SCTE® | STANDARDS

Digital Video Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 250 2022

Real-time Event Signaling and Management API

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards and Operational Practices (hereafter called "documents") are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interoperability, interchangeability, best practices, and the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members.

SCTE assumes no obligations or liability whatsoever to any party who may adopt the documents. Such adopting party assumes all risks associated with adoption of these documents and accepts full responsibility for any damage and/or claims arising from the adoption of such documents.

NOTE: The user's attention is called to the possibility that compliance with this document may require the use of an invention covered by patent rights. By publication of this document, no position is taken with respect to the validity of any such claim(s) or of any patent rights in connection therewith. If a patent holder has filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license, then details may be obtained from the standards developer. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Patent holders who believe that they hold patents which are essential to the implementation of this document have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE web site at https://scte.org.

140 Philips Road
Exton, PA 19341

Document Types and Tags

Document Type: Specification

Document Tags:

☐ Test or Measurement

☒ Architecture or Framework

☐ Procedure, Process or Method

☐ Checklist

☐ Metric

☒ Cloud

☐ Facility

☒ Access Network

☐ Customer Premises

Document Release History

Release	**Date**
SCTE 250 2019	*01/28/2019*
SCTE 250 2020	*02/18/2022*
SCTE 250 2022	*August 2022*

Note: Standards that are released multiple times in the same year use: a, b, c, etc. to indicate normative balloted updates and/or r1, r2, r3, etc. to indicate editorial changes to a released document after the year.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Executive Summary

This standard defines the Real-time Event Signaling and Management (ESAM) API, an interface that allows a Signal Acquisition System (e.g. encoder, transcoder, packager, stream switcher, etc.) to submit signals to a Signal Decision System and receive relevant instructions for processing the signal or associated content. Furthermore, the Signal Decision System has the ability to initiate a set of instructions based on a schedule or event that is not signaled in the content.

There are several kinds of acquisition systems and the possible instruction set will vary based on the purpose and capabilities of the system. An encoder/transcoder will typically receive instructions for conditioning the content for splices or changes in content for either advertising insertion or alternative content due to blackout restriction. A linear stream switcher is responsible for managing an outbound linear stream based on content/blackout requirements for an intended audience. A packager will typically receive instructions about how to segment content and compose a manifest. All systems are eligible to receive instructions about status reporting or how to add or remove in-band signals.

Table 1 – Types of Instructions

Instruction	Encoder / Transcoder	Linear Stream Switcher	Packager
Remove original in-band signal	✓	✓	✓
Insert new in-band signal for downstream consumption	✓		✓
Receive status information regarding a given signal	✓		✓
Receive and use an endpoint for reporting status	✓	✓	✓
Condition stream for a splice (either in or out)	✓		
Switch content	✓	✓	✓
Customize an ABR manifest			✓
Perform instructions on a repeating / ongoing basis	✓		✓

A given environment may be comprised of multiple Signal Acquisition Systems (SAS) and multiple Signal Decision Systems (SDS). Multiple SAS may be employed to process different content or to redundantly process the same content as other systems. Multiple SDS may be employed to handle load of decisions or to provide redundancy.

In a linear acquisition model, the linear stream is acted on by multiple systems, which are capable of acquiring an in-band signal. The SAS extracts the signal and uses it as the basis to request instructions from the SDS (Figure 1). The SDS will respond with instructions appropriate for the particular SAS and its capabilities.

For example, a real-time transcoder acting as a SAS submits an SCTE 35 splice insert message to the SDS. The SDS may consult an SCTE 130 Placement Opportunity Information Service (POIS) to confirm the validity of the signal and return instructions to the transcoder to identify and update the start/end times of the signaled region, condition the video stream at the appropriate splice points and insert a more descriptive signal into the stream for downstream consumption.

The packager will encounter the enhanced signal that was previously confirmed at the transcoder and, again, submit the signal to the SDS. This time, the SDS could provide manifest-specific conditioning instructions. Not shown in the figure is that downstream from the CDN, the client (or a manifest manipulator acting on behalf of a client) will use the manifest instructions to initiate decisions for ad insertion or alternative content.

The QAM (or multicast IP) use case is to employ an acquisition system, which will traffic streams to geographic zones. The primary instruction for this use case will be a switch from primary content stream to an alternative stream or slate.

