



FlipFactory®

USING VANC INSERTION WITH FLIPFACTORY

This App Note applies to FlipFactory versions 7.0 & later

Synopsis2

 Introduction2

 VANC Data Types2

 VANC Storage.....4

Using VANC In FlipFactory8

 Processing VANC With FlipFactory.....8

 Using VANC Data Insertion.....9

 Creating XML Files10

Copyright and Trademark Notice..... 14

Limited Warranty and Disclaimers 14

Synopsis

Introduction

Ancillary (ANC) data is a means of embedding non-video information, such as audio and metadata, in a serial digital transport stream. Ancillary data packets are located in horizontal (HANC) or vertical (VANC) blanking of a video signal. HANC is used to embed uncompressed audio data in the SDI or HD-SDI stream. VANC is used to embed low-bandwidth data -- information updated on a per-field or per-frame basis. Closed caption data (CCD) and Active Format Description (AFD) are examples of metadata stored as VANC. SMPTE 291m describes the details of Ancillary data packet structures.

FlipFactory can preserve all VANC data and transform it into a variety of output file formats. FlipFactory can also insert certain types of data into a VANC data stream during a transcode operation.

VANC Data Types

Closed Captioning (608 vs. 708)

Caption data is encoded according to CEA-608-B and CEA-708-B. When caption data is decoded it displays on-screen information related to the caption data.

CEA-608 defines how line 21 carries the CC information in an NTSC broadcast. This standard encodes CC information as seven bits, plus one parity bit, at 120 characters per second (cps). This produces a data rate of 960 bits per second (bps). Closed captions at 120 cps equate to 2 bytes per field of caption data. These two bytes are encoded on line 21 as a waveform that is described in CEA-608-B. Multiple caption services, text services, and XDS data can be carried in these two fields (CEA-608 describes these services in detail). This caption information can also be carried digitally in several ways including multiple types of MPEG2 user data packets as described in SCTE 20, SCTE 21, ATSC A/53, etc. CEA-608 captions are required to be a part of the CEA-708-B caption packets.

CEA-708-B defines coding of DTVCC in an ATSC A/53 specified bitstream. This standard significantly enhances DTV display and formatting features (CEA-CEB-10-A discusses implementation details). CEA-708-B also defines caption distribution packets (CDP's) that hold DTVCC data, 608 caption data, caption service information, and (optionally) time code. This facilitates decoding digital cable signals and insertion of 608 captions into line 21 of the NTSC output of an STB. The first packet(s) in a 708 CDP contain 608 compatibility bytes causing 608 caption data to be part of the 708 standard. ATSC A/53 describes how the CEA-708 CDP is carried as MPEG User data in an ATSC Transport Stream (see Figure 1 below).

