%" y="70%" />
      <Opacity level="0%" />
      <Fade duration="00:00:01.000" />
    </Tail>
  </Image>
</Segment>
```

Elements (CML Hierarchy Map)

Optionally, one each of these material elements may be added to a Segment:

- [Advisory](#)
- [Title](#)
- [Image](#)
- [Audio](#)
- [Video](#)
- [Canvas](#)

Optionally, the [Sequence](#) may also be added to a Segment element, to create an independent timeline for additional segments.

Attributes

These attributes may be applied to a Segment element.

Table 26. Segment Attributes

Name	Description and Values
identifier (optional)	A user-assigned SCTE 35 ID for this segment. May be a description or an ad insertion ID, for example.
instructions (optional)	Keyword which instructs the encoder on placement of keyframes and splice points. Keywords: <i>keyframe</i> <i>splice</i> <i>keyframe</i> —instructs the x.264 encoder to insert an IDR frame and the MPEG 2 encoder to add black frames to complete the GOP around the splice point, so that the segment is aligned to an IDR or I-frame. This feature supports the Black Arrow dynamic ad insertion specification. <i>splice</i> —adds an SCTE 35 dynamic ad flag to the output stream. This flag is tagged with an ID that is passed with the SCTE 35 location to the transport stream output.
time (As Run)	As Run attribute. This is the starting time of the segment in the output file, after the composition has been conformed. This value is set by the Conform action and inserted in the As Run composition by the Chronicle action, and should not be set by you.
transcode (As Run)	As Run attribute. This attribute advises if this segment was direct converted or not.

Sequence

You can add one or more Sequence elements to a [Composition](#), or as sub-sequences, to a [Segment](#). You add a sequence when you need to establish a separate, independent timeline in your composition, upon which to place segments of material.

Example

A sequence may be used as a complex material type to combine multiple material elements. In this example, a graphic overlay spans two video clips within the segment by creating a complex video clip using the Sequence element.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="\\share\path\Mystic River Seg 1.mov" />
  </Source>
  <Source identifier="2">
    <File location="\\share\path\3_FTD_Start at_16.21.mov" />
    <Mask left="12.5%" right="87.5%" top="0%" bottom="100%" />
  </Source>
  <Sequence layer="0">
    <Segment>
      <Image align="both" adjust="body" fill="loop" layer="1"
location="\\share\path\logo.png" frames="1" layout="none"
offset="00:00:02.000">
    </Image>
    <Sequence>
      <Segment>
        <Video source="1" >
          <Tail>
            <Edit time="00:00:8.933"/>
          </Tail>
        </Video>
      </Segment>
      <Segment>
        <Video source="2" >
          <Tail>
            <Edit time="00:00:10.330"/>
          </Tail>
        </Video>
      </Segment>
    </Sequence>
  </Segment>
</Sequence>
</Composition>
```

Elements (CML Hierarchy Map)

One or more [Segments](#) may be added to a Sequence element.

Attributes

The layer attribute may be applied to a Sequence element.

Table 27. Sequence Attributes

Name	Description and Values
layer (optional)	<p>An integer value describing the ordinal position in the composite of all material in this segment. Default: 0.</p> <p>Example: <code><Sequence layer = "10"... /></code></p> <p>Material in a lower layer sequence may be obscured by material in a higher layer sequence.</p>

Shadow

One Shadow element can optionally be added to a [Title](#) to apply a shading effect to your title text. The shadow is a replication of the Title text using the same attributes, but offsetting the text, applying a different color, and adjusting opacity through the softness attribute.

There are no elements in a Shadow element.

Note: For a practical example of adding a shadow to a title, see [Adding Titles](#).

See also [Title](#), [Area](#).

Example

Here is a Title sample with every attribute, including an Area and Shadow element.

```
<Title align="head" adjust="edge" fill="none" layer="2"
duration="00:00:04.00" offset="00:00:01.000" font="Helvetica"
size="50pt" style="italic" weight="bold" foreground-color="bisque"
background-color="transparent" wrap="false" horizontal-
align="left" vertical-align="middle" overflow="overflow">
  <Area top="70%" left="20%" bottom="90%" right="80%" />
  <Shadow color="gold" softness="10%" vertical-offset="6%"
horizontal-offset="6%" />
  This is My Story...
</Title>
```

Attributes

These attributes may be applied to a Shadow element to control shading.

Table 28. Shadow Attributes

Name	Description and Values
color (optional)	<p>This is the color that the shadow is rendered in.</p> <p>Keywords: <i>transparent</i> <i>black</i> <i>white</i> <code><color></code> <i>ARGB</i></p> <p>Default: <i>black</i>.</p> <p>Supported color names are defined in .NET. See Color Table.</p> <p>Example: <code><Shadow color="10"... /></code></p> <p><i>ARGB</i>—The RGB + Alpha color, in 0xAARRGGBB format, where the high-order byte (AA) contains the alpha channel and the remaining bytes contain color components for red, green and blue, respectively. The alpha channel specifies the opacity of the pixel, where a value of 0x00 represents a pixel that is fully transparent and a value of 0xFF represents a fully opaque pixel.</p>

