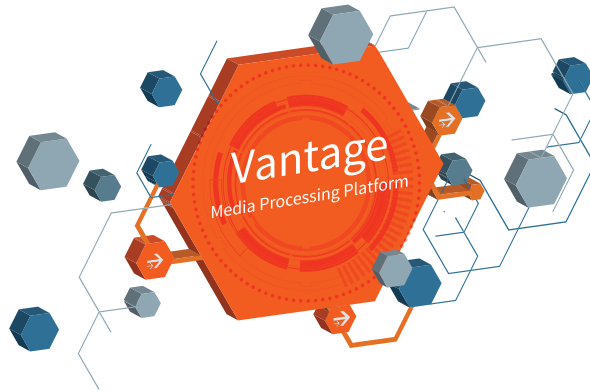




Timed Text CML  
Developer Guide



---

## Timed Text CML Developer Guide

---

Vantage 8.0 UP3 and later  
Timed Text Flip 8.0.6 and later



## Copyrights and Trademark Notices

**Telestream.** Copyright © June 2021 Telestream, LLC and its Affiliates. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, altered, or translated into any languages without written permission of Telestream, LLC. Information and specifications in this document are subject to change without notice and do not represent a commitment on the part of Telestream. Specifications subject to change without notice.

Telestream, CaptionMaker, Cerify, Episode, Flip4Mac, FlipFactory, Flip Player, Gameshow, GraphicsFactory, Lightspeed, MetaFlip, Post Producer, Prism, ScreenFlow, Split-and-Stitch, Switch, Tempo, TrafficManager, Vantage, VOD Producer, and Wirecast are registered trademarks and Aurora, Cricket, e-Captioning, Inspector, iQ, iVMS, iVMS ASM, MacCaption, Pipeline, Sentry, Surveyor, Vantage Cloud Port, CaptureVU, Cerify, FlexVU, Prism, Sentry, Stay Genlock, Aurora, and Vidchecker are trademarks of Telestream, LLC and its Affiliates. All other brand, product, and company names are the property of their respective owners and are used only for identification purposes.

**Adobe.** Adobe® HTTP Dynamic Streaming Copyright © 2014 Adobe Systems. All rights 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.

**CoreOS.** Developers of ETCD.

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

**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|Server 2016, Windows 7, Windows 8, Windows 10, Media Player, Media Encoder, .Net, Internet Explorer, SQL Server 2005|2008|Server 2012, and Windows Media Technologies are trademarks of Microsoft Corporation.

**NLOG, MIT, Apache, Google.** NLog open source code used in this product under MIT License and Apache License is copyright © 2014-2016 by Google, Inc., © 2016 by Stabzs, © 2015 by Hiro, Sjoerd Tieleman, © 2016 by Denis Pushkarev, © 2015 by Dash Industry Forum. All rights reserved.

**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.

**Swagger.** Licensed from SmartBear.

**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  
Android™



**SCREEN**  
SYSTEMS



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, LLC (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.

## Warranty Remedies

Your sole remedies under this limited warranty are as follows:

*Hardware and 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, LLC 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, LLC 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, LLC  
848 Gold Flat Road  
Nevada City, CA 95959 USA

You can call Telestream during U. S. business hours via telephone at (530) 470-1300.

## Regulatory Compliance

**Electromagnetic Emissions:** FCC Class A, EN 55022 Class A, EN 61000-3-2/-3-3, CISPR 22 Class A

**Electromagnetic Immunity:** EN 55024/CISPR 24, (EN 61000-4-2, EN 61000-4-3, EN 61000-4-4, EN 61000-4-5, EN 61000-4-6, EN 61000-4-8, EN 61000-4-11)

**Safety:** CSA/EN/IEC/UL 60950-1 Compliant, UL or CSA Listed (USA and Canada), CE Marking (Europe)

**California Best Management Practices Regulations for Perchlorate Materials:**  
This Perchlorate warning applies only to products containing CR (Manganese Dioxide) Lithium coin cells. Perchlorate Material-special handling may apply. See [www.dtsc.ca.gov/hazardouswaste/perchlorate](http://www.dtsc.ca.gov/hazardouswaste/perchlorate).

# Contacting Telestream

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

Resource	Contact Information
Vantage Technical Support	Web Site: <a href="http://www.telestream.net/telestream-support/Vantage/support.htm">http://www.telestream.net/telestream-support/Vantage/support.htm</a> Support Email: <a href="mailto:support@telestream.net">support@telestream.net</a> Enterprise Telephone Support: U. S. Toll Free: (877) 257-6245 U. S. from outside U.S.: (530) 470-2036 Europe   Middle East   Africa   Asia   Pacific: +49 228 280 9141 Terms and times of support services vary, per the terms of your current service contract with Telestream.
Vantage Information, Assistance, FAQs, Forums, & Upgrades	Web Site: <a href="http://www.telestream.net/telestream-support/Vantage/support.htm">http://www.telestream.net/telestream-support/Vantage/support.htm</a> Support Email: <a href="mailto:support@telestream.net">support@telestream.net</a>
Telestream, LLC	Web Site: <a href="http://www.telestream.net">www.telestream.net</a> Sales and Marketing Email: <a href="mailto:info@telestream.net">info@telestream.net</a> Telestream, LLC 848 Gold Flat Road, Suite 1 Nevada City, CA USA 95959
International Distributor Support	Web Site: <a href="http://www.telestream.net">www.telestream.net</a> See the Telestream Web site for your regional authorized Telestream distributor.
Telestream Technical Writers	Email: <a href="mailto:techwriter@telestream.net">techwriter@telestream.net</a> 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.





# Contents

Copyrights and Trademark Notices	2
Contacting Telestream	7

## Timed Text CML Overview 11

Introduction	12
Timed Text CML Requirements and Limitations	14
Troubleshooting	15

## Timed Text Conform Applications 17

Timed Text Conform Workflow	19
Timed Text Conform Action Configuration Panel	20
Timed Text CML for Conforming Caption or Subtitle Sidecar Files	22
Timed Text CML for Conforming Captions from Media Files	25
Timed Text CML for Conforming Captions/Subtitles with Associated Movie Files	27
Advanced Workflow Combining Timed Text Conform, Post Producer Conform, and Timed Text Flip	30

## Timed Text CML Reference 31

Timed Text CML Elements in Alphabetic Order	31
Timed Text CML Hierarchy Map	32
Timed Text CML Implementation Details	33
Case Sensitivity is Critical	33
Non-Timed Text CML is Ignored	33
Specifying File References in Timed Text CML	33
Comment	34
Example	34
Composition	35
Child Elements	35
Attributes	35
Example	35
Edit	36

Attributes	36
Example	37
File	38
Attributes	39
Example	39
Head	41
Child Elements	41
Example	41
Segment	42
Child Elements	42
Example	42
Sequence	43
Child Elements	43
Example	43
Source	45
Child Elements	45
Attributes	46
Example	46
Subtitle	47
Child Elements	47
Application	47
Example	48
Tail	49
Child Elements	49
Example	49
Target	50
Child Elements	50
Example	50
Timecode	51
Attributes	51
Example	51
Video	52
Child Elements	52
Attributes	52
Example	52

# Timed Text CML Overview

Timed Text Composition Markup Language (CML) is a specialized dialect of the Vantage family of CML, which describes media in a manner suitable for stitching associated caption | subtitle files in preparation for further processing in Vantage workflows. Timed Text CML is ideally suited for automating repetitive, caption file stitching tasks in a production environment.

---

**Note:** In the context of this guide, the terms caption and subtitle are used interchangeably.

---

Timed Text is an optional, licensed feature of Vantage. A Timed Text Flip license is required to utilize Timed Text CML in Vantage Timed Text workflows.