Figure 1. CEA708 VANC Data Packet Structure in a Transport Stream

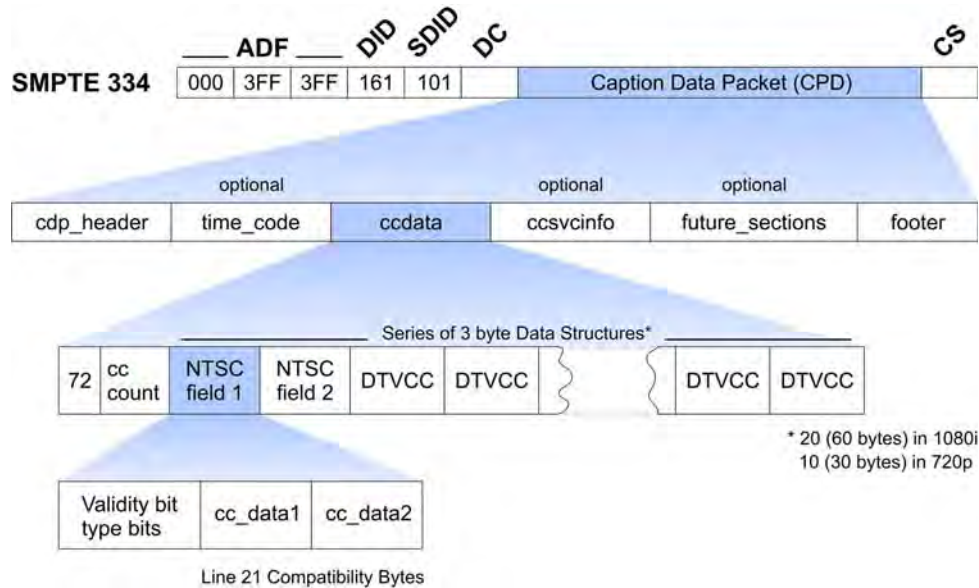


Diagram courtesy of Ensemble Designs

FlipFactory preserves 608 caption data from SD file formats to HD file formats during an up-convert operation. In this case the 608 data is preserved in the 708 CDP as bytes that are 608 compatible, but the 608 data is not transformed to full 708 caption data. FlipFactory also transforms 30 fps caption data from SD or HD source file formats to HD source file formats at 60 fps, and vice versa.

Time Code (SMPTE 12m-2)

SMPTE 12m-2 forms a portion of one of the oldest SMPTE standards for television. SMPTE Time Code was developed originally for analog television recording systems and thus dealt only with interlaced television systems operating with frame rates up to 30 frames per second. It is, however, flexible enough in design to be used in digital television systems, both standard definition and high definition. The actual transport of the codewords for digital systems varies significantly from the transport methods defined for analog systems, and is defined in SMPTE 12m-2 (formerly named SMPTE RP 188). The time and control code information is carried as the payload of ancillary packets which are located in the Ancillary data space of a digital television signal.

FlipFactory can preserve time code information present in a VANC data stream during a transcode operation. Note that some formats may require VANC based timecode to be stripped from the VANC data stream and stored in a different manner. Omneon is an example of this, where timecode should be stored in MPEG2 user data via SMPTE 328m (described later).

Active Format Description (SMPTE 2016)

Active Format Description (AFD) is a standard set of codes that can be sent in the MPEG video stream or in the baseband SDI video signal that carries information about their aspect ratio and active picture characteristics. AFD has been used by television broadcasters to enable both 4:3 and 16:9 television sets to optimally present pictures transmitted in either format. It has also been used by broadcasters to dynamically control how down-conversion equipment formats widescreen 16:9 pictures for 4:3 displays.

Standard AFD codes provide information to video devices about where in the coded picture the active video is and also the "protected area", which is the area that needs to be displayed. Outside of the protected area, edges at the sides or the top can be removed without the viewer missing anything significant. Video decoders and display devices can then use this information, together with knowledge of the display shape and user preferences, to choose a presentation mode.

FlipFactory can preserve AFD information present in a VANC data stream and can insert AFD information into a VANC data stream during a transcode operation.

Dolby E Audio Metadata

Audio data-rate reduction technologies use metadata to describe the encoded audio and convey information that precisely controls downstream encoders and decoders. This metadata is multiplexed into the encoded audio bitstream.

Audio metadata is created during program creation or mastering. It can be carried in the VANC data space of a digital television signal. Relevant standards are: SMPTE RDD06 - Description and Guide to the Use of the Dolby® E Audio Metadata Serial Bitstream, and SMPTE 2020 - Format of Audio Metadata and Description of the Asynchronous Serial Bitstream Transport.

FlipFactory can preserve audio metadata information present in a VANC data stream and can insert Audio Metadata of limited types into a VANC data stream during a transcode operation.

Teletext/EU (SMPTE RDD08, Free TV OP-47)

Free TV Operational Practice (OP-47) describes the technical/operational practices associated with the storage and distribution of closed caption/subtitling data in the VANC space of the 10-bit serial HD-SDI Signal. This complies with ITU-R BT.1120-7.

