OpenCable™ Specifications

MUX 3.0 Specification

OC-SP-MUX3.0-I01-130215

ISSUED

Notice

This OpenCable specification is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by CableLabs®. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document.

Neither CableLabs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein.

DISCLAIMER

This document is published by Cable Television Laboratories, Inc. ("CableLabs®").

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various agencies; technological advances; or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein. CableLabs makes no representation or warranty, express or implied, with respect to the completeness, accuracy, or utility of the document or any information or opinion contained in the report. Any use or reliance on the information or opinion is at the risk of the user, and CableLabs shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

This document is not to be construed to suggest that any affiliated company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any cable member to purchase any product whether or not it meets the described characteristics. Nothing contained herein shall be construed to confer any license or right to any intellectual property, whether or not the use of any information herein necessarily utilizes such intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	OC-SP-MUX3.0-I01-130215			
Document Title:	MUX 3.0 Specification			
Revision History:	I01 - Released 2/15/13			
Date:	February 15, 2013			
Status:	~~Work in Progress~~	~~Draft~~	**Issued**	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

Contents

Figures

Tables

1 SCOPE

1.1 Introduction and Purpose

The introduction of enhanced services, such as ETV, delivered on the cable network creates a new set of requirements on the system elements of the delivery network. In order to fully scale video with enhanced services, this specification addresses operational and technical areas within the service delivery network systems and infrastructure. A given cable system will receive services, programming, or advertising containing enhanced services from multiple network sources and may augment those services with locally-inserted enhancements (e.g., local enhanced ad insertion).

The enhancements delivered within each service will change throughout the day based on programming and advertising schedules. Enhancements include signaling, application and data carried in PIDs in a transport stream for a given service. Cable systems must manage the enhancements from acquisition through to delivery in output transport streams based on rules configured by each operator's business systems.

An essential device within the cable plant is a multiplexer (MUX). Multiplexers have been used within cable plants to manage both video and data services. Historically, the focus for multiplexers has been to configure and optimize video and splice in local advertising. For video services, MUXs primarily perform bandwidth management through rate shaping to allow more video streams to be delivered. More operators are launching a broad range of enhanced services for advertising, programming, and other media (such as enhanced VOD content). A next generation multiplexer ("MUX 3.0") compliant with this specification will provide the ability to manage enhanced services along with the video services. With the growth in demand of both video and enhanced services, cable operators have an increasing need to easily configure, manage, and collect information from MUX 3.0 devices.

This specification defines a common set of interfaces and functionality for the configuration of, management of, and reporting from a MUX 3.0 compliant device.

This specification defines the following for MUX 3.0 devices:

- A configuration interface which extends the existing industry specification Converged Cable Access Platform Operations Support System Interface Specification ([CCAP OSSI]).
- A data model to be carried within the SCTE 130-3 message framework, which enables dynamic configuration and associated reporting.
- A reporting interface, which includes object definitions and protocol conformance, based upon the existing industry IPDR Protocol Specification.

1.2 Conformance Notation

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This word means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

All references are subject to revision, and parties to agreement based on this specification are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

[AM 1.0.1]	Enhanced TV Application Messaging Protocol 1.0, OC-SP-ETV-AM1.0.1-120614, June 14, 2012, Cable Television Laboratories, Inc.
[AM I05]	Enhanced TV Application Messaging Protocol 1.0, OC-SP-ETV-AM1.0-I05-091125, November 25, 2009, Cable Television Laboratories, Inc.
[CCAP OSSI]	Converged Cable Access Platform Operations Support System Interface Specification, CM-SP-CCAP-OSSI-I03-120809, August 9, 2012, Cable Television Laboratories, Inc.
[IPDR/XDR]	IPDR/XDR Encoding Format 3.8 (http://www.tmforum.org/DownloadCenter/7549/home.html#ipdr).
[IPDR/SP]	IPDR Streaming Protocol (IPDR/SP) Specification Version 2.8 (http://www.tmforum.org/browse.aspx?linkID=47351&docID=16236).
[MUX3.0 CONFIG]	MUX3.0 Configuration Schema, OC-MUX3.0-CONFIG-1.0.0.xsd.
[MUX3.0 UNPARSED]	MUX 3.0 IPDR Service Definition for Unparsed Data Reports, OC-MUX3.0-UNPARSED-1.0.0.xsd.
[MUX3.0 BANDWIDTH]	MUX 3.0 IPDR Service Definition for Bandwidth Reports, OC-MUX3.0-BANDWIDTH-1.0.0.xsd.
[MUX3.0-GEN]	Auxiliary schema defining elements to be included in IPDR Records reported by a MUX 3.0 device, OC-MUX3.0-GEN-1.0.0.xsd.
[SCTE 35]	ANSI/SCTE 35 2012, Digital Program Insertion Cueing Message for Cable.
[SCTE 30]	ANSI/SCTE 30 2009, Digital Program Insertion Splicing API.
[SCTE 54]	ANSI/SCTE 54 2009, Digital Video Service Multiplex and Transport System Standard for Cable Television.
[SCTE 130-3]	ANSI/SCTE 130-3 2010, Digital Program Insertion–Advertising Systems Interfaces Part 3 Ad Management Service (ADM) Interface.

2.2 Informative References

This specification uses the following informative references.

[CoDF]	OpenCable Content Definition Format (CoDF), OC-SP-CoDF-I02-110325, March 25, 2011, Cable Television Laboratories, Inc.
[DOCSIS 3.0]	DOCSIS 3.0 Operations Support System Interface Specification, CM-SP-OSSIv3.0-I20-121113, November 13, 2012, Cable Television Laboratories, Inc.
[EBIF 1.0.1]	Enhanced TV Binary Interchange Format, OC-SP-ETV-BIF1.0.1-120614, June 14, 2012, Cable Television Laboratories, Inc.
[IEC 13818-6]	ISO/IEC 13818-6:1998(E), Information technology—Generic coding of moving pictures and associated audio information: Extensions for DSM-CC, 1998.
[OPS GL]	Enhanced TV Operational Guidelines, OC-GL-ETV-OG-V02-091223, December 23, 2009, Cable Television Laboratories, Inc.

[SCTE 130-2]	ANSI/SCTE 130-2 2008a, Digital Program Insertion–Advertising Systems Interfaces Part 2 Core Data Elements.
[SCTE 130-7]	ANSI/SCTE 130-7 2009, Digital Program Insertion–Advertising Systems Interfaces Part 7 Message Transport.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027;
  Phone +1-303-661-9100; Fax +1-303-661-9199; http://www.cablelabs.com
- SCTE, Society of Cable Telecommunications Engineers Inc., 140 Philips Road, Exton, PA 19341
  Phone: 610-363-6888 / 800-542-5040; Fax: 610-363-5898; http://www.scte.org
- TM Forum - IPDR, 240 Headquarters Plaza East Tower, 10th Floor Morristown, NJ 07960-6628
  Phone: 973-944-5100; Fax: 973-944-5110; http://www.tmforum.org/InternetProtocolDetail/4501/home.html
- YANG Central – Tools and examples related to YANG

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Augmentation	In a MUX, Augmentation refers to making changes to applications that are streaming on the plant, for example, changing out an image or a block of text to make it applicable to a local zone.
Authorization	In a MUX, Authorization refers to the condition that results in an action of passing or dropping an application to the output based on a set of filters that are configured.
Configuration	The way in which a MUX is set up to define how input transport streams should be combined and processed into output transport streams.
Flight Window	Date/time range during which an enhancement is delivered.
Flow Point	Within a MUX, a logical element that represents an entity through which one or more transport streams (SPTS or MPTS) passes. A program within a transport stream can contain any combination of PIDs (video, audio, ETV signaling, ETV data or other).
Grooming	The process of mapping an input stream to an output stream through a MUX that may include rate shaping, PID modifications, or other processing of the stream.
IPDR	Standardized interface commonly used in the telecommunications industry to provide information about IP-based service usage, in the form of Internet Protocol Detail Records.
Late Bind	The concept of inserting an enhancement at a point downstream from the original distribution, typically at the MSO headend or local advertising insertion point.
Multiple System Operator	A corporate entity that owns and/or operates more than one cable system.
Network	Broadcast or cable programming network (e.g., TNT or Style).
Pre-Bind	Content that is delivered with enhancements already bound within the audio/video asset. Pre-bound content may be streamed (ex. from a network) or file-based (ex. an enhanced advertisement on an ad server).
Session	In the context of MUX 3.0, the term session refers to video elements that could be a CCAP OSSI program-session, pid-session, or mpts-passthrough-session.
Source Management	The ability to manage video and enhanced data from multiple input sources to multiple outputs defined by CCAP OSSI and extensions.
Splicing	The process of replacing, adding, or dropping part or all of an input stream (ex. a network stream) with another stream (ex. an advertising stream from an ad server).

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

ADM	Ad Management (service)
ADS	Ad Decision Service
BSS	Business Support Systems
CCAP OSSI	Converged Cable Access Platform Operations Support System Interface Specification
CIP	SaFI Campaign Information Package
CoDF	Content Definition Format
DDB	DownloadDataBlock
DII	DownloadInfoIndication
DNS	Domain Name System
DOCSIS	Data-Over-Cable Service Interface Specifications
DSMCC	Digital Storage Medium Command and Control
EBIF	Enhanced Binary Interchange Format
EISS	ETV Integrated Signaling Stream
ETV	Enhanced Television
IAF	SaFI Interactive Application Fulfillment
IAM	SaFI Interactive Application Messaging
IPDR	Internet Protocol Data Record
IPDR-EF	Internet Protocol Data Record Encoding Format
MPTS	Multiple Program Transport Stream
MSO	Multiple System Operator
MUX	Multiplexer
MUX 3.0	Next generation Multiplexer device with ability for Configuration, Management, and Reporting
NVRAM	Non-volatile RAM
OSS	Operations Support Systems
PAT	Program Association Table
PEID	Program Enhancement ID
PES	Program Elementary Stream
PID	MPEG-2 Packet Identifier
PMT	Program Map Table
PSN	Placement Status Notification (SCTE 130)
SaFI	Stewardship and Fulfillment Interfaces
SDT	Service Descriptor Table
SPTS	Single Program Transport Stream
STB	Set-top Box
UA	ETV User Agent
URI	Uniform Resource Identifier
XDR	External Data Representation
XML	Extensible Markup Language
XSD	XML Schema Definition

5 OVERVIEW

A MUX 3.0 compliant device will allow an operator to manage and monitor the delivery of enhanced services to subscribers using the interfaces and functionality defined in this specification.

For configuration, applicable video elements (video-input-ts, session, and video-output-ts) as currently defined by CCAP OSSI (see [CCAP OSSI]) will be used in combination with the MUX 3.0 configuration schema to specify the management of enhanced services. This includes authorization of enhanced applications, augmentation of enhanced applications, reporting of enhanced services-related information, and source management rules applied to enhanced video services.

The MUX 3.0 schema also serves as a data model to be used within an SCTE 130-3 messaging framework. This can be used to dynamically update authorization filters, augmentation, and processing rules within the MUX 3.0. In addition, it provides tracking values to be delivered as part of SCTE 130 Placement Status Notifications (PSNs).

The MUX 3.0 reporting interface defines a standard data payload sent by MUX 3.0 devices. IPDR [IPDR/XDR] will be used to carry the data payload.

5.1 Functions

This specification defines a standard format for multiplexers to support Authorization, Augmentation, Source Management, and Reporting functions for enhanced services.

5.1.1 Authorization

This specification defines a standard configuration and format for operator systems to authorize which enhancements are passed through as part of a transport stream for a service. Specifically, the configuration mechanism defines:

- The ability to configure a whitelist of enhancements that defines:
  - The enhancements that are approved to be passed in the transport stream
  - A mapping of approved enhancements to specific services
  - Approved instances of enhancements
- The ability to configure a blacklist of enhancements to prevent applications from entering the plant
  - The enhancements that should be removed from the transport stream
  - A removal of disallowed enhancements on specific services
  - Disallowed instances of enhancements
- The ability to prioritize the evaluation of configured whitelist and blacklist, in the event multiple configuration entries are present

5.1.2 Augmentation

This specification defines the ability to handle local updates of enhanced applications, including such tasks as replacing an image, text, or an entire application, through standard mechanisms.

5.1.3 Source Management

This specification defines the ability to control enhanced services that source from various Flow Points. This includes functions such as splicing, dropping, grooming, and merging of enhanced services. Enhancements for a given service may be delivered along with the network programming and may also be inserted locally. Furthermore, the enhanced service source may change based on business rules or placement decisions. Therefore, enhancements may require a MUX 3.0 compliant device to support dynamic grooming/splicing operations similar to those provided for video services.

This specification provides a standard mechanism to define:

- The ability to configure a MUX to apply rule-based processing to enhancements originating from multiple sources at different times during the day (e.g., switch between enhancements within a national program to a locally-inserted enhancement during a local ad splice, and back upon completion of the ad splice).
- The ability to define different processing rules and prioritize the rules to manage enhanced services (for example, rules for merging of multiple enhancement sources within a single Flow Point defining an output service delivered to subscriber devices.)

5.1.4 Reporting

This specification defines IPDR Service Definitions to provide consistent reports to configured endpoints. It also defines the ability to configure the frequency of records being reported.

5.2 Interfaces

5.2.1 Configuration Interface

Static configuration of the functions described in the previous section is provided by extending CCAP OSSI at various touch points in the schema. Functions that can be configured using this interface include Authorization, Augmentation, Source Management, and Reporting.

5.2.2 SCTE 130-3 Interface

The MUX 3.0 schema can be used to define the payload within an SCTE 130-3 message exchange to dynamically update processing rules for Authorization, Augmentation, or Source Management on a Flow Point. For example, rather than statically configuring whether to pass ETV Applications from a source service to an output transport, a given delivery system may want to authorize the ETV Applications passed based on the time of day or a specific campaign. The SCTE 130-3 interface provides the ability to specify which enhancements are passed based on dynamic campaign, business, and operational rules.

5.2.3 Reporting Interface

This specification defines a reporting interface to enable communication of Source Management, Authorization, and Augmentation decisions made by the MUX to business and technical operational systems.

This specification defines a schema for reporting:

- Broadcast times and events for each enhancement
- Application of processing rules (ex. authorization policies)
- Bandwidth utilization of the enhancements

5.3 Reference Architecture

The following diagram provides a reference system architecture from content origination through a cable operator's delivery system. Physical network architectures will vary and a given installation may implement logical functions in different physical devices. MUX 3.0 reporting data will be utilized by both Technical Operations and Business Operations (OSS/BSS). The functionality and users of OSS and BSS systems will also vary across operators.

The green boxes represent systems performing MUX 3.0 processing. The red lines indicate enhanced services flows and blue lines indicate management and reporting flows.

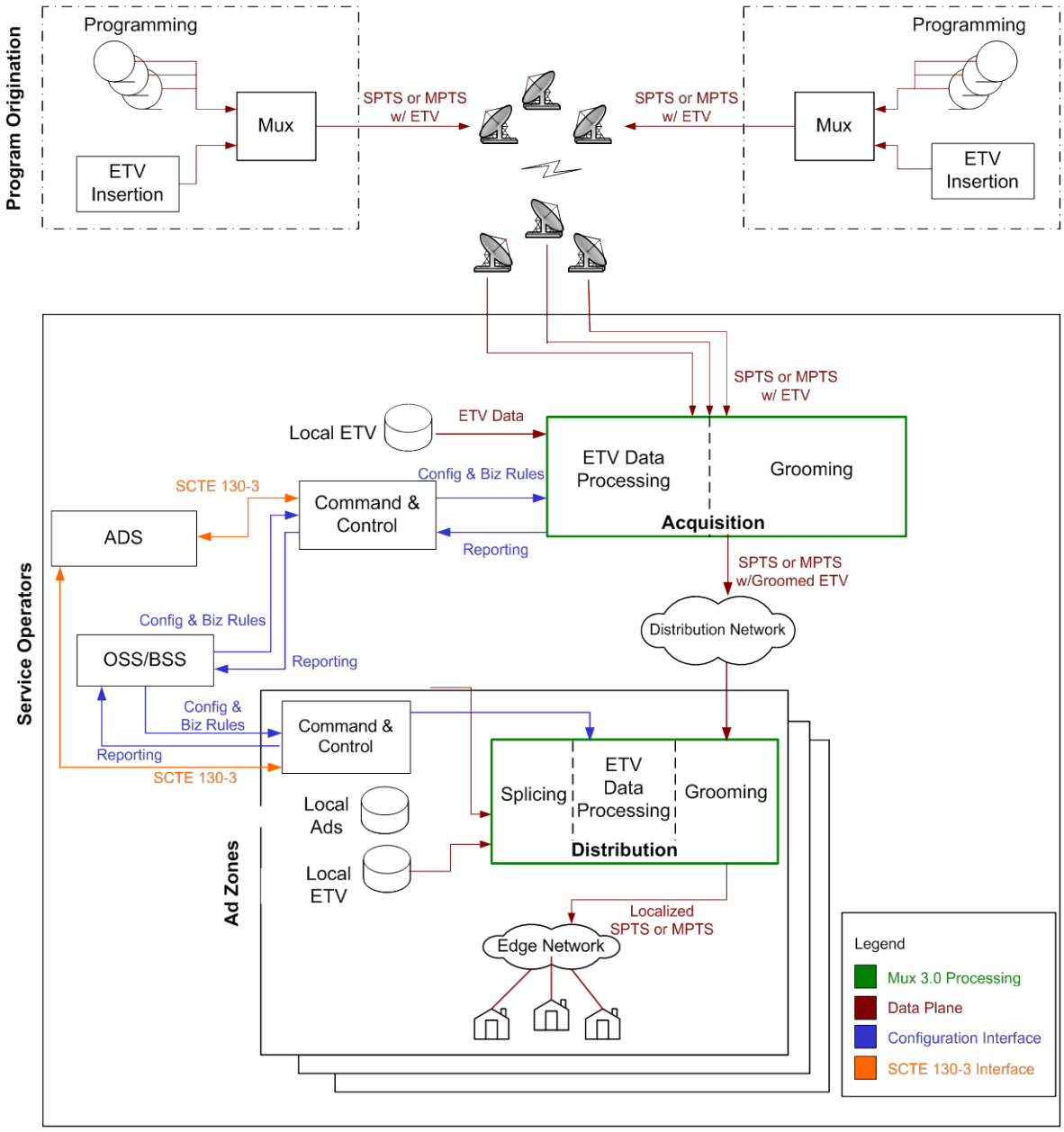


Figure 1 - MUX 3.0 Reference Architecture

6 TECHNICAL SPECIFICATION

6.1 General Configuration

Configuration of a MUX 3.0 device includes general level and more specific information required for management and definition of enhanced services and processing. The general level configuration can be performed in accordance with the CCAP OSSI specification. This includes defining video services (multiple inputs and outputs). Key functionalities required for configuration and management of enhanced services, such as Authorization, Augmentation, and Source Management, are defined as extensions of CCAP OSSI in the MUX 3.0 configuration schema [MUX3.0 CONFIG].

The MUX 3.0 device MUST support export of an XML configuration file using CCAP OSSI with extensions defined by [MUX3.0 CONFIG].

The MUX 3.0 device MUST support import of an XML configuration file using CCAP OSSI with extensions defined by [MUX3.0 CONFIG].

MUX 3.0 MAY be applied to other configuration models. Such configuration models are out of scope of this specification.

The CCAP OSSI schema defines functionality for configuring a MUX 3.0 device including:

- Identification of input transport streams that are received by the MUX.
- Identification of individual streams (defined by applicable video session elements) at the MPTS/SPTS, Program (PMT) or PID level that pass through internal Flow Points within the MUX.
- Identification of output transports that are passed from the MUX.
- Mapping of video session elements to applicable video-output-ts.

  PID values may be renumbered or remapped as they pass through program or PID session Flow Points.

  PID values can also be created or groomed-in prior to the MUX applying additional processing rules at applicable Flow Points.

The MUX 3.0 schema defines functionality for configuring a compliant MUX including:

- Define whitelists, blacklists, and processing rules (e.g., augmentation rules, mapping/processing of multiple sources).
- Identification and configuration of external interface points for management, reporting, etc.
- Support for IP outputs.

6.1.1 Assumptions

- Sources and destinations for each video transport stream are assumed to be configured using CCAP OSSI.
- The MUX 3.0 configuration schema does not configure device or line-card level parameters.

6.1.2 Informative Reference

There are 2 models for creating the configuration files:

- The MUX 3.0 device will export a file in a well-defined format (CCAP OSSI + defined OC_MUX_3_0 extensions), which will create an XML file that represents a snapshot of the current configuration of the MUX device. This configuration file can then be changed and imported into the MUX 3.0 device to modify the configuration.
- Use YANG to create the configuration file. Currently there are various tools in the market for working with YANG. Refer to the link in Section 2.3 for more information.

The long term goal is to use a third party tool to configure the MUX 3.0 devices using the CCAP OSSI model with MUX 3.0 extensions.

Appendix I provides an example configuration of inputs, outputs, and program sessions using [CCAP OSSI].