The purpose of this guide is to help producers, editors, operators, and others involved in media processing learn how to use Timed Text Composition Markup Language to automatically concatenate—*stitch*—separate captions together to generate a new caption file that corresponds to the media clip specifications, directly in Timed Text workflows.

---

**Note:** Users familiar with CML will recognize Timed Text CML fundamentally as a dialect of Post Producer CML, with variations designed specifically for caption stitching using Timed Text Conform actions.

---

Learning and using Timed Text CML requires familiarity with both Vantage and the Timed Text Conform action. Because the language is implemented in XML format, a basic understanding of XML is also helpful.

In this chapter, you'll learn what Timed Text CML is designed to do, and how it works.

- [Introduction](#)
- [Timed Text CML Requirements and Limitations](#)
- [Troubleshooting](#)

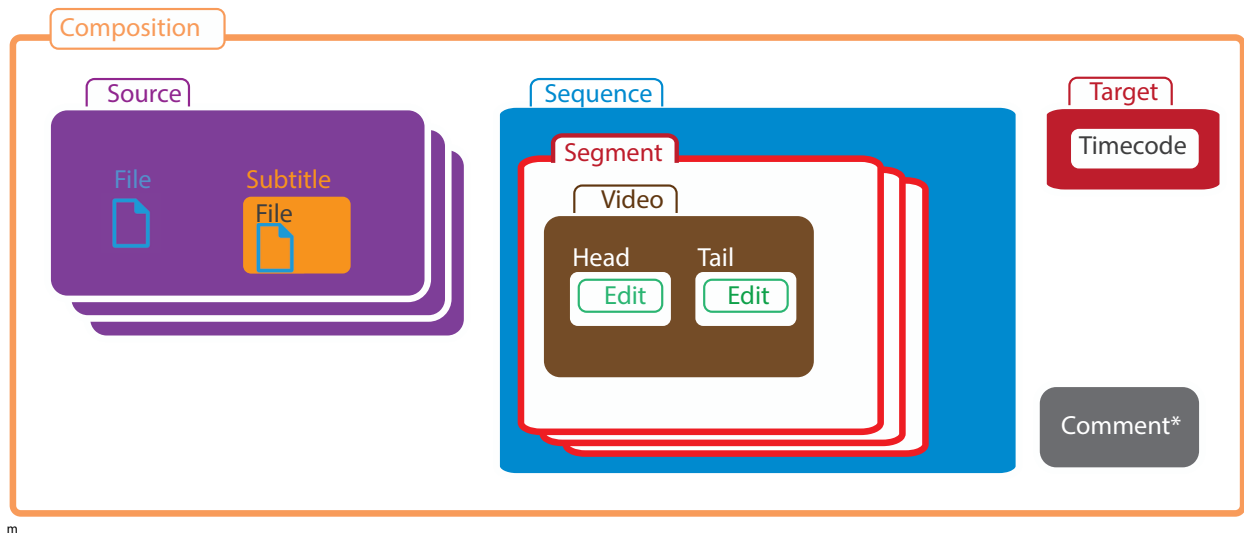
# Introduction

Timed Text Composition Markup Language (TTCML) is a media definition language designed specifically for utilizing Telestream's Timed Text Conform action in Vantage workflows to trim and stitch captions files to match the associated media trimming.

Timed Text CML is an efficient way to connect multiple, sequential caption file segments using a Timed Text workflow to produce a single captions file that aligns with the associated media.

A typical application is to compile a thirty-minute program with a bumper (remapping audio if required), three segments with ads and a trailer, and submit them with their associated captions files to a Timed Text workflow that trims and stitches them to produce a single captions output file. This file may be submitted with its associated media to a Flip64, Post Producer and/or Timed Text Flip workflow for further processing.

Timed Text CML uses XML syntax. Each of its elements identifies a particular aspect of media, along with attributes to configure it as required. The elements function as building blocks, enabling you to create and organize the components to describe the media you want to generate. This diagram illustrates the relationship between the elements. Click on any element to learn about it:



Stacked elements indicates multiple instances are permitted.

**Note:** Timed Text CML processing should be approached from a design and implementation perspective as a separate, pre-processing step, performed directly in Timed Text Conform before transcoding the media and the captions file generated by Timed Text Conform. When Timed Text ingests CML, it parses it to generate (conform) the specified caption file for processing by other Vantage actions. Thus, in the context of this guide, the term *output* represents the dynamically-generated output from

Timed Text Conform; not the output of Timed Text Flip. The output of Timed Text Conform is *input* only in the context of the Timed Text Flip.

---

## Timed Text CML Requirements and Limitations

The following requirements and limitations apply to Timed Text CML processing:

- CML files should have an appropriate extension (for example, `MyTTCMLfile.xml` or `MyTTCMLfile.cml`) to be easily recognized as a CML file and processed properly in Timed Text Conform.
- Timed Text only processes files on Windows (NTFS) platforms—using a drive letter or UNC path. If your workflow is watching for new files on a server other than a Windows server, you must add a Copy or other transport action to locate the file on a Windows server accessible to the Timed Text service processing it.

# Troubleshooting

During the design phase, it's typical to discover bugs in your CML. One excellent way of isolating the problem is to use the CML [Comment](#) element or XML comments `<!--` and `-->` to surround portions of your CML to prevent it from executing, in order to isolate the problem code.

As you design, develop and process Timed Text CML, here are some common problems you may encounter:

- CML vs Timed Text Conform—Determine whether the error is originating from the CML or in the Timed Text Conform action directly.
- XML formatting—make sure that your CML conforms to XML syntax.
- Ignored elements and attributes—elements and attributes that are misspelled or capitalized incorrectly are ignored. Make sure keywords are spelled and capitalized correctly as well.
- Smart quotes—be sure to use straight double quotes to surround values.
- Segments missing or out of order—Segments are placed on the Sequence timeline, based on their ordinal position in the CML—their order is not explicitly stated by an attribute value. Make sure your elements are spelled and capitalized correctly. Make sure your paths are correct.
- Cannot access files referenced in the CML—Only files located on Windows servers and accessible to the Vantage Timed Text service can be used.





# Timed Text Conform Applications

This chapter provides examples of compositions and associated workflows that may be used in a wide range of Timed Text Conform applications using Timed Text CML.

## Topics

- [Timed Text Conform Workflow](#)
- [Timed Text CML for Conforming Caption or Subtitle Sidecar Files](#)
- [Timed Text CML for Conforming Captions from Media Files](#)
- [Timed Text CML for Conforming Captions/Subtitles with Associated Movie Files](#)
- [Advanced Workflow Combining Timed Text Conform, Post Producer Conform, and Timed Text Flip](#)

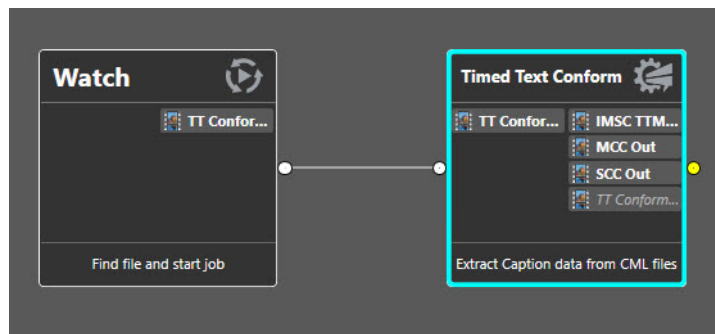
All of the CML examples in this guide are complete compositions. To test them, you can copy and paste the example into a text editor and change the references to the media files you are supplying then save the file. In your Vantage domain, configure workflows using Timed Text Conform, then activate them for submission of your Timed Text CML files. Sample Timed Text CML and associated media can be found under [Vantage Timed Text Workflow Examples](#) located on the Telestream website.