FlipFactory preserves Teletext information present in a VANC data stream during a transcode operation.

Note: *This is presently untested by Telestream, but is part of FlipFactory's development roadmap.*

VANC Storage

VANC data is stored in media files in many different ways. Generally speaking, the data can be stored with each frame of data or as a separate data track within a file. Listed below are track-based and frame-based methods for storing VANC data with their file wrapper formats.

Track-based Methods

MXF (SMPTE 436)

MXF format is covered by a variety of SMPTE standards. SMPTE 436m describes the embedding of VANC data or standard definition uncompressed VBI data within an MXF file. The standard specifies a variety of formats for this data.

FlipFactory can preserve 8-bit VANC data from MXF files and can create a SMPTE 436m 8-bit VANC data track in an output file. The 8-bit VANC data stream is common to Sony XDCAM HD format.

GXF (SMPTE 360, SMPTE RDD-14)

GXF format is the native format for Grass Valley Profile and K2 broadcast video servers. This format is covered by SMPTE 360m and RDD-14 standards. SMPTE RDD-14 is the latest description of GXF file format, which describes how a VANC data track is embedded in the file format.

FlipFactory can preserve VANC data from GXF files and can create a VANC data track in an output GXF file.

MPEG2 Transport Stream (SMPTE RDD-11)

Registered Disclosure Document RDD 11 describes a bitstream syntax and semantics used to transmit SMPTE 291 formatted ancillary data (VANC and HANC) from the input of an encoder, through an MPEG-2 Transport Stream, and to reconstruct the ancillary data accurately at the output of a decoder. SMPTE RDD-11 describes a method for preserving full 10-bit VANC data in an MPEG2 Transport Stream as a separate data track in the transport multiplex. FlipFactory can preserve VANC data from an MPEG2 Transport Stream file as a source only. Creation of VANC data in an MPEG2 Transport Stream is not supported at this time.

Frame-based Methods

Frame based VANC data carriage methods are either associated with a specific video codec or a proprietary media file format.

MPEG2 User Data types

There are many different methods for storing closed caption or VANC data in a data packet within an MPEG2 compressed video elementary bitstream. Listed below are just a few of the possibilities:

SMPTE 328m (Omneon). This standard describes methods for preserving multiple types of information in MPEG2 user data constructs. This includes timecode and VANC data. This format is used by Omneon to preserve timecode and VANC data in their MPEG2 files.

Preservation of VANC data (other than captions) is on the Telestream development roadmap. FlipFactory can preserve timecode from Omneon MPEG2 source files that use SMPTE 328m and can create timecode and VANC data in an output Omneon file.

ATSC A/53. This standard in conjunction with CEA-708 specifies how DTV Closed Caption data should be stored in MPEG2 User data.

FlipFactory can preserve this type of caption data from the following file formats:

- MPEG2 Transport Streams containing MPEG2 video
- MPEG2 Program Streams containing MPEG2 video
- Omneon files containing MPEG2 video.

FlipFactory can create this type of caption data in the following file formats:

- MPEG2 Transport Streams containing MPEG2 video (VOD Encoder)
- MPEG2 Transport Streams containing H.264 video (IPTV Encoder)
- MPEG2 Program Streams containing MPEG2 video (Program Stream Encoder)
- Omneon files containing MPEG2 video
- Avid MediaStream files containing MPEG2 video
- Seachange BML files containing MPEG2 video.

Note: *SCTE 21 describes the same constructs as ATSC A/53 for closed captions.*

SCTE 20. This standard also describes how closed caption data should be stored in MPEG2 Picture User data. It is similar to ATSC A/53 and SCTE 21.

FlipFactory can preserve this type of caption data from the following file formats:

- MPEG2 Transport Streams containing MPEG2 video
- MPEG2 Program Streams containing MPEG2 video.

FlipFactory can create this type of caption data in the following file formats:

- MPEG2 Transport Streams containing MPEG2 video (VOD Encoder)
- MPEG2 Program Streams containing MPEG2 video (Program Stream Encoder).