Table 28. Shadow Attributes

Name	Description and Values
horizontal-offset (optional)	<p>Specifies the horizontal offset of the shadow, as pixels (px), percent (%), or ratio (no unit designator). Positive numbers adjust the shadow text to the right.</p> <p>Default: 4 px.</p> <p>Example: <code><Shadow horizontal-offset="10px"... /></code></p>
softness (optional)	<p>Specifies the relative blurriness of the shadow as a ratio or percent. Larger numbers increase softness.</p> <p>Default: 0.5 (50%).</p> <p>Example: <code><Shadow softness="10%"... /></code></p> <p>The value may be an integer or real number to 3 digits, as a ratio (no unit designator) or percent (%).</p>
vertical-offset (optional)	<p>Specifies the vertical offset of the shadow, as pixels (px), percent (%), or ratio (no unit designator). Positive numbers adjust the shadow text downward.</p> <p>Default: 4 px.</p> <p>Example: <code><Shadow horizontal-offset="20px"... /></code></p>

Source

One or more Source elements may be added to a [Composition](#). Source elements by convention are the first elements in a composition, because it is common to change the file names each time a job is run.

You add one source element for each file you want to identify, to be used as material in a [Video](#), [Audio](#), and [Image](#) element.

Example

This snippet illustrates the typical use of a Source element—to reference an input file being used in one or more segments in the composition.

```
<Source identifier="1">  
  <File location="//share/path/Prime_Time_Show.mov" />  
</Source>
```

Elements (CML Hierarchy Map)

These optional elements may be added to a Source element. Elements with a limit (1 only) may be used multiple times.

- [File](#)
- [Crop](#) (1 only)
- [Mask](#) (1 only)
- [Mix](#)
- [Route](#)
- [DolbyE](#)
- [Subtitle](#)
- [Processor](#)

Attributes

These attributes may be applied to a Source element.

Table 29. Source Attributes

Name	Description and Values
identifier (required)	<p>An integer value, which must be unique among all Source elements.</p> <p>Example: <code><Source identifier="1"></code></p> <p>This identifier value is used in the <i>source</i> attribute of Canvas, Image, Title, Audio, and Advisory elements, to reference this specific file.</p>

Table 29. Source Attributes

Name	Description and Values
timecode (optional)	<p>Keyword to specify the type of timecode in the file. Default: <i>none</i>.</p> <p>Keywords: <i>none</i> <i>ltc</i> <i>vitc1</i> <i>vitc2</i> <i>atc</i></p> <p>Example: <code><Source timecode="vitc1"></code></p> <p><i>none</i>—an empty (" ") value indicates that the source does not contain a time code, and the first frame time is 00:00:00:00.</p> <p><i>ltc</i>—longitudinal timecode</p> <p><i>vitc1</i>—VITC timecode, in line 1</p> <p><i>vitc2</i>—VITC timecode, in line 2</p> <p><i>atc</i>—ancillary timecode</p>

Subtitle

Optionally, you can add one or more Subtitle elements to a [Source](#). The Subtitle element uses a File element to associate the source media file with a closed caption (.scc) file. If the media and the .scc file are not already timecode-aligned, they can be aligned in the composition directly, using the *offset* parameter.

Only one subtitle file can be played at a time; when multiple files (and subtitle files) are present, use the order attribute to specify the subtitle file to use.

See also [File](#).

Example

In this example, the subtitle file (HD_movie.scc) is assigned order number 1 (there's only one in this example), with an offset of 1 hour in relation to the source media file, HD_movie.mov.

```
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share/path/HD_movie.mov"/>
    <Subtitle order="1" offset="01:00:00:00">
      <File location="//share/path/HD_movie.scc"/>
    </Subtitle>
  </Source>
  <Sequence layer="0">
    <Segment>
      <Video source="1" />
    </Segment>
  </Sequence>
</Composition>
```

Timecode Example

In this example, the .scc file has a start timecode one hour later than the media file, which must be reconciled. In addition, the media is edited to have an in point 1 minute into the clip, which must also be taken into consideration.

- Media start timecode 00:00:00:00
- .scc file start timecode 01:00:00:00
- In point edit 00:01:00:00 (<Edit time="00:01:00:00"/>)
- Offset 01:00:00:00
- Result 01:01:00:00

The .scc file's offset value is added to the timecode of the media files' in point before it is applied to the .scc file. The result is that the edit point in the .scc file specifies the frame that is one minute into the .scc file, so that it aligns with the media:

00:01:00:00 + 01:00:00:00 = 01:01:00:00

In a second example, the media file has a one hour start time, but the .scc file starts at zero. The media file also has the in point 1 minute into the clip, as in the first example.

- Media start timecode 01:00:00;00
- .scc file start timecode 00:00:00;00
- In point edit 01:01:00;00 (<Edit time="00:01:00;00"/>)
- Offset 23:00:00;00
- Result 00:01:00;00

In this example, a 23 hour offset is added to the specified in point to create a matching in point in the .scc file at 0 hours (as in a 24-hour clock, where 2400 is actually 0000 hours):

01:01:00;00 + 23:00:00;00 = 00:01:00;00

Elements (CML Hierarchy Map)

One [File](#) can be added to a Subtitle element.

Attributes

These attributes may be applied to a Subtitle element.