Timed Text CML files support a subset of elements and commands typically associated with CML creation and configuration that you are already familiar with when using Vantage Post Producer and NLEs such as Adobe Premiere. See the [Timed Text CML Reference](#) for the extent and limitation of supported CML syntax for Timed Text CML. Post Producer and Flip64 can be used in conjunction with Timed Text CML and Timed Text Conform actions but require uniquely configured CML input files which comply with the specific syntax limitations of those actions. See the Flip64 and Post Producer CML developer guides, available on the Telestream website, for configuration details.

## Timed Text Conform Workflow

This workflow is a basic Timed Text Conform workflow, where you submit a composition file for caption/subtitle stitching.

You can use this basic workflow as the basis of all other caption/subtitle conforming workflows. Add and configure other actions to create a workflow that meets your operational requirements.

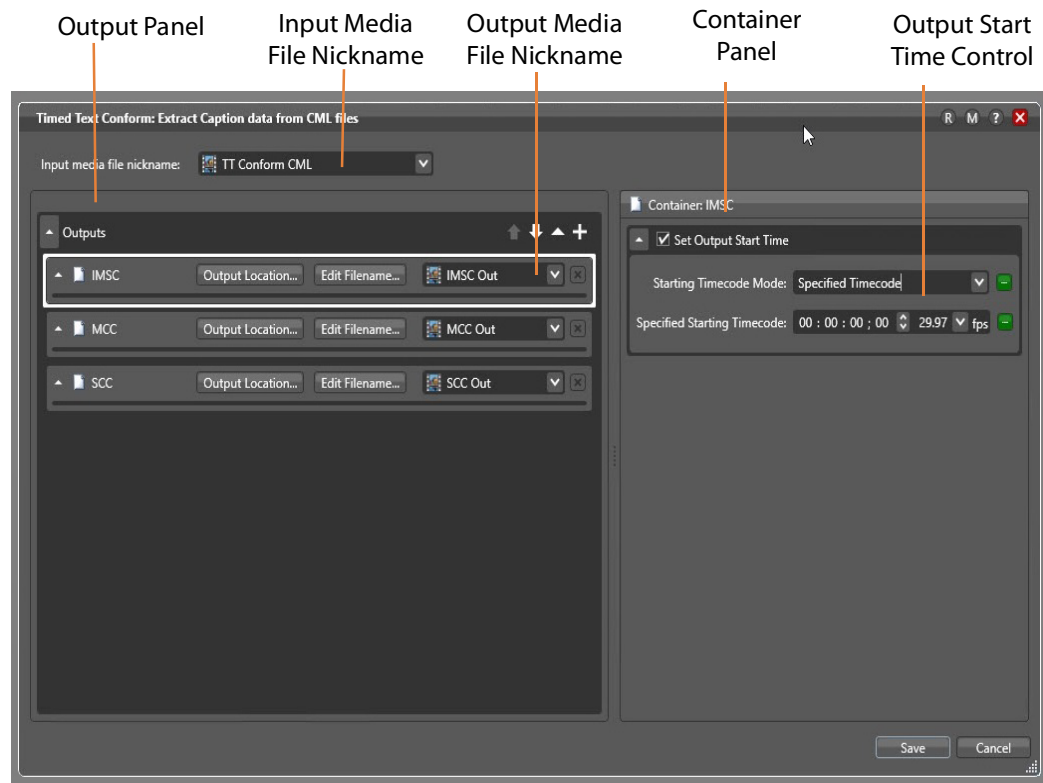


This workflow utilizes the following actions to perform these tasks:

*Watch*—The Watch action starts a job when you submit a composition file for processing. You can configure the Watch action to monitor a wide variety of systems for files; commonly a conforming workflow monitors a Windows computer or share for CML files. Under the File Match Pattern, you typically 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.

*Timed Text Conform*—The Timed Text Conform action processes the Timed Text CML file, generating a stitched SCC, MCC, or IMSC TTML output file. Downstream actions of more advanced workflows can utilize this output file (see [Advanced Workflow Combining Timed Text Conform, Post Producer Conform, and Timed Text Flip](#) workflow) or it can be re-purposed for use outside of Vantage. Configuration is similar to other Vantage actions you are familiar with and includes the option to set Output Start Time based on CML source timecode or user specified values.

## Timed Text Conform Action Configuration Panel



**Output Panel.** Specify output caption or subtitle type: SCC | MCC | IMSC (TTML).

**Input Media File Nickname.** Specifies the CML file to process, by nickname. The file must have been ingested or created upstream in this workflow and identified by a nickname.

---

**Note:** Only properly formatted CML files are supported as inputs. Attempting to submit other file types will return an error.

---

**Output Media File Nickname.** Specifies the nickname to assign to the output caption or subtitle file.

**Container Panel.** Configures container options for selected files.

**Set Output Start Time.** When unchecked (default), a zero-based start timecode is applied in the output. When checked, allows selection of a Starting Timecode Mode.

- **Starting Timecode Mode.** Specify Starting Timecode Mode to use:
  - **Specified Timecode.** Specified value and framerate is applied to the output. This is similar to a ripple timecode feature so for timecode values starting at zero, the amount of time specified in the Specified Starting Timecode field is added.
  - **Use CML Target Timecode.** The timecode value is transferred from the CML to the output. If the CML does not have a timecode, a default, zero-based timecode is used.

## Timed Text CML for Conforming Caption or Subtitle Sidcar Files

Multiple caption or subtitle sidcar files may be stitched together, in whole or in part, using a properly configured CML ingested by Timed Text Conform.

---

**Note:** Timed Text CML does not allow the combined specification of caption and subtitle files for stitching in the same CML. All sources must be either captions or subtitle file types. Use Timed Text Flip to convert captions to subtitles or vice versa.

---

In this example, there are four caption files (`mccfile1.mcc`, `mccfile2.mcc`, `sccfile1.scc`, `sccfile2.scc`) which each contain 10 minutes of captions. All captions from `mccfile1`, the last 5 minutes of captions from `mccfile2`, and the first 5 minutes from each of the SCC files are stitched together in the order specified. The CML is submitted to a Timed Text Conform workflow configured to create an SCC output with default container settings (see [Timed Text Conform Action Configuration Panel](#)).

```
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <!--Specifies source 1, a caption sidcar file.-->
    <Subtitle>
      <File location="C:\CML1\mccfile1.mcc"/>
      <!--Specifies location of source 1.-->
    </Subtitle>
  </Source>
  <Source identifier="2">
    <Subtitle>
      <File location="C:\CML1\mccfile2.mcc"/>
    </Subtitle>
  </Source>
  <Source identifier="3">
    <Subtitle>
      <File location="C:\CML1\sccfile1.scc"/>
    </Subtitle>
  </Source>
  <Source identifier="4">
    <Subtitle>
      <File location="C:\CML1\sccfile2.scc"/>
    </Subtitle>
  </Source>
  <Sequence>
    <Segment>
      <Video source="1">
        <Head>
          <Edit mode="absolute" time="00:00:00:00"/>
        </Head>
        <Tail>
          <Edit mode="absolute" time="00:10:00:00"/>
        </Tail>
        <!--All 10 minutes of captions from source 1 are used for
this segment of the sequence. For standalone caption files, at
least a tail time and ideally also a head time are required.-->
      </Video>
    </Segment>
  </Sequence>
</Composition>
```

```

</Segment>
<Segment>
  <Video source="2">
    <!--Edit points specify that the last 5 minutes of captions
    from source 2 are used for this segment of the sequence.-->
    <Head>
      <Edit mode="absolute" time="00:05:00:00"/>
    </Head>
    <Tail>
      <Edit mode="absolute" time="00:10:00:00"/>
    </Tail>
  </Video>
</Segment>
<Segment>
  <Video source="3">
    <Head>
      <Edit mode="absolute" time="00:00:00:00"/>
    </Head>
    <Tail>
      <Edit mode="absolute" time="00:05:00:00"/>
    </Tail>
  </Video>
</Segment>
<Segment>
  <Video source="4">
    <Head>
      <Edit mode="absolute" time="00:00:00:00"/>
    </Head>
    <Tail>
      <Edit mode="absolute" time="00:05:00:00"/>
    </Tail>
  </Video>
</Segment>
</Sequence>
</Composition>

```

The resulting output is an SCC file with a duration of 25 minutes at 29.97 fps.