Note: A MUX 3.0 device may not be aware of an advertising server until the server has initiated an SCTE 30 Init_Request. For this reason, a configuration export performed prior to any communications with ad servers may not contain a video-input-ts for the dynamic insertion service. When configuring a program-session Flow Point that is associated with an ad server, it is recommended to specify the session-input-ts as '0' (zero), since it is required by the ccap schema, but not statically available for the ad server at the time of configuration.

6.2 General Management Configuration

The MUX 3.0 schema provides the ability to configure general settings, such as default Authorization, Reporting settings, and ADS registration information. These features can be configured statically and are not modifiable via the SCTE 130-3 interface.

The MUX 3.0 Configuration element MUST be supported as an extension to the ccap/management element (see [CCAP OSSI]).

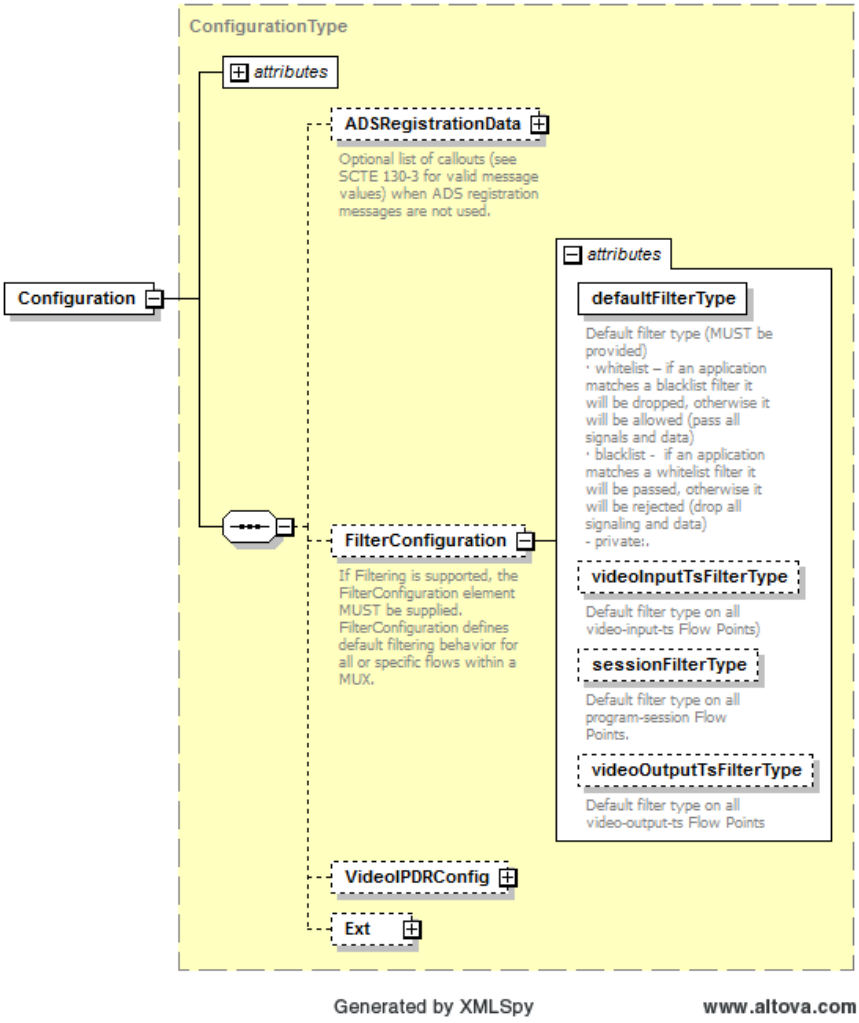


Figure 2 - Configuration Element

6.2.1 Default Authorization

Default authorization rules can be configured using the FilterConfiguration element as part of the Configuration element.

The FilterConfiguration element contains attributes which specify default behaviors on a MUX3.0 compliant device. A default behavior (whitelist or blacklist) for all input, session, and output elements under the video element in CCAP OSSI can be specified by using the defaultFilterType attribute; one for all video input sessions using the videoInputTsFilterType attribute; one for all program, pid, and mpts-passthrough session Flow Points by using the sessionFilterType; and one for all video output Flow Points by using the video outputTsFilterType attribute.

In all cases, these can be specified to be a whitelist type or a blacklist type. These default filters take effect if, during processing through the Flow Points, there are no filter matches that result in the explicit action of passing or dropping the application. If the default filter type is defined to be a whitelist, the application will be passed. Conversely, a blacklist default will drop the application. Section 6.5 describes this behavior in more detail.

Appendix II provides an example of a default authorization configuration.

6.2.2 Reporting Configuration

Reporting from a MUX 3.0 device is via IPDR, as discussed in Section 6.8. The MUX 3.0 configuration schema provides the ability to configure IPDR Collectors, Streaming Sessions, and Exporters to receive reports based on MUX 3.0 activity. This is defined in the Configuration element, pictured below:

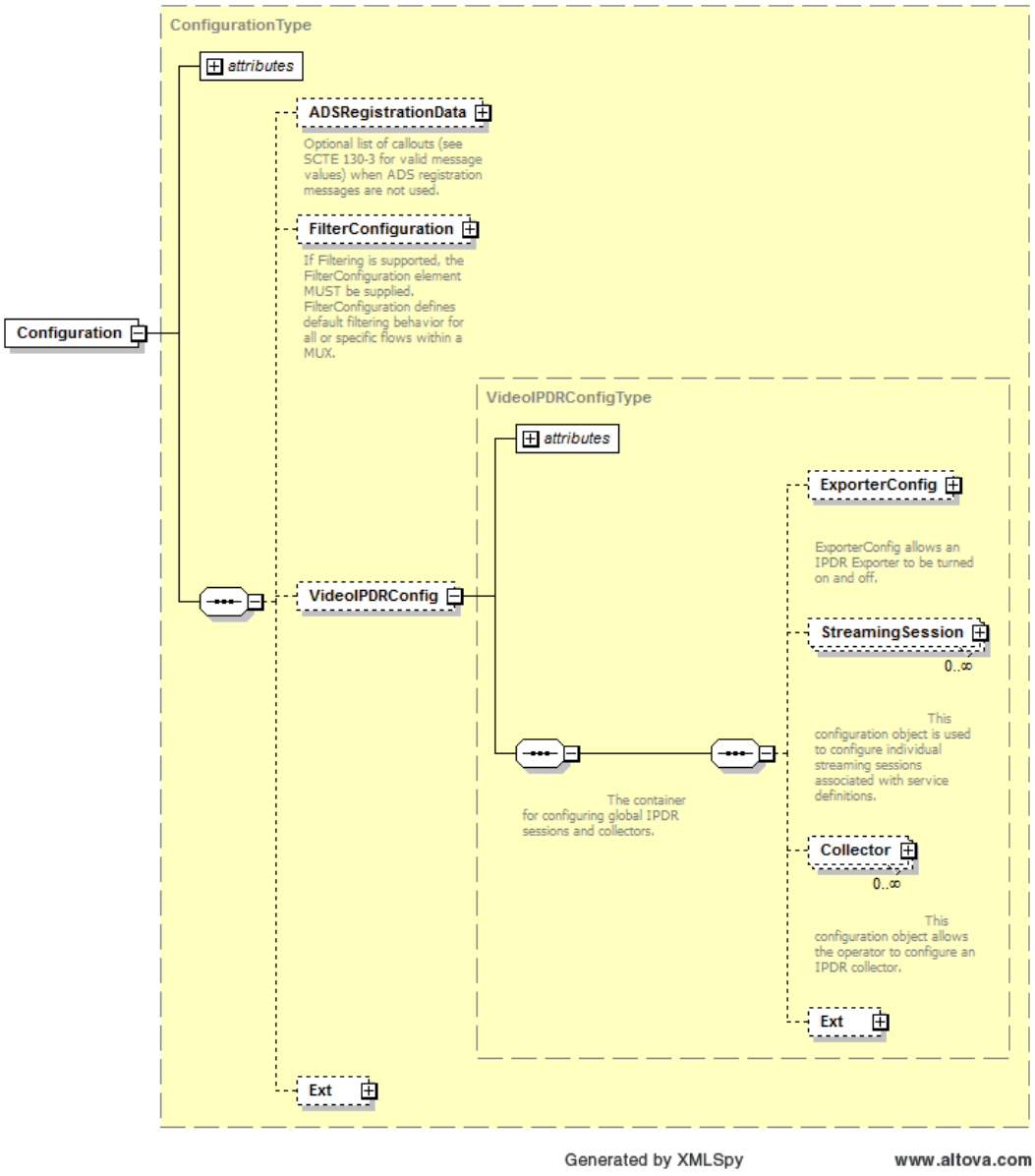


Figure 3 - VideoIPDRConfig Element

Refer to Appendix III for an example IPDR configuration.

The following objects are used to configure IPDR collectors and sessions:

- ExporterConfig – allows an IPDR exporter to be turned on and off.
  - Enabled – the Enabled element specifies the IPDR reporting state on a global level. If false, no IPDR reporting will occur. If true, then the reporting state is further specified by the reporting state attributes. The Enabled element is true by default.
- StreamingSession – this configuration object is used to configure individual streaming sessions associated with service definitions and collectors.
  - SessionId – id for this session.

- KeepAliveInterval – the interval, in seconds, at which IPDR "keepalives" are sent from the MUX 3.0 IPDR exporter to the collector.
- AckTimeInterval – the interval, in seconds, in which the MUX 3.0 exporter waits for an acknowledgement.
- AckSequenceInterval – the maximum number of unacknowledged records that can be sent by the MUX 3.0 IPDR exporter before receiving an acknowledgement.
- CollectionInterval - Where streaming is of the type timeInterval, this object configures the interval in seconds at which IPDR information is extracted from the MUX 3.0 management objects and transmitted to the collector. Where streaming is of the type timeEvent, this object identifies the interval at which the MUX 3.0 IPDR exporter will close the IPDR session to allow IPDR session processing to occur. Records created by Service Definitions supporting timeEvent are sent when the event is generated.
- StreamingType – the type of IPDR streaming used for this session. An enumeration with the options:

  other (1) – used when a vendor extension has been implemented for this object

  time-interval (2)

  ad-hoc (3)

  event (4)

  time-event (5)
- Enabled – controls whether this IPDR session is enabled or disabled; defaults to true.
- ServiceDefinitionType – the service definition type for this IPDR session. This is a string that specifies the URI of the service definition schema.
- CollectorReference – a reference to the CollectorId that receives this IPDR stream.

- Collector – this configuration object allows the operator to configure an IPDR collector.
  - CollectorId – a unique identifier for this collector.
  - CollectorIp – the IP address of the collector. The collector establishes a connection to the MUX 3.0.
  - CollectorName – optional name for the IPDR collector.
  - CollectorPort – the port used by the collector to communicate with the MUX 3.0. If not specified, defaults to 4737.
  - Priority – this object configures the priority of the IPDR collector. The priority is used to elect the primary and active collector. The collector with the highest priority (lowest number) will be elected.

In addition to configuring which types of records are sent to which collectors, the VideoIPDRConfig element provides the ability to set up default reporting states for several report components. These defaults can be overridden at each Flow Point, using the IPDRConfig element in FlowConfig. Following is a list of the reporting states. In all cases except bwSampleInterval and bwNumSamples, TRUE means report; FALSE means don't report, and if not configured, these are all FALSE by default.

- defaultReportingState – the default for all local ReportingAttributeGroup usage. If FALSE, any other reporting control in this attribute group MUST be ignored; otherwise, each additional reporting control MUST be applied.
- pmtReportingState – the default state of reporting for the PMT.
- eissReportingState – the default state of reporting for the ETV EISS.
- diiReportingState – the default state of reporting for the ETV DII.
- ddbReportingState – the default state of reporting for the ETV DDB.
- sdtReportingState – the default state of reporting for the SDT.

- scte35ReportingState – the default state of reporting for SCTE 35 sections.
- bwReportingState – the default state of reporting for bandwidth.
- bwSampleInterval (unsigned int) – the sampling interval (in seconds) for bandwidth collection. If not specified, the default is 5 (seconds).
- bwNumSamples (unsigned int) – the number of sample intervals for which the MaxBandwidth and AvgBandwidth are calculated. If not specified, the default is 6 (sample intervals).

Note: Although the bwSampleInterval and bwNumSamples can be specified for each flow point, there may be some measurement tools that require these values to be consistent across all flow points, especially if the tool is aggregating sample data over all flow points that make up a transport stream.

6.2.3 ADS Configuration and Defaults

The ADSRegistrationData element provides the ability to configure adsCallOutState defaults and ADS endpoint addresses.

The MUX 3.0 device MUST support registration with one or more decision systems using at least one of the following methods:

1. Standard SCTE 130 registration by the decision system. Refer to SCTE 130 Part 3 documentation [SCTE 130-3] for more information.

   This will be used to configure ADS Callout endpoints at the ccap:video level.

2. Configuration of messages and addresses for one or more decisions system using the ADSRegistrationData element.

   An attribute, adsCallOutState, provides the ability to specify at the system level if all Flow Points will send the designated messages to the designated addresses. This can be overridden at the FlowConfig level. If not specified, defaults to “false”.

   - Setting the system level adsCallOutState attribute to “true” within the ADSRegistrationData element will result in all Flow Points performing the specified call out if the Callout is configured.
   - Setting the adsCallOutState attribute to “true” within the FlowConfig element will result in a call out from the Flow Point if the Callout is configured, regardless of the system level default setting.
   - The adsCallOutState is only applicable for configured Callout endpoints, not Callout endpoints established through SCTE 130 registration.

   The ADSCallOut Element specifies a message type and address.

   - The message can identify a specific message type (e.g., PlacementRequest, PlacementResponse, PlacementStatusNotification, and PlacementUpdateNotification), or apply to all messages if left blank.
   - The address represents the address of the endpoint to receive the specific messages.

Figure 4 illustrates the schema components used in configuring the ADS Callout endpoints manually, and Appendix IV contains an example configuration of an ADS Registration endpoint.

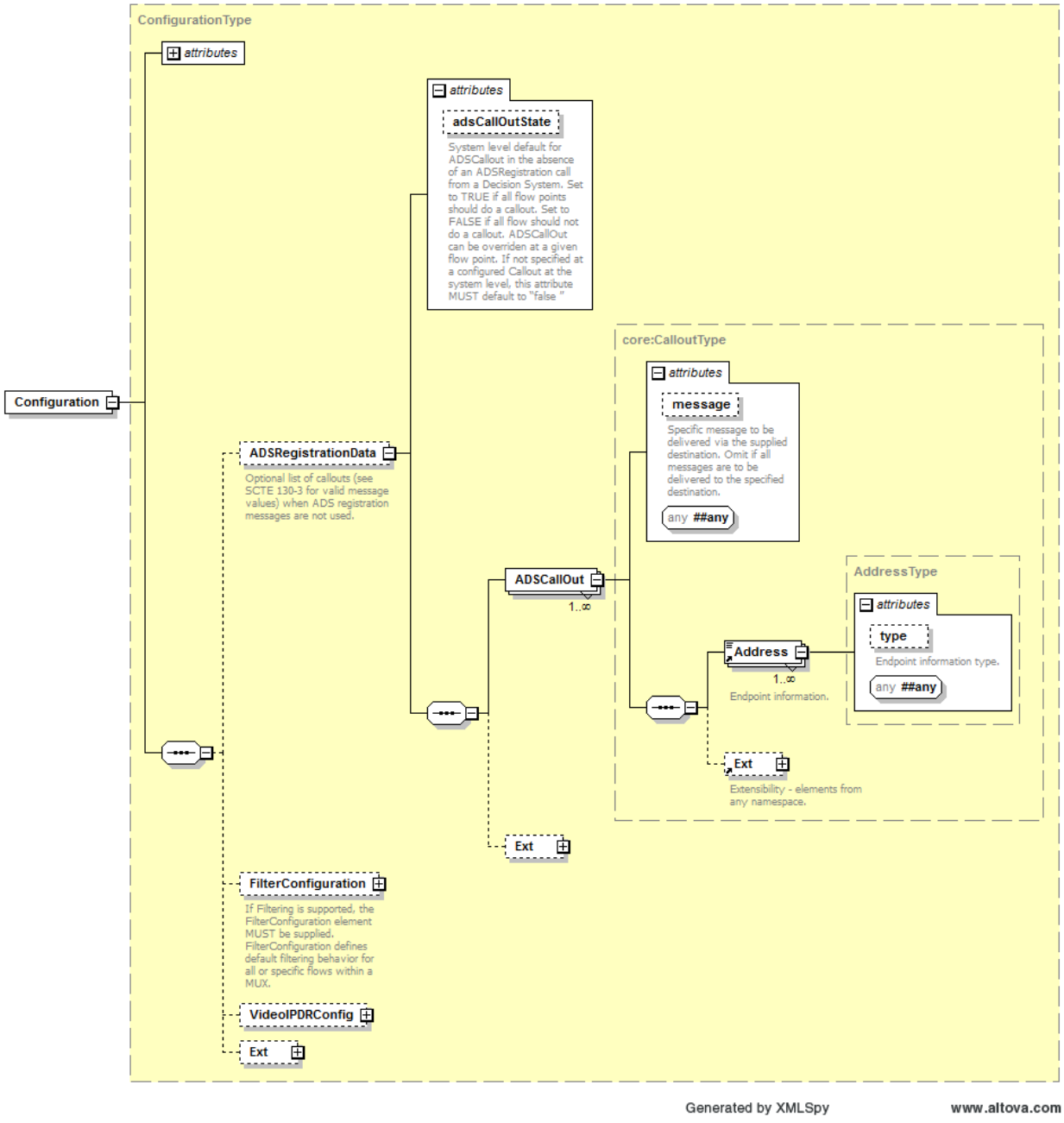


Figure 4 - Configuration of ADS Endpoints

6.2.4 Identifying an IP Video Output

The CCAP OSSI specification does not currently support (as of version I03) defining an IP (multicast or unicast) output for the video element. To address the current shortcoming, the MUX 3.0 configuration schema provides this capability through the VideoOutputTs element in a FlowConfig as an extension on a ccap video-output-ts element.

It is expected that the IP video output will be configured through the vendor-specific configuration mechanism. The MUX 3.0 VideoOutputTs element is used to identify a specific video output Flow Point, on which Authorization application filters and/or Source Management rules can be applied. The VideoOutputTs element is not intended to be used to configure output streams on the MUX 3.0.

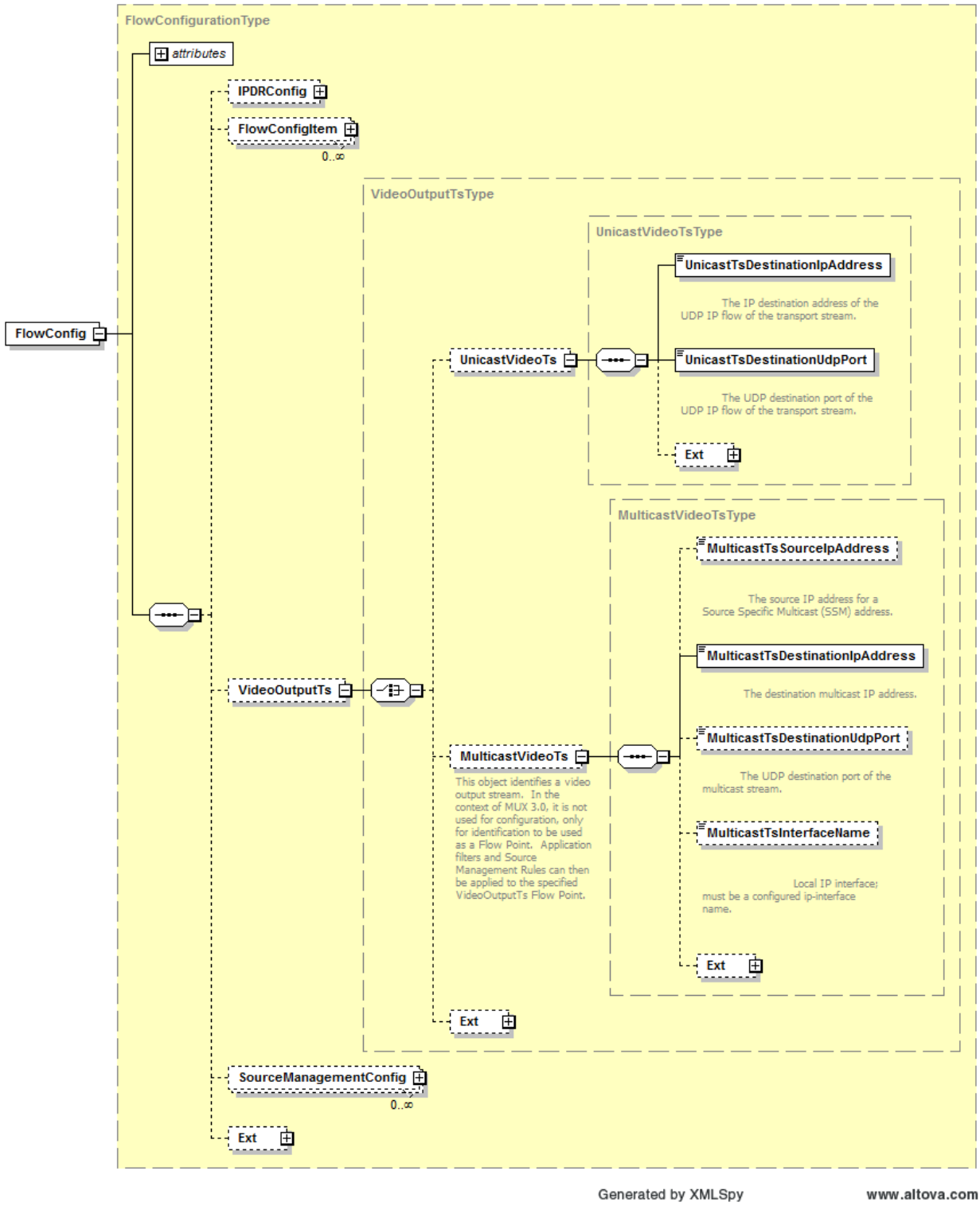


Figure 5 - Configuration of an IP-based Video Output Transport Stream

Figure 5 illustrates the VideoOutputTs element of the FlowConfig element. Either a unicast address and port or a multicast address and port can be specified for the destination. When extending the video-output-ts element in

CCAP OSSI, a video-down-channel-ref, specifying the slot, ds-rf-port, and down-channel is required, so it needs to be reflected in the export.