Table 30. Subtitle Attributes

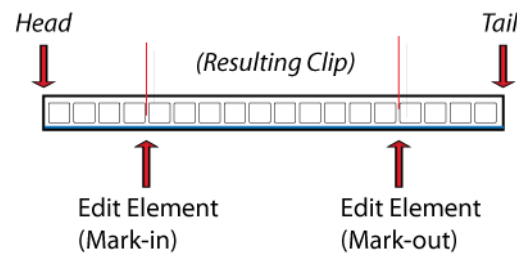
Name	Description and Values
offset (optional)	<p>The amount of is time required to align non-matching timecode references of source media and the closed caption file. The offset value is added to the media timecode reference to adjust for the difference. The offset has a range of 24 hours.</p> <p>Time values: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS:FF@FPS</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i><i>FF</i>) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated media file source.</p> <p>Positive offsets are indicated by adding the time to zero: a 10-minute offset is 00:10:00.000. Negative offsets are subtracted from 2400. Thus, a 1 hour negative offset s indicated as 23:00:00.000.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Example: <code><Subtitle offset = "01:43:05.333"... /></code></p> <p>Default: "00:00:00.000".</p>
order (optional)	<p>Integer (default: 0), which specifies the rank of this subtitle in relation to other subtitle files, if present. Only the highest-ranking subtitle file is actually played out.</p>

Tail

The Tail element is the point (or frame) on a material item (an Advisory, Image, Title, Canvas, or Media element) at the end (or mark-out point, when using an [Edit](#)) of the instance.

The Tail (like the [Head](#) and the [Body](#)) may contain effects elements (opacity, scaling translation, rotation and volume) that control the audio and video composition. If you do not supply a Fade with a duration, the effects change instantaneously with the time transition from the Body to the Tail—the Tail effects will not be applied to any video, because the end of the Body and Tail are at the same moment—the end of the file.

Figure 26. The Tail is the End of the File or the or the Tail's Edit Element



You define a Tail when you want to perform effects on the end of the material.

There are no attributes in a Tail element.

See also [Head](#), [Body](#).

Example

In this sample, the video is trimmed by 4 seconds at the end.

```
<Sequence layer="0">
  <Segment>
    <Image align="tail" adjust="body" fill="none" layer="2"
      location="\\share\path\1C_Arrow_Title-PNGs\frame-000508.png"
      frames="72" duration="00:00:00.000" layout="none">
    </Image>
    <Video adjust="edge" fill="none" source="1" layer="0">
      <Tail>
        <Edit time="00:00:04.000" />
      </Tail>
    </Video>
  </Segment>
```

Elements (CML Hierarchy Map)

One each of these elements may be optionally added to a Head, Body, or Tail element:

- [Edit](#)
- [Fade](#)

One each of these effects elements may be optionally added to a Head, Body, or Tail element:

- *Opacity*
- *Scaling*
- *Translation*
- *Rotation*
- *Volume*

Target

Optionally, one Target element may be added to a composition.

During the conforming process, the Conform action creates an uncompressed video stream and an uncompressed audio stream, containing up to 16 audio channels. The video and audio streams are then encoded as specified to produce the media file. The Target element can be used to specify the properties of the uncompressed media streams.

You should add and configure a Target when you want the properties to be specified directly in the composition. Otherwise, they are determined at run time, when the composition is conformed—based on the encoder and settings configured in the Conform action.

There are no elements in a Target.

Example

Here is the Example

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share\path\source_video.mov" />
  </Source>
  <Sequence layer="1">
    <Segment>
      <Video adjust="edge" fill="none" source="1" layer="0" />
    </Segment>
  </Sequence>
  <Target aspect-ratio="16:9" color-space="BT.709" frame-
rate="29.97" width="1920" height="1080" sample-rate="14000" />
</Composition>
```

Attributes

These attributes may be applied to a Target element. No default values are specified for any of these attributes.

Table 31. Target Attributes

Name	Description and Values
aspect-ratio (optional)	Specifies the aspect ratio of output media. Example: <Target aspect-ratio = "16:9"... /> It may be expressed as a ratio (4:3), a fraction (4/3), or a decimal (1.333).
channels (optional)	Specifies the number of audio channels in the output. Example: <Target channels = "6"... />

Table 31. Target Attributes

Name	Description and Values
color-space (optional)	Specifies the color space definition to use. Keywords: <i>BT.601</i> <i>BT.709</i> <i>BT.2020</i> Example: <Target color-space = "BT.709"... />
frame-rate (optional)	Specifies the frame rate, in frames per second. Example: <Target frame-rate = "29.97"... /> It may be expressed as a ratio, a fraction, or a decimal.
height (optional)	Specifies the height of the output frame as pixels (px), percent (%), or ratio (no unit designator). Example: <Target height = "1080"... />
sample-rate	Specifies the audio sample rate as an integer, in bits per second. Example: <Target sample-rate = "14000"... />
width (optional) (optional)	Specifies the width of the output frame, as pixels (px), percent (%), or ratio (no unit designator). Example: <Target width = "1920"... />

Title

When you want to display rendered text in your video, add a Title element to the segment. There may be multiple Title elements per segment.

The Title element provides a text string, with optional formatting. You specify the text string as the value of the element. For example: `<Title>Up Next...</Title>`.

The Title element has two optional child elements: [Area](#) and [Shadow](#). The Area element enables you to position the title text exactly where you want on the frame. The shadow element allows you to shadow the text.

Note: For a practical example of adding a title, see [Adding Titles](#).

See also [Area](#), [Shadow](#).

Example

Here is a Title sample with every attribute, plus an Area and Shadow element.