DV / DVC PRO

DV video can carry timecode and CEA-608 closed captioning within the Video Auxiliary (VAUX) space of a compressed frame.

FlipFactory can preserve CEA-608 CC data from DV video wrapped in QuickTime files or Grass Valley GXF files. It can also create closed caption data in DV video frames wrapped in QuickTime, Grass Valley GXF, or Omneon files.

DVC Pro HD (SMPTE 375m)

The SMPTE 375m standard specifies the mapping of VANC data packets into the payload area of the DV based 100 Mbps digital interface format (DIF) structure VAUX DIF blocks. This is defined in SMPTE 370M.

At this time FlipFactory does not preserve or create VANC data from DVCPRO HD.

Apple Pro Res

Apple Pro Res carries VANC data in each video frame using a proprietary format. Pro Res is carried in QuickTime wrappers or in Telestream TIFO files. TIFO files are created by Telestream Pipeline and when containing Pro Res video, it can also carry VANC data separate from the video frames.

FlipFactory preserves VANC data from TIFO files containing Pro Res video.

Avid DNxHD (SMPTE 2019m VC3)

Avid DNxHD is assumed to carry VANC data in each video frame using a proprietary format. Telestream is working with Avid to obtain this information. DNxHD is carried in QuickTime wrappers, MXF wrappers, in Telestream TIFO files. TIFO files are created by Telestream Pipeline, and when containing DNxHD video, it can also carry VANC data separate from the video frames.

FlipFactory preserves VANC data from TIFO files containing DNxHD video.

Harris Nexio LXF Format

LXF format is the native format for Harris Nexio broadcast video servers. It is a proprietary format that stores VANC or VBI data on a frame-by-frame basis.

FlipFactory preserves VANC data from LXF files and creates a VANC data track in an output LXF file.

Seachange

Seachange format is the native format for Seachange broadcast video servers. This format is very similar to MPEG2 Program Streams. Seachange VANC preservation capability is a proprietary MPEG2 user database format that stores VANC or VBI data on a frame-by-frame basis.

Seachange BML server can also use SCTE 21 or ATSC A/53 user data constructs with MPEG2 standard definition files.

FlipFactory creates VANC data tracks in output Seachange files.

Quicktime

QuickTime is Apple's file wrapping format for a vast variety of video and audio formats. Some video formats carried in QuickTime wrappers can also include VANC data in each video frame. This includes many HD and SD video formats. When FlipFactory decodes a QuickTime file, VANC or caption data is preserved.

HD Formats:

- DVCPro HD (not supported)
- Apple Pro Res (VANC is preserved if present in the source compressed video frames)
- Avid DNxHD (VANC is preserved if present in the source compressed video frames).

SD Formats:

- D10/IMX - Captions in 512 line VBI are extracted by FlipFactory or created in a D10 QuickTime file
- DV - User data captions can be preserved and created.

QuickTime supports a closed caption track, but this is not currently supported by FlipFactory.

360 Systems JPEG 2000

360 Systems MAXX HD file format is the native format for 360 Systems' MAXX-HD server family of products. This format is a proprietary format that stores VANC data in a separate file on the server. FlipFactory support for this format is currently under development.

Telestream Intermediate Format (TIFO)

Telestream Intermediate Format (TIFO) is a proprietary intermediate file format. This format is used primarily for integration between Telestream Pipeline and FlipFactory products. Any VANC data on the incoming HD-SDI to Pipeline can be preserved in a TIFO output file produced by the Pipeline Application.

FlipFactory can preserve VANC data from a TIFO source file.