A MUX 3.0 MUST export the configuration of an IP output using the VideoOutputTs extension.

Refer to Appendix V for a detailed example, which shows configuration of inputs, outputs, and program sessions for a MUX device using [CCAP OSSI] and a multicast IP video output.

6.3 Flow Point Configuration

[MUX3.0 CONFIG] defines elements that can be used for static configuration as well as within an SCTE 130-3 message exchange to support Authorization (whitelists and blacklists), Augmentation, and Source Management of enhanced services at specific Flow Points.

Note: the ccap/video element can be considered a container of Flow Points (not a Flow Point itself), and a FlowConfig element can be applied to it. Typically this would be a filter that is defined globally for all ccap/video input, session and output Flow Points of a MUX 3.0 compliant device.

FlowConfig/FlowConfigItem elements MUST be supported at the following CCAP OSSI elements:

- video
- session elements (program-session, pid-session and mpts-passthrough-session)

FlowConfig/FlowConfigItem elements SHOULD be supported at the following CCAP OSSI elements:

- video-input-ts
- video-output-ts

FlowConfig/SourceManagementConfig elements MUST be supported at the following CCAP OSSI element:

- session elements (program-session, pid-session and mpts-passthrough-session)

FlowConfig/SourceManagementConfig elements SHOULD be supported at the following CCAP OSSI elements:

- video
- video-input-ts
- video-output-ts

FlowConfig/VideoOutputTS elements MUST be supported at the following CCAP OSSI element:

- video-output-ts

FlowConfig/IPDRConfig elements MUST be supported at the following CCAP OSSI element:

- session elements (program-session, pid-session and mpts-passthrough-session)

FlowConfig/IPDRConfig elements SHOULD be supported at the following CCAP OSSI elements:

- video-input-ts
- video-output-ts

The following table outlines the required level of support for the FlowConfig elements:

Table 1 - Configuration Support for FlowConfig Elements

	FlowConfig Item	***SourceManagementConfig***	***VideoOutputTS***	***IPDRConfig***
video	MUST	SHOULD	NA	NA
video-input-ts	SHOULD	SHOULD	NA	SHOULD
session	MUST	MUST	NA	MUST
video-output-ts	SHOULD	SHOULD	MUST	SHOULD

For SCTE 130-3 message exchanges:

- the AppList element MUST be supported
- the SourceManagementConfig element MUST be supported

Section 6.4 describes the SCTE 130-3 message exchange in more detail.

Figure 6 shows the schema elements and attributes used to configure and control the authorization function.

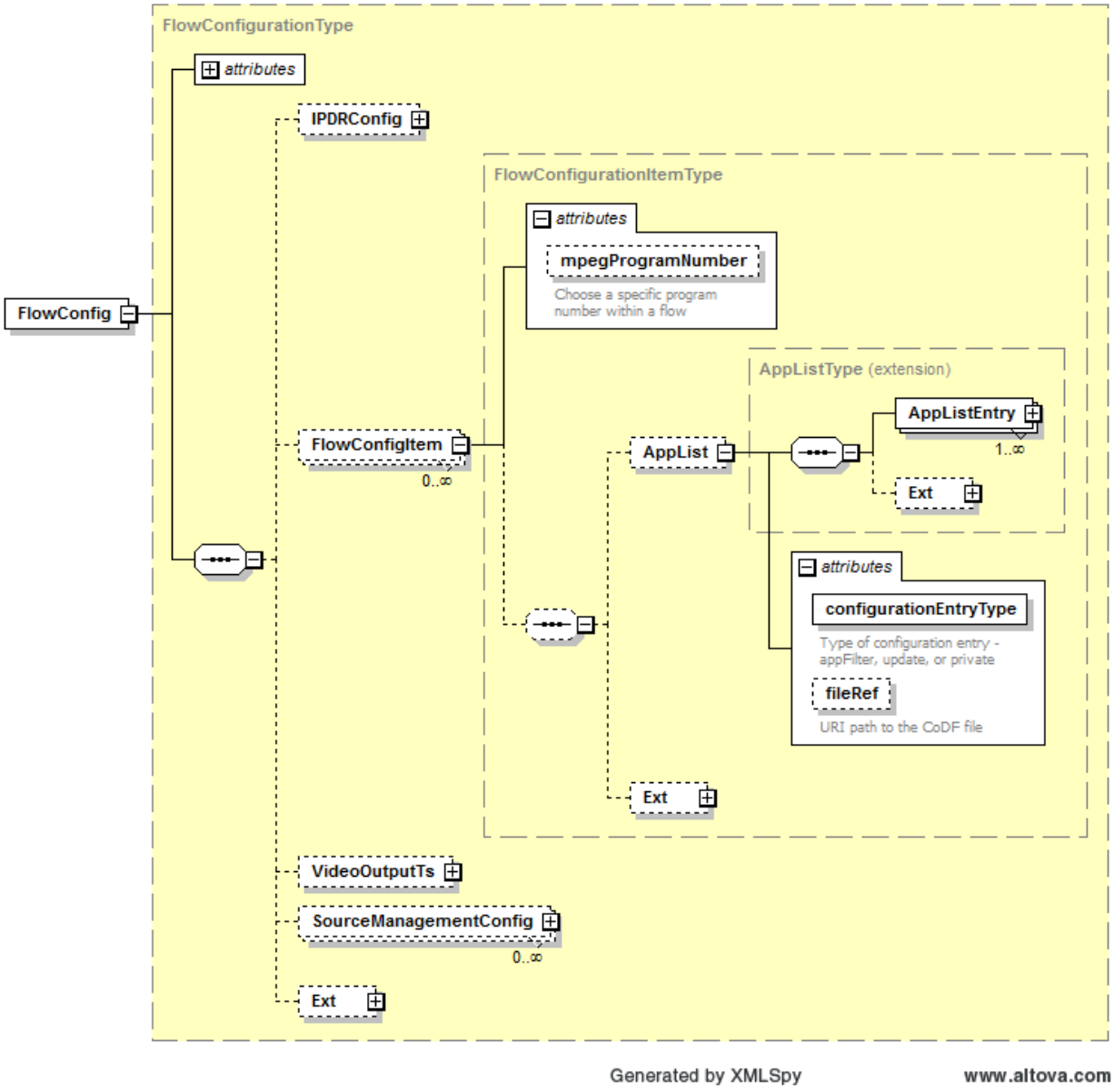


Figure 6 - Authorization Configuration Schema

6.3.1 Authorization

Authorization of specific applications are configured using a FlowConfig element as follows:

- A FlowConfig element can include zero, one or more FlowConfigItem elements.
- The FlowConfigItem element contains an optional attribute to identify the mpegProgramNumber. To define a FlowConfig for all video elements, extend the CCAP OSSI video element and do not specify the mpegProgramNumber.
- A FlowConfigItem element can include an optional AppList element.
- An AppList element contains one or more AppListEntry elements.

- An AppList element contains a configurationEntryType attribute that MUST be set to "appFilter" to notify the MUX 3.0 device that this configuration describes a list of applications to be used for authorization.
- If configurationEntryType is set to "appFilter", the optional AppList element attribute fileRef MUST be ignored if present.
- The AppListEntry element contains attributes that define an application filter. Applications may be identified by matching against the following attributes:
  - organizationID is optional and specifies a 32-bit organization ID. If organizationID is not specified, any other application identifiers in this list MUST be ignored.
  - applicationID is optional and specifies a 16-bit application ID.
  - majorVersion is optional and specifies an integer (0-255) application major version.
  - minorVersion is optional and specifies an integer (0-255) application minor version for matching.
  - applicationInstanceID is optional and specifies an application instance identifier for matching.
  - applicationPriority is optional and specifies the application priority (0-255) for matching. Used in combination with the other application specific attributes.
  - testFlag is optional and specifies an 8-bit integer which specifies a test flag field. Per [EBIF 1.0.1], the usage of this field is entirely defined by the MSO.
  - privateData is optional and reflects an ETV application descriptor private data field. It is a HEX string value.
- The AppListEntry element contains attributes that define processing filters: priorityOfEntry and typeOfEntry.
  - The typeOfEntry attribute is required if defining an authorization filter, and is used to identify the action to be taken on applications that match the filter criteria. typeOfEntry can be whitelist, blacklist or private.

    If typeOfEntry is specified as whitelist, an application that matches the filter defined by the AppListEntry MUST be passed in the stream (unless dropped as part of another process in the MUX). If typeOfEntry is specified as blacklist, an application that matches the filter defined by the AppListEntry MUST be dropped.
  - The priorityOfEntry attribute specifies the order entries are processed. It is used when more than one AppListEntry exists in the AppList table. The default value is priority 1. The lower the value, the higher the priority. If more than one AppListEntry has the same priority, the order of interpretation is not specified. Once an application is matched to an AppListEntry, the MUX 3.0 MUST process the application according to the typeOfEntry, and discontinue further filter matching on that Flow Point.

In an AppListEntry element, if organizationID is not specified, all other application identifiers are not relevant. If organizationID is not specified, all applications present at the Flow Point will match this filter. This can be used to define a Flow Point to "pass all" or "block all" applications that do not explicitly match a higher priority filter. It is recommended to configure filters such as this (that match all applications) with a higher priorityOfEntry value (which makes it a lower priority), so that other filters are applied first.

Refer to Appendix VI for an example which configures a whitelist.

6.3.2 Augmentation

The configuration schema contains elements that are defined to provide support for augmentation of enhanced services.

The MUX 3.0 MUST be used to support the following augmentation functions of enhanced services using the [MUX3.0 CONFIG]:

- configuration to replace an existing resource in an enhancement
- configuration to add or modify a resource to an existing enhancement
- configuration to replace an entire enhancement

The following provides a flow of the elements that are used to support augmentation.

- Augmentation of specific applications is also configured through the a FlowConfig element, but in the augmentation case, the AppList element will have configurationEntryType set to "update", and the fileRef attribute will supply the location and name of the application or resource which will be used by the MUX 3.0 to perform the augmentation.

  The MUX 3.0 device MUST support CoDF-formatted files pointed to by the "fileRef" attribute.

  The MUX 3.0 device MAY cache the file after it has been downloaded from the external server.

  The AppListEntry element attributes will be used to identify the application to be augmented. Applications may be identified by matching against the following attributes:

  - organizationID is optional and specifies a 32-bit organization ID.
  - applicationID is optional and specifies a 16-bit application ID.
  - majorVersion is optional and specifies an integer (0-255) application major version.
  - minorVersion is optional and specifies an integer (0-255) application minor version for matching.
  - applicationInstanceID is optional and specifies an application instance identifier for matching.
  - applicationPriority is optional and specifies the application priority (0-255) for matching. Used in combination with the other application-specific attributes.
  - testFlag is optional and specifies an 8-bit integer which specifies a test flag field. Per [EBIF 1.0.1], the usage of this field is entirely defined by the MSO.
  - privateData is optional and reflects an ETV application descriptor private data field. It is a HEX string value.

6.3.3 Source Management

The configuration schema contains elements that support Source Management at Flow Points within a MUX.

The SourceManagementConfig element provides the ability to specify rules for handling one or more PID types during splice and non-splice time that are inputs to a Flow Point. The rules specify when PID types are to be passed through a Flow Point or not. If more than one PID of a given type is passed through a Flow Point, the processing of those PIDs by the receiving Flow Point will be based on the PID type.

Initially, only PID types of EISS and ETV-DATA are to be supported; however, PID type and PID number are included as attributes for the rules in case the scope expands to accommodate other PID types. If multiple EISS PIDs are to be passed, a merge compliant with [AM 1.0.1] will be performed by the Flow Point. For example, if EISS PIDs are passed from two or more Flow Points, the output Flow Point is expected to perform the merge. (Merging of ETV-DATA PIDs is out of scope of this specification.)

In the descriptions throughout this section, the term "present", when applied to rule definitions, implies that the specified PID type exists in the transport stream and is compliant with SCTE 54, Section 5.5, [SCTE 54] or, in the case of a splice insert, the PID exists in the insertion stream, or is included in the Splice_Request message as defined in SCTE 30 Section 7.5 [SCTE 30]. It also implies it has not been removed by any authorization filtering.

Also note, a splice event starts when content from another Flow Point replaces the content for a Flow Point receiving some type of signaling (e.g., SCTE 35). The splice event ends when the delivery of content from the original Flow Point resumes. In the typical use case using SCTE 35 and SCTE 30, a splice event is bounded by one or more SCTE 30 Splice_Request messages sent to a MUX. The time value of the first Splice_Request message defines the start of a splice event and the duration value defines the end of the splice event. Additional Splice_Request messages can be sent to the MUX to override a previously communicated end of splice event. See [SCTE 30] for more details on how SCTE 30 Splice_Request messages are to be interpreted by a MUX.

For Flow Points that receive insertion signals (e.g., a broadcast session receiving SCTE 35 cue messages), if no rules are defined on those Flow Points, the following default behavior MUST be applied:

1. During non-splice time (outside of a splice event), the MUX 3.0 MUST pass all audio, video, EISS (if present) and ETV-DATA (if present) to the output.
2. During a splice event, the MUX 3.0 MUST drop all audio, video, EISS (if present) and ETV-DATA (if present) from the output.

For Flow Points that do not receive insertion signals (e.g., an insertion session from an ad server), the default behavior MUST be to pass PID types received. The actual PIDs that are spliced MUST be compliant with applicable standards (SCTE 30 and SCTE 35) or specifications. For example, in SCTE 30 Section 7.5, the Splice_Request is further described and contains information of the stream type for an elementary stream defined. This is used by the MUX 3.0 device to verify it matches the stream type of an elementary stream in the output (network) PMT. Put more simply, only spliced input elementary streams that match the elementary streams defined in the output PMT are passed.

6.3.3.1 Rules

Rules MUST be interpreted as follows:

- keep: pass the inputs to the output on the specified Flow Point outside of a splice event if present.
- drop: do not pass any present (EISS and ETV-DATA) inputs to the output on the specified Flow Point outside of a splice event.
- splice-keep: pass the inputs to the output during splice event if present.
- splice-drop: do not pass any present (EISS and ETV-DATA) during a splice-event.
- splice-exclusive: keep ETV-DATA and EISS from the highest priority Flow Point if present; if not, pass the ETV-DATA and EISS PIDs from next priority Flow Point if present.
- persistent: at all times (non-splice and during a splice event), pass the EISS and ETV-DATA, if present, from the input to the output (note EISS is merged, but priority needs to be applied to select the ETV-DATA on the output).
- create: if not present, create ghost PIDs on the output.

Some additional details on rules:

- Rules MUST only be applied to pidTypes of EISS and ETV-DATA as of this specification issue.
- Typically splice-drop and splice-keep are applied to Flow Points that receive splice signaling (e.g., SCTE-35).
- More than one rule can be defined on a Flow Point. For example, keep and splice-drop explicitly states that the PID will be passed during non-splice time and dropped during splice time.
- If more than one Flow Point has a rule that results in EISS passed to the output, the MUX 3.0 MUST merge the EISS packets into a single EISS stream.
- If more than one Flow Point has a rule that results in ETV-DATA passed to the output, the MUX 3.0 MUST select the ETV-DATA Flow Point with the highest priority rule (which is the lowest number).
- A rule can be defined within one Flow Point for another Flow Point using the nameOfFlowPoint attribute. The purpose for doing this is to define a rule that takes effect at splice time for a Flow Point that does not receive insertion signals, such as SCTE 35 cues.
- If conflicting rules are defined, the rule with the highest priority (lowest number) MUST be applied.

Rules have the following attributes:

- tracking

  This attribute is required. Tracking uniquely identifies this entry. It is a GUID that can be passed in SCTE 130 PSNs or in reporting entries.

- nameOfFlowPoint

  Optional name of the Flow Point on which to apply the rule. If omitted, the rule applies to the Flow Point where this entry is placed.

- pidType

  Optional elementary stream type. If omitted, the rule applies to all EISS and ETV-DATA PIDs in the specified Flow Point. If the Flow Point has multiple programs, all PIDs of this type will be processed unless otherwise constrained to a specific program number.

- pid

  Optional 13-bit packet ID (PID). If omitted, the rule applies to all PIDs of the specified pidType in the specified Flow Point.

- programNumber

  Optional program number within the designed stream on which to apply the rule. If omitted, the rule applies to all programs in the specified Flow Point. This also applies to all PIDs within the program unless constrained otherwise.

The following provides a diagram of the elements defined in the MUX 3.0 configuration schema to support Source Management.

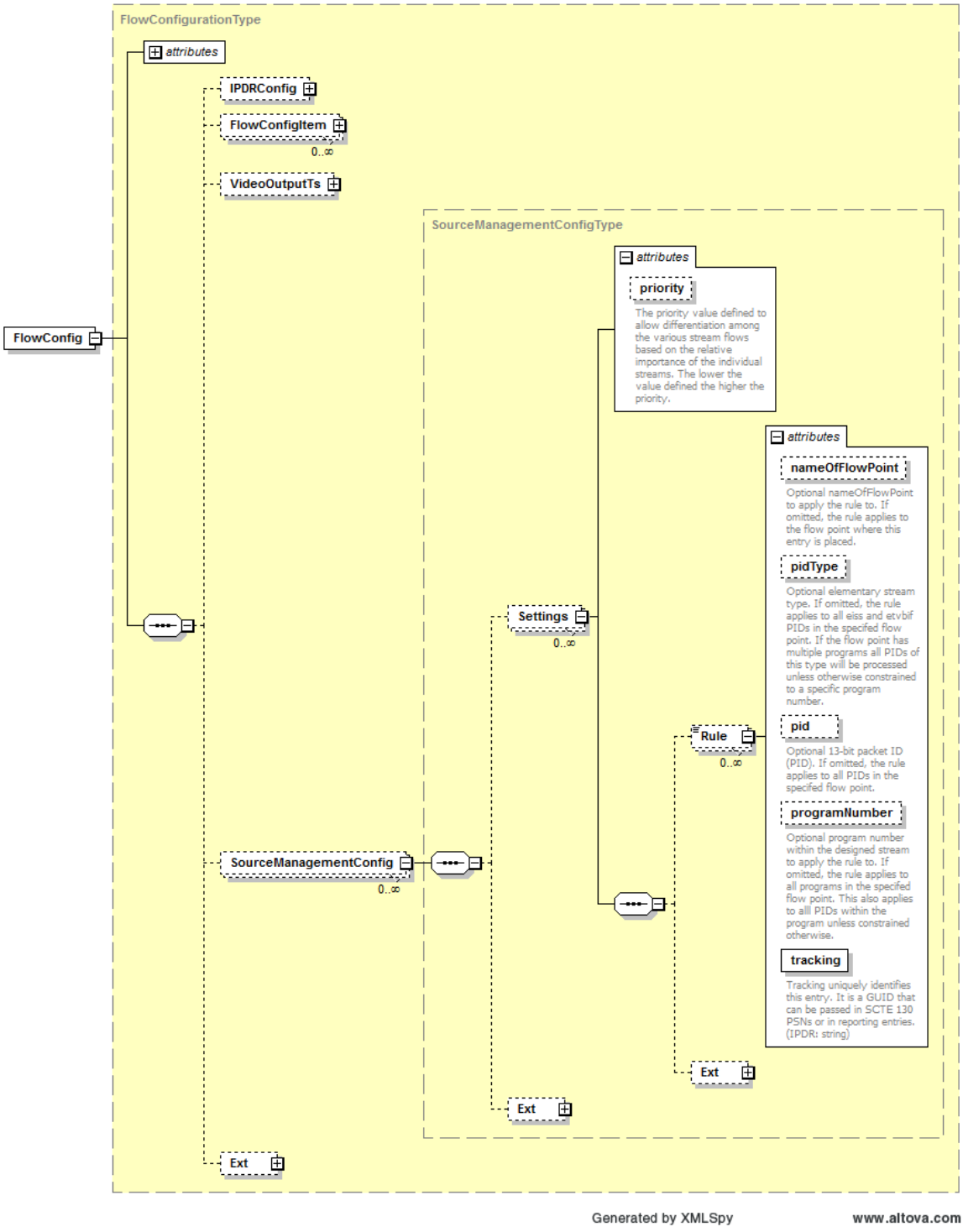


Figure 7 - Source Management Configuration

6.3.3.2 Examples

The following Source Management scenarios are provided as examples in the Appendices:

- Pre-bind Splice-drop (Appendix VII)

- Pre-bind Rules (Ghost PIDs) (Appendix VIII)
- Late-bind Splice-drop (Appendix IX)
- Late-Bind, merge broadcast (Appendix X)
- Rules Splice Exclusive (Appendix XI)

6.3.3.3 Informative Reference

Refer to Appendix XVIII for more informative references.