```
<Segment>
  <Title align="head" adjust="edge" fill="none" layer="2"
    duration="00:00:04.00" offset="00:00:01.000" font="Helvetica"
    size="50pt" style="italic" weight="bold"
    foreground-color="bisque" background-color="transparent"
    wrap="false" horizontal-align="left" vertical-align="top"
    overflow="overflow">
    <Area top="70%" left="20%" bottom="90%" right="80%" />
    <Shadow color="gold" softness="10%" vertical-offset="6%"
      horizontal-offset="6%" />This is My Story...</Title>
</Segment>
```

Elements (CML Hierarchy Map)

One each of these optional elements may be added to control the title text:

- [Area](#)—specifies the rectangle in which to render the text.
- [Shadow](#)—specifies an optional shadow on the text.

One each of these optional elements may be added to apply effects on a Title:

- [Head](#)
- [Body](#)
- [Tail](#)

Attributes

These attributes (some required, some optional) may be applied to a Title element. The rating and advisory attributes are required. Note that all attribute values must be surrounded by straight double quotes. For example, `<Advisory adjust="body" />`.

Table 32. Title Attributes

Name	Description and Values
adjust (optional)	<p>Specifies which portion of the material may be extended within the segment when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>edge</i> <i>body</i></p> <p>Example: <code><Title adjust = "body"... /></code></p> <p><i>edge</i> (default)—The material is adjusted by extending the edge that is <i>not</i> identified by the <i>align</i> property and is applied to the duration of the head and tail of the material.</p> <p><i>body</i>—The material is adjusted by extending the body of the material. It is applied to the end of the body that is <i>not</i> identified by the <i>align</i> property.</p>
align (optional)	<p>Specifies which end (or both) to (anchor) the material to the segment, when the duration of the material is less than the duration of the segment.</p> <p>Keywords: <i>head</i> <i>tail</i> <i>both</i>.</p> <p>Example: <code><Title align = "tail"... /></code></p> <p><i>head</i> (default)—Anchors the beginning of the material to the beginning of the segment.</p> <p><i>tail</i>—Anchors the end of the material to the end of the segment.</p> <p><i>both</i>—Anchors the center of the material at the center of the segment, so that the Head and Tail are equi-distant from the beginning and end of the segment.</p> <p>Note: If an offset is used in this situation, it is also applied to both ends.</p>

Table 32. Title Attributes

Name	Description and Values
background-color (optional)	<p>Specifies the color that the area is painted, over which the text is rendered. Default: <i>transparent</i>.</p> <p>Keywords: <i>transparent</i> <i>black</i> <i>white</i> <i>ARGB</i></p> <p>The supported colors are defined in .NET. For details, see Color Table.</p> <p>ARGB—The RGB + Alpha color, in 0xAARRGGBB format, where the high-order byte (AA) contains the alpha channel and the remaining bytes contain color components for red, green and blue, respectively. The alpha channel specifies the opacity of the pixel, where a value of 0x00 represents a pixel that is fully transparent and a value of 0xFF represents a fully opaque pixel.</p>
compose (required)	<p>Specifies that compositing of images should be performed at either the source or output resolution.</p> <p>Keywords: <i>output-size</i> <i>source-size</i></p> <p>Example: <Title compose="source-size"... /></p> <p>output-size—Renders the image in the output frame at the output frame size.</p> <p>source-size—Renders the image in the output frame at the input frame size.</p> <p>null (" ")—By default, Post Producer will use the optimal method of compositing. If you do not want to specify a size, specify an empty string: <Audio compose=""... />.</p> <p>Note: An As-run CML supplies this attribute in an As-run CML to specify how a given image sequence was rendered. This is helpful if you did not specify a value.</p>

Table 32. Title Attributes

Name	Description and Values
duration (optional)	<p>A time to specify the display time of the advisory image. The seconds value may be an integer or real number. Default: 0.</p> <p>Time values: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS:FF@FPS</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i> <i>FF</i>) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p>
fill (optional)	<p>When the length of the material is less than the length of the segment, the fill attribute determines if and how the item continues to play:</p> <p>Keywords: <i>none</i> <i>hold</i> <i>loop</i>.</p> <p>Example: <code><Title fill = "loop"... /></code></p> <p><i>none</i> (default)—extend with empty media (no visual or auditory material).</p> <p><i>hold</i>—continue playing the first or last frame to play the material the entire duration of the segment. If <i>align="head"</i>, the last frame is repeated. If <i>align="tail"</i>, the first frame is repeated.</p> <p><i>loop</i>—play the media repeatedly for the duration of the segment.</p>
font (optional)	<p>The family name of the font, which must be present on the system where the Vantage service executing the Conform action is executing. Default: Helvetica.</p> <p>For details, see .NET Font Family.</p>

Table 32. Title Attributes

Name	Description and Values