Using VANC In FlipFactory

Processing VANC With FlipFactory

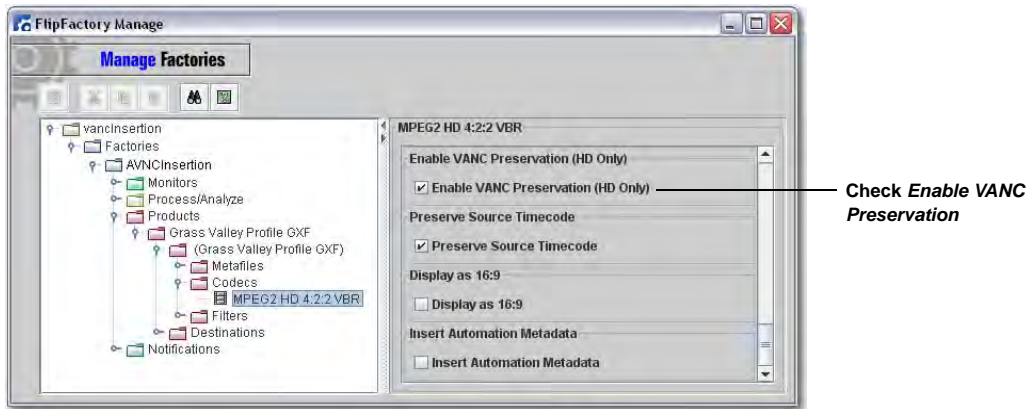
VANC data is processed in FlipFactory by creating a factory with a product that supports VANC data preservation. FlipFactory supports many common file formats and has several more currently in development. The table below lists the type of metadata and decode/encode availability in FlipFactory for each file format:

| FILE FORMAT | METADATA TYPE | DECODE | ENCODE |
|---------------------|-------------------------------|-------------------------------------|-------------------------------------|
| MXF | SMPTE 436 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| MPEG2 TS | Captions (ATSC, SCTE20, etc.) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| MPEG2 TS | SMPTE RDD-11 (VANC) | <input checked="" type="checkbox"/> | * |
| MPEG2 PS | Captions (ATSC, SCTE20, etc.) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Omneon | Captions in MPEG2 (ATSC) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Omneon | SMPTE 328 in MPEG2 (Timecode) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Omneon | SMPTE 328 in MPEG2 (VANC) | * | <input checked="" type="checkbox"/> |
| Omneon | SMPTE 314 DV | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Omneon | SMPTE 374 DVCPPro HD | * | * |
| Grass Valley GXF | SMPTE 360, SMPTE RDD-14 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Harris LXF | Proprietary | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Seachange | Proprietary | * | <input checked="" type="checkbox"/> |
| MediaStream | ATSC captions | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| QuickTime | Captions in DV (SMPTE 314) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| QuickTime | CEA-608 caption track | * | * |
| 360 Systems JPEG 2K | Proprietary | * | * |
| TIFO | Proprietary | <input checked="" type="checkbox"/> | * |

* *This functionality is currently in development*

Note: *File formats used in FlipFactory to decode or encode VANC data must be supported by FlipFactory, as indicated in the table above.*

To process VANC data in FlipFactory, submit a file containing VANC data to be decoded by the factory. Using any product that preserves VANC data, a new file is created with encoded VANC data in it. The configuration of a typical factory to process VANC data is shown below. In this example, the codec selected is an MPEG2 HD for a Grass Valley Profile (GFX), and is configured to enable VANC preservation:



Using VANC Data Insertion

VANC data Insertion inserts VANC data packets into the processing stream during a transcode operation in FlipFactory. VANC data Insertion enables a factory to add specific VANC data packets into the media output file.

To set up VANC data Insertion, an XML data file (which is used to describe the contents and type of VANC data packets) must be identified. The XML file has been created to give the user control over the data values that will be added to the VANC data stream. A sample XML data file for VANC data Insertion is available from Telestream.

The output file type must support some form of VANC data preservation. Additionally, if the output HD encoder is set up to preserve VANC data, the original VANC data from the source file and the newly-inserted VANC data will be present in the output file.