6.4 SCTE 130-3 Interface

The MUX 3.0 will provide management functionality by using a standard SCTE 130-3-based Request/Response model. Within SCTE 130-3, the MUX will send messages that are analogous to an Ad Decision Manager (ADM).

The MUX 3.0 device MUST support SCTE 130-3-based PlacementRequest, PlacementResponse, PlacementStatusNotification, and PlacementUpdateNotification messages, enabling an operator's ability to:

- Provide updates for Authorization.
- Perform dynamic Augmentation.
- Dynamically modify Source Management Rules.

Message flow via the SCTE 130-3 interface can operate a number of different ways. One scenario is shown below.

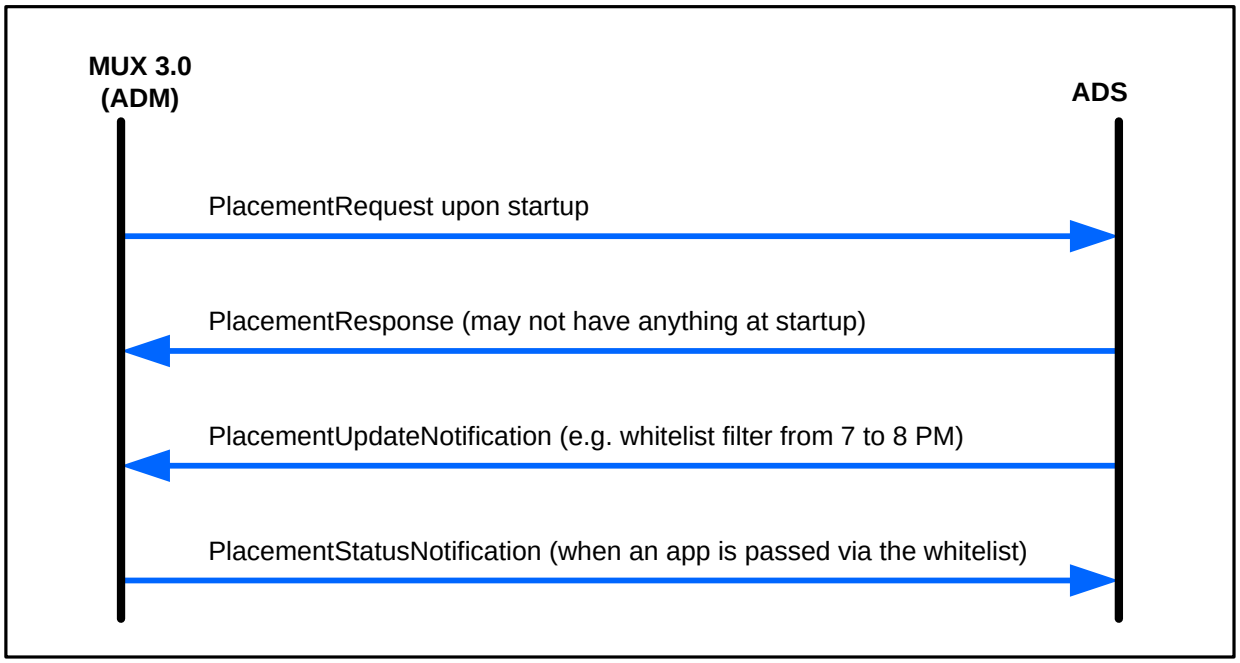


Figure 8 - Example message flow for SCTE 130-3 messaging

6.4.1 Authorization via the SCTE 130-3 Interface

Whitelist and blacklist filters may be added or modified using the SCTE 130-3 interface. The AppList element is added as an extension to the Placement element within the SCTE 130-3 schema. The AppList element provides the interface to define one or more filters with the AppListEntry element, using the application identifiers (e.g., organizationID, applicationID, applicationInstanceID) to specify the applications.

Identical to a filter created with the configuration interface, the filters use the typeOfEntry attribute to identify if it is a blacklist or a whitelist, and priorityOfEntry to address scenarios where applications match more than one filter.

The SCTE 130-3 interface also provides the ability to configure flight windows using the PlacementConstraints element within SCTE 130-3.

When filters are created via the SSCTE 130-3 interface, the MUX 3.0 also sends PlacementStatusNotifications (PSNs), if PSN Callout endpoints are configured and enabled or established via the SCTE 130-3 registration. The MUX 3.0 MUST send PSNs under these conditions when applications are passed via a whitelist action. Note that the MUX 3.0 does not send PSNs for statically configured filters or blacklist actions. Appendix XII contains a sample of a placement status notification.

Appendix XIII contains a sample of a placement request. Appendix XIV contains a sample of a placement response that contains a whitelist definition, and Appendix XV contains a sample of a placement response that contains a blacklist definition. Appendix XVII contains a sample of a placement response that contains a rule defined by using a SourceManagementConfig element.

6.4.2 Augmentation via the SCTE 130-3 Interface

To perform augmentation via the SCTE 130-3 interface, the decision system sends an augmentation in the SCTE 130-3 PlacementResponse with the AppList element, which can contain either a reference to a standard CoDF formatted file or inline standard CoDF formatted data within the message. Appendix XVI provides a sample of a placement response that contains an augmentation.

To perform updates to an application, the SCTE 130-3 interface MUST support:

- configuration of replacement of an existing resource in an enhancement.
- configuration of adding a resource to an existing enhancement.
- configuration of replacement of an entire enhancement.

6.4.3 Source Management via the SCTE 130-3 Interface

Rules can also be updated dynamically via the SCTE 130-3 interface by inserting a SourceManagementConfig element in the SCTE 130 Placement element. Similar to AppList filters, rules specified via the SCTE 130-3 interface take precedence over statically configured rules.

6.5 Filters and Rules: Inheritance and Processing Sequence

Application filters and rules may exist on Flow Points as well as at the overall CCAP OSSI video element level. In addition, there may be filters and rules that are configured statically, and other filters and rules that are currently in effect on the MUX due to a dynamic update via the SCTE 130-3 interface. This section illustrates how these actions interact when specified at multiple levels.

At each level, the MUX 3.0 MUST apply filters created via the SCTE 130-3 interface prior to applying static filters created via the configuration interface.

At each level, the MUX 3.0 MUST apply Source Management Rules created via the SCTE 130-3 interface prior to applying Source Management Rules created via the configuration interface.

The MUX 3.0 MUST apply Authorization filters starting at the current Flow Point, and if the application does not match any filter criteria, apply filters at the ccap/video level until either a filter is applied or a default is applied.

If Source Management Rules are supported at the ccap/video level, the MUX 3.0 MUST apply Source Management rules starting at the current Flow Point and then apply rules at the video level.

The following pseudo-code describes the processing sequence for filters and rules defined on the MUX.

```
for each Flow Point:
{
    do until the application matches a filter:
    {
        Match app to Flow Point SCTE 130-3 Filters in priority order
        Match app to Flow Point statically configured filters in priority order
        Match app to video element SCTE 130-3 filters in priority order
        Match app to video element statically configured filters in priority order
        if a Flow Point-level default is specified
   {
     Apply the Flow Point-level default
        }
        else
        {
            Apply the defaultFilterType
        }
    }
    Apply SCTE 130-3 Rules defined on the Flow Point
    Apply statically configured Rules defined on the Flow Point
    Apply SCTE 130-3 Rules defined at the video level
    Apply statically configured Rules defined at the video level
}
```

The processing sequence is further illustrated with the following example.

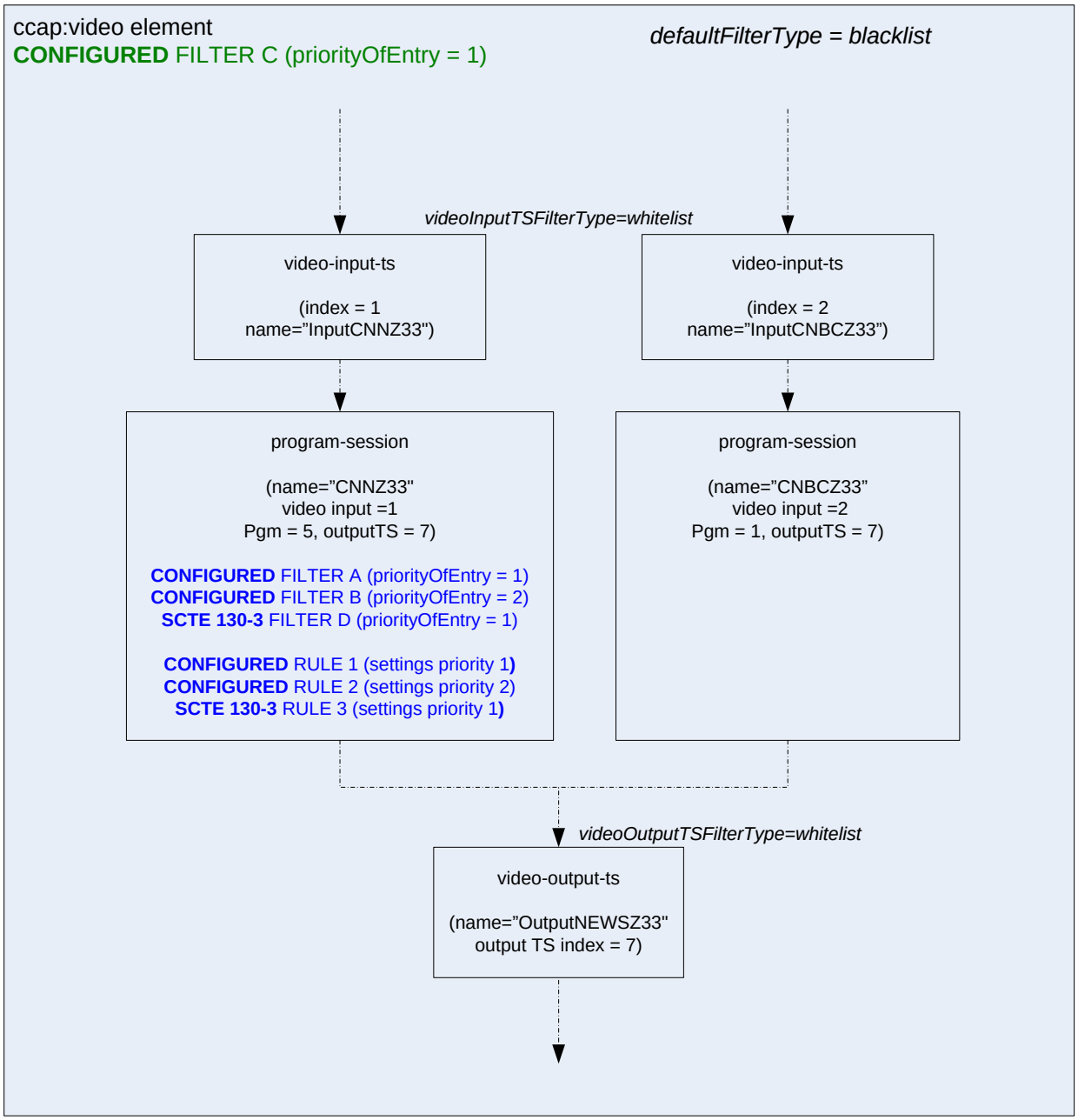


Figure 9 - Example of Filter Inheritance

Figure 9 illustrates a total of five Flow Points: two video-input Flow Points, flowing to two program-session Flow Points, and multiplexed to a video-output Flow Point. In this example, two filters (CONFIGURED FILTER A and CONFIGURED FILTER B) are statically defined on the program-session "CNNZ33" Flow Point. An overall filter is defined at a Flow Point on the video element level (CONFIGURED FILTER C). Another filter, SCTE 130-3 FILTER D, is defined on the program-session Flow Point through the SCTE 130-3 interface.

Also, there is a videoInputTSFilterType defined of whitelist for all video-input Flow Points, a videoOutputTSFilterType of whitelist for all video-output Flow Points, and an overall defaultFilterType of blacklist for the video element.

When an enhanced transport stream enters the MUX at the video-input-ts "InputCNNZ33", no rules or filters are defined. So, the application is evaluated at the next outer element, the video element. In this case, the application will be evaluated for a match with FILTER C. If there is no match, then the current Flow Point default will be evaluated. In this case, a videoInputTSFilterType has been defined as whitelist, so the application will be passed out of the Flow Point. For the purposes of this example, assume the application was not dropped by a rule, so it is passed out of the Flow Point.

The enhanced transport stream is next evaluated at the program-session "CNNZ33" Flow Point. First, the incoming enhancement is matched against FILTER D, because filters created via the SCTE 130-3 interface are evaluated before statically configured filters at each level. Next, the configured filters are evaluated in priority order, FILTER A first, then FILTER B. Once the filters on the program-session Flow Point are evaluated, the SCTE 130-3 defined filters on the next outer element (the video element) would be evaluated. Since there are no SCTE 130-3-defined filters on the video element, the configured Filter C is evaluated next. If there still isn't a match, the default filter types are applied. In this example, there is no session-level default filter, so the defaultFilterType at the video level will take effect. Since the defaultFilterType is defined as blacklist, the application would be dropped. For the purposes of this example, assume the application was passed out of the program-session Flow Point by matching a white list filter.

After applying the filters, the rules are applied on the program-session "CNNZ33" Flow Point. Rule 3 is applied first, because it was created via the SCTE 130-3 interface. RULE 1 will be applied next, because it has a higher priority (the lower the number the higher the priority). RULE 2 will be applied next. The Rules are used to determine when enhancements are passed, dropped or merged during splice and non-splice times.

If the application is passed out of the "CNNZ33" program-session Flow Point, it will enter the video-output-ts Flow Point. There are no rules or filters defined on this Flow Point, so the FILTER C at the video level will be applied. If there is not a match, the default for the Flow Point, the videoOutputTSFilterType of whitelist will be applied.

6.6 Extensibility

The MUX 3.0 schema makes use of extensions defined in the CCAP OSSI schema, and also provides similar extension opportunities within the MUX 3.0 schema.

- ext elements within the CCAP OSSI are used to define the additional functionality for a MUX 3.0 device.
- ext elements are defined in the MUX 3.0 configuration schema as well to allow for extensibility of this schema.
- private:.+ is included in some elements or attributes when restricting to certain string patterns. Extensions are allowed by prefacing it with "private:". For example, if a string is specified with the pattern restriction "red|blue|green|private:.+", a valid private extension is "private:purple".

6.7 Stream Compliance

Specifications for enhancements place constraints on composition of elementary streams that need to be observed when processing enhancements. For example, PES packets for a given table section are delivered in order on the same PID. Any processing of an enhancement by an implementation has to result in a compliant output stream.

The MUX 3.0 device MUST generate MPEG-2 compliant transport streams at the output.

The MUX 3.0 device MUST create output transport streams that comply with the OpenCable ETV AM specifications [AM 1.0.1] and [AM I05].

The MUX 3.0 device MUST remove the EISS table from the stream if incoming EISS data is non-compliant (see [AM 1.0.1]).

6.8 Reporting

The MUX 3.0 device MUST report data records based on the Service Definitions defined in this specification, encoded in IPDR/XDR Encoding Format per the [IPDR/XDR] specification.

The MUX 3.0 device MUST support the IPDR Streaming Protocol as specified in [IPDR/SP] as the transport mechanism for all MUX 3.0 Service Definitions.

Note: An informative description about IPDR can be found in section 6.2 of [DOCSIS 3.0]. Refer to [IPDR/SP] for the IPDR/SP Security recommendations. The IPDR/SP Security Model is out of the scope of this specification.

The MUX 3.0 device MUST allow limiting the amount of information that is reported through configuration changes. For further control, each Flow Point within the MUX 3.0 device can be set to either send or not send reporting information, as the MUX 3.0 device performs essential processing of video and enhanced data services. (Refer to Section 6.2.2 for more information on reporting configuration.)

6.8.1 MUX 3.0 Unparsed Data Type Service Definition

MUX3.0-UNPARSED-TYPE is an IPDR Service Definition schema defining a report from a MUX 3.0 where the EISS, DII, DDB, PMT, or SCTE 35 information is included in an Unparsed Data packet.

The MUX 3.0 MUST support Unparsed Data Type records as defined by the MUX3.0-UNPARSED-TYPE Service Definition, in [MUX3.0 UNPARSED].

The MUX 3.0 MUST support streaming of the MUX3.0-UNPARSED-TYPE record collections as an event session.

The MUX3.0-UNPARSED-TYPE Service Definition includes the following elements from MUX3.0-GEN:

- ReportItemId
- TimeStamp
- HostName
- InformativeText
- ReportSourceName
- ProgramNumber
- Pid
- SDTServiceName
- DataId
- Action
- ReasonCode
- ActionId
- UnparsedDataType
- UnparsedData

6.8.2 MUX 3.0 Bandwidth Type Service Definition

MUX3.0-BANDWIDTH-TYPE is an IPDR Service Definition schema defining a report from a MUX 3.0 to periodically transmit bandwidth statistics. This service definition should be configured for a time-interval-based streaming session.

The MUX 3.0 MUST support Bandwidth Type records as defined by the MUX3.0-BANDWIDTH-TYPE Service Definition, in [MUX3.0 BANDWIDTH].

The MUX 3.0 MUST support streaming of the MUX3.0-BANDWIDTH-TYPE record collections as a time-interval session.

If bandwidth is reported (AvgBandwidth and/or MaxBandwidth) and the PID is passed to the output of a Flow Point, the report MUST be based on the data rate of the PID at the output of the associated Flow Point.

If bandwidth is reported (AvgBandwidth and/or MaxBandwidth), the report MAY be based on the data rate of the PID at the input of the associated Flow Point when the PID is not passed onto the output of the Flow Point.

The MUX3.0-BANDWIDTH-TYPE Service Definition includes the following elements from MUX3.0-GEN:

- ReportItemId
- TimeStamp
- HostName
- ReportSourceName
- ProgramNumber
- Pid
- AvgBandwidth
- MaxBandwidth

6.8.3 MUX3.0-GEN Auxiliary Schema

The MUX3.0-GEN [MUX3.0-GEN] auxiliary schema defines general elements that are used in the MUX 3.0 IPDR reports.

Table 2 - MUX3.0-GEN Elements

Category	Attribute Name	Type	Presence	Permitted Values
What	ReportItemId	ipdr:UUID	Required	
When	TimeStamp	ipdr:dateTimeMsec	Required	
Who	HostName	string	Required	FQDN or IP Address
What	InformativeText	string	Required	May be zero length
What	AvgBandwidth	unsignedLong	Required	May be 0
What	MaxBandwidth	unsignedLong	Required	May be 0
Who	ReportSourceName	string	Required	
Who	ProgramNumber	unsignedShort	Required	May be 0
Who	Pid	unsignedShort	Required	May be 0
Who	SDTServiceName	string	Required	May be zero length
What	DataId	ipdr:UUID	Required	
What	Action	string	Required	Passed|Dropped|Updated|private:.+
What	ReasonCode	integer	Required	applied|mismatch|default|private:.+
What	ActionId	ipdr:UUID	Required	May be zero length.
What	UnparsedDataType	string	Required	0x00|0x02|0xe0|0xe3|0xe4|0x42|private:.+
What	UnparsedData	base64Binary	Required	May be zero length

6.8.3.1 ReportItemId

A unique id for this report.

6.8.3.2 TimeStamp

Timestamp (UTC date/time) of when the report item was created or captured.

6.8.3.3 HostName

The fully qualified domain name or the IP address of the host MUX 3.0 device.

6.8.3.4 InformativeText

Optional informative text supplied by an implementation.

6.8.3.5 AvgBandwidth

Average bandwidth for the specified flow, characterized by Flow Point, Program Number and PID, calculated by the total bits collected during the total period (bwNumSamples multiplied by bwSampleInterval) divided by this total period. The bwSampleInterval and bwNumSamples are configured globally but can also be overridden for each Flow Point. The default value for bwSampleInterval is 5 seconds. The default value for bwNumSamples is 6. Please refer to Section 6.2.2 for more information on how to configure the bwSampleInterval and bwNumSamples. Reported in bits per second.

6.8.3.6 MaxBandwidth

Maximum bandwidth detected for the specified flow, characterized by Flow Point, Program Number and PID. The MaxBandwidth is the maximum data rate calculated among the bwNumSamples intervals. Refer to Section 6.2.2 for more information on how to configure the bwSampleInterval and bwNumSamples. Reported in bits per second.