---

**Note:** If the Edit time attribute does not contain an explicit @ framerate value (for example, 00:05:00:00@23.976), the default 29.97 fps drop frame rate is applied to the output sequence. The frame rate of the output Sequence may also be determined by the frame rate of the first Segment in the Sequence. If any subsequent Segment uses a Source which has a different frame rate, those captions/subtitles are converted to the Sequence frame rate. In the example CML above, if source 1 has an explicit @23.976 framerate value specified then the 23.976 frame rate is applied to the output Sequence.

---



---

**Note:** Some subtitle file types (such as SRT, WebVTT, and some TTML files) can use milliseconds or other non-frame-based time values. If the Source is such a type, and a Head or Tail time is specified using a timecode format that includes a frame rate (for example, "HH:MM:SS:FF@23.976") then that frame rate defines the frame rate of the segment. If the Head/Tail edits do not describe a frame rate, the frame rate of the

---

segment defaults to 29.97 drop frame, unless it can be otherwise determined from the source file.

---

## Timed Text CML for Conforming Captions from Media Files

Embedded captions in a media file can be extracted and stitched, in whole or in part, using a properly configured CML ingested by Timed Text Conform.

---

**Note:** Timed Text CML does not allow the combined specification of caption and subtitle files for stitching in the same CML. All sources must be either captions or subtitle file types. Use Timed Text Flip to convert captions to subtitles or vice versa.

---

In this example, there are three media files (clip1.mov, clip2.mov, clip3.mov) with a duration of 30 minutes each. All captions from clip1, the last 15 minutes from clip2, and the middle 10 minutes from clip3 are stitched together in the order specified in the CML. Clips 1 and 2 have starting timecode of 00:00:00:00 but clip 3 has a starting timecode of 01:00:00:00. The CML is submitted to a Timed Text Conform workflow configured to create an MCC output with Specified Starting Timecode of 10:00:00:00@59.94 (see [Timed Text Conform Action Configuration Panel](#)).

```
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <!--Specifies source 1, a media file with embedded captions.-->
    <File location="C:\CML2\clip1.mov"/>
  </Source>
  <Source identifier="2">
    <File location="C:\CML2\clip2.mov"/>
  </Source>
  <Source identifier="3">
    <File location="C:\CML2\clip3.mov"/>
  </Source>
  <Sequence>
    <Segment>
      <Video source="2">
        <!--Edit points specify that the last 15 minutes of captions
        from source 2 are used for this segment of the sequence.-->
        <Head>
          <Edit mode="absolute" time="00:15:00:00@23.976"/>
          <!--Framerate is set to 23.976 which governs subsequent
          framerate of sequence.-->
          <!--Mode is set to absolute which assumes a start
          timecode of zero.-->
        </Head>
        <Tail>
          <Edit mode="absolute" time="00:30:00:00@23.976"/>
        </Tail>
      </Video>
    </Segment>
    <Segment>
      <Video source="1"/>
      <!--Since no edit points are specified, all 30 minutes of
      embedded captions from source 1 are used for this segment of the
      sequence.-->
    </Segment>
  </Sequence>
</Composition>
```



```

<Video source="3">
  <Head>
    <Edit mode="relative" time="00:10:00:00"/>
    <!--Mode is set to relative so that regardless of start
time code, the middle 10 minutes of embedded captions from source 3
will be extracted.-->
  </Head>
  <Tail>
    <Edit mode="relative" time="00:20:00:00"/>
  </Tail>
</Video>
</Segment>
</Sequence>
</Composition>

```

The resulting output is an MCC file with a duration of 55 minutes at 59.94 fps with a start timecode of 10:00:00:00 @ 59.94.

---

**Note:** The source order of segments stitched in the CML was 2,1,3 and the Edit attributes for segment 2 explicitly indicate @23.976 thus forcing the framerate of the sequence to be 23.976. However, the Timed Text Conform action has a Specified Start Timecode of 10:00:00:00@59.94 so the actual output of the action has a framerate of 59.94.

---



---

**Note:** Clip 3 has a known duration of 30 minutes and some type of offset start time (in this case 1hr). Using the *relative* edit attribute extracts the middle 10 minutes of captions.

---

## Timed Text CML for Conforming Captions/Subtitles with Associated Movie Files

In cases where a movie file (with or without captions) plus an associated sidecar caption/subtitle file are used, the movie file is parsed to determine frame rate and duration and the caption/subtitles are read from the associated caption/subtitle file.

In this example, there are numerous sources: a 5 second *black slate* movie file with no captions; two 30 second commercial clips with associated scc files; and a 30 minute television show with an associated scc file. The CMLstitches (in order):

The 5 second black slate.

The initial 10 minutes of the show

The two 30 second commercials.

The middle 10 minutes of the show.

The two 30 second commercials again.

The last 10 minutes of the show.

The CML is submitted to a Timed Text Conform workflow configured to use the CML Target Timecode (see [Timed Text Conform Action Configuration Panel](#)) and create SCC, MCC, and IMSC outputs all at 29.97fps.

```
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <!--Specifies source 1. This is a slate only with no associated
captions.-->
    <File location="C:\CML3\black_slate.mov"/>
  </Source>
  <Source identifier="2">
    <!--Specifies source 2. Consists of an scc file with associated
media file. Media file is used to determine framerate and duration
of caption file.-->
    <File location="C:\CML3\commercial_1.mov"/>
    <Subtitle>
      <File location="C:\CML3\commercial_1.scc"/>
    </Subtitle>
  </Source>
  <Source identifier="3">
    <File location="C:\CML3\commercial_2.mov"/>
    <Subtitle>
      <File location="C:\CML3\commercial_2.scc"/>
    </Subtitle>
  </Source>
  <Source identifier="4">
    <File location="C:\CML3\30min_show.mov"/>
    <Subtitle>
      <File location="C:\CML3\30min_show.scc"/>
    </Subtitle>
  </Source>
</Sequence>
<Segment>
```

```

    <Video source="1"/>
    <!--Since there are no embedded captions, has the effect of
    inserting 5 seconds of timecode at the beginning of the stitched
    file.-->
  </Segment>
  <Segment>
    <Video source="4">
      <!--First 10 minutes of captions from scc file, based on
      framerate and duration as determined from associated media file
      from source 4.-->
      <Head>
        <Edit mode="absolute" time="01:00:00:00"/>
      </Head>
      <Tail>
        <Edit mode="absolute" time="01:10:00:00"/>
      </Tail>
    </Video>
  </Segment>
  <Segment>
    <Video source="2"/>
    <!--Since no edit points are specified, all captions from
    associated scc file will be extracted.-->
  </Segment>
  <Segment>
    <Video source="3"/>
  </Segment>
  <Segment>
    <Video source="4">
      <Head>
        <Edit mode="absolute" time="01:10:00:00"/>
      </Head>
      <Tail>
        <Edit mode="absolute" time="01:20:00:00"/>
      </Tail>
    </Video>
  </Segment>
  <Segment>
    <Video source="3"/>
  </Segment>
  <Segment>
    <Video source="2"/>
  </Segment>
  <Segment>
    <Video source="4">
      <Head>
        <Edit mode="absolute" time="01:20:00:00"/>
      </Head>
      <Tail>
        <Edit mode="absolute" time="01:30:00:00"/>
      </Tail>
    </Video>
  </Segment>
</Sequence>
<Target>
  <Timecode time = "02:00:00;00@29.97"/>
  <!--Specified target start timecode of 02:00:00;00 @29.97 is
  applied.-->