foreground-color (optional)	<p>Keyword or ARGB value to define the color that the font should be rendered in. Default: white.</p> <p>Keywords: <i>black</i> <i>transparent</i> <i>white</i></p> <p>Example: <code><Title foreground-color = "black"... /></code></p> <p>The supported colors are defined in .NET. For details, see Color Table.</p> <p>ARGB—The RGB + Alpha color, in 0xAARRGGBB format, where the high-order byte (AA) contains the alpha channel and the remaining bytes contain color components for red, green and blue, respectively. The alpha channel specifies the opacity of the pixel, where a value of 0x00 represents a pixel that is fully transparent and a value of 0xFF represents a fully opaque pixel.</p>
horizontal-align (optional)	<p>A keyword describing how the text is rendered on the horizontal plane, relative to the area. Default: left.</p> <p>Keywords: <i>left</i> <i>right</i> <i>center</i> <i>justify</i></p> <p>Example: <code><Title horizontal-align="right"... /></code></p>
layer (optional)	<p>An integer value describing the ordinal position in the composite of all material in this segment. Default: 0.</p> <p>Example: <code><Title layer = "10"... /></code></p> <p>Material with a lower layer value may be obscured by material with a higher layer value in the same segment.</p> <p>You should supply a layer value higher than the video layer—by default, layer is 0; thus, the video will obscure the graphic.</p>

Table 32. Title Attributes

Name	Description and Values
offset (optional)	<p>Specifies the amount of time to wait from the beginning of the segment, to display the material. Default 0.</p> <p>Time Value: <i>HH:MM:SS</i> <i>HH:MM:SS.sss</i> <i>HH:MM:SS:FF</i> <i>HH:MM:SS;FF</i></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (<i>sss</i>); frames (<i>;</i>FF) as standard (<i>:</i>) or drop-frame (<i>:</i>), and optional frame rate (<i>@FPS</i>), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Example: <code><Title duration = "00:00:05.000"... /></code></p>
overflow (optional)	<p>Specifies what to do when the rendered text extends beyond the bounds of the area: allow it to overflow, clip it, or resize the text downward until it fits. Default: <i>overflow</i>.</p> <p>Keywords: <i>overflow</i> <i>clip</i> <i>resize</i></p> <p>Example: <code><Title overflow="resize"... /></code></p>
size (optional)	<p>An integer value describing the nominal height of the font in points (pt). Default: 36 points.</p> <p>Example: <code><Title size="72pt"... /></code></p>
style (optional)	<p>Specifies the style in which the font face is rendered. Default: <i>normal</i>.</p> <p>Keywords: <i>normal</i> <i>italic</i> <i>underline</i> <i>strikeout</i> <i>overline</i></p> <p>Example: <code><Title style="italic"... /></code></p>
vertical-align (optional)	<p>Specifies where the text is rendered, relative to the area. Default: <i>top</i>.</p> <p>Keywords: <i>top</i> <i>middle</i> <i>bottom</i></p> <p>Example: <code><Title vertical-align="bottom"... /></code></p>

Table 32. Title Attributes

Name	Description and Values
weight (optional)	Specifies the thickness of the stroke of each character. Default: <i>normal</i> . Keywords: <i>normal</i> <i>bold</i> <i>light</i> <i>medium</i> <i>heavy</i> Example: <Title weight="medium"... />
wrap (optional)	Specifies how text is treated when it extends beyond the area: wrap it or not. Default: <i>false</i> . Boolean: (<i>true</i> <i>false</i>) Example: <Title wrap="false"... />

Translation

One Translation element may optionally be added to a [Body](#), [Head](#), or [Tail](#).

Add a Translation when you want to move the image along the X/Y axes. Positive values move the image to the right/bottom. Negative values move the image to the left/top.

If you don't supply a Translation element, the image is displayed in the upper left corner of the output frame.

For this effect to be of practical value in a Head or Tail element, the Fade element must be present, and its duration must be greater than 0.

The Translation element is often used in conjunction with the [Scaling](#).

There are no elements in a Translation element.

See also [Volume](#), [Opacity](#), [Scaling](#), [Rotation](#).

Example

The following composition places an image onto an underlying video clip. It locates the upper left hand corner of the image at a location, measured from the top, left corner of the frame that is 80% from the left edge and 70% from the top edge.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="\\share\path\Segment 1.mov" />
  </Source>
  <Source identifier="2">
    <File location="\\share\path\Segment 2.mov" />
    <Mask left="12.5%" right="87.5%" top="0%" bottom="100%"/>
  </Source>
  <Sequence layer="0">
    <Segment>
      <Image align="both" adjust="body" fill="loop" layer="1"
location="\\share\path\Vantage-logo_small.png" frames="1"
duration="00:00:15.000" offset="00:00:02.000" layout="none">
        <Head>
          <Translation x="80%" y="70%" />
          <Opacity level="0%"/>
          <Fade duration="00:00:01.000"/>
        </Head>
        <Body>
          <Translation x="80%" y="70%" />
          <Opacity level="100%"/>
        </Body>
        <Tail>
          <Translation x="80%" y="70%" />
          <Opacity level="0%"/>
          <Fade duration="00:00:01.000"/>
        </Tail>
      </Image>
    </Segment>
  </Sequence>
</Composition>
```



```

</Image>
<Video source="1" layer="0" >
  <Tail>
    <Edit time="00:00:8.933"/>
  </Tail>
</Video>
</Segment>
</Sequence>
</Composition>