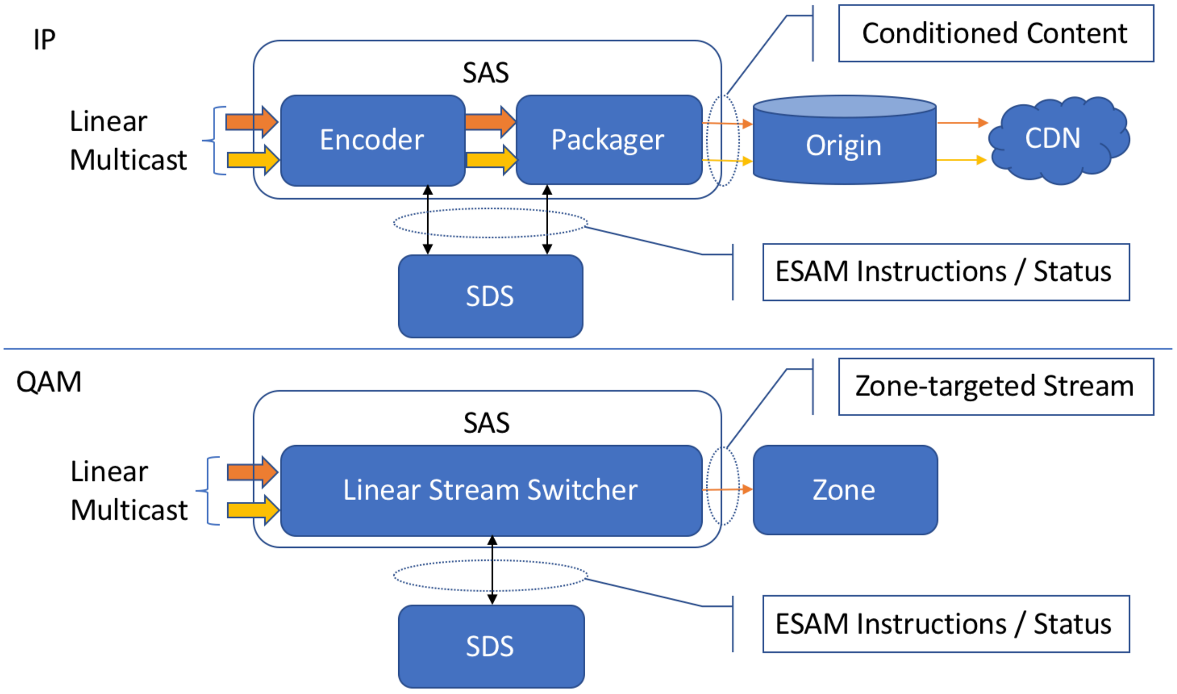

Figure 1 - Example Real-time Video Applications

For VOD content, a transcoder can operate in one of two operational modes (Figure 2). In mode A, the transcoder is triggered to call the SDS to obtain signal regions for content in a file. In mode B, the transcoder is directed to process the file-based content based on directives in the request.

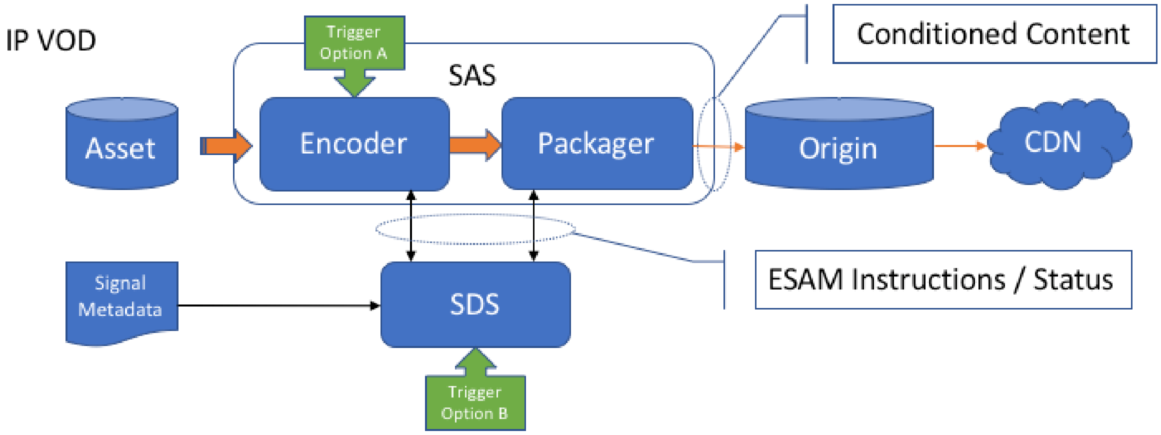

Figure 2 - Example file-based IP Video Application

To fully understand the ABR conditioning portions of this document, the reader is expected to be familiar with and understand the different ABR delivery formats and their individual terminology.

The Event Signaling and Management API supports both JSON and XML event and notification message payload formats with the caller controlling the payload format using standard HTTP semantics.