Example:

When bwSampleInterval is 5 (seconds) and bwNumSamples is 6 (sample periods), then the bandwidth is measured for six 5-second sample intervals. MaxBandwidth is the maximum of these 6 values. AvgBandwidth is the total number of bits collected during the 6 sample intervals divided by the total duration of the 6 sample intervals (30 seconds in this example).

Please note:

- MaxBandwidth and AvgBandwidth are not "rolling" values. In the example above, new values for each of these are calculated and reported only once for each 30-second period, not for every 5-second period as each 5-second period rolls off and a new 5-second period rolls on.
- When configuring this streaming session, it is recommended the collection-interval should be a multiple (1 or more) of the total period (bwSampleInterval multiplied by bwNumSamples) (30 seconds in the example).
- If collection-interval is greater than the total period (bwSampleInterval multiplied by bwNumSamples) then multiple IPDR records MAY be sent during one collection-interval.

6.8.3.7 ReportSourceName

The name of the Flow Point associated with the report.

6.8.3.8 ProgramNumber

In the case of a MPTS input or output, the ProgramNumber further qualifies the identity of the PID.

6.8.3.9 Pid

The PID value.

6.8.3.10 SDTServiceName

If present in the transport stream the name of the service from the Service Descriptor Table (SDT).

6.8.3.11 DataId

The UUID associated with a previous record, where the "data" is identical. For example, if two events occur based on the same EISS stream, in the record reporting on the second event, it can reference the ReportItemId of the previous record and then not populate the Unparsed Data of the record. In this case the Unparsed Data in the second record would be zero length.

6.8.3.12 Action

Action taken as a result of a filter or rule application (Passed, Dropped, Updated, or private).

6.8.3.13 ReasonCode

A string type which defines the reason for the event:

"applied": a rule or process was applied

"mismatch": PID match failed (refer to SCTE 30 Section 7.5 more for information)

"default: a default filter was applied

"private:" to be used for additional private reasons

6.8.3.14 ActionId

Tracking value from the rule or process that resulted in passing or modifying the signal (e.g., if passed, the white list ID from OC_MUX or tracking value from the SCTE 130-3 message).

6.8.3.15 UnparsedDataType

Applicable stream_type value as defined in [AM I05] or later, or the applicable table_id value as assigned in the every MPEG-2 private table (for example, PAT (0x00), PMT (0x02), ETV_EISS (0xe0), DII (0xe3), DDB (0xe4), SDT (0x42), SCTE 35 (0xfc)).

6.8.3.16 UnparsedData

Raw data block(s) in base64 binary and as a fully assembled and complete MPEG section.

If UnparsedDataType is EISS, the UnparsedData field MUST contain a complete EISS section, fully-assembled (i.e., no longer in 188-byte TS format, but a complete and valid MPEG section). The EISS section is defined in [AM 1.0.1].

If UnparsedDataType is DII, the UnparsedData field MUST contain a complete DSMCC section containing the DII Data Download Message, as defined in Table 9-2 of [IEC 13818-6].

If UnparsedDataType is DDB, the UnparsedData field MUST contain the following header information from the DownloadDataBlock (as defined in [IEC 13818-6]) (note this does not mandate including the data bytes):

- dsmccDownloadDataHeader()
- moduleId
- moduleVersion
- blockNumber

Appendix I CCAP

The following is an example export of an input, multiple program sessions, and an output on a MUX 3.0 device using [CCAP OSSI].

Defined are:

1 Video Input

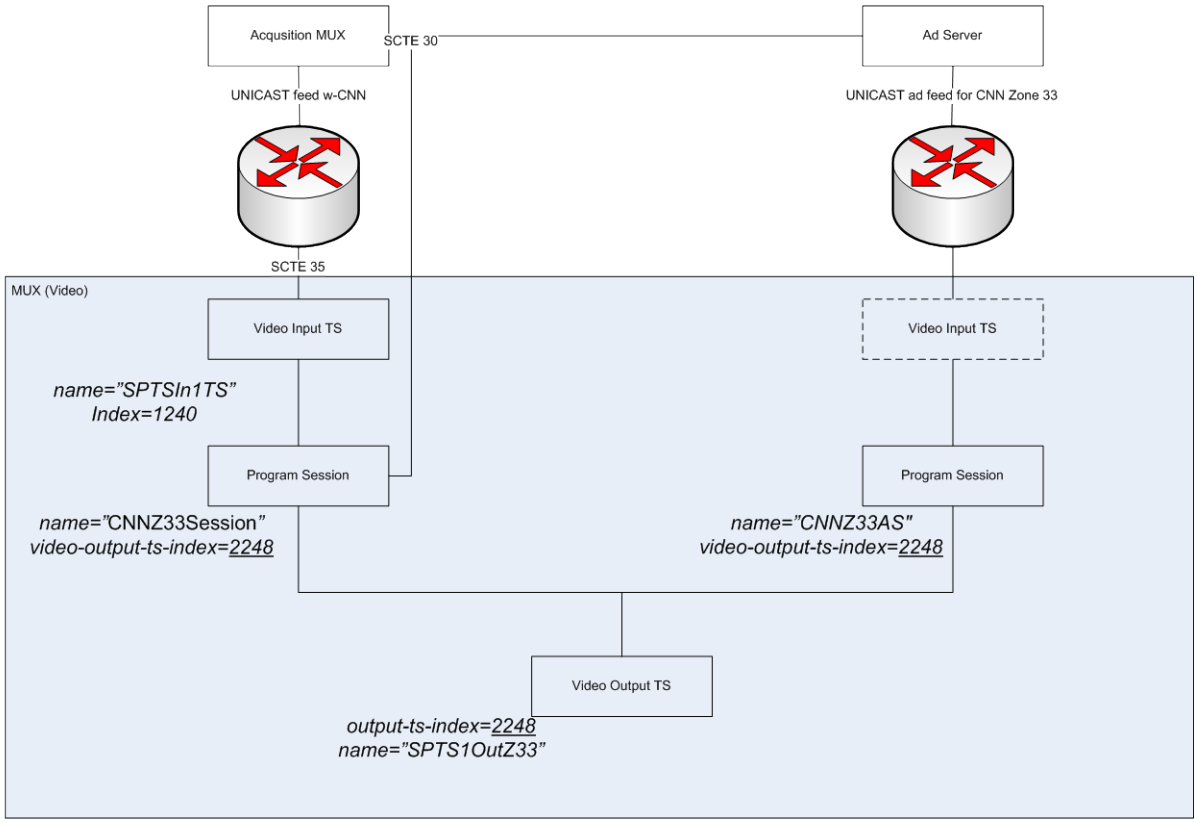
- a Unicast input (SPTSIn1TS) (e.g., a network CNN feed)

2 Program Sessions

- Program Session for network (CNNZ33Session)
- Program Session for both inputs

1 Video Output - Video Output (SPTS1OutZ33)

High level diagram



XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="merge" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <video>
    <!--CNN Source -->
    <!--Define the Input TS that carries the CNN source - the source can be multi-cast or uni-cast.  -->
```

```
    <video-input-ts>
      <input-ts-index>1240</input-ts-index>
      <input-ts-name>SPTSIn1TS</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.12.1.1</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>423</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>
    <!-- Define the CNN Program for Zone 33 -->
    <program-session>
      <session-index>5</session-index>
      <session-name>CNNZ33Session</session-name>
      <session-input-ts>1240</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
    </program-session>
    <!-- Define the CNN pre-bind Ad Server program session for zone 33. The session-name MUST be used when
configuring the Ad Server.  Note:  The session-input-ts index might not be known by the MUX; however, it is
required by CCAP.  A value of 0 has been defined.-->
    <program-session>
      <session-index>6</session-index>
      <session-name>CNNZ33AS</session-name>
      <session-input-ts>0</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
    </program-session>
    <!-- Define the Output MPTS for Zone 33 that contains the CNN feed  -->
    <video-output-ts>
      <output-ts-index>2248</output-ts-index>
      <output-ts-name>SPTS1OutZ33</output-ts-name>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
    </video-output-ts>
  </video>
</ccap:ccap>
```

Appendix II Default Filter Configuration

The following is an example which configures the default filtering rules for ALL Flow Points to whitelist.

XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="replace" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <management>
    <ext>
      <!-- Define vendorControlValue and using Configuration element, specify filtering mode as whitelist for video,
session, and output using the MUX namespace-->
      <mux:Configuration revisionDateTime="2012-11-13T09:30:47.0Z" vendorControlValue="30dd879c-ee2f-
11db-8314-0800200c9a69" revision="2" xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0 OC-
MUX3.0-CONFIG-1.0.0.xsd" xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0">
        <mux:FilterConfiguration
          videoInputTsFilterType="whitelist"
          sessionFilterType="whitelist"
          videoOutputTsFilterType="whitelist"
          defaultFilterType="whitelist"/>
        </mux:Configuration>
    </ext>
  </management>
</ccap:ccap>
```

Appendix III IPDR Configuration

This example demonstrates a configuration of an IPDR collector and two MUX 3.0 service definitions.

XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This example configures an IPDR collector and exporter functions.  This sets up the bandwidth service definition records to be sent as a time-interval session, and
    UnparsedData records to be sent as event based records. Some of the parameters are not specified if the defaults were acceptable. -->
<ccap:ccap operation="replace" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <management>
    <ext>
      <!-- Define the IPDR collector for the video-related records from a video MUX 3.0 device-->
      <mux:Configuration revisionDateTime="2012-11-12T09:30:47.0Z" vendorControlValue="30dd879c-ee2f-11db-8314-0800200c9a80" revision="1" xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0 OC-MUX3.0-CONFIG-1.0.0.xsd" xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0">
      <mux:VideoIPDRConfig defaultReportingState="true">
        <mux:ExporterConfig>
          <mux:Enabled>true</mux:Enabled>
        </mux:ExporterConfig>
        <mux:StreamingSession>
          <mux:SessionId>1</mux:SessionId>
          <mux:CollectionInterval>30</mux:CollectionInterval>
          <mux:StreamingType>time-interval</mux:StreamingType>
          <mux:Enabled>true</mux:Enabled>
          <mux:ServiceDefinitionType>http://www.cablelabs.com/namespaces/opencable/xsd/ipdr/MUX3.0-BANDWIDTH-TYPE-1.0.xsd</mux:ServiceDefinitionType>
        </mux:StreamingSession>
        <mux:StreamingSession>
          <mux:SessionId>2</mux:SessionId>
          <mux:CollectionInterval>30</mux:CollectionInterval>
          <mux:StreamingType>event</mux:StreamingType>
          <mux:Enabled>true</mux:Enabled>
          <mux:ServiceDefinitionType>http://www.cablelabs.com/namespaces/opencable/xsd/ipdr/MUX3.0-UNPARSED-TYPE-1.0.xsd</mux:ServiceDefinitionType>
        </mux:StreamingSession>
        <mux:Collector>
          <mux:CollectorId>1</mux:CollectorId>
          <mux:CollectorIp>10.76.0.94</mux:CollectorIp>
          <mux:Priority>1</mux:Priority>
        </mux:Collector>
      </mux:VideoIPDRConfig>
      </mux:Configuration>
    </ext>
  </management>
</ccap:ccap>
```

Appendix IV ADS Registration Configuration

The following XML example illustrates how to configure the MUX 3.0 to send PlacementStatusNotification messages to a specific ADS.

XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="replace" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap"
xmlns:core="http://www.scte.org/schemas/130-2/2008a/core">
  <name>MUX5</name>
  <management>
    <ext>
      <!-- Using Configuration element, allow a call out for Placement Status Notification on all flow points, using
mux namespace by setting the adsCallOutState to true and defining the message and Address for the CallOut.-->
      <mux:Configuration revisionDateTime="2001-12-17T09:30:47.0Z" vendorControlValue="30dd879c-ee2f-
11db-8314-0800200c9a69" revision="2" xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0 OC-MUX3.0-
CONFIG-1.0.0.xsd" xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0">
        <mux:ADSRegistrationData adsCallOutState="true">
          <mux:ADSCallOut message="PlacementStatusNotification">
            <core:Address type="HTTP">http://localhost:8080/adm</core:Address>
          </mux:ADSCallOut>
        </mux:ADSRegistrationData>
      </mux:Configuration>
    </ext>
  </management>
</ccap:ccap>
```

Appendix V CCAP – Multicast IP video output

The following is an example export of a simple configuration with a multicast output.

Defined are:

1 Video Input

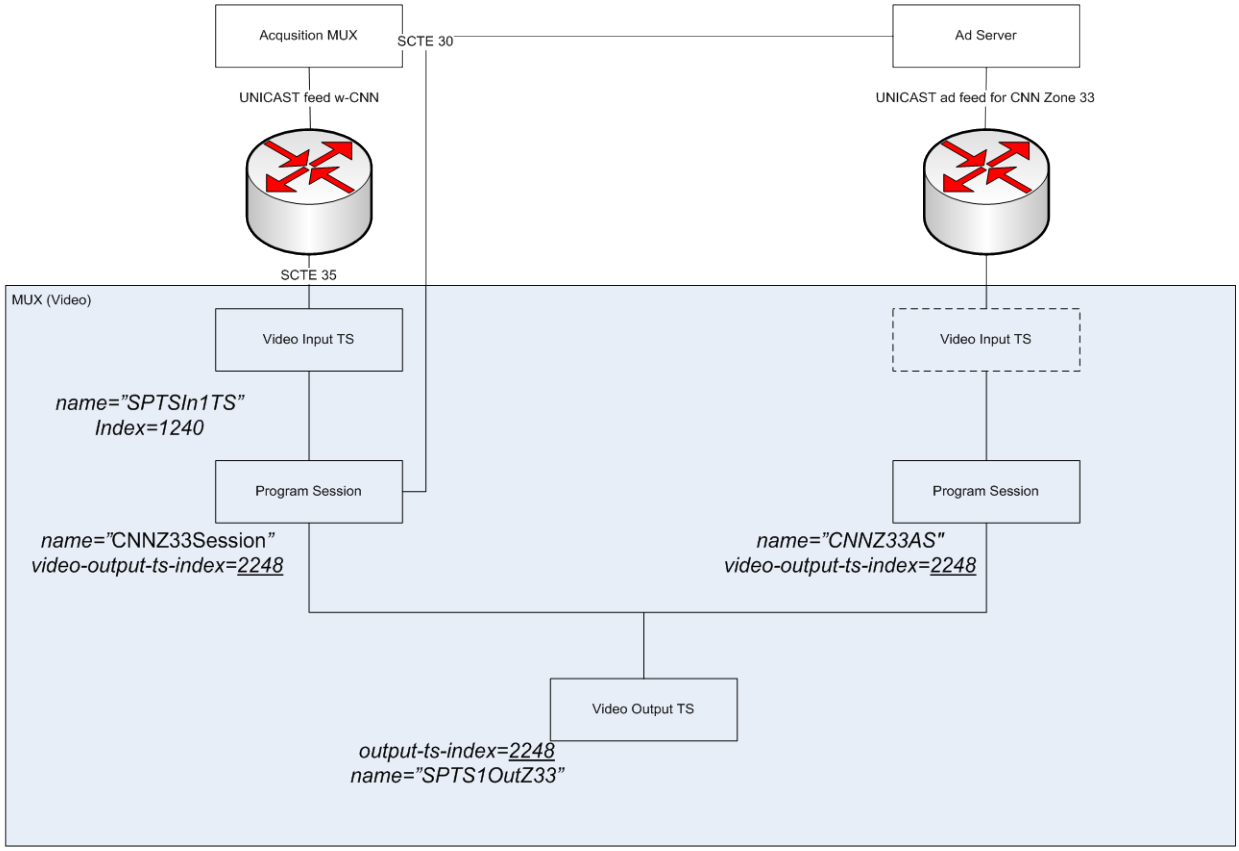
- a Unicast input (SPTSIn1TS) (e.g., a network CNN feed)

2 Program Sessions

- Program Session for network (CNNZ33Session)
- Program Session for both inputs

1 Multicast Video Output – Multicast Video Output (SPTS1OutZ33)

High level diagram



XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="merge" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <video>
    <!--CNN Source -->
    <!--Define the Input TS that carries the CNN source - the source can be multi-cast or uni-cast.  -->
```

```
    <video-input-ts>
      <input-ts-index>1240</input-ts-index>
      <input-ts-name>SPTSIn1TS</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.12.1.1</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>5423</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>
    <!-- Define the CNN Program for Zone 33 -->
    <program-session>
      <session-index>5</session-index>
      <session-name>CNNZ33Session</session-name>
      <session-input-ts>1240</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
    </program-session>
  <!-- Define the CNN pre-bind Ad Server program session for zone 33. The session-name MUST be used when
configuring the Ad Server.  Note:  The session-input-ts index might not be known by the MUX; however, it is
required by CCAP.  A value of 0 has been defined.-->
      <program-session>
      <session-index>6</session-index>
      <session-name>CNNZ33AS</session-name>
      <session-input-ts>0</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
    </program-session>
    <!-- Define the Output MPTS for Zone 33 that contains the CNN feed  -->
    <video-output-ts>
      <output-ts-index>2248</output-ts-index>
      <output-ts-name>SPTS1OutZ33</output-ts-name>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
      <ext>
        <mux:FlowConfig revisionDateTime="2012-09-26T09:30:47.0Z" revision="1"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd"
xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <mux:VideoOutputTs>
            <mux:MulticastVideoTs>
              <mux:MulticastTsSourceIpAddress>192.168.45.3</mux:MulticastTsSourceIpAddress>
              <mux:MulticastTsDestinationIpAddress>225.0.0.33</mux:MulticastTsDestinationIpAddress>
              <mux:MulticastTsDestinationUdpPort>5033</mux:MulticastTsDestinationUdpPort>
              <mux:MulticastTsInterfaceName>eth2</mux:MulticastTsInterfaceName>
            </mux:MulticastVideoTs>
          </mux:VideoOutputTs>
        </mux:FlowConfig>
```

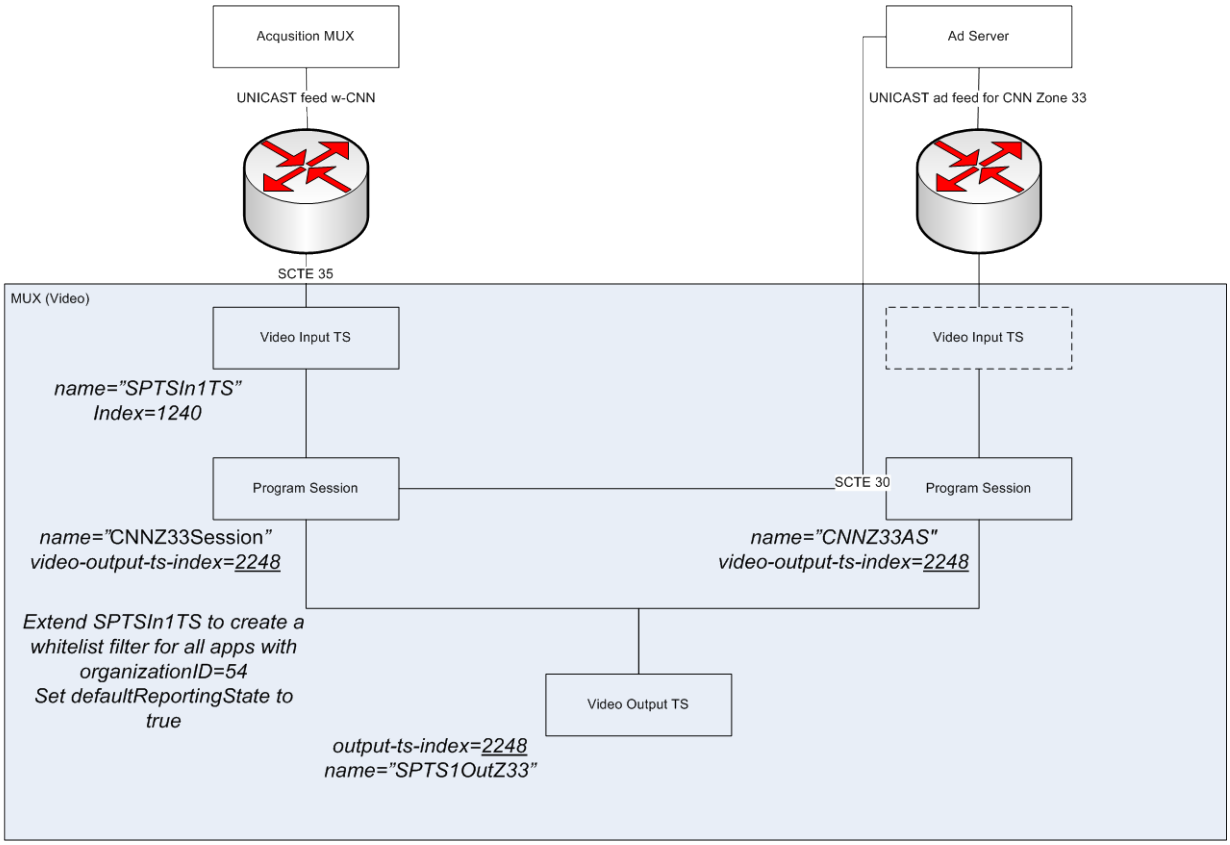
```
      </ext>
    </video-output-ts>
  </video>
</ccap:ccap>
```

Appendix VI Flow Configuration – Whitelist

The following is an example to allow all CableLabs applications carried on the program session (CNNZ33Session) to pass through. To do so, a whitelist filter for applications with an organization ID set to 54 (CableLabs) on the input program session Flow Point (CNNZ33Session)) is created.

Note: This assumes the videoInputTsFilterType and videoOutputTsFilterType are configured to default to "whitelist".

High level diagram



XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="merge" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <video>
    <!--CNN Source -->
    <!--Define the Input TS that carries the CNN source - the source can be multi-cast or uni-cast.  -->
    <video-input-ts>
      <input-ts-index>1240</input-ts-index>
      <input-ts-name>SPTSIn1TS</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.12.1.1</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>423</unicast-ts-destination-udp-port>
```