```

## 28 | **Timed Text Conform Applications**

Timed Text CML for Conforming Captions/Subtitles with Associated Movie Files

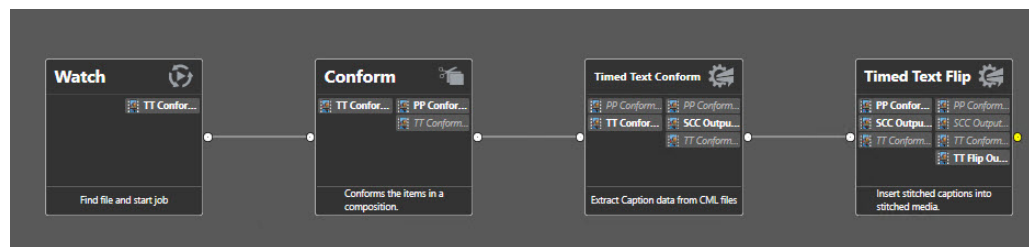
```
</Target>  
</Composition>
```

The resulting outputs are SCC, MCC, and IMSC files with a duration of 32 minutes and 5 seconds with a start timecode of 02:00:00:00 @ 29.97.

# Advanced Workflow Combining Timed Text Conform, Post Producer Conform, and Timed Text Flip

Now that you are familiar with Timed Text Conform, let's take a look at a workflow that conforms captions and media then combines them into an output file with synchronized and embedded captions.

**Note:** Although a single CML can be created that would observe the proper CML syntax for Post Producer Conform, Timed Text Conform, and Flip64 CML, it is more typical to create CML files tailored to the syntax of the action ingesting them. For purposes of this example, the same CML file can be ingested by Post Producer Conform and Timed Text Conform to create the desired outputs.



This workflow utilizes the following actions to perform these tasks:

*Watch*—The Watch action starts jobs when you submit the CML file for processing. You can replace Watch with a Receive action, so that you can submit jobs from other workflows, programs, or systems using the Vantage SDK.

*Post Producer Conform*—The Post Producer Conform action processes the CML, generating a stitched output file. This output file (PP Conform Output) is utilized in the downstream Timed Text Flip action as Media Source. See the Post Producer Developer's Guide and Post Producer Cookbook available on the Telestream website for Post Producer CML configuration details.

*Timed Text Conform*—The Timed Text Conform action processes the CML file, generating a stitched SCC output file. This output file (SCC Output from TTC) is utilized in the downstream Timed Text Flip action as an SCC Source.

*Timed Text Flip*—The Timed Text Flip action inserts the captions into the media file using a Same As Source output.

Activate the workflow and submit the same CML specified in [Timed Text CML for Conforming Captions/Subtitles with Associated Movie Files](#). When the job is complete open the output in your media player, such as Switch, and begin playback with captions enabled to view your now stitched captions.

## 30 | **Timed Text Conform Applications**

Advanced Workflow Combining Timed Text Conform, Post Producer Conform, and Timed Text Flip

# Timed Text CML Reference

The purpose of this chapter is to identify all elements in the Timed Text Composition Markup Language and their relationship, and describe each element in detail, including its attributes, along with simple examples to illustrate typical usage.

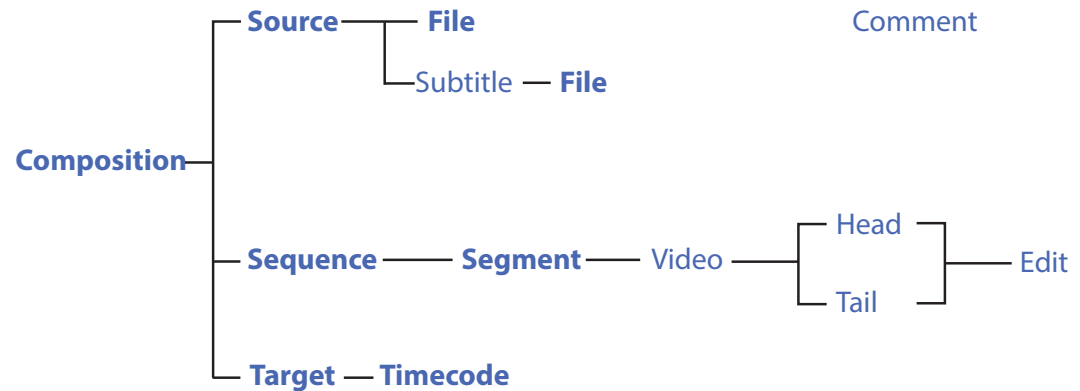
- [Timed Text CML Hierarchy Map](#)
- [Timed Text CML Implementation Details](#)

## Timed Text CML Elements in Alphabetic Order

- [Comment](#)
- [Composition](#)
- [Edit](#)
- [File](#)
- [Head](#)
- [Segment](#)
- [Sequence](#)
- [Source](#)
- [Subtitle](#)
- [Tail](#)
- [Target](#)
- [Timecode](#)
- [Track](#)
- [Video](#)

## Timed Text CML Hierarchy Map

This tree structure illustrates the hierarchical relationship between the Timed Text CML elements, expressed as a tree. (Click on any element to display its reference topic.)



Elements in **bold text** are required in a composition, in context. For example, [File](#) is required in [Source](#). In Timed Text CML, [Subtitle](#) is an optional element but if used, [File](#) is then required.

[Comment](#) is an optional child of all elements. [Comments](#) are unique; they have no effect on the CML, you can use them in any element, and use as many as you require.



# Timed Text CML Implementation Details

These topics provide information about creating Timed Text CML files for processing in Timed Text workflows.

- [Case Sensitivity is Critical](#)
- [Specifying File References in Timed Text CML](#)

## Case Sensitivity is Critical

XML is case-sensitive.

CML elements must be entered in title case—with the first letter in upper-case—for example, *Composition*, not *COMPOSITION* or *composition*—to be valid. If you don't capitalize an element (or you misspell it), it is ignored—and no error is displayed.

Although XML attributes are also by convention title cased, attributes in CML are all lower case.

## Non-Timed Text CML is Ignored

XML is permissive, in the sense that elements and attributes that are not part of the CML specification being utilized are ignored by the processor; they are not identified as errors.

Thus, you can create Timed Text CML compositions that contain Flip64 or Post Producer CML elements and attributes so that a single composition can be processed in two different workflows—first, a Timed Text Conform workflow to correctly trim subtitles—and then, a second workflow using Flip64 or Post Producer Conform—to generate the production media.

Of course, you can also mis-type an element or attribute, or capitalize it incorrectly—and it too, will be ignored.

## Specifying File References in Timed Text CML

Files referenced in Timed Text CML must be located on Windows servers, because Timed Text only accesses files on Windows (NTFS) platforms. If your workflow is monitoring a server with a different operating system (HTTP, FTP, S3, etc.) for new files, you must provide a Copy or other transport action in the workflow to relocate the file on a Windows server, and it must be accessible to the Timed Text service processing it.

References to (potentially relocated) files may be specified using a Windows drive letter or a UNC path.