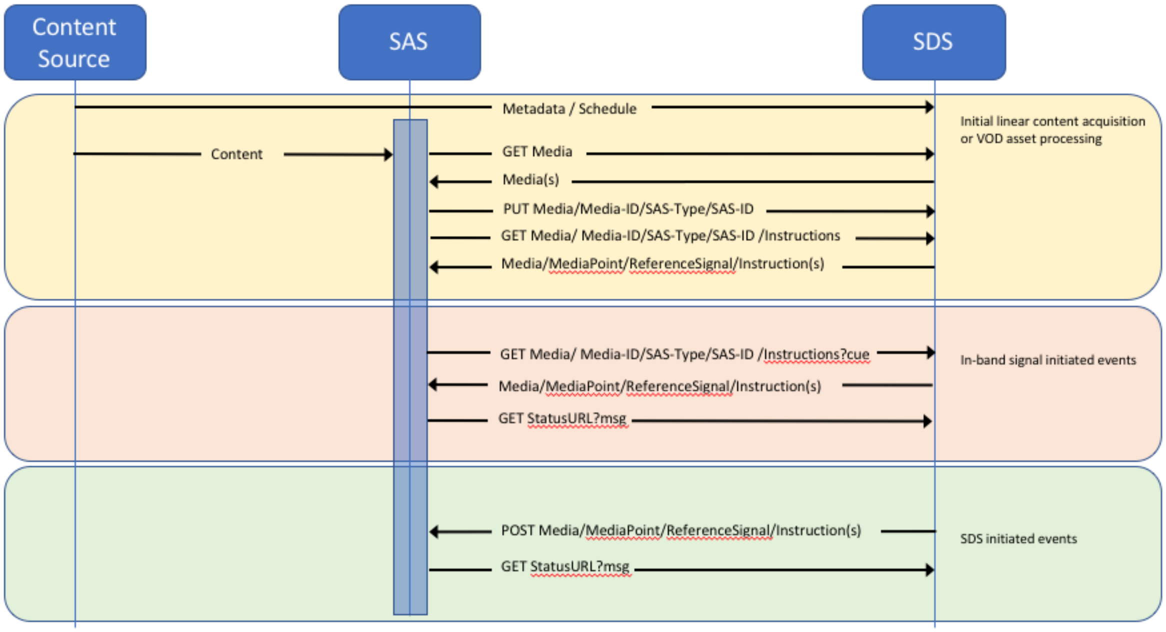
Content Source
SAS
SDS
Metadata / Schedule
Initial linear content acquisition or VOD asset processing
Content
GET Media
Media(s)
PUT Media/Media-ID/SAS-Type/SAS-ID
GET Media/ Media-ID/SAS-Type/SAS-ID /Instructions
Media/MediaPoint/ReferenceSignal/Instruction(s)
GET Media/ Media-ID/SAS-Type/SAS-ID /Instructions?cue
In-band signal initiated events
Media/MediaPoint/ReferenceSignal/Instruction(s)
GET StatusURL?msg
POST Media/MediaPoint/ReferenceSignal/Instruction(s)
SDS initiated events
GET StatusURL?msg

Figure 3 – Interaction between SAS and SDS

2. Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of this document. At the time of subcommittee approval, the editions indicated were valid. All documents are subject to revision; and while parties to any agreement based on this document are encouraged to investigate the possibility of applying the most recent editions of the documents listed below, they are reminded that newer editions of those documents might not be compatible with the referenced version.

2.1. SCTE References

- ANSI/SCTE 35 2019, Digital Program Insertion Cueing Message for Cable.
- ANSI/SCTE 214-1 2016, MPEG DASH for IP-Based Cable Services Part 1: MPD Constraints and Extensions

2.2. Standards from Other Organizations

- ISO 8601:2004, Data elements and interchange formats -- Information interchange -- Representation of dates and times (Coordinated Universal Time).

- ISO 8601-1:2019, Date and time — Representations for information interchange — Part 1: Basic rules

- IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. T. Berners-Lee, R. Fielding, L. Masinter. January 2005.

- IETF RFC 4648, The Base16, Base32, and Base64 Data Encodings. S. Josefsson. October 2006.

- IETF RFC 7230, Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. R. Fielding, J. Reschke. June 2014.

- IETF RFC 7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. R. Fielding, J. Reschke. June 2014.

- W3C XML Base (Second Edition). W3C Recommendation 28 January 2009. http://www.w3.org/TR/xmlbase/.

- W3C XML Schema Part 2: Datatypes Second Edition. W3C Recommendation 28 October 2004. http://www.w3.org/TR/xmlschema-2/.

- W3C XML Path Language (XPath) 2.0 (Second Edition). W3C Recommendation 14 December 2010. http://www.w3.org/TR/xpath20/.

- W3C XML Linking Language (XLink) Version 1.1. W3C Recommendation 06 May 2010. http://www.w3.org/TR/xlink11/.

2.3. Published Materials

3. Informative References

The following documents might provide valuable information to the reader but are not required when complying with this document.

3.1. SCTE References

- ANSI/SCTE 67, Recommended Practice for SCTE 35 Digital Program Insertion Cueing Message for Cable.

- ANSI/SCTE 172, Constraints on AVC and HEVC Structured Video Coding for Digital Program Insertion.

- ANSI/SCTE 236, Content Metadata

- SCTE 224, Event Scheduling and Notification Interface (ESNI).

3.2. Standards from Other Organizations

- IETF RFC 8216, HTTP Live Streaming, https://tools.ietf.org/html/rfc8216, R. Pantos, August 2017.

- IETF RFC 6749, The OAuth 2.0 Authorization Framework. D. Hardt. October 2012.

3.3. Published Materials

- Flash® Media Manifest (F4M) Format Specification Version 3.0 FINAL, http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/hds/pdfs/adobe-media-manifest-specification.pdf

- JavaScript Object Notation (JSON), http://www.json.org.

4. Compliance Notation

shall	This word or the adjective "***required***" means that the item is an absolute requirement of this document.
shall not	This phrase means that the item is an absolute prohibition of this document.
forbidden	This word means the value specified shall never be used.
should	This word or the adjective "*recommended*" 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 weighted 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 or the adjective "*optional*" 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.
deprecated	Use is permissible for legacy purposes only. Deprecated features may be removed from future versions of this document. Implementations should avoid use of deprecated features.