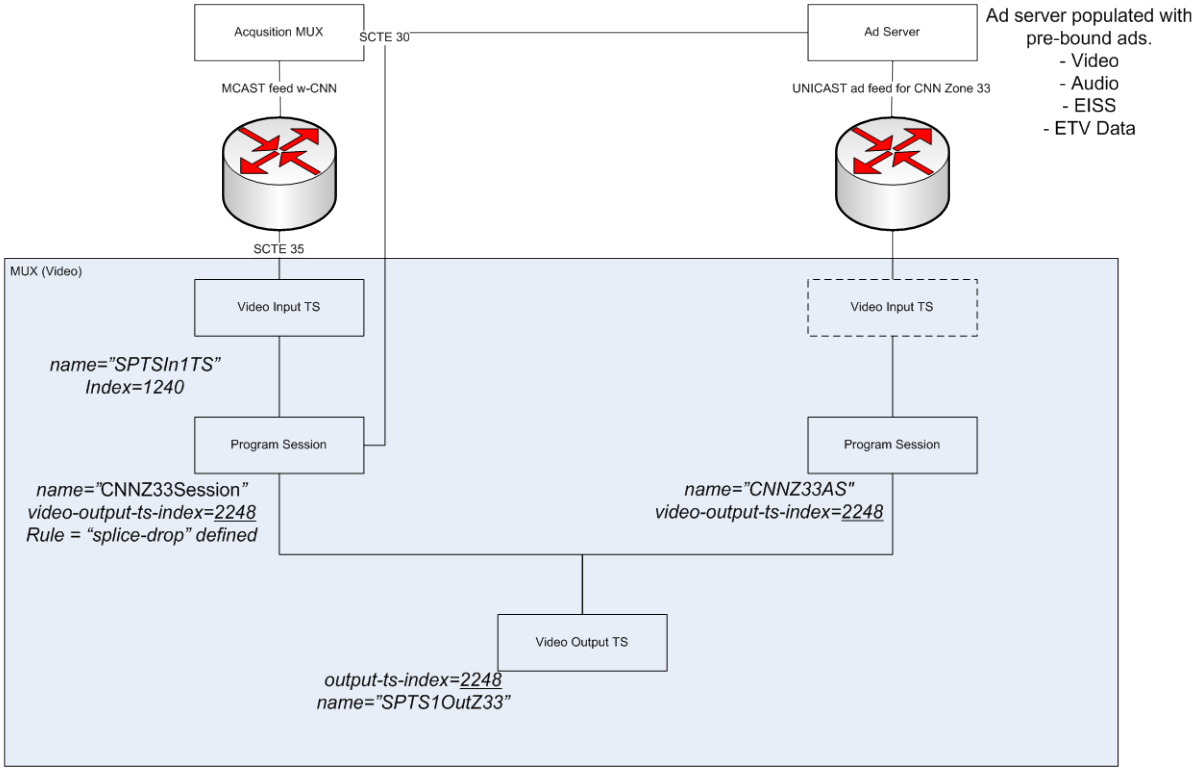
```
      </unicast-video-input-ts>
    </video-input-ts>
    <!-- Define the CNN Program for Zone 33 -->
    <program-session>
      <session-index>5</session-index>
      <session-name>CNNZ33Session</session-name>
      <session-input-ts>1240</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
      <ext>
    <!--Since this extension is placed into the CNNZ33Session Flow Point, all changes will only impact this Flow
Point-->
    <!--Using the FlowConfig element set the defaultReportingState to true so MUX reports-->
    <!--Create a white-list appFilter for the Cablelabs organization (54) by creating an AppList of
configurationEntryType of appFilter and an AppListEntry which has an organizationID of 54 and is set to be a
whitelist, by setting the typeOfEntry as whitelist-->
        <mux:FlowConfig revisionDateTime="2001-12-17T09:30:47.0Z" revision="0" adsCallOutState="false"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd"
xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <mux:IPDRConfig defaultReportingState="true"/>
        <mux:FlowConfigItem>
        <mux:AppList configurationEntryType="appFilter">
            <mux:AppListEntry organizationID="54" typeOfEntry="whitelist" tracking="ec1ebd51-1b9b-4cbd-9f5b-
310ff5c13983">
            </mux:AppListEntry>
        </mux:AppList>
        </mux:FlowConfigItem>
        </mux:FlowConfig>
      </ext>
    </program-session>
    <!-- Define the CNN pre-bind Ad Server program session for zone 33. -->
    <program-session>
      <session-index>6</session-index>
      <session-name>CNNZ33AS</session-name>
      <session-input-ts>0</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
    </program-session>
    <!-- Define the Output MPTS for Zone 33 that contains the CNN feed  -->
    <video-output-ts>
      <output-ts-index>2248</output-ts-index>
      <output-ts-name>SPTS1OutZ33</output-ts-name>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
    </video-output-ts>
```

```
  </video>
</ccap:ccap>
```

Appendix VII Source Management – Pre-bind Splice-drop

The following is an example where the PIDs are dropped for the CNN input (network) feed during an SCTE-35 splice event; during the splice event (e.g., ad break) the pre-bound ad containing audio/video/eiss/etv PIDs are passed. To do so, Source Management Settings and Rules are defined on the program session (CNNZSession).

High level diagram



XML Example

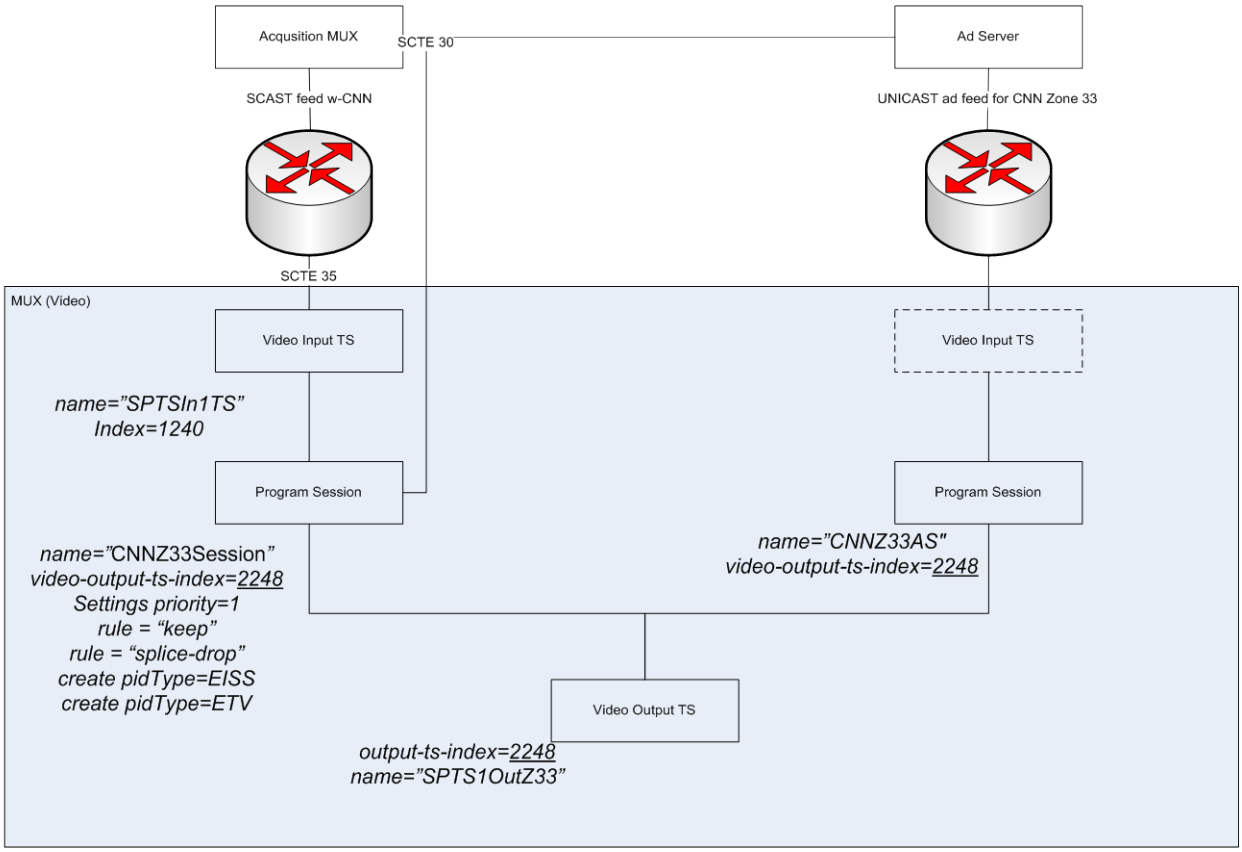
```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="merge" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <video>
    <!-- CNN Source -->
    <!-- Define the Input TS that carries the CNN source - the source can be multi-cast or uni-cast.  -->
    <video-input-ts>
      <input-ts-index>1240</input-ts-index>
      <input-ts-name>SPTSIn1TS</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.12.1.1</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>423</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>
    <!-- Define the CNN Program for Zone 33 -->
    <program-session>
      <session-index>5</session-index>
```

```
      <session-name>CNNZ33Session</session-name>
      <session-input-ts>1240</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
      <ext>
        <!-- Use the MUX 3.0 schema to create a Source Management Rule to perform a splice-drop on the current
Flow Point: Program Session CNNZ33Session-->
        <mux:FlowConfig revisionDateTime="2001-12-17T09:30:47.0Z" revision="0" adsCallOutState="false"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd"
xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <mux:SourceManagementConfig>
            <mux:Settings>
              <!-- On a splice (SCTE 35 out point) stop passing PIDs (not specified, so applies to EBIF and ETV)
from the source feed to the output TS -->
              <mux:Rule tracking="b08935b5-56af-441b-bd24-cc260b1a1811"
nameOfFlowPoint="CNNZ33Session">splice-drop</mux:Rule>
            </mux:Settings>
          </mux:SourceManagementConfig>
        </mux:FlowConfig>
      </ext>
    </program-session>
    <!-- Define the CNN pre-bind Ad Server program session for zone 33. -->
    <program-session>
      <session-index>6</session-index>
      <session-name>CNNZ33AS</session-name>
      <session-input-ts>0</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
    </program-session>
    <!-- Define the Output MPTS for Zone 33 that contains the CNN feed  -->
    <video-output-ts>
      <output-ts-index>2248</output-ts-index>
      <output-ts-name>SPTS1OutZ33</output-ts-name>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
    </video-output-ts>
  </video>
  </ccap:ccap>
```

Appendix VIII Source Management – Pre-bind Rules (Ghost PIDs)

The following is an example where the MUX 3.0 device only passes thru enhancements, if present, from the Network broadcast when not splicing. During a splice, the MUX 3.0 device will only pass thru enhancements, if present, from the Advertising source. Ghost PIDs are created on the program session of the broadcast.

High level diagram



XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="merge" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <video>
    <!--CNN Source -->
    <!--Define the Input TS that carries the CNN source - the source can be multi-cast or uni-cast.  -->
    <video-input-ts>
      <input-ts-index>1240</input-ts-index>
      <input-ts-name>SPTSIn1TS</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.12.1.1</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>5423</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>
    <!-- Define the CNN Program for Zone 33 -->
```

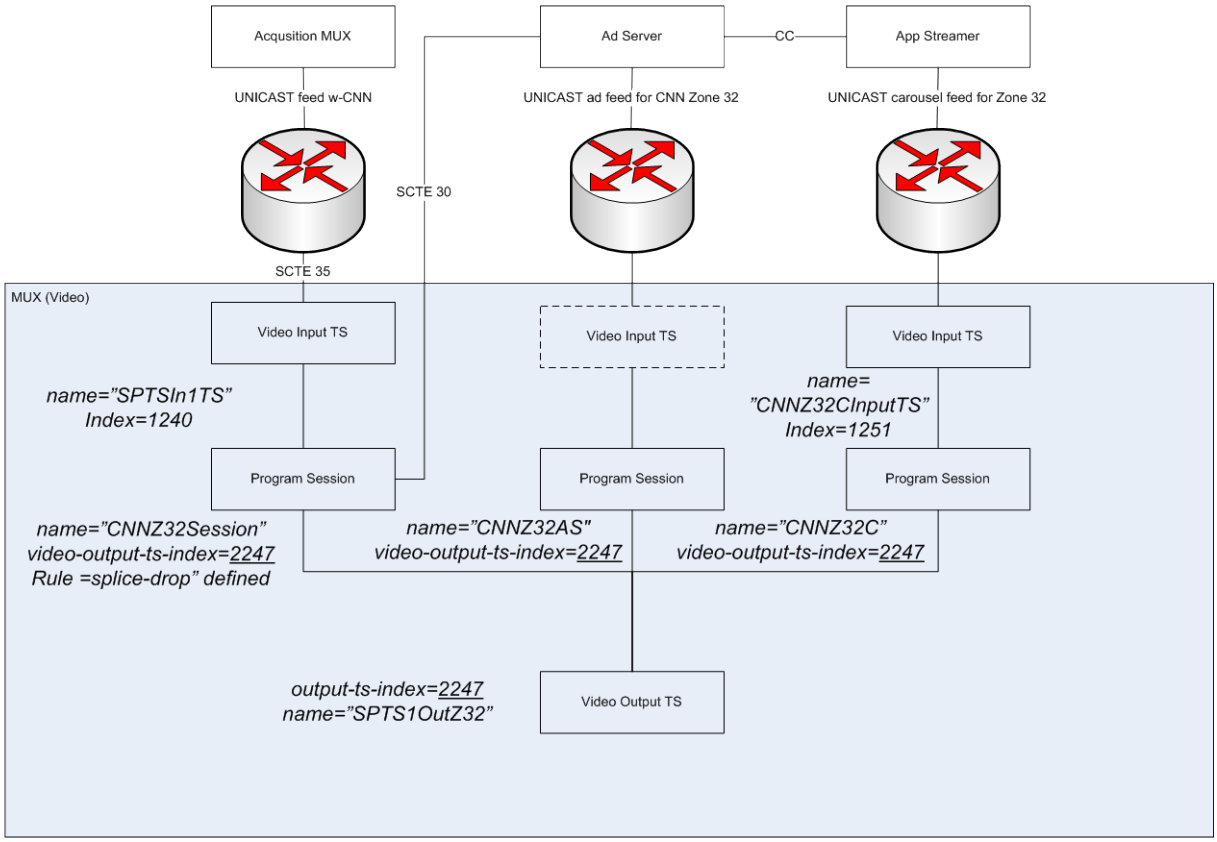
```
    <program-session>
      <session-index>5</session-index>
      <session-name>CNNZ33Session</session-name>
      <session-input-ts>1240</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
      <!-- Defining splice rules on the program session - this is the primary channel that receives SCTE 35 cues.  If
the nameOfFlowPoint is not specified in the rule, it applies to this CNNZ33Session. -->
      <!-- This example actually represents the expected default behavior, in the event no rules are defined. -->
      <ext>
        <mux:FlowConfig revisionDateTime="2012-09-26T09:30:47.0Z" revision="0" adsCallOutState="false"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd"
xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <mux:IPDRConfig defaultReportingState="true"/>
          <mux:SourceManagementConfig>
            <mux:Settings priority="1">
              <!-- During non-splice, keep the etv pids, but drop on splice -->
              <mux:Rule tracking="1156f69b-3282-40ba-b3c5-0e1d3a4b77ea">keep</mux:Rule>
              <mux:Rule tracking="b983fbf2-c6d2-4f9b-8a89-f8f15f9b3735">splice-drop</mux:Rule>
              <!-- create ghost pids on the during non-splice time if the etv pids are not already there (1301 and 1302
are arbitrary) -->
              <mux:Rule tracking="831dbf07-3df9-4901-8e30-57b4b7473c19" pidType="eiss"
pid="1301">create</mux:Rule>
              <mux:Rule tracking="4eed828f-de9c-45ba-a519-da129409a40b" pidType="etv-data"
pid="1302">create</mux:Rule>
              <!-- The next two rules on the AS flow point are not completely necessary but helps to be more
prescriptive -->
              <mux:Rule tracking="e85961d4-409f-4cc5-8c9a-d8ff13639d5e"
nameOfFlowPoint="CNNZ33AS">drop</mux:Rule>
              <mux:Rule tracking="c756506e-166c-4058-a7a8-6572a1d8a2b7"
nameOfFlowPoint="CNNZ33AS">splice-keep</mux:Rule>
              <!-- create ghost pids on the during splice time if the etv pids are not already there (1301 and 1302 are
arbitrary but note they are consistent with ) -->
              <mux:Rule tracking="95146671-f4b5-4b50-8ff5-9d70dca9c470" nameOfFlowPoint="CNNZ33AS"
pidType="eiss" pid="1301">create</mux:Rule>
              <mux:Rule tracking="84e21d77-9ce7-4ceb-8ace-beb3663703e6" nameOfFlowPoint="CNNZ33AS"
pidType="etv-data" pid="1301">create</mux:Rule>
            </mux:Settings>
          </mux:SourceManagementConfig>
        </mux:FlowConfig>
      </ext>
    </program-session>
    <!-- Define the CNN pre-bind Ad Server program session for zone 33. -->
    <program-session>
      <session-index>6</session-index>
      <session-name>CNNZ33AS</session-name>
      <session-input-ts>0</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
```

```
      <requested-bandwidth>0</requested-bandwidth>
    </program-session>
    <!-- Define the Output MPTS for Zone 33 that contains the CNN feed  -->
    <video-output-ts>
      <output-ts-index>2248</output-ts-index>
      <output-ts-name>SPTS1OutZ33</output-ts-name>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
    </video-output-ts>
  </video>
</ccap:ccap>
```

Appendix IX Source Management – Late-bind Splice-drop

The following is an example where PIDs are dropped for the CNN (network) feed during a SCTE-35 splice event; during the splice event (e.g., ad break), the ad containing audio/video PIDs are passed from the Ad Server, and the EISS/ETV PIDs from the carousel are passed. To do so, Source Management Settings and Rules are defined on the program session (CNNZ32Session).

High level diagram



XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="merge" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <video>
    <!-- CNN Source -->
    <!-- The Input TS that carries the CNN source - the source can be multi-cast or uni-cast.  -->
    <video-input-ts>
      <input-ts-index>1240</input-ts-index>
      <input-ts-name>SPTSIn1TS</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.12.1.1</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>423</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>
```