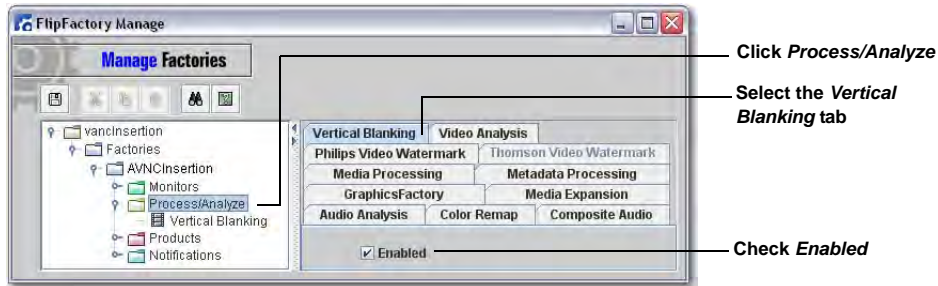
VANC data insertion currently implements insertion of:

- Vertical Ancillary Data Mapping of Active Format Descriptor (SMPTE 2016)
- Vertical Ancillary Data Mapping of Audio Metadata, Method B (SMPTE 2020-3). The implementation of SMPTE 2020 is limited to the description of a single audio stream.
- XDS data insertion of Program Name field (EIA-608 describes XDS data).

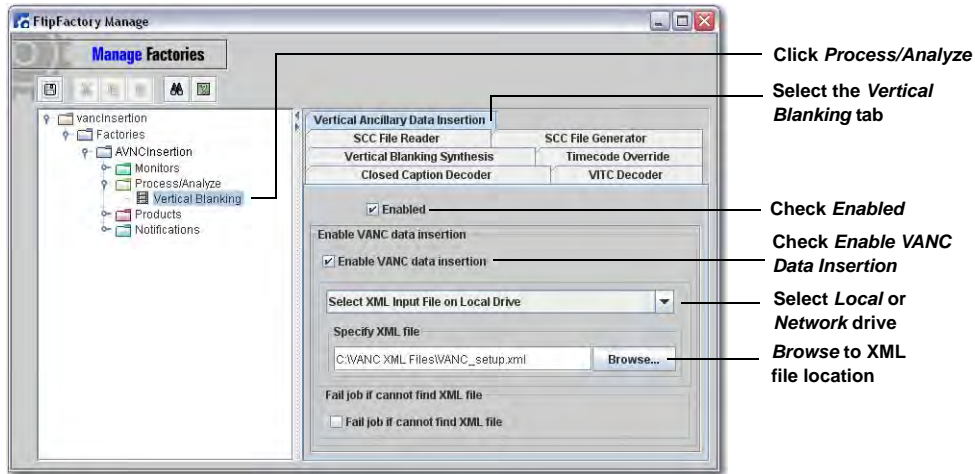
To create a Vertical Blanking Process/Analyze Tool and configure VANC data Insertion, follow these steps:

Note: *This example assumes that you know how to create and configure factories in FlipFactory and how to submit jobs. Review the FlipFactory User's Guide for details.*

1. Add a Vertical Blanking Process/Analyze tool: Click *Process/Analyze*, select the *Vertical Blanking* tab, and check *Enabled*:



2. Configure VANC data insertion: Click *Vertical Blanking*, select the *Vertical Ancillary Data Insertion* tab, and check *Enabled*. Then check *Enable VANC Data Insertion*, select *Local* or *Network* drive from the pull-down menu, and *Browse* to the XML file location:



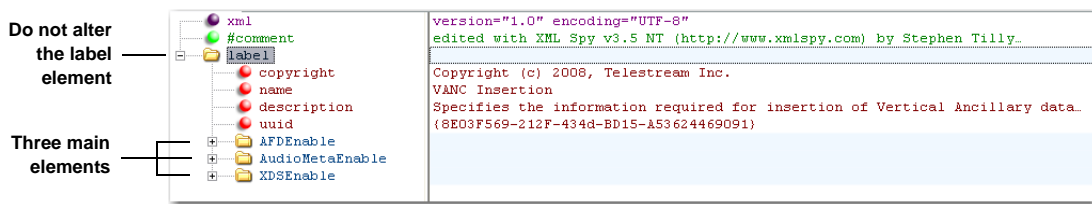
Your factory is now ready to perform VANC data insertion.