5. Abbreviations and Definitions

5.1. Abbreviations

ABR	adaptive bitrate
API	application programming interface
DASH	MPEG Dynamic Adaptive Streaming over HTTP
GUID	Global Unique Identifier
HDS	HTTP Dynamic Streaming (Adobe Zeri)
HLS	HTTP Live Streaming (Apple)
HSS	HTTP Smooth Streaming (Microsoft Smooth)
HTTP	Hypertext Transfer (or Transport) Protocol
JSON	JavaScript Object Notation
LAN	local area network
NTP	Network Time Protocol
POIS	Placement Opportunity Information System
REST	Representational state transfer
SAS	Signal Acquisition System
SPS	Signal Processing System
URI	Uniform Resource Identifier

URL	Uniform Resource Locator
UTC	Coordinated Universal Time
VOD	video on demand
WAN	wide area network
XML	eXtensible Markup Language

5.2. Definitions

Base64 Binary	Binary contents coded in Base64 format
Segment	Unit of data associated with an HTTP URL and optionally a byte range that is specified by a manifest.
Media	Any form of on-demand, time-shifted or linear content.
HLS Media Segment	SegmentThe M3U8 extended record commencing with the #EXTINF tag through to its paired URI line inclusive of any line in between that starts with #EXT.
RESTful	Web service that conforms to the REST architectural style
Signal Point	A particular point of interest within a video essence.
Signal Region	A region of interest within a video essence.

6. Common Conventions

Note 1: Unless noted as optional, all attributes, elements, and objects are required.

Note 2: In all cases, unrecognized attributes, elements, and objects are to be ignored.

As a convention used throughout this API, a JSON array is identified using a plural name like 'spots' or 'segments'. The XML equivalent utilizes an XML element sequence with a singular element tag that is capitalized., for example, 'Spot' or 'Segment'. In documentation situations, herein where duplicating the element name is redundant or confusing, the XML value is utilized and the JSON equivalent is to be assumed by the reader.

6.1. XML Namespaces

The XML namespace for the API defined in this specification is **`http://www.scte.org/schemas/250/bn1.`**

The XML schema version corresponding to this specification is “20220419”.

To improve readability, all XML examples in this document are assumed to have the following namespace declarations:

```
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xmlns="http://www.scte.org/schemas/250/bn1"
```

6.2. Standard data formats

All duration and date and time values in this API ***shall*** use the formats specified in ISO 8601.

Binary contents ***shall*** be coded in Base64 format per section 6.8 of IETF RFC 4648.

7. Data Model

The data model provides a set of structured messages for use in providing instructions from an SDS to an SAS.

7.1. Media

The term, Media, is purposely broad and is applicable to all forms of on-demand, time-shifted and linear content, as well as other forms of media not explicitly mentioned here. Media is instantiated as a global **Media** element, which may contain a collection of **MediaPoint** elements.

The primary identification mechanism of the **Media** element is the @id attribute. The @description attribute provides expanded descriptive text.

Table 2 - Semantics of the Media element

Element or Attribute Name	Use	Description
@id	CM	This value *may* directly identify the content feed that it represents, but it ***shall not*** be presumed to do so.
@description	O	A common description used for the **Media**, which *may* be a call-sign commonly used for a linear network such as "ESPN" or an asset title such as "Titanic".
@batch	O	An identifier used to relate this request to a batch of requests. Batch requests are used to group processing of VOD assets.
MediaPoint	0..N	A set of temporal media points within the media. See 7.2.
Instruction	0..N	A set of Media-level instructions, which ***shall*** only be present for a VOD asset. See 7.5.
AcquisitionSystem	0..N	A set of Acquisition Systems that are registered with an SDS for the parent Media stream. This ***shall not*** be populated together with Instructions.
Ext	0..1	
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example **Media** (namespace declarations omitted for readability):

```
<Media description="XYZ Linear Network" id="esam/media/xyz">
    <Encoder endpoint="http://10.10.10.10:1010/enc1" id="esam/media/xyz/encoder/enc1"/>
</Media>
```

7.2. MediaPoint

A MediaPoint represents an event within a Media stream or asset. It *shall* be associated with one of a point along the Media timeline. When a MediaPoint is provided in response to a request for instructions from a SAS then the corresponding ReferenceSignal (for a SCTE 35 signal) or ReferenceWatermark that triggered the request shall be provided. When a MediaPoinit is given to the SAS proactively (also referred to as out-of-band) then a MatchSignal (for a SCTE 35 signal), MatchWatermark or MatchTime shall be provided to indicate the trigger that the SAS will use in the future. In any case the associated instructions are provided.

Table 3 - Semantics of the MediaPoint element

Element or Attribute Name	Use	Description
ReferenceSignal	0..1	A reference to the SCTE-35 signal that was used to trigger this MediaPoint and its associated instructions
MatchSignal	0..1	Criteria for matching a future SCTE 35 signal for the purpose of applying this MediaPoint and its associated instructions
MatchTime	0..1	Criteria for matching a future time for the purpose of applying this MediaPoint and its associated instructions.
ReferenceWatermark	0..1	A reference to the Watermark signal that was used to trigger this MediaPoint and its associated instructions
MatchWatermark	0..1	Criteria for matching a future Watermark signal for the purpose of applying this MediaPoint and its associated instructions
Instruction	0..N	A set of instructions that apply to this specific MediaPoint. See 7.5.
Ext	0..1	
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example **MediaPoint** (namespace declarations omitted for readability):

```
<Media description="XYZ Linear Network" id="esam/media/xyz">
  <MediaPoint>
    <ReferenceSignal
remove="true">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyM
WLvc/g=</ReferenceSignal>
    <Signal offset="PT0.000S">/DAWAAAAAAAA///wBQb/CXVCAgAA28P+7g==</Signal>
    <Condition direction="OUT" offset="PT0.000S" />
    <Condition direction="IN" offset="PT0.000S" />
  </MediaPoint>
</Media>
```

7.3. ReferenceSignal

All instructions provided by an SDS in response to a signal request are associated with a reference signal. This signal may either be acquired in-band from the underlying content or initiated asynchronously by the SDS in response to a set of out-of-band break points or some other event.