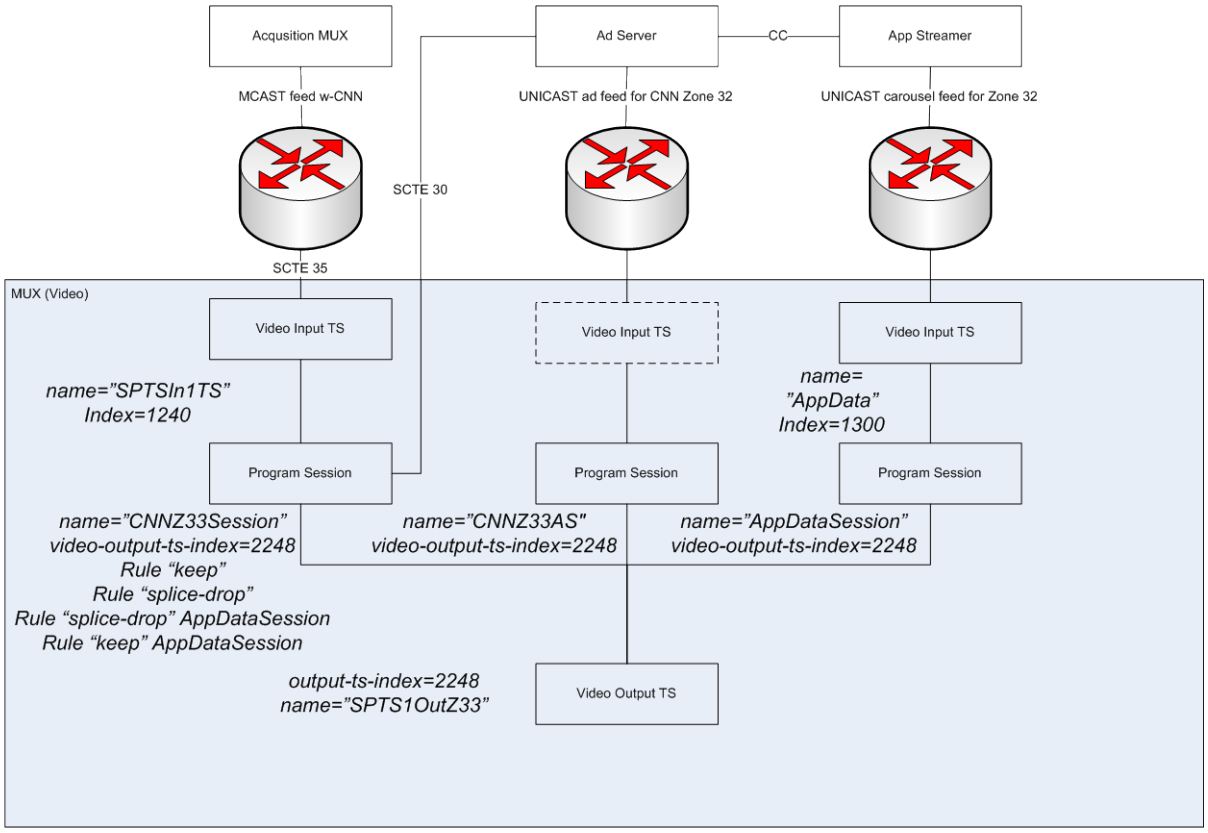
```
<!-- The CNN Carousel Server input for zone 32 - the source can be multi-cast or uni-cast.  -->
<video-input-ts>
  <input-ts-index>1251</input-ts-index>
  <input-ts-name>CNNZ32CInputTS</input-ts-name>
  <unicast-video-input-ts>
    <unicast-ts-destination-ip-address>10.16.1.1</unicast-ts-destination-ip-address>
    <unicast-ts-destination-udp-port>420</unicast-ts-destination-udp-port>
  </unicast-video-input-ts>
</video-input-ts>
<!-- CNN Program for Zone 32 -->
<program-session>
  <session-index>1</session-index>
  <session-name>CNNZ32Session</session-name>
  <session-input-ts>1240</session-input-ts>
  <session-output-ts>
    <session-output-ts-index>2247</session-output-ts-index>
  </session-output-ts>
  <input-mpeg-program-number>1</input-mpeg-program-number>
  <output-mpeg-program-number>1</output-mpeg-program-number>
  <requested-bandwidth>0</requested-bandwidth>
  <ext>
    <!-- Use the MUX 3.0 schema to create a Source Management Rule to perform a splice-drop on the current
Flow Point: Program Session CNNZ33Session-->
    <mux:FlowConfig revisionDateTime="2001-12-17T09:30:47.0Z" revision="0" adsCallOutState="false"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd"
xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <mux:SourceManagementConfig>
        <mux:Settings priority="1">
          <!-- On a splice (SCTE 35 out point) stop passing PIDs from the source feed to the output TS -->
          <mux:Rule tracking="03cdc36a-730d-49dd-894b-853f4d0c552e"
nameOfFlowPoint="CNNZ32Session">splice-drop</mux:Rule>
          <!-- During non-splice time, keep any ETV and EISS PIDs flowing -->
          <mux:Rule tracking="9939305c-7433-45cd-84db-3ff483499666"
nameOfFlowPoint="CNNZ32Session">keep</mux:Rule>
          <!-- During splice, pass PIDs from the late-bind carousel -->
          <mux:Rule tracking="c738fc84-6233-4df4-b941-849eb777c561"
nameOfFlowPoint="CNNZ32C">splice-keep</mux:Rule>
          <!-- During non-splice time, drop the ETV and EISS PIDs from the late-bind carousel -->
          <mux:Rule tracking="d2b43625-eb03-4279-add8-1addb9a0ee2c"
nameOfFlowPoint="CNNZ32C">drop</mux:Rule>
        </mux:Settings>
      </mux:SourceManagementConfig>
    </mux:FlowConfig>
  </ext>
</program-session>
<!-- The CNN Ad Server program session for zone 32. -->
<program-session>
  <session-index>2</session-index>
  <session-name>CNNZ32AS</session-name>
  <session-input-ts>0</session-input-ts>
  <session-output-ts>
    <session-output-ts-index>2247</session-output-ts-index>
  </session-output-ts>
  <input-mpeg-program-number>1</input-mpeg-program-number>
  <output-mpeg-program-number>1</output-mpeg-program-number>
  <requested-bandwidth>0</requested-bandwidth>
```

```
    </program-session>
    <!-- The CNN Carousel server program for zone 32. -->
    <program-session>
      <session-index>3</session-index>
      <session-name>CNNZ32C</session-name>
      <session-input-ts>1251</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2247</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
    </program-session>
    <!-- The output MPTS for Zone 32 that contains the CNN feed  -->
    <video-output-ts>
      <output-ts-index>2247</output-ts-index>
      <output-ts-name>MPTS1OutZ32</output-ts-name>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
    </video-output-ts>
  </video>
  </ccap:ccap>
```

Appendix X Source Management – Late-Bind, merge broadcast

The following is an example where the broadcast session CNN input (network) feed can have an enhancement, (EISS and ETV-DATA). In addition, a local late-bind carousel is signaling EISS that is merged during non-splice time. The inserted stream from the Ad Server does not have an enhancement. Source Management Settings and Rules are defined on the program session (CNNZSession).

High level diagram



XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="merge" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
<!-- In this example, the broadcast session can have an enhancement, (EISS and ETV-DATA). In addition, a local late-bind carousel is signaling EISS that is merged during
     non-splice time, but dropped during splice time. The inserted stream from the AD Server does not have an enhancement-->
  <name>MUX5</name>
  <video>
    <!--CNN Source -->
    <!--Define the Input TS that carries the CNN source - the source can be multi-cast or uni-cast. -->
    <video-input-ts>
      <input-ts-index>1240</input-ts-index>
      <input-ts-name>SPTSIn1TS</input-ts-name>
      <unicast-video-input-ts>
```

```
        <unicast-ts-destination-ip-address>10.12.1.1</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>5423</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>
    <!--Define the EISS late-bind carousel carrying EISS.  -->
    <video-input-ts>
      <input-ts-index>1300</input-ts-index>
      <input-ts-name>AppData</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.14.1.4</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>5425</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>

    <!-- Define the CNN Program for Zone 33 -->
    <program-session>
      <session-index>5</session-index>
      <session-name>CNNZ33Session</session-name>
      <session-input-ts>1240</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
      <!-- Defining splice rules on the program session - this is the primary channel that receives SCTE 35 cues.  If
the nameOfFlowPoint is not specified in the rule, it applies to this CNNZ33Session.Rules are not defined on the
CNNZ33AS session because it is not configured to carry enhancements.  The local carousel EISS will be merged
with
the program session during non-splice time, but dropped during splice time.  The carousel is configured with settings
priority 2, so that if ETV-DATA is present in the carousel it
will be dropped and the ETV-DATA from the primary session will be passed. -->
      <ext>
        <mux:FlowConfig revisionDateTime="2012-09-26T09:30:47.0Z" revision="0" adsCallOutState="false"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd"
xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <mux:IPDRConfig defaultReportingState="true"/>
        <mux:SourceManagementConfig>
            <mux:Settings priority="1">
              <mux:Rule tracking="1156f69b-3282-40ba-b3c5-0e1d3a4b77ea">keep</mux:Rule>
              <mux:Rule tracking="b983fbf2-c6d2-4f9b-8a89-f8f15f9b3735">splice-drop</mux:Rule>
              <mux:Rule tracking="c07ad6c3-52f0-4485-bf68-7d118269f4d3"
nameOfFlowPoint="AppDataSession">splice-drop</mux:Rule>
            </mux:Settings>
            <mux:Settings priority="2">
              <mux:Rule tracking="b4740b13-97db-4a61-99f0-dfebea614de3"
nameOfFlowPoint="AppDataSession">keep</mux:Rule>
            </mux:Settings>
          </mux:SourceManagementConfig>
        </mux:FlowConfig>
      </ext>
    </program-session>
    <!-- Define the CNN pre-bind Ad Server program session for zone 33. -->
    <program-session>
      <session-index>6</session-index>
      <session-name>CNNZ33AS</session-name>
```

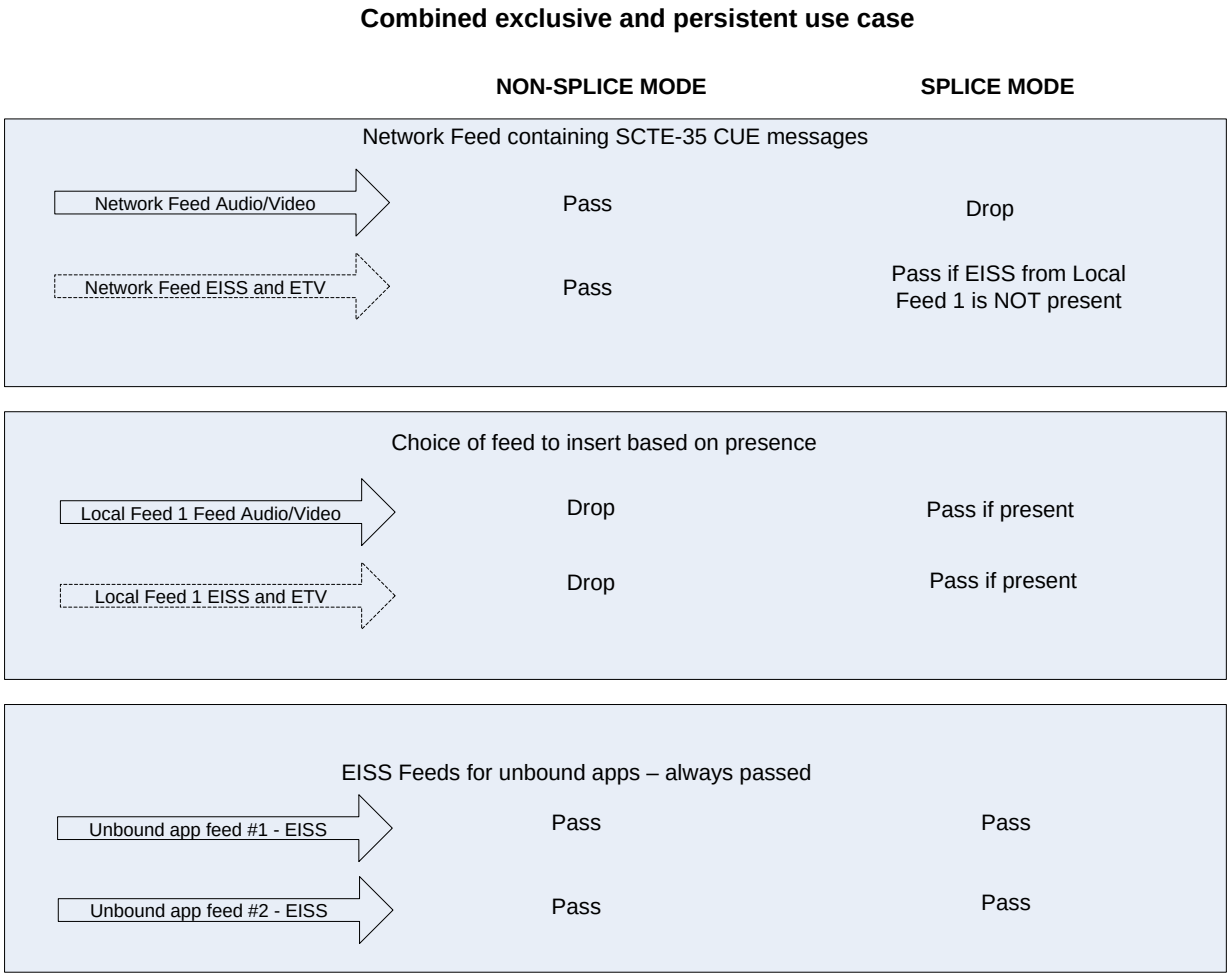
```
        <session-input-ts>0</session-input-ts>
        <session-output-ts>
          <session-output-ts-index>2248</session-output-ts-index>
        </session-output-ts>
        <input-mpeg-program-number>1</input-mpeg-program-number>
        <output-mpeg-program-number>1</output-mpeg-program-number>
        <requested-bandwidth>0</requested-bandwidth>
      </program-session>
      <!-- Define the EISS App  program session. -->
      <program-session>
        <session-index>7</session-index>
        <session-name>AppDataSession</session-name>
        <session-input-ts>1300</session-input-ts>
        <session-output-ts>
          <session-output-ts-index>2248</session-output-ts-index>
        </session-output-ts>
        <input-mpeg-program-number>1</input-mpeg-program-number>
        <output-mpeg-program-number>1</output-mpeg-program-number>
        <requested-bandwidth>0</requested-bandwidth>
      </program-session>

      <!-- Define the Output MPTS for Zone 33 that contains the CNN feed  -->
      <video-output-ts>
        <output-ts-index>2248</output-ts-index>
        <output-ts-name>SPTS1OutZ33</output-ts-name>
        <video-down-channel-ref>
          <slot>1</slot>
          <ds-rf-port>1</ds-rf-port>
          <down-channel>1</down-channel>
        </video-down-channel-ref>
      </video-output-ts>
    </video>
</ccap:ccap>
```

Appendix XI Source Management – Rules Splice Exclusive

The following is an example where multiple unbound applications are fed into a broadcast using multiple carousels. Use the splice-exclusive and persistent rule on the broadcast session so the unbound applications are merged along with possible enhancements during the broadcast or during the splice.

High level diagram



XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ccap:ccap operation="merge" SchemaVersion="0000-00-00"
xsi:schemaLocation="urn:cablelabs:params:xml:ns:yang:ccap ccap_2012_08_09.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccap="urn:cablelabs:params:xml:ns:yang:ccap">
  <name>MUX5</name>
  <video>
    <!--CNN Source -->
    <!--Define the Input TS that carries the CNN source - the source can be multi-cast or uni-cast.  -->
    <video-input-ts>
      <input-ts-index>1240</input-ts-index>
      <input-ts-name>SPTSIn1TS</input-ts-name>
      <unicast-video-input-ts>
```

```
        <unicast-ts-destination-ip-address>10.12.1.1</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>5423</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>
    <!--Define the first EISS late-bind carousel carrying persistent EISS for an unbound application - the source can be multi-cast or uni-cast.  -->
    <video-input-ts>
      <input-ts-index>1300</input-ts-index>
      <input-ts-name>App1Data</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.14.1.4</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>5425</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>
    <!--Define the second EISS late-bind carousel carrying persistent EISS for an unbound application - the source can be multi-cast or uni-cast.  -->
    <video-input-ts>
      <input-ts-index>1301</input-ts-index>
      <input-ts-name>App2Data</input-ts-name>
      <unicast-video-input-ts>
        <unicast-ts-destination-ip-address>10.14.1.5</unicast-ts-destination-ip-address>
        <unicast-ts-destination-udp-port>5425</unicast-ts-destination-udp-port>
      </unicast-video-input-ts>
    </video-input-ts>

    <!-- Define the CNN Program for Zone 33 -->
    <program-session>
      <session-index>5</session-index>
      <session-name>CNNZ33Session</session-name>
      <session-input-ts>1240</session-input-ts>
      <session-output-ts>
        <session-output-ts-index>2248</session-output-ts-index>
      </session-output-ts>
      <input-mpeg-program-number>1</input-mpeg-program-number>
      <output-mpeg-program-number>1</output-mpeg-program-number>
      <requested-bandwidth>0</requested-bandwidth>
      <!-- Defining splice rules on the program session - this is the primary channel that receives SCTE 35 cues.  If the nameOfFlowPoint is not specified in the rule, it applies to this CNNZ33Session. -->
      <!-- This example represents a primary program session that has an enhancement, and an ad server insertion during splice that may or may not have an enhancement.
           The rule here is set up to play the enhancement of the inserted ad if one exists, but if not, pass the enhancement associated with the primary program session.
           In addition, two additional EISS carousels are defined that only send stream events (or triggers) targeted to unbound apps. In this case, they are defined to be persistent,
           as we want them to pass during splice and non-splice time.  These EISS carousels will be merged with the EISS from either the primary program session or the inserted ad
           session on one PID.-->
      <ext>
        <mux:FlowConfig revisionDateTime="2012-09-26T09:30:47.0Z" revision="0" adsCallOutState="false" xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd" xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <mux:IPDRConfig defaultReportingState="true"/>
          <mux:SourceManagementConfig>
            <mux:Settings priority="1">
              <mux:Rule tracking="1156f69b-3282-40ba-b3c5-0e1d3a4b77ea">keep</mux:Rule>
```

```
            <mux:Rule tracking="361c3362-fafa-4146-9e59-d0ac57ffe27b"
nameOfFlowPoint="CNNZ33AS">splice-exclusive</mux:Rule>
            <mux:Rule tracking="d78bd0a0-6361-4b19-9f4a-86dd87716e50"
nameOfFlowPoint="CNNZ33AS">drop</mux:Rule>
            <mux:Rule tracking="b4740b13-97db-4a61-99f0-dfebea614de3"
nameOfFlowPoint="App1DataSession">persistent</mux:Rule>
            <mux:Rule tracking="c07ad6c3-52f0-4485-bf68-7d118269f4d3"
nameOfFlowPoint="App2DataSession">persistent</mux:Rule>
          </mux:Settings>
          <mux:Settings priority="2">
            <mux:Rule tracking="b983fbf2-c6d2-4f9b-8a89-f8f15f9b3735"
nameOfFlowPoint="CNNZ33Session">splice-exclusive</mux:Rule>
          </mux:Settings>
        </mux:SourceManagementConfig>
      </mux:FlowConfig>
    </ext>
  </program-session>
  <!-- Define the CNN pre-bind Ad Server program session for zone 33. -->
  <program-session>
    <session-index>6</session-index>
    <session-name>CNNZ33AS</session-name>
    <session-input-ts>0</session-input-ts>
    <session-output-ts>
      <session-output-ts-index>2248</session-output-ts-index>
    </session-output-ts>
    <input-mpeg-program-number>1</input-mpeg-program-number>
    <output-mpeg-program-number>1</output-mpeg-program-number>
    <requested-bandwidth>0</requested-bandwidth>
  </program-session>
  <!-- Define the EISS App 1 program session. -->
  <program-session>
    <session-index>7</session-index>
    <session-name>App1DataSession</session-name>
    <session-input-ts>1300</session-input-ts>
    <session-output-ts>
      <session-output-ts-index>2248</session-output-ts-index>
    </session-output-ts>
    <input-mpeg-program-number>1</input-mpeg-program-number>
    <output-mpeg-program-number>1</output-mpeg-program-number>
    <requested-bandwidth>0</requested-bandwidth>
  </program-session>
  <!-- Define the EISS App 2 program session. -->
  <program-session>
    <session-index>8</session-index>
    <session-name>App2DataSession</session-name>
    <session-input-ts>1301</session-input-ts>
    <session-output-ts>
      <session-output-ts-index>2248</session-output-ts-index>
    </session-output-ts>
    <input-mpeg-program-number>1</input-mpeg-program-number>
    <output-mpeg-program-number>1</output-mpeg-program-number>
    <requested-bandwidth>0</requested-bandwidth>
  </program-session>
  <!-- Define the Output MPTS for Zone 33 that contains the CNN feed  -->
  <video-output-ts>
    <output-ts-index>2248</output-ts-index>
```

```
      <output-ts-name>SPTS1OutZ33</output-ts-name>
      <video-down-channel-ref>
        <slot>1</slot>
        <ds-rf-port>1</ds-rf-port>
        <down-channel>1</down-channel>
      </video-down-channel-ref>
    </video-output-ts>
  </video>
</ccap:ccap>
```