In the examples in this guide, files are typically shown as a UNC path (for example: `\\share\path\myvideo.mov`) to reinforce the reliable use of shares to access network-based files by services operating in the Vantage domain. For the CML examples, drive letters are shown (for example: `C:\CML3\black_slate.mov`)

# Comment

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

The optional *Comment* element can be placed in any element in the composition. The *Comment* element can span multiple lines.

The purpose of a *Comment* is to provide documentation for the composition or to pass information (for example, a key-pair value or other metadata) into a workflow, which an action can extract for use in the workflow. You can use *Comment* to explain the purpose of the various elements of the [Composition](#), which aid in development, troubleshooting, debugging and maintenance.

There are no child elements or attributes in a *Comment* element. You can place any text in the comment element that are valid value characters in XML; except that you can't place XML syntax in a *Comment*.

---

**Note:** You can't use *Comment* as a surrounding element for other XML elements that you want to disable. The workflow will fail with an error: "The <XYZ> start tag... does not match end Comment tag."

The *Comment* element is not an HTML | XML comment. To disable multi-line portions of the CML, you can use the multi-line XML comment elements <!-- and -->.

---

## Example

In this [Sequence](#), the comments (in bold) contain key pair values, which can be extracted in the Timed Text Conform workflow for use in the workflow as necessary.

```
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share/path/My_TV_Show_Promo.mov" />
  </Source>
  <Sequence>
    <Segment>
      <Comment>TRP ID = urn:uuid:06a4-e204-407b-ac8b</Comment>
      <Comment>TRP Hash = emtXD7kMraxeTojicE6Ofva2HAc</Comment>
      <Comment>TRP Size = 224957607</Comment>
      <Video source="1" />
    </Segment>
  </Sequence>
  ...
</Composition>
```

To extract values from *Comment* elements, the best practice is to use a Metadata action to extract the values, parse them as needed, and update variables. The variables are automatically passed downstream where other actions can utilize the variables as needed.

# Composition

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

The *Composition* element identifies this XML-formatted text as a Composition object. It is the root element of every composition; one *Composition* element is required. The *Composition* element is the container for all other CML elements.

## Child Elements

- [Source](#) (one or more; one required)
- [Sequence](#) (one; required)
- [Target](#) (one, optional)

## Attributes

Name	Description
created (optional)	String; a date/time stamp, for informational purposes only.
name (optional)	String; a practical name for this composition file for informational purposes only.
version (optional)	String; a version number for informational purposes only; not used in Vantage.
xmlns	String; specifies utilized name spaces. Must include this namespace: xmlns="Telestream.Soa.Facility.Playlist". Omission results in an error. When using XML prefixes in CML, a namespace for each prefix must be defined.

## Example

This simplistic example demonstrates the basic functionality of a *Composition*: to contain all other elements. Here, the CML consists of a [Source](#), and a [Sequence](#) with a single [Segment](#) which includes [Video](#).

```
<Composition xmlns="Telestream.Soa.Facility.Playlist">
  <Source identifier="1">
    <File location="//share\path\My_TV_Show_Promo.mov" />
  </Source>
  <Sequence>
    <Segment>
      <Video source="1"/>
    </Segment>
  </Sequence>
</Composition>
```

# Edit

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

The *Edit* element (one required) is used in the [Head](#) and [Tail](#) of [Video](#). The *Edit* element creates a smaller clip from a source—by specifying a mark-in or mark-out point—for trimming 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 end of the clip from its default position to the point specified.

---

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

---

There are no child elements in an *Edit* element.

## Attributes

Name	Description
mode (required)	<p>Specifies how timecode is applied to a <a href="#">Head</a> or <a href="#">Tail</a>.</p> <p>Keywords: <i>absolute</i>   <i>relative</i></p> <p><i>absolute</i>—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 <i>Head</i>) or ending (in a <i>Tail</i>) of the source, irrespective of the timecode (if present).</p> <p>Example: <code>&lt;Edit mode = "relative" time = "00:10:00" /&gt;</code></p>
time (optional)	<p>Specifies the location of the edit point by timecode. Drop frame and non-drop frame references are valid.</p> <p>If you are specifying a relative timecode for video with a timecode track, use these formats:</p> <p>HH:MM:SS:FF@FPS   HH:MM:SS:FF@FPS   HH:MM:SS:FF   HH:MM:SS:FF</p> <p>Timecode references are applied to the timecode track specified in the associated Source element's file.</p> <p>Example:</p> <p><code>&lt;Edit mode = "relative" time = "00:00:10:00@23.976" /&gt;</code></p>

## Example

This [Segment](#) illustrates using an *Edit* to trim the *Video's Tail* at exactly 8 seconds by the timecode.

```
<Segment>
  <Video source = "3">
    <Head>
      <Edit mode = "absolute" time = "00:00:00:00@23.976"/>
    </Head>
    <Tail>
      <Edit mode = "absolute" time = "00:00:08:00@23.976"/>
    </Tail>
  </Video>
</Segment>
```

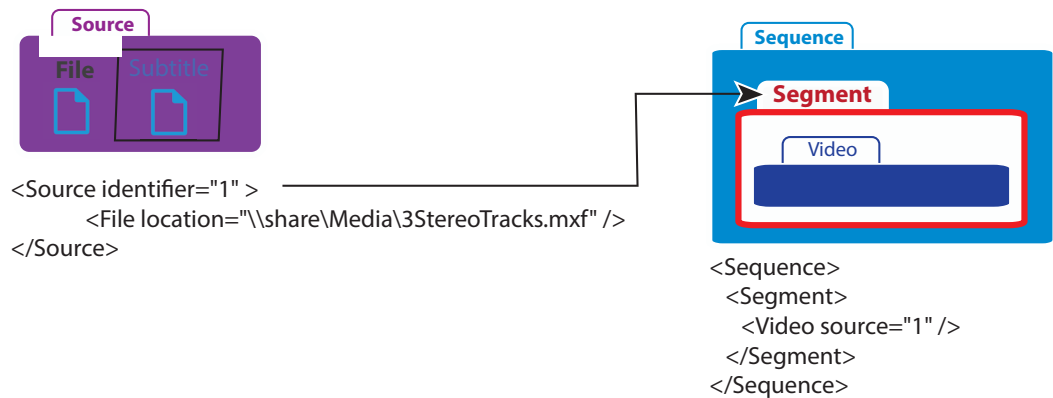
# File

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

A *File* element is required in a *Source*, to specify the path to the file, so that it may be utilized in a given *Segment*.

**Note:** Files referenced in Timed Text CML must be on Windows servers, because Timed Text only accesses files on Windows (NTFS) platforms. If your workflow is watching for new files on a server with a different operating system (HTTP, FTP, S3, etc.), you must add a Copy or other transport action to relocate the file on a Windows server accessible to the Transcode service processing it on behalf of Timed Text.

The association between a *Source* and its *Segment* is the *Video* element's *source* attribute's value in the *Source* which matches the *Segment* element's identifier attribute value, as depicted here:



Here, the *Video* references *Source* identifier 1, where the *File* is `//share/Media/3StereoTracks.mxf`.

There are no child elements in a *File* element.

## Attributes

This attribute is required for a *File* element.