A ReferenceSignal has the following attributes:

Table 4 - Semantics of the ReferenceSignal element

Element or Attribute Name	Use	Description
`@remove`	O	A Boolean value that instructs the SAS to remove the original signal from the content stream if set to true. When this attribute is not present, the default value ***shall*** be assumed to be false.
`#text`	M	The Base64 encoded version of the original signal being referenced.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example **`ReferenceSignal`** (namespace declarations omitted for readability):

```
<ReferenceSignal remove="true">
/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=
</ReferenceSignal>
```

7.4. MatchSignal

MatchSignal provides the context to evaluate and identify any component of a fully expressed XML representation of a qualified SCTE 35 signal as a set of XPath 2.0 assertions. When the assertions are evaluated and found to return an affirmative response, the Policy related to the **`MatchSignal`** would then also be evaluated. The **`MatchSignal`** wraps a sequence of **`Assert`** XPath expressions and if the set of XPath 2.0 assertions yield a negative result, the Policy related to the **`MatchSignal`** ***shall not*** be removed or applied. The `@match` attribute needs to be consulted for the level of inclusivity/exclusivity.

The **`MatchSignal`** would be a match which results in application (or removal) of the Policy if all of the **`Assert`** Xpath expressions evaluate to true when the `@match` is set to "ALL".

The **`MatchSignal`** would be a match which results in application (or removal) of the Policy if any of the **`Assert`** Xpath expression evaluates to true when the `@match` is set to "ANY".

The **`MatchSignal`** would be a match which results in application (or removal) of the Policy only if each **`Assert`** Xpath expression evaluates to false when the `@match` is set to "NONE".

Table 5 - Semantics of the MatchSignal element

Element or Attribute Name	Use	Description
`@match`	OD default: `ALL`	One of the following values: `ALL` – All **`Assert`** statements ***shall*** evaluate to true `ANY` – At least one **`Assert`** statement ***shall*** evaluate to true `NONE` – None of the **`Assert`** statements ***shall*** evaluate to true
`@effective`	O	This value, when provided, indicates the earliest time that this MatchSignal shall be considered by the SAS. It is always expressed as a date and time in UTC. When omitted, there shall be no early boundary for applying this MatchSignal criteria.
`@expires`	O	This value, when provided, indicates the latest time that this MatchSignal shall be considered by the SAS. It is always

		expressed as a date and time in UTC. When omitted, there shall be no late boundary for applying this MatchSignal criteria.
Assert	1..n	This element carries an XPath 2.0 expression as if it were going to be evaluated against the SCTE 35 schema compliant XML document. An implementation *should* evaluate the expression against the SCTE 35 schema compliant document but *may* choose other methods outside the scope of this standard. An expression, when evaluated, results in a positive match for the **Assert**, returning true as if the fn:boolean() had been applied, false otherwise.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

The *TemplateResponse* ***shall*** be able to use macro substitutions in the *ManifestResponse* which instruct the packager about how to condition the manifests using the SCTE 35 message as a source. These macros are delimited by the '$' character in the notification response. Here is a list of the possible macros along with the origin of the data for that macro. If a specified macro element is not present in the SCTE 35 message, that element of the notification *should* be ignored.

Table 6 – Standard template parameters

acquisitionPointIdentity	Configured on the packager
acquisitionSignalID	Randomly generated per signal
segmentationEventId	Extracted from segmentation_event_id in the segmentation descriptor
segmentationTypeId	Extracted from segmentation_type_id in the segmentation descriptor
segmentationUpid	Extracted from segmentation_upid() name in the segmentation descriptor
duration	Extracted from segmentation_duration in the segmentation descriptor
hdsDuration	Duration expressed as fractional seconds
availNum	Extracted from splice insert command or segment num in the segmentation descriptor
availExpected	Extracted from splice insert command or segments expected in the segmentation descriptor
subSegmentNum	Extracted from the sub_segment_num in the segmentation descriptor
subSegmentsExpected	Extracted from the sub_segments_expected in the segmentation descriptor
utcPoint	Expected wall clock time of signal point in UTC timestamp (XML dateTime)

ptsTime	Expected PTS value of signal point
smoothTime	Expected Smooth timestamp of signal point
hdsTime	Expected signal point time expressed as fractional seconds
binarySignal	Base64 encoded representation of the SCTE 35 signal

7.5. ReferenceWatermark

All instructions provided by an SDS in response to a Watermark signal request are associated with a reference Watermark.