Creating XML Files

To set up VANC data insertion, an XML data file can be used to describe the contents and type of VANC data packets. This XML file can be created to provide user control over the data values that will be added to the VANC data stream. A sample XML data file is available by request from Telestream.

To create an XML file, make a copy of the sample XML file obtained from Telestream, and rename it as desired. Make all modifications to this new file and keep the original sample file as a template for future use.

The top level element in the XML file is the label element and should not be altered. It contains a UUID which is verified for each transcode. There are three main elements under the label element: AFDEnable, AudioMetaEnable, and XSDEnable:



Note: All XML screen shots are taken from XML Notepad, a free utility application from Microsoft.

Subframe1Data (and *Subframe2Data*) - In addition to the common VancLineNumber and VancField elements, Subframe1Data contains five other elements: revision_id, originator_id, originator_address, DolbyE_Complete_Metadata_Data_Segment, and DolbyD_Complete_Program_Metadata_XBSI_Payload:

The screenshot displays a tree view on the left and a values pane on the right. The tree view shows the following structure:

- AudioMetaEnable
 - name
 - description
 - type
 - #text
- MetadataInsertMethod
 - Subframe1Data
 - name
 - description
 - VancLineNumber
 - VancField
 - revision_id
 - name
 - description
 - type
 - #text
 - originator_id
 - name
 - description
 - type
 - #text
 - originator_address
 - name
 - description
 - type
 - #text
 - DolbyE
 - DolbyE_Complete_Metadata_Data_Segment
 - DolbyD
 - DolbyD_Complete_Program_Metadata_XBSI_Payload
 - Subframe2Data

The values pane shows the following details for Subframe 1 Data:

- Audio Metadata Enabled: boolean, true
- Subframe 1 Data: Includes the data segments for Subframe 1.
- revision_id: numeric, 0 (Set revision_id)
- originator_id: numeric, 2 (Set originator_id)
- originator_address: numeric, 33408 (Set originator_address)

Each Dolby metadata segment (DolbyE and DolbyD) has a text value that can be set *true* or *false*. When set *true*, the Dolby metadata is inserted. Each segment also contains a MetadataParams element that contains a list of named elements. Each name corresponds to an element in a Dolby metadata segment and has a description, number of valid bits, type (*numeric* is an integer value, *fixed* is a set integer value, *text* is a character string), and a value:

The screenshot displays a tree view on the left and a values pane on the right. The tree view shows the following structure:

- DolbyE_Complete_Metadata_Data_Segment
 - name
 - description
 - type
 - #text
 - MetadataParams
 - name
 - description
 - elementCount
 - program_config
 - name
 - description
 - bits
 - type
 - #text

The values pane shows the following details for Dolby E Complete Metadata:

- Dolby E Complete Metadata: boolean, true (Set true to enable)
- Dolby E Complete Metadata Data Segment. Enable flag..: boolean, true
- Metadata Parameters: Holds the metadata parameters, 38
- program_config: description, program_config element, only single programs.. (description); 6 (Number of bits); numeric (type); 11 (Set value)

Note: *description*, *bits*, and *type* are all taken directly from the standards documents (SMPTE 2020 and SMPTE RDD-6) and should not be changed. Only the *text* value can be changed, and only changed to valid values as specified in the standards.

Subframe2Data - (See **Subframe1Data** above and SMPTE 2020 and RDD 6 for details).