```

Attributes

These attributes may be applied to a Translation element. Although both are optional, supplying neither would be impractical, and result in no change to the location.

Table 33. Translation Attributes

Name	Description and Values
x (optional)	<p>Specifies the horizontal translation relative to the left edge of the frame, as a ratio (no designator), a percent (%), or as an absolute position in pixels (px).</p> <p>Default: 0.</p> <p>Example: <Translation y="9%" x="16%"/></p>
y (optional)	<p>Specifies the vertical translation relative to the top edge of the frame, as a ratio (no designator), a percent (%), or an absolute position in pixels (px).</p> <p>Default: 0.</p> <p>Example: <Translation y="9%" x="16%"/></p>

Video

The Video element adds video (with audio, if present) to a [Segment](#), from a file identified by a specific [Source](#).

A segment's length is defined by the duration of the longest material element within the segment. A Video element typically defines the Segment length; other image, title or video overlays that are shorter in length are superimposed on the underlying video.

Picture-in-picture effects can be created by scaling and translating video clips on upper layers within the segment.

Examples

This CML snippet illustrates the basic use of a video clip in a segment. This example plays a 10-second video clip, with 1-second fades at the beginning and end, in a segment.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share\path\Post_Prod_Summary.mov" />
  </Source>
  <Sequence layer="0">
    <Segment>
      <Video align="head" adjust="body" fill="none" source="1"
layer="0">
        <Head>
          <Edit time="00:00:00:00" />
          <Opacity level="0%"/>
          <Fade duration="00:00:01.000"/>
        </Head>
        <Body>
          <Opacity level="0%"/>
        </Body>
        <Tail>
          <Edit time="00:00:10:00" />
          <Opacity level="0%"/>
          <Fade duration="00:00:01.000"/>
        </Tail>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

In this example, Source 2—the credit roll—is resized, repositioned, and sped up to play in half the time of its original duration.

When rate is employed, frames are added to or removed from the clip to achieve the desired duration, at the frame rate specified for the entire composition. For example:

- Source 2 is 2 seconds long with a frame rate of 15 FPS for a total of 30 frames

- The composition output frame rate is 59.94 FPS
- The rate setting is =200% for this source

```
<Composition>
  <Source identifier="1">
    <File location="//share\path\Background.mov" />
  </Source>
  <Source identifier="2">
    <File location="//share\path\creditroll.mov" />
  </Source>
  <Sequence>
    <Segment>
      <Video align="head" adjust="edge" fill="none" source="1"
layer ="0" />
      <Video align="head" adjust="edge" fill="none" source="2"
layer="1" rate="200%">
        <Head>
          <Scaling x="30%" y="30%"/>
          <Translation x="60%" y="60%"/>
        </Head>
        <Body>
          <Scaling x="30%" y="30%"/>
          <Translation x="60%" y="60%"/>
        </Body>
        <Tail>
          <Scaling x="30%" y="30%"/>
          <Translation x="60%" y="60%"/>
        </Tail>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

During encoding, the source is converted to 120 frames (2 seconds at 60 FPS) and the rate setting is applied to reduce the total frame count to 60 so that it plays for 1 second: 50% of the original duration.

Elements (CML Hierarchy Map)

One each of these optional elements may be added to a Video element:

- *Head*
- *Body*
- *Tail*

Attributes

These attributes (some required, some optional) may be applied to a Video element.

Note: The *adjust*, *align*, and *fill* attributes are closely related. Together, they specify how the material is aligned, and how it is adjusted along the timeline in the segment.

Table 34. Video Attributes

Name	Description and Values
adjust (optional)	<p>When the duration of the material is less than the duration of the segment, the material may be extended within the segment.</p> <p>This provides a flexible method to define the behavior of overlay material so that it is consistently applied, regardless of the duration of the underlying content. For example, using the <i>adjust</i> attribute, you can consistently display a branding bug on underlying video of arbitrary duration.</p> <p>There are 3 things to consider when modifying material:</p> <ol style="list-style-type: none"> 1. What to adjust—the body or the edge (using <i>adjust</i>) 2. Which end to adjust from (based on <i>align</i> attribute) 3. How to adjust—hold, loop, none (based on <i>fill</i> attribute). <p>Keywords: <i>edge</i> (default) <i>body</i></p> <p>Example: <code><Video adjust = "body"... /></code></p> <p><i>edge</i> (default)—The material is adjusted by extending the edge that is <i>not</i> identified by the <i>align</i> property and is applied to the duration of the head and tail of the material.</p> <p><i>body</i>—The material is adjusted by extending the body of the material. It is applied to the end of the body that is <i>not</i> identified by the <i>align</i> property.</p>

Table 34. Video Attributes

Name	Description and Values
align (optional)	<p>When the duration of the material is less than the duration of the segment, use the appropriate keyword to pin (anchor) the material to the segment as you intend.</p> <p>There are 3 things to consider when modifying material:</p> <ol style="list-style-type: none"> 1. What to adjust—the body or the edge (using <i>adjust</i>) 2. Which end to adjust from (based on <i>align</i> attribute) 3. How to adjust—hold, loop, none (based on <i>fill</i> attribute) <p>Keywords: <i>head</i> <i>tail</i> <i>both</i>.</p> <p>Example: <code><Video align = "tail"... /></code></p> <p>Default: <i>head</i>.</p> <p><i>head</i> (default)—Anchors the beginning of the material to the beginning of the segment.</p> <p><i>tail</i>—Anchors the end of the material to the end of the segment.</p> <p><i>both</i>—Anchors the center of the material at the center of the segment, so that the Head and Tail are equi-distant from the beginning and end of the segment.</p> <p>Note: If an offset is used in this situation, it is also applied to both ends.</p>
fill (optional)	<p>When the length of the material is less than the length of the segment, the fill attribute determines if and how the item continues to play:</p> <p>Keywords: <i>none</i> (default) <i>hold</i> <i>loop</i></p> <p>Example: <code><Video fill = "loop"... /></code></p> <p>There are 3 things to consider when modifying the material:</p> <ol style="list-style-type: none"> 1. What to adjust—the body or the edge (using <i>adjust</i>) 2. Which end to adjust from (based on <i>align</i> attribute) 3. How to adjust—hold, loop, none (based on <i>fill</i> attribute) <p><i>none</i> (default)—extend with empty media (no visual or auditory material). From a practical perspective, no extension.</p> <p><i>hold</i>—continue playing the first or last frame the entire duration of the segment. If <i>align</i>="head", the last frame is repeated. If <i>align</i>="tail", the first frame is repeated.</p> <p><i>loop</i>—play the media repeatedly for the duration of the segment.</p>

Table 34. Video Attributes

Name	Description and Values
layer (optional)	<p>An integer value describing the ordinal position in the composite of all material in this segment. Default: 0.</p> <p>Example: <code><Video layer = "10"... /></code></p> <p>Material with a higher layer value obscures material with a lower layer value in the same segment.</p>
offset (optional)	<p>Specifies a time value which changes the location of the beginning or end of the material, relative to the segment.</p> <p>The offset is applied to the end of the content that is specified by the align property. If align="both" the offset is applied at the defined value to both the head and tail of the material.</p> <p>Default: 0.</p> <p>Time Value: <code>HH:MM:SS</code> <code>HH:MM:SS.sss</code> <code>HH:MM:SS:FF</code> <code>HH:MM:SS;FF</code></p> <p>Time is measured on a 24-hour clock. Time may include milliseconds (sss); frames (;FF) as standard (:) or drop-frame (;), and optional frame rate (@FPS), as an integer, decimal number, or rational (for example, 30000/1001). If FPS is not specified, it is inferred from the associated source.</p> <p>Rates supported include: 23.976, 24, 25, 29.97, 30, 50, 59.94.</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>A semicolon preceding the FF field indicates drop-frame format: The first 2 (29.97 fps) or 4 (59.94 fps) time code numbers are dropped every minute except minutes divisible by 10. Drop frame format is equivalent to non-drop frame format for all other frame rates.</p> <p>Example: <code><Video offset="00:10:00"... /></code></p>
rate (optional)	<p>Specifies the frame rate of the video within a composition, relative to its original speed. May be specified as a percent (% units designator), or as a ratio (no units designator.) You can not specify an absolute speed.</p> <p>Used for changing the duration of a video clip, such as shortening the time required to roll credits in a promo.</p> <p>When the original rate is changed, volume (if any) is set to 0.</p> <p>Example: <code><Video rate="250%"... /></code></p> <p>In this example, this video is played at 2.5 times the original speed.</p>

Table 34. Video Attributes

Name	Description and Values
source (required)	Integer, corresponding to the Source element's identifier value, to identify the file to be used in this instance.

Volume

The Volume element is an element that allows you to adjust the volume of audio in certain material.

One Volume element may optionally be added to a [Body](#), [Head](#), or [Tail](#) in a [Video](#) or an [Audio](#) element.

It is important to note that the default value in a Body element is 100%, while in a Head or Tail, it is 0%. For this effect to be of practical value, the Fade element must be present, and its duration must be greater than 0. If you are using Fade for effects other than volume (that is, you do not want to fade the volume), you must supply a volume level explicitly.

You can adjust volume by percent, or by decibels. Positive values apply a gain, negative values attenuate the audio. The level is applied equally to all channels targeted by default or by a [Mix](#). (This element does not effect any channels targeted by a [Route](#).)

There are no elements in a Volume element.

See also [Opacity](#), [Scaling](#), [Translation](#), [Rotation](#).

Example

This example applies a 1-second audio fade-in and fade-out to the beginning and end of the clip.