A ReferenceWatermark has the following attributes:

Table 7 - Semantics of the ReferenceWatermark element

Element or Attribute Name	Use	Description
`#text`	M	The Base64 encoded version of the original ATSC 3.0 compliant Watermark being referenced.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example **`ReferenceWatermark`** (namespace declarations omitted for readability):

```
<ReferenceWatermark>
…
</ReferenceWatermark>
```

7.6. MatchWatermark

MatchWatermark provides the context to evaluate and identify any component of a fully expressed XML representation of a qualified ATSC Watermark signal as a set of XPath 2.0 assertions. When the assertions are evaluated and found to return an affirmative response, the Policy related to the **`MatchWatermark`** would then also be evaluated. The **`MatchWatermark`** wraps a sequence of **`Assert`** XPath expressions and if the set of XPath 2.0 assertions yield a negative result, the Policy related to the **`MatchWatermark`** ***shall not*** be removed or applied. The `@match` attribute needs to be consulted for the level of inclusivity/exclusivity.

The **`MatchWatermark`** would be a match which results in application (or removal) of the Policy if all of the **`Assert`** Xpath expressions evaluate to true when the `@match` is set to “ALL”.

The **`MatchWatermark`** would be a match which results in application (or removal) of the Policy if any of the **`Assert`** Xpath expression evaluates to true when the `@match` is set to “ANY”.

The **`MatchWatermark`** would be a match which results in application (or removal) of the Policy only if each **`Assert`** Xpath expression evaluates to false when the `@match` is set to “NONE”.

Table 8 – Semantics of the MatchSignal element

Element or Attribute Name	Use	Description
`@match`	OD default:	One of the following values: `ALL` – All **`Assert`** statements ***shall*** evaluate to true

	ALL	ANY – At least one **Assert** statement ***shall*** evaluate to true NONE – None of the **Assert** statements ***shall*** evaluate to true
@effective	O	This value, when provided, indicates the earliest time that this MatchSignal shall be considered by the SAS. It is always expressed as a date and time in UTC.
@expires	O	This value, when provided, indicates the latest time that this MatchSignal shall be considered by the SAS. It is always expressed as a date and time in UTC.
@schema	M	The XML schema to be used for processing the Watermark
Assert	1..n	This element carries an XPath 2.0 expression as if it were going to be evaluated against the provided schema. An implementation *should* evaluate the expression against the provided schema compliant document but *may* choose other methods outside the scope of this standard. An expression, when evaluated, results in a positive match for the **Assert**, returning true as if the fn:boolean() had been applied, false otherwise.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

7.7. Instruction

Instructions are a set of child elements within a Media or MediaPoint. Some instructions are typically applicable only to transcoders (e.g. Condition), some are applicable only to packagers (e.g. Tag) and others may apply to either (e.g. Signal).

When an instruction must be precisely timed within the video, an offset attribute shall be provided. This offset is expressed as a duration and shall be frame accurate. The offset is referenced from one of the following:

- When the instruction is a child of a Media element then the offset is from the beginning of the Media.
- When the instruction is a child of a MediaPoint element and a ReferenceSignal is provided then the offset is added to the PTS point referenced by the signal, not the location where the signal was discovered in the stream.
- When the instruction is a child of a Template element and a MatchSignal is provided then the offset is added to the PTS point of the matched signal, not the location where the signal was discovered in the stream.

An instruction *may* be specified as repeating when the @interval and @end attributes are present. In this case, the instruction ***shall*** be executed at the time specified by @offset and again at the time specified by adding @interval to @offset. This *should* be continued until reaching the duration of @offset + @end.