XDS insertion

XDS insertion allows a type code and a string of XDS information to be inserted into the field 2 caption stream (See CEA-608 for details). The XDSEnable element contains an extra attribute that specifies the number of services that are included. This should be set to 1. Currently the IsciService element is the only supported XDS service. This allows an XDS typecode and string to be placed into the field 2 data stream at a rate defined by repetitionRate element. The IsciService element contains three elements: *typeCode* (the XDS type code), *IsciString* (the string value), and *repetitionRate* (the repetition rate for the string to be inserted). Text is the frame count value:

| Element | Property | Value | Annotation |
|-------------|----------------|----------|--------------------------------|
| XDSEnable | numServices | 1 | Set numberServices to 1 |
| | IsEnabled | true | |
| IsciService | IsEnabled | true | Set true to enable IsciService |
| | typeCode | 3 | Set the typeCode |
| IsciString | Value | ABCD1234 | Set IsciString value |
| | repetitionRate | 60 | Set repetitionRate |

Known issues with VANC Insertion

- AFD insertion: The AFD value will be inserted into both fields regardless of the field chosen.
- Audio Metadata insertion: Only a single program & Method B metadata insertion is supported
- XDS insertion: Limitations exist for preserving an existing field 2 data stream. The XDS insertion may overwrite portions of an existing field 2 data stream.

Copyright and Trademark Notice

©2010 Telestream, Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, altered, or translated into any languages without written permission of Telestream, Inc. Information and specifications in this document are subject to change without notice and do not represent a commitment on the part of Telestream.

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

All other brand, product, and company names are the property of their respective owners and are used only for identification purposes.

Limited Warranty and Disclaimers

Telestream, Inc. warrants to you, as the original licensee only, that the software you licensed will perform as stated below for a period of one (1) year from the date of purchase of the software by you:

The software will operate in substantial conformance with its specifications as set forth in the applicable product user's guide/published specifications/product description. Telestream does not warrant that operation of the software will be uninterrupted or error-free, will meet your requirements, or that software errors will be corrected. Telestream's sole liability under Section 1 of this Limited Warranty shall be to use reasonable commercial efforts to bring the Software's performance into substantial conformance with the specifications in the applicable product user's guide/ published specifications/product description.

FlipFactory has been designed for professionals skilled in the art of digital media transformation and workflow automation, to facilitate the automation of complex media operations and workflow that require a multitude of input and output media formats, delivery to numerous types of media devices and file systems, and notification of media systems including broadcast automation systems and media asset management systems.

The FlipFactory architecture and user interface is designed to provide maximum flexibility in the setup and configuration of these complex media transformations and workflow. In providing this high degree of flexibility, it is possible for media transformation and workflow processes to be configured that are impractical, likely to result in unexpected or unintended results, or beyond the limits of FlipFactory to perform satisfactorily. Additionally, FlipFactory may be executed on a platform that lacks the performance or capacity to perform the media transformations and workflow you've configured, which is your responsibility to specify. Telestream has chosen to implement FlipFactory to provide the greatest flexibility without limiting its functionality to only those transformations and workflow that are known with certainty to be within its performance capabilities, including those limits imposed by the platform upon which you have installed FlipFactory.

Therefore, you acknowledge that you may create transformations and workflow that are impractical or beyond your FlipFactory installation's limits, and Telestream does not warrant that each transformation or workflow you specify or use will complete without error.

Limitations of Warranties. EXCEPT AS EXPRESSLY SET FORTH IN SECTION 1 ABOVE, NO OTHER WARRANTY, EXPRESS OR IMPLIED, IS MADE WITH RESPECT TO THE SOFTWARE, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OF THIRD PARTY RIGHTS AND THOSE ARISING FROM A COURSE OF DEALING OR USAGE OF TRADE. NO WARRANTY IS MADE THAT USE OF THE SOFTWARE WILL BE ERROR FREE OR UNINTERRUPTED, THAT ANY ERRORS OR DEFECTS IN THE LICENSED MATERIALS WILL BE CORRECTED, OR THAT THE SOFTWARE'S FUNCTIONALITY WILL MEET YOUR REQUIREMENTS.