Name	Description
location	<p>Specifies the fully-qualified path to a file on a Windows (NTFS) server, including file name. The location must be accessible to the Timed Text service which is executing the Timed Text action. You may use a drive letter or a UNC path—full is required in a Video element; full or relative is permitted in a Subtitle element). For example:</p> <ul style="list-style-type: none"><li>• <i>Drive Letter</i>—D:\pathname\filename.ext</li><li>• <i>Full UNC Path</i>—\\share\path\filename.ext</li><li>• <i>Relative UNC Path</i>—path\filename.ext or filename.ext or ..\upone\filename.ext</li></ul> <p>Relative paths are annotated relative to the location of the CML file being processed.</p> <p>Shares are recommended to ensure access by Vantage services, especially in an array where the service that executes the action may be on different nodes from job to job.</p>

## Example

This example illustrates using a *File* element to specify a movie file on a local drive, without a subtitle file. If this location is directly on the server where the Timed Text service is executing the Timed Text Conform action, the service can access the file. If its on another, the path must be a share, or you should change it to a UNC path.

```
<Source identifier="1">  
  <File location="D:\path\Mystic_River_Music.mov" />  
</Source>
```

This example illustrates using a *File* element to specify a movie file and a subtitle file; both are using a UNC file path.

```
<Source identifier="2">  
  <File location="\\share\path\Mystic_River.mov" />  
  <Subtitle>  
    <File location="\\share\path\subtitles.scc" />  
  </Subtitle>  
</Source>
```

This example illustrates using a *File* element to specify a movie file and a subtitle file; both are using a relative (to the target CML file) file path.

```
<Source identifier="2">  
  <File location="media\Mystic_River\Mystic_River.mov" />  
  <Subtitle>  
    <File location="media\Mystic_River\subtitles.scc" />  
  </Subtitle>
```

```
</Source>
```

This example illustrates using a *File* element to identify a subtitle file without a movie file; the path is UNC.

```
<Source identifier="2">  
  <Subtitle>  
    <File location="//share/path/subtitles.scc" />  
  </Subtitle>  
</Source>
```

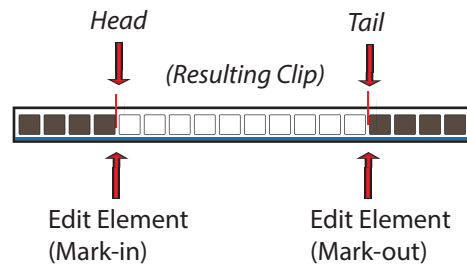


# Head

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

The *Head* element is the mark-in point (the first frame) in a *Video* element. *Heads* and *Tails* have no duration; they represent either edge of the material. You only use a *Head* element when you want to modify the mark-in point of the *Video*.

In this example, the *Head* is equal to the timecode of its *Edit*; *Tail* is also equal to its *Edit*, and the associated subtitle file is clipped accordingly:



There are no attributes in a *Head* element.

See also [Tail](#).

## Child Elements

One *Edit* element is required in a *Head* and *Tail* element:

## Example

In this *Segment*, the *Video* is clipped by timecode in the media: in the beginning (*Head* > *Edit*) it is clipped at one hour—presumably, the timecode begins at one hour, so there is no adjustment—although the element is useful from a clarity standpoint. The *Segment* is clipped at the end at 1:03:43:15 into the clip, with the *Tail* > *Edit*.

```
<Segment>
  <Video source = "1">
    <Head>
      <Edit mode = "absolute" time = "01:00:00:00@23.976"/>
    </Head>
    <Tail>
      <Edit mode = "absolute" time = "01:03:43:15@23.976"/>
    </Tail>
  </Video>
</Segment>
```

# Segment

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

At least one *Segment* must be included in a *Sequence*; multiple are permitted. A *Segment* represents one clip on the timeline and is comprised of one *Video* element. A clip can be used in its entirety (the full duration of the original source) or it can represent a sub clip selected from the source.

The duration of a *Segment* is determined by the duration of the *Video*.

*Segments* organize their material ordinally along the timeline relative to other *Segments*. The order of *Segment* elements dictates the organization of clips.

The frame rate of a *Segment* is determined by the frame rate of the *Source* it contains: a *Source* of a movie file has the frame rate of the movie file. Some subtitle file types (such as SRT, WebVTT, and some TTML files) use milliseconds or other non-frame-based time values. If the *Source* is such a type, and a *Head* or *Tail* time is specified using a timecode format that includes a frame rate (for example: "HH:MM:SS:FF@23.976") then that frame rate defines the frame rate of the *Segment*. If the *Head/Tail* > *Edit* elements do not describe a frame rate, the frame rate of the *Segment* default to 29.97 drop frame, unless it can be otherwise determined from the *Source* file.

There are no attributes in a *Segment* element.

## Child Elements

One *Video* element must be present in a *Segment*.

## Example

In this example, the segment consists of a *Video* from the source 1 file, and it is clipped per the edits shown:

```
<Segment>
  <Video source = "1">
    <Head>
      <Edit mode = "absolute" time = "01:00:00:00@23.976"/>
    </Head>
    <Tail>
      <Edit mode = "absolute" time = "01:03:43:15@23.976"/>
    </Tail>
  </Video>
</Segment>
```

# Sequence

[Timed Text CML Elements in Alphabetical Order](#) | [Timed Text CML Hierarchy Map](#)

A *Sequence* is used to organize media (material elements) on an independent timeline, using one or more *Segment* elements, which contain the media streams. It's often used to join multiple material elements ordinally.

One *Sequence* element is required in a *Composition*. There are no attributes in a *Sequence*.

The frame rate is determined by the frame rate of the first *Segment* in the *Sequence*. If any subsequent *Segment* uses a *Source* with a different frame rate, those subtitles are converted to the *Sequence* frame rate.

By default, the output sequence begins at time 00:00:00:00, unless the *Target* element specifies a different start time. The start time can also be overridden via a control in the Timed Text Conform action.

## Child Elements

One or more *Segments* must be added to a *Sequence* element.

## Example

This *Sequence* consists of three *Segments*, arranged ordinally by source 3, 1, and 2. Each is clipped.

```
<Sequence>
  <Segment>
    <Video source = "3">
      <Head>
        <Edit mode = "absolute" time = "00:00:00:00@23.976"/>
      </Head>
      <Tail>
        <Edit mode = "absolute" time = "00:00:08:00@23.976"/>
      </Tail>
    </Video>
  </Segment>
  <Segment>
    <Video source = "1">
      <Head>
        <Edit mode = "absolute" time = "01:00:00:00@23.976"/>
      </Head>
      <Tail>
        <Edit mode = "absolute" time = "01:03:43:15@23.976"/>
      </Tail>
    </Video>
  </Segment>
  <Segment>
    <Video source = "2">
      <Head>
        <Edit mode = "absolute" time = "00:00:00:00@23.976"/>
      </Head>
      <Tail>
        <Edit mode = "absolute" time = "00:00:00:00@23.976"/>
      </Tail>
    </Video>
  </Segment>
</Sequence>
```

```
</Head>  
<Tail>  
  <Edit mode = "absolute" time = "00:00:05:00@23.976"/>  
</Tail>  
</Video>  
</Segment>  
</Sequence>
```

# Source

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

A *Source* specifies a media | subtitle file to be used in *Segments* that comprise the *Sequence*. Each *Source* includes a unique identifier, the file, and subtitle information.

At least one *Source* must be defined in your *Composition*; multiple are permitted. *Source* elements by convention (but not required) are typically listed first in a *Composition*. *Source* elements may be in any order.

A *Source* requires a *File* element, a *Subtitle* element, or both.

Each *Source* represents one of the following:

- **A movie file**—The movie file is parsed to determine its frame rate and duration. If the movie contains embedded captions, they are extracted from the movie. A movie file with no subtitles just adds time to the subtitles timeline for the duration.
- **A movie file with a sidecar subtitle file**—The movie file is parsed to determine its frame rate and duration, but subtitles are read from the subtitle file. If subtitles are embedded, they are ignored in the presence of a sidecar subtitle file.
- **A subtitle sidecar file only**—The duration of a subtitle-only source cannot be reliably determined, and thus, any *Segment* consisting of a subtitle-only source must specify at least a *Tail* time, and ideally, it should also include a *Head* time.