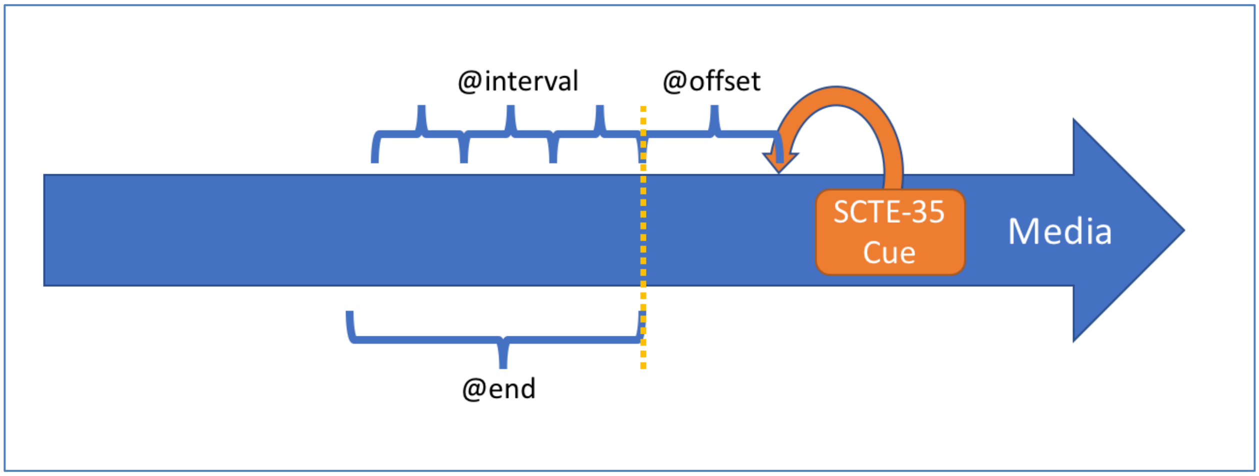

Figure 4 – Use of repeating instructions

7.7.1. Signal

Table 9 - Semantics of the Signal element

Element or Attribute Name	Use	Description
@offset	CM	The offset from the referenced or matched signal, accurate to milliseconds.
@pid	O	Specifies the MPEG-2 Transport Stream PID number on which the Signal shall be placed. This value shall be specified when multiple output PIDs are used for carrying SCTE 35 information and may be provided if there is only one output PID. When not specified, the SAS should use the same PID as the incoming SCTE 35 was found or in the case of an out-of-band instruction, choose the lowest number output PID used for SCTE 35 cues.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Insert a signal at the specified offset. The element value contains a Base-64 encoding of the binary signal.

```
<Signal offset="PT0S">U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl</Signal>
<Signal offset="PT30.000S">U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl</Signal>
```

7.7.2. Condition

Condition the stream for a potential splice at the specified offset.

Table 10 - Semantics of the Condition element

Element or Attribute Name	Use	Description
@offset	CM	The offset from the referenced or matched signal, accurate to milliseconds.
@direction	M	**BOTH** – Condition for either in our out **IN** – Condition as an in-point only **OUT** – Condition as an out-point only
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

```
<Condition direction="OUT" offset="PT0S"/>
<Condition direction="IN" offset="PT30.000S"/>
```

7.7.3. Content

The Content element instructs the SAS to switch the content to another stream or asset. The details for how content is switched is out of scope for this document.

Table 11 - Semantics of the Content element

Element or Attribute Name	Use	Description
@offset	CM	The offset from the referenced or matched signal, accurate to milliseconds.
@zone	O	The identity of the zone for which this content switch should apply. When absent, then the content switch applies for all outputs.
#text	M	A URI reference to the content to be switched.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example (namespace declarations omitted for readability):

```
<Content offset=”PT0S”>/sources/content-feed-3</Content>
```

Example **zone**(namespace declarations omitted for readability):

```
<Content offset=”PT0S” zone=”zone1”>SLATE_ID1</Content>
```

7.7.4. Tag (HLS)

A Tag element provides one or more manifest tags (lines) to be directly inserted into the output HLS manifest file. Each line is explicitly controlled, via the "adapt" attribute, as to whether the line element is to be placed directly into the manifest file unaltered or if the packager is to enhance the line before insertion. If the @adapt attribute is set to true, the packager ***shall*** fill in the placeholder substitution keyword framed by the start substitution delimiter '${' and ending delimiter '}'. Thus, the full substitution sequence is ${keyword}. Table 11 lists the substitution keywords and their descriptions. The line insertion location is controlled by the @location attribute and lines having the same @location are positioned in returned document order.

Table 12 - Semantics of the Tag element

Element or Attribute Name	Use	Description
@segment	M	**FIRST** – Manifest lines that are inserted at the signal point start HLS Media Segment (i.e., the signal splice start location). The HLS Media Segment's start record marker (i.e. #EXTINF line) and URI are not altered. **SPAN** – Manifest lines that are inserted for each HLS Media Segment between the first HLS Media Segment and the last HLS Media Segment, excluding the first and last HLS Media Segments, which are independently specified using FIRST and LAST respectively. Typically used only for linear streaming. **LAST** – Manifest lines that are inserted with the last HLS Media Segment identified as the end location. The HLS Media Segment's start record marker (i.e., #EXTINF line) and URI are not altered.
@adapt	O	Boolean indicating if the value attribute's string is to be modified by the packager before placement into the manifest file. If the attribute is omitted, the default value is "false."
@location	O	Enumeration specifying the line location relative to the HLS Media Segment. **BEFORE** – The line is to be placed before the Media Segment's #EXTINF tag line. **WITHIN** – The line is to be placed in between the Media Segment's start #EXTINF line and the Media Segment's ending URI line. **AFTER** – The line is to be placed after the Media Segment's URI line **REPLACE** – The line is to replace an existing line with the same tag name (e.g. #EXTINF).
#text	M	Line to be placed into the manifest file. Typically beginning with "#EXT".
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Table 13 - Manifest Line Substitution Keywords

Substitution Keyword	Value Description
timeFromSignal	Offset from the CUE (typically a start segmentation type) of the earliest presentation time of the HLS Media Segment that follows. If an implementation removes fragments from the manifest file (ex. live application), the ELAPSED value ***shall*** be adjusted by the duration of the HLS Media Segments removed. Elapsed is expressed in seconds to millisecond accuracy formatted as an ISODuration [ISO 8601].
timeFromSignalFS	Offset from the CUE (typically a start segmentation type) of the earliest presentation time of the HLS Media Segment that follows. If an implementation removes fragments from the manifest file (ex. live application), the ELAPSED value ***shall*** be adjusted by the duration of the HLS Media Segments removed. Elapsed is expressed in seconds to millisecond accuracy. (ex. 30.234)
segmentID	The original Media Segment URI.
streamID	The configured streamID value.