```
<?xml version="1.0" encoding="utf-8"?>
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="\\share\path\Post Prod Summary.mov" />
  </Source>
  <Sequence layer="0">
    <Segment>
      <Video align="head" adjust="body" fill="none" source="1"
layer="0">
        <Head>
          <Edit time="00:00:00:00" />
          <Volume level="0%" />
          <Fade duration="00:00:01.000" />
        </Head>
        <Body>
          <Volume level="0%" />
        </Body>
        <Tail>
          <Edit time="00:00:10:00" />
          <Volume level="0%" />
          <Fade duration="00:00:01.000" />
        </Tail>
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

</Composition>

Attributes

These attributes may be applied to a Volume element.

Table 35. Volume Attributes

Name	Description and Values
level (optional)	A positive or negative integer or decimal number representing the volume level as a ratio (no designator), percent (%) or in decibels (dB) of the original material. Default (in Head and Tail elements): 0%. Default (in Body elements): 100%. Example: <Volume level="30%" />

Post Producer Formats

This chapter identifies the supported input formats for both video and audio, and images, and the supported output formats for the Conform action.

- [Input Video Formats](#)
- [Input Audio Formats](#)
- [Input Image Formats](#)
- [Output Formats](#)

Input Video Formats

The following table lists the video formats that the Conform action can process, and their containers..

Table 36.