Appendix XII SCTE 130-3 Interface: Placement Status Notification Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<PlacementStatusNotification resend="a" messageId="a" identity="a" system="a" version="a"
xsi:schemaLocation="http://www.scte.org/schemas/130-3/2008a/adm SCTE_130-3_2010.xsd"
xmlns="http://www.scte.org/schemas/130-3/2008a/adm" xmlns:core="http://www.scte.org/schemas/130-
2/2008a/core" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PlayData eventRangeStartDateTime="2001-12-17T09:30:47.0Z" identityADS="a" identityADM="a"
eventRangeEndDateTime="2001-12-17T09:30:47.0Z" systemADM="a">
    <SystemContext>
      <Network>//mso.com/MUX5/CNNZ33</Network>
    </SystemContext>
    <Events>
      <PlacementStatusEvent type="status" time="2001-12-17T09:30:47.0Z">
        <!-- class - set to "0" if successfully process the action (white/black list or augmentation). "1" may be
reported (for example, can't replace a resource successfully on the carousel -->
        <core:StatusCode class="0">
          <!-- Description of what was passed. -->
          <core:Note>Passed trigger for xxx. OR Blocked resource xxx</core:Note>
        </core:StatusCode>
        <Spot>
          <!-- provide linkage back to the AppListEntry -->
          <core:Tracking>77e41edd-a1ae-41a8-9aff-a4abef6876dd</core:Tracking>
          <core:Content/>
        </Spot>
        <core:Ext>
          <!-- Put the binary reporting entry here using the MUX 3.0 reporting schema - life cycle elements and
triggers that are whitelisted or blacklisted -->
        </core:Ext>
      </PlacementStatusEvent>
      <!-- When the EffectiveEndDateTime is reached -->
      <PlacementStatusEvent type="endEvents" time="2001-12-17T09:30:47.0Z">
        <!-- class - set to "0" if successfully process the action (white/black list or augmentation). "1" may be
reported (for example, can't replace a resource successfully on the carousel -->
        <core:StatusCode class="0">
          <!-- Description of what was passed. -->
          <core:Note>Reached end of time on placement constraint</core:Note>
        </core:StatusCode>
        <Spot>
          <!-- provide linkage back to the AppListEntry -->
          <core:Tracking>77e41edd-a1ae-41a8-9aff-a4abef6876dd</core:Tracking>
          <core:Content/>
        </Spot>
      </PlacementStatusEvent>
    </Events>
  </PlayData>
        <core:Ext/>
</PlacementStatusNotification>
```

Appendix XIII SCTE 130-3 Interface: Placement Request Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<PlacementRequest messageId="a" identity="a" version="a" updatesAllowed="true"
xsi:schemaLocation="http://www.scte.org/schemas/130-3/2008a/adm SCTE_130-3_2010.xsd"
xmlns="http://www.scte.org/schemas/130-3/2008a/adm" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <SystemContext>
    <!-- Adjust to address flow point of interest -->
    <Network>//mso.com/MUX5/CNNZ33</Network>
  </SystemContext>
  <PlacementOpportunity id="2aa07706-4219-4d58-bcdb-d5bcd07eea19
" serviceRegistrationRef="MUX3.0">
<!-- Fetch all interactive actions that apply during the Window -->
    <LinearAvailBinding opportunityType="interactive">
    <!-- Adjust to cover testing interval of interest -->
      <Window start="2011-08-26T04:00:00.0Z" end="2011-08-27T04:00:00.0Z"/>
    </LinearAvailBinding>
  </PlacementOpportunity>
</PlacementRequest>
```

Appendix XIV SCTE 130-3 Interface Placement Response Sample (whitelist)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Preload MUX with black list as default operation -->
<PlacementResponse messageRef="a" messageId="a" identity="a" version="a"
xsi:schemaLocation="http://www.scte.org/schemas/130-3/2008a/adm SCTE_130-3_2010.xsd"
xmlns="http://www.scte.org/schemas/130-3/2008a/adm" xmlns:core="http://www.scte.org/schemas/130-
2/2008a/core" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <core:StatusCode class="0"/>
  <SystemContext>
    <Network>//mso.com/MUX5/CNNZ33</Network>
  </SystemContext>
  <PlacementDecision id="528573CD-1485-13A6-56B778222E2B2936" placementOpportunityRef="2aa07706-
4219-4d58-bcdb-d5bcd07eea19">
    <Placement id="528573CD-1485-13A6-56B778222E2B2937" action="appFilter">
      <core:Tracking>77e41edd-a1ae-41a8-9aff-a4abef6876dd</core:Tracking>
      <PlacementConstraints>
        <!-- Allow any app from organization 1234 to run during this effective period -->
        <EffectiveStartDateTime>2011-08-26T04:00:00.0Z</EffectiveStartDateTime>
        <EffectiveEndDateTime>2011-08-27T04:00:00.0Z</EffectiveEndDateTime>
      </PlacementConstraints>
      <core:Ext>
          <AppList xmlns="urn:cablelabs:opencable:xsd:mux:1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd">
            <AppListEntry tracking="77e41edd-a1ae-41a8-9aff-a4abef6876dd" typeOfEntry="whitelist"
organizationID="1234" priorityOfEntry="3"/>
            <!-- Other derivatives -
            Add a second white list item
            <AppListEntry tracking="77e41edd-a1ae-41a8-9aff-a4abef6876dd" typeOfEntry="whitelist"
organizationID="5678" priorityOfEntry="3"/>
            Use a specific applicationID
            <AppListEntry tracking="77e41edd-a1ae-41a8-9aff-a4abef6876dd" typeOfEntry="whitelist"
organizationID="1234" applicationID="5678" priorityOfEntry="3"/>
            Add a specific instance (I06)
            <AppListEntry tracking="77e41edd-a1ae-41a8-9aff-a4abef6876dd" typeOfEntry="whitelist"
organizationID="1234" applicationID="5678" applicationInstanceID="1234" priorityOfEntry="3"/>
            Other variants possible
-->
          </AppList>
      </core:Ext>
    </Placement>
  </PlacementDecision>
  <PlacementDecision id="528573CD-1485-13A6-56B778222E2B2936" placementOpportunityRef="2aa07706-
4219-4d58-bcdb-d5bcd07eea19">
    <Placement id="528573CD-1485-13A6-56B778222E2B2937" action="appFilter">
      <core:Tracking>6a53754f-8743-4e52-aad0-26f8883a744a</core:Tracking>
      <PlacementConstraints>
        <!-- Allow app 5678 from organization 0xABCD to run from 7 - 8pm ET -->
        <EffectiveStartDateTime>2011-08-26T23:00:00.0Z</EffectiveStartDateTime>
        <EffectiveEndDateTime>2011-08-27T00:00:00.0Z</EffectiveEndDateTime>
      </PlacementConstraints>
      <core:Ext>
```

```
            <AppList xmlns="urn:cablelabs:opencable:xsd:mux:1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
              <AppListEntry typeOfEntry="whitelist" organizationID="43981" applicationID="5678"
priorityOfEntry="12"/>
            </AppList>
        </core:Ext>
      </Placement>
    </PlacementDecision>
</PlacementResponse>
```

Appendix XV SCTE 130-3 Interface: Placement Response Sample (blacklist)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Preload MUX with white list as default operation -->
<PlacementResponse messageRef="a" messageId="a" identity="a" version="a"
xsi:schemaLocation="http://www.scte.org/schemas/130-3/2008a/adm SCTE_130-3_2010.xsd"
xmlns="http://www.scte.org/schemas/130-3/2008a/adm" xmlns:core="http://www.scte.org/schemas/130-
2/2008a/core" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <core:StatusCode class="0"/>
  <SystemContext>
    <Network>//mso.com/MUX5/CNNZ33</Network>
  </SystemContext>
  <PlacementDecision id="528573CD-1485-13A6-56B778222E2B2936" placementOpportunityRef="2aa07706-
4219-4d58-bcdb-d5bcd07eea19">
    <Placement id="528573CD-1485-13A6-56B778222E2B2937" action="appFilter">
      <core:Tracking>77e41edd-a1ae-41a8-9aff-a4abef6876dd</core:Tracking>
      <PlacementConstraints>
        <!-- Allow any app from organization 1234 to run during this effective period -->
        <EffectiveStartDateTime>2011-08-26T04:00:00.0Z</EffectiveStartDateTime>
        <EffectiveEndDateTime>2011-08-27T04:00:00.0Z</EffectiveEndDateTime>
        <core:Ext>
          <AppList xmlns="urn:cablelabs:opencable:xsd:mux:1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd">
            <AppListEntry tracking="77e41edd-a1ae-41a8-9aff-a4abef6876dd" typeOfEntry="blacklist"
organizationID="1234" priorityOfEntry="3"/>
            <!-- Other derivatives -
            Add a second black list item
            <AppListEntry tracking="77e41edd-a1ae-41a8-9aff-a4abef6876dd" typeOfEntry="blacklist"
organizationID="5678" priorityOfEntry="3"/>
            Use a specific applicationID
            <AppListEntry tracking="77e41edd-a1ae-41a8-9aff-a4abef6876dd" typeOfEntry="blacklist"
organizationID="1234" applicationID="5678" priorityOfEntry="3"/>
            Add a specific instance (I06)
            <AppListEntry tracking="77e41edd-a1ae-41a8-9aff-a4abef6876dd" typeOfEntry="blacklist"
organizationID="1234" applicationID="5678" applicationInstanceID="1234" priorityOfEntry="3"/>
            Other variants possible
-->
          </AppList>
        </core:Ext>
      </PlacementConstraints>
    </Placement>
  </PlacementDecision>
  <PlacementDecision id="528573CD-1485-13A6-56B778222E2B2936" placementOpportunityRef="2aa07706-
4219-4d58-bcdb-d5bcd07eea19">
    <Placement id="528573CD-1485-13A6-56B778222E2B2937" action="appFilter">
      <core:Tracking>6a53754f-8743-4e52-aad0-26f8883a744a</core:Tracking>
      <PlacementConstraints>
        <!-- Allow app 5678 from organization 0xABCD to run from 7 - 8pm ET -->
        <EffectiveStartDateTime>2011-08-26T23:00:00.0Z</EffectiveStartDateTime>
        <EffectiveEndDateTime>2011-08-27T00:00:00.0Z</EffectiveEndDateTime>
        <core:Ext>
          <AppList xmlns="urn:cablelabs:opencable:xsd:mux:1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
            <AppListEntry typeOfEntry="blacklist" organizationID="43981" applicationID="5678"
priorityOfEntry="2"/>
          </AppList>
        </core:Ext>
      </PlacementConstraints>
    </Placement>
  </PlacementDecision>
  <PlacementDecision id="528573CD-1485-13A6-56B778222E2B2936" placementOpportunityRef="2aa07706-
4219-4d58-bcdb-d5bcd07eea19">
    <Placement id="528573CD-1485-13A6-56B778222E2B2937" action="update">
      <core:Tracking>36654f4d-dbb7-450c-b630-ce35ac1849f1</core:Tracking>
      <core:Content>
        <core:AssetRef providerID="MSO.COM" assetID="localtext.bxf"/>
      </core:Content>
      <PlacementConstraints>
        <!-- UPDATE app 5678 from organization ABCD  from 7:15 - 7:30pm ET -->
        <EffectiveStartDateTime>2011-08-26T23:15:00.0Z</EffectiveStartDateTime>
        <EffectiveEndDateTime>2011-08-26T23:30:00.0Z</EffectiveEndDateTime>
        <core:Ext>
          <AppList xmlns="urn:cablelabs:opencable:xsd:mux:1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <AppListEntry organizationID="43981" applicationID="5678"  priorityOfEntry="12"/>
          </AppList>
        </core:Ext>
      </PlacementConstraints>
    </Placement>
  </PlacementDecision>
</PlacementResponse>
```

Appendix XVI SCTE 130-3 Interface: Placement Response Sample (augmentation)

```
<?xml version="1.0" encoding="UTF-8"?>
<PlacementResponse messageRef="a" messageId="a" identity="a" version="a"
xsi:schemaLocation="http://www.scte.org/schemas/130-3/2008a/adm SCTE_130-3_2010.xsd"
xmlns="http://www.scte.org/schemas/130-3/2008a/adm" xmlns:core="http://www.scte.org/schemas/130-
2/2008a/core" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <core:StatusCode class="0"/>
  <SystemContext>
    <Network>//mso.com/MUX5/CNNZ33</Network>
  </SystemContext>
  <PlacementDecision id="528573CD-1485-13A6-56B778222E2B2936" placementOpportunityRef="2aa07706-
4219-4d58-bcdb-d5bcd07eea19">
    <Placement id="528573CD-1485-13A6-56B778222E2B2937" action="update">
      <core:Tracking>36654f4d-dbb7-450c-b630-ce35ac1849f1</core:Tracking>
      <core:Content>
        <!-- Use CoDF reference -->
        <core:AssetRef providerID="MSO.COM" assetID="localtext.codf"/>
      </core:Content>
      <PlacementConstraints>
        <!-- UPDATE app 5678 from organization ABCD  from 7:15 - 7:30pm ET -->
        <EffectiveStartDateTime>2011-08-26T23:15:00.0Z</EffectiveStartDateTime>
        <EffectiveEndDateTime>2011-08-26T23:30:00.0Z</EffectiveEndDateTime>
        <core:Ext>
          <AppList xmlns="urn:cablelabs:opencable:xsd:mux:1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <AppListEntry organizationID="43981" applicationID="5678" priorityOfEntry="1"/>
          </AppList>
        </core:Ext>
      </PlacementConstraints>
    </Placement>
  </PlacementDecision>
</PlacementResponse>
```

Appendix XVII SCTE 130-3 Interface: Placement Response Sample (SourceManagementConfig)

```
<?xml version="1.0" encoding="UTF-8"?>
<PlacementResponse messageRef="a" messageId="a" identity="a" version="a"
xsi:schemaLocation="http://www.scte.org/schemas/130-3/2008a/adm SCTE_130-3_2010.xsd"
xmlns="http://www.scte.org/schemas/130-3/2008a/adm" xmlns:core="http://www.scte.org/schemas/130-
2/2008a/core" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <core:StatusCode class="0"/>
  <SystemContext>
    <Network>//mso.com/MUX5/CNNZ33Session</Network>
  </SystemContext>
  <PlacementDecision id="528573CD-1485-13A6-56B778222E2B2936" placementOpportunityRef="2aa07706-
4219-4d58-bcdb-d5bcd07eea19">
    <Placement id="528573CD-1485-13A6-56B778222E2B2937" action="appFilter">
      <core:Tracking>77e41edd-a1ae-41a8-9aff-a4abef6876dd</core:Tracking>
      <PlacementConstraints>
        <!-- Allow any app from organization 1234 to run during this effective period -->
        <EffectiveStartDateTime>2011-08-26T04:00:00.0Z</EffectiveStartDateTime>
        <EffectiveEndDateTime>2011-08-27T04:00:00.0Z</EffectiveEndDateTime>
        <core:Ext>
        <mux:FlowConfig revisionDateTime="2001-12-17T09:30:47.0Z" revision="0" adsCallOutState="false"
xsi:schemaLocation="urn:cablelabs:MUX3.0:xsd:config:1.0OC-MUX3.0-CONFIG-1.0.0.xsd"
xmlns:mux="urn:cablelabs:opencable:xsd:mux:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <mux:SourceManagementConfig>
            <mux:Settings>
              <!-- Change the rule so on a splice (SCTE 35 out point) stop passing all PIDs from the source feed to
the output TS -->
              <mux:Rule tracking="b08935b5-56af-441b-bd24-cc260b1a1811"
nameOfFlowPoint="CNNZ33Session">splice-drop</mux:Rule>
            </mux:Settings>
          </mux:SourceManagementConfig>
        </mux:FlowConfig>
        </core:Ext>
      </PlacementConstraints>
    </Placement>
  </PlacementDecision>
</PlacementResponse>
```

Appendix XVIII Source Management

The set of active enhancements delivered in a Network stream may transition throughout the day. All enhancements originated within Network programming are carried on the same PIDs. For example, when a Network switches from one enhancement to another, the content and rate of the data streams will change, but the transport stream configuration (PMT) will remain the same.

NOTE: Since current efforts are focused on baseline profiles, the PMTs remain constant. As advanced profiles are introduced, the number of enhanced data PIDs identified in the PMT will expand and may change over time.

A cable operator may have multiple sources for enhancements on a given channel. There may be times where two or more sources are competing for both bandwidth and, more importantly, enhancement delivery. In addition to competing sources, two sources may need to be merged in order to produce a compliant output MUX. Business and operational rules may necessitate the removal of one of the enhancements from the downstream MUX.

Once an output MUX is groomed for EBIF, the signaling and data PIDs remain in the output program's PMT regardless of the presence of EBIF data at the source. In the examples it is assumed that each independent source may or may not be providing EBIF signaling and data at any given time. In existing cable operational practices, the feed from the telecast source (broadcast or cable network) is groomed at the acquisition point to create the desired PMT structure and to map incoming PIDs to that PMT structure. The acquisition point can be an acquisition MUX at a centralized MSO facility, a similar MUX at a master/regional headend, or a smaller headend where the acquisition MUX may include other functions (ex. ad splicing, rate shaping, etc.). The grooming process may include adding PIDs to the PMT that may not necessarily exist in the telecast source. For example, if a telecast source does not pass enhancements, the operator may add ETV PIDs, signaling and data, to the PMT to allow downstream splicing to operate per SCTE 30/35.

XVIII.1 Data Stream Selection

A MUX 3.0 device may need to select from data sources coming from the broadcast programming, from a local ad splice that contains a pre-bind enhancement, or from a local streamer inserting data that will be combined with broadcast or spliced audio/video content (late-bind enhancement).The streamer function may be implemented as an external function/sub-system, integral to the MUX functionality, or in some other manner. In the case of EBIF, once an output MUX is groomed to carry ETV data, the data PIDs remain in the PMT despite the fact that PES packets for the EBIF data PIDS may or may not be actively streaming from the source.

On a source by source basis, EBIF PIDs (EISS or ETV-DATA) are groomed, dropped or merged within an output MUX. In each scenario, the output MUX PMT is initially generated with a static descriptor list. Throughout the various use cases, the PMT remains constant.

When two or more EISS PIDs are merged, they create a single EISS PID on the output MUX representing both EISS sources. Merging EISS streams preserves the content of each source through table boundary multiplexing.

XVIII.2 Use Case examples

The following table describes some typical use cases which would need to be handled by the MUX 3.0 devices.

ID	Description
Broadcast Enhancement	
Broadcast Drop on Splice	The MUX 3.0 device should drop the broadcast data PIDs, if present, when splicing to a local audio/video feed (e.g., during local ad insertion). The local audio/video feed may or may not be enhanced. This will be the typical configuration.
Broadcast Keep on Splice	The MUX 3.0 device should preserve the broadcast data PIDs when splicing to a local audio/video feed. If the local audio/video feed has an enhancement, the EISS will be merged. This is an uncommon configuration, but may be used on a Network that is running a 24x7 channel enhancement with permission to execute across local ad breaks.
Conditional Broadcast Keep on Splice	The MUX 3.0 device should preserve the broadcast data PIDs when splicing to a local audio/video feed if the local audio/video feed does not include an enhancement. This is an uncommon configuration, but may be used on a Network that is running a 24x7 channel enhancement with permission to conditionally execute enhancements across local ad breaks. The "splice-exclusive" rule is used to configure this use case.
Late-Bind Enhancement	
Late-Bind Broadcast, Keep on Splice	The MUX 3.0 device will insert an enhanced signal and data from a streamer on both the broadcast feed and spliced feed. This might be used by systems that are locally enhancing the broadcast programming as well as locally spliced ads. The content of the enhanced signal and data stream may change between splice events based on control messages, but that will be transparent to the MUX 3.0 device.
Late-Bind Splice Only	The MUX 3.0 device will only insert enhanced signal and data from a streamer during a local splice. If the broadcast feed has an enhancement, it will be dropped. No merging takes place here.
Late-Bind Broadcast Only	The MUX 3.0 device will insert enhanced signal and data from a streamer when passing the broadcast feed, but will not pass the broadcast enhanced signal and data when splicing local ads. If a streaming source is present while splicing a local ad, it will be dropped.
Broadcast and Late-Bind Splice	The MUX 3.0 device will pass enhanced signal and data along with the broadcast programming, and will switch to enhanced signal and data from a streamer when splicing local ads.
Pre-Bind Enhancement	
Broadcast and Pre-Bind Splice	The MUX 3.0 device will switch between the enhanced signals and data streams provided within the broadcast transport and enhanced signals and data streams provided within the local ad transport.

Appendix XIX Acknowledgements

We wish to thank the lead MSO participants contributing directly to this document:

Author	Company
Jay Patel	CableLabs
Steve Newkirk	Charter
Walt Michel	Comcast
Robert Steinberg	Comcast
Joe Davis	Cox
Barry Pratt	Rogers
Niem Dang	TWC

Additional MSO contributors and reviewers include:

Contributor/ Reviewer	Company
Debbie Fitzgerald	CableLabs
Peter Caramanica	Cablevision
John Markowski	Cablevision
Wilbert Caban	Comcast
Adam Larose	Comcast
Steve Reynolds	Comcast
David DeAndrade	Comcast
Roy Pereira	Cox
Chip Paryzek	Cox
John Civiletto	Cox
George Hart	Rogers
Craig Mohanchek	TWC
Vipul Patel	TWC

Additional Vendor contributors and reviewers include:

Contributor/ Reviewer	Company
Andrew Sundelin	Applied Broadband
Jason Schnitzer	Applied Broadband
Joel Daly	Arris
Aviad Inger	Arris
Eyol Komemi	Arris

Contributor/ Reviewer	Company
Majid Chelehmal	Arris
David Casselano	Arris
Ashraf Albert	Cisco
Joris Lammers	Cisco
Pete Chave	Cisco
Ives Wauters	Cisco
Kristof Margodt	Cisco
Larry Westerman	Ensequence
Allan Nicholson	Harmonic
Yaron Raz	Harmonic
Pat Weddell	Harmonic
Lior Morad	Imagine
Steve Musallam	Motorola
Kevin McKinnon	Motorola
Dnyanesh Chati	Motorola
Aviral Pandey	Motorola
Fabrice Quinard	Motorola
David Brenner	Motorola
Dwight Burchak	Motorola
Avi Braslavski	Motorola
Paul Braun	RGB
Ben Leftowitz	RGB
Adrian Fowkes	S&T
JD Burke	S&T
Adam Mauger	Softel