Example **`Tag`** (namespace declarations omitted for readability):

```
<Tag adapt="false" location="BEFORE" segment="FIRST">#EXT-X-SPLICE-
EXIT:SpliceDescriptors=0x786df876dffa87687,Time={timeFromSignalFS}</Tag>
```

7.7.5. DateRange (HLS)

Table 14 - Semantics of the DateRange element

Element or Attribute Name	Use	Description
`@id`	O	An ID value associated with the DateRange. Recommend that the SCTE35 segmentation_event_id is used when present.
`@class`	O	A type associated with the DateRange. Although this can be any value, it should use the SCTE35 segmentation_type_id when present.
`@duration`	O	This value should only be used when the duration of the DateRange is known to be fixed and cannot change. Otherwise, populate the plannedDuration value.
`@plannedDuration`	O	This value should be used to indicate the expected duration of a DateRange whose actual duration is not yet known.
`@endOnNext`	O	This Boolean attribute, when set to True, indicates that the end of the DateRange containing it is indicated by the start of a subsequent DateRange that has the same @class attribute.
`@scte35Cmd`	O	
`@scte35Out`	O	
`@scte35In`	O	
`X-<any>`	O	Any additional attributes may be defined provided that the attribute name begins with “X-“. These attributes shall be carried forward into the HLS manifest using the given names and values.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example **`DateRange`** (namespace declarations omitted for readability):

```
<DateRange id=”2044334492” class=”Distributor opportunity” duration=”60”
scte35Cmd=”/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=”/>
```

7.7.6. SparseTrack (HSS)

Data to be embedded in an HSS sparse track. The data blob is an adaptable string similar to the tags/value attribute.

Table 15 - Semantics of the SparseTrack element

Element or Attribute Name	Use	Description
`@name`	O	Track name identifier. If omitted, the first sparse track is to be assumed.
`#text`	M	String to be placed into the Smooth sparse track.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example (namespace declarations omitted for readability):

```
<SparseTrack
name="ad_marker">PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPEFjcXVp
cmVkU2lnbmFsIHhtbG5zPSJodHRwOi8vd3d3LmNvbWNhc3QuY29tL3NjaGVtYXMvTkdPRC9TaWduY
WwvMjAxMC9SMVYwIiB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLW
luc3RhbmNlIiB4c2k6c2NoZW1hTG9jYXRpb249Imh0dHA6Ly93d3cuY29tY2FzdC5jb20vc2NoZW1
hcy9OR09EL1NpZ25hbC8yMDEwL1IxVjAgQ0MtTkdPRC1TSUdOQUxJTkctUjFWMC0xMDEyMTAueHNk
Ij4KCTxBY3F1aXNpdGlvblBvaW50SW5mbyBBY3F1aXNpdGlvblBvaW50SWRlbnRpdHk9IkUhQ01DU
0EzIiBBY3F1aXNpdGlvblNpZ25hbElEPSI1YjQ4ZjdmZi1hMTJlLTQ0ZWEtOGIxZC1iODA5OGZjZD
EwZmEiLz4KCTxVVENPZlN3aXRjaFBvaW50IFVUQ1BvaW50PSIyMDEyLTAyLTI5VDIxOjE5OjA5WiI
vPgoJPFN0cmVhbVRpbWVzPgoJCTxTdHJlYW1UaW1lIFRpbWVUeXBlPSJTbW9vdGgiIFRpbWVWYWx1
ZT0iMjM3NTk0NDAxMDg4OCIvPgoJCTxTdHJlYW1UaW1lIFRpbWVUeXBlPSJQVFMiIFRpbWVWYWx1Z
T0iNDIwMzYyNjkxNCIvPgoJPC9TdHJlYW1UaW1lcz4KPC9BY3F1aXJlZFNpZ25hbD4K
</SparseTrack>
```

7.7.7. SecurityMetadata (HSS)

Security data to provide instructions for encryption key rotations and/or data to be forwarded downstream.

Table 16 - Semantics of the SecurityMetadata element

Element or Attribute Name	Use	Description
`#text`	M	
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example (namespace declarations omitted for readability):

```
<SecurityMetadata>z4KPEFjcXVpcmVkU2lnbmFsIHhtbG5zPSJodHRwOi8vd3d3LmNvbWNhc3Qu
Y29tL3NjaGVtYXMvTkdPRC9TaWduYWwvMjAxMC9SMVYwIiB4bWxuczp4c2k9Imh0dHA6Ly93d3cud
zMub3JnLzIwMDEv</SecruityMetadata>
```

7.7.8. Event (DASH)

Insert a DASH MPD Event in an EventStream as described in SCTE 214-1, section 6.7.

Example (namespace declarations omitted for readability):

```
<dash:Event xmlns:dash="urn:mpeg:dash:schema:mpd:2011" presentationTime="0" duration="60"
id="0">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=</dash:Event>
```

7.7.9. Tracking

Provide a URL for submitting status updates. The SAS ***shall*** post an update at the completion of all instructions for the Media or MediaPoint. The SAS may post intermediate updates during processing of instructions where practical.

7.7.10. Template

A Template instruction is a container for one of the other instruction elements along with conditions on how it may be triggered.

When a Template is provided as part of a Media element, a MatchSignalStart shall be provided to define when the template instructions will take effect. When a Template is provided as a child of a MediaPoint element, the MediaPoint defines the timing.

For repeating instructions, such as repeated creation of a signal, the interval attribute shall define how often the instruction shall be repeated from its initial time along with a trigger to end the repeat cycle. The ending