Container	DV ¹	IMX	MPEG2 ²	ProRes	DNxHD	AVCI	J2K	H.2 64
TIFO	•	•		•	•			
AVI	•							
LXF			•					
MOV	•	•		•	•			•
MP4								•
GXF	•	•	•			•		
MPEG2 PS			•					
MPEG2 TS			•					
MXF OP-1a	•	•	•		•	•	•	
MXF AS02 ³	•	•	•			•	•	

1. DV includes DV25, DVCPro25, DV50/DVCPro50, and DVCProHD.

2. MPEG2 includes Long-GOP and I-Frame only formats.

3. Files produced by Omneon Media Subsystem (6.1) and Amberfin ICR (7.8) were tested.

Input Audio Formats

The following table lists the audio formats that the Conform action can process, and their containers.

Table 37. Post Producer Audio Input Formats

Container	PCM	MPEG	302M	331M	AAC	AC3	Dolby E
TIFO	•						•
AC3						•	
AVI	•						
LXF	•						
MOV	•				•		•
MP4	•				•		•
GXF	•						•
MPEG2 PS		•					
MPEG2 TS	•	•	•		•		•
MXF OP-1a	•			•			•
MXF AS02 ¹	•						•
WAVE	•						•

1. AC3 includes Dolby Digital and Dolby Digital Plus (E-AC3).

Input Image Formats

Supported raster image formats are:

- PNG
- JPEG
- TIFF
- TARGA

Output Formats

The following table lists the output formats that the Conform action can produce, and their containers. The output files produced by the Conform action may be distributed directly, or they can be utilized by any Vantage encoder (in Flip, IPTV VOD, or Multiscreen actions) to produce any supported output format.

Table 38. Post Producer Video Input Formats

Container	ProRes	DNxHD	JP2K	MP1	MP2	H.264	Uncompressed
TIFO	•	•	•	•	•	•	•
MOV	•	•	•	•	•	•	•
MP4	•	•	•	•	•	•	•

Index

A

ad insertion **25**
 Advisory
 Advisory Filter, enabling in Workflow Designer **97**
 rating reasons table **104**
 rating system ratings table **103**
 rating systems **103**
 Advisory Filter, enabling in Workflow Designer **97**
 Anystream (AN2) files, processing in Post Producer **25**
 Area element, specifications **105**
 As run CML, workflow **94**
 Audio element, specifications **107**
 audio input formats **189**

B

BlackArrow SCTE-35 ad insertion workflow **94**
 Body element, specifications **111**

C

Canvas element, specifications **112**
 Chronicle action, generally **25**
 CML file, processing **28, 45**
 CML hierarchy map **96**
 comments, making **15**
 Compose action, generally **25**
 composition development, approaches to **37**
 Composition element, specifications **116**
 composition hierarchy **30**

composition hierarchy, simplified **30**
 Composition Markup Language, specifications **95**
 compositions, generally **29**
 Conform action
 generally **24**
 supported formats **25**
 contact information **15**
 copyright notice **3**
 Crop element, specifications **117**

D

decibels, specifying values as **41**
 DolbyE
 Mix element in **143**
 Route element in **150**
 DolbyE element, specifications **121**

E

Edit element, specifications **123**

F

Fade element, specifications **125**
 File element, specifications **128**
 formats
 audio input **189**
 output **190**
 video input **188**

H

Head element, specifications **129**

I

Image element, specifications **131**
input formats **188**

J

job, submitting **28, 45**

L

Lightspeed server **24**

M

Manzanita Systems **25**
Mask element, specifications **140**
material, generally **33**
Mix element, specifications **143**
MOV format **25**
MP4 format **25**
MPEG disclaimers **6**

N

notices, legal, generally **3**

O

Opacity element, specifications **145**
output formats **190**

P

percent, specifying values as **41**
pixels, specifying values as **41**
points, specifying values as **41**
Post Producer
 actions **24**
 Basic conforming workflow **27**
 Chronicle action, generally **25**
 Compose action, generally **25**
 Conform action, generally **24**
 described, generally **18**
 how it works, generally **20**
 using for TSEDL and AN2 (Anystream)
 file processing **25**
 using in Vantage **23**
Post Producer, generally **18**
processing a CML file, generally **28, 45**
Processor element, specifications **147**

Q

QuickTime MOV format **25**
quotes, straight, required in XML **41**

R

rating reasons table **104**
rating system ratings table **103**
rating systems **103**
ratio, specifying values as **41**
Rotation element, specifications **148**
Route element, specifications **150**

S

Scaling element, specifications **152**
SCTE35 **25**
SCTE-35 BlackArrow ad insertion
 workflow **94**
Segment element, specifications **154**
segments, generally **32**
Sequence elements, specifications **156**
sequences, generally **31**
Shadow element, specifications **158**
Source element, specifications **160**
straight quotes, using in CML **41**
Subtitle element, specifications **162**

T

Tail element, specifications **165**
Target element, specifications **167**
tech support, obtaining **15**
Telestream
 contacting **8, 15**
 International **15**
 sales and marketing **15**
 technical support **15**
 Web site **15**
Telestream, contacting **15**
TIFO format **25**
Title element, specifications **169**
Title, Area element in **105**
Title, Shadow element in **158**
trademark notices **3**
Translation element, specifications **176**
TSEDL files, processing in Post
 Producer **25**

U

unit designators, using to define attribute values **41**

V

Vantage SDK **28**

Vantage, using Post Producer in **23**

Video element, specifications **178**

video input formats **188**

Video Processing Library **25**

Volume element, specifications **184**

W

warranty **7**

Warranty and Disclaimers **7**

workflow

as run CML **94**

Workflow Designer, Post Producer actions in **24**

workflows

BlackArrow SCTE-35 ad insertion **94**