When a *Source* consists of a movie file with embedded subtitles or a movie file plus a subtitle file, the duration of the movie can be determined, and thus the duration of the *Segment* does not need to be specified in the CML when the entire duration of the source is being used. An optional *Head* and/or *Tail* can be used to trim the source and the resulting sub-clip is placed on the *Sequence*. If the *Head* is not specified, it is the first frame of the movie file. If the *Tail* is not specified, it is the last frame of the movie file. The duration of the resulting *Segment* is the interval between the *Head* and *Tail* time.

However, if a *Source* consists of only a sidecar subtitle file with no movie file, the duration of the resulting *Segment* cannot be reliably determined because subtitle files generally don't specify a Start of Program / End of Program time, and it is not guaranteed that the first subtitle begins on the first frame of the program, nor is it guaranteed that the last subtitle ends on the last frame of the program.

Therefore, when using a sidecar subtitle file without a movie as a *Source*, the *Segment* MUST specify at least the *Tail* time (in "absolute" mode), to define the duration of the *Segment*. The *Head* time, if not specified, is 00:00:00:00 by default.

## Child Elements

Each of these elements are child elements of a *Source*—one or the other is required, both are permitted:

- *File*—One permitted; optional
- *Subtitle*—One permitted; optional.

## Attributes

Name	Description
identifier	<p>An integer value, which must be unique among all <i>Source</i> elements.</p> <p>Example: <code>&lt;Source identifier="1"&gt;</code></p> <p>This identifier value is used in the <i>source</i> attribute of <i>Video</i> elements to reference and utilize media from this file, for placement in the output.</p>

## Example

This example illustrates the typical use of a *Source* element to identify the file providing media, including *Subtitle* processing:

```
<Source identifier = "1">  
  <File location = "\\SHARE\Path\CapeAndClown.mxf"/>  
  <Subtitle>  
    <File location="\\SHARE\Path\CapeAndClown_fr.mxf"/>  
  </Subtitle>  
</Source>
```

# Subtitle

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

Optionally, you can provide one *Subtitle* in a [Source](#). The purpose of the *Subtitle* element is identify the specified sidecar file on a source-by-source basis, to utilize captions in the sidecar file instead of the embedded captions in the [Source](#) file itself.

If you want to propagate embedded captions from the movie source, you omit the *Subtitle* element entirely.

Subtitle file supported frame rates: 23.976 | 25 (SCC) | 29.970 | 59.94 fps.

There are no attributes in a *Subtitle*.

See also [File](#).

## Child Elements

One [File](#) must be provided in a *Subtitle* element.

## Application

In this example, a caption 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 one minute into the clip, which must also be taken into consideration.

Here are the details:

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

The caption file's offset value is added to the timecode of the media file's in point before it is applied to the caption file. The result is that the edit point in the file specifies the frame that is one minute into the caption 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 caption file starts at zero. The media file also has the in point one minute into the clip, as in the first example.

Here are the details:

- Media start timecode      01:00:00;00
- Caption file start timecode 00:00:00;00
- In point edit              01:01:00;00      (<Edit time="00:01:00;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 caption 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

## Example

In this example, a movie file and the associated caption file are identified.

```
<Composition>
  <Source identifier="1">
    <File location="//share/path/Chronicle_St_Lucia_TC.mov" />
    <Subtitle>
      <File location="//share/path/Chronicle_St_Lucia_TC.scc" />
    </Subtitle>
  </Source>
</Sequence>
  <Segment>
    <Video source="1" />
  </Segment>
```

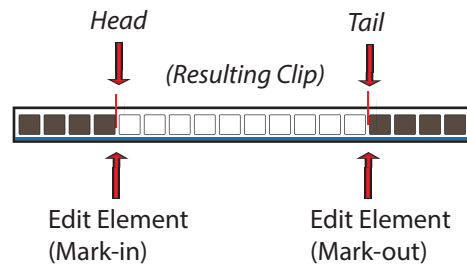


# Tail

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

The *Tail* element is the mark-out point (the last frame + 1) in a *Segment* of *Video* material. *Heads* and *Tails* have no duration; they represent either edge of the material. You only use a *Head* element when you want to modify the mark-in point of the *Video*.

In this example, the *Head* is equal to the timecode of its *Edit*; *Tail* is also equal to its *Edit*, and the associated subtitle file is clipped accordingly:



There are no attributes in a *Tail* element.

See also the corollary element, *Head*.

## Child Elements

One *Edit* element is required in a *Head* or *Tail* element.

## Example

In this example, the *Segment* is eight seconds long, starting at the beginning of the clip (since there is no *Head* > *Edit*), with an audio fade out.

```
<Segment>
  <Video source = "3">
    <Head>
      <Edit mode = "absolute" time = "00:00:00:00@23.976"/>
    </Head>
    <Body/>
    <Tail>
      <Edit mode = "absolute" time = "00:00:08:00@23.976"/>
    </Tail>
  </Video>
</Segment>
```

## Target

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

One *Target* element is optional in a *Composition*. You should only include *Target* to override the timecode. By default, the output sequence begins at time 00:00:00:00, unless the *Target* element is used to specify a different start time.

Optionally, you can override the start time via a control in the Timed Text Conform action.

There are no attributes in a *Target*.

## Child Elements

- *Timecode* (required; one permitted)

## Example

In this example, the time is set to 1 hour, and the frame rate is 29.97.

```
<Target>  
  <Timecode time = "01:00:00;00@29.97"/>  
</Target>
```

# Timecode

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

One *Timecode* is required in a *Target*; you use it to specify the timecode of the generated media. When *Timecode* is used, the Timecode filter in Timed Text Conform should be disabled or set to *Use Source Timecode*. In this manner, the *Timecode* specification is propagated directly to the final Timed Text Conform output.

There are no child elements of *Timecode*.

See also *Target*.

## Attributes

These attributes must be applied to a *Timecode* element.

Name	Description
time	Timecode; specifies the starting timecode to utilize, using these formats: HH:MM:SS:FF@FPS   HH:MM:SS;FF@FPS   HH:MM:SS:FF   HH:MM:SS;FF Rates supported: 23.976, 24, 25, 29.97, 30, 50, 59.94.

## Example

In this example, the time is set to 1 hour, and the frame rate is 29.97.

```
<Target>  
  <Timecode time = "01:00:00;00@29.97"/>  
</Target>
```

# Video

[Timed Text CML Elements in Alphabetic Order](#) | [Timed Text CML Hierarchy Map](#)

One *Video* element is required in a *Segment*; its duration dictates how captions are clipped in the output. The *Video* element is a material element from a file identified by a specific *Source*, by using the *source* attribute (required).

By adding *Video* to a *Segment*, you are placing the corresponding source(s) on the *Sequence*'s timeline (accounting for optional *Head* and *Tail* trim points).

One *Video* element is required per *Segment*.

## Child Elements

One each of these elements may be added to apply a *Video* element:

- *Head*—when you want to clip the start of the file
- *Tail*—when you want to clip the end of the file.

## Attributes

These attributes are required in a *Video* element.

Name	Description
source	Integer; corresponding to the <i>Source's identifier</i> attribute value, to identify the media file to process.

## Example

In this example, the *Video* is used from the file identified in source 2.

```
<Segment>
  <Video source = "2">
    <Head>
      <Edit mode = "absolute" time = "00:00:00:00@23.976"/>
    </Head>
    <Tail>
      <Edit mode = "absolute" time = "00:00:05:00@23.976"/>
    </Tail>
  </Video>
</Segment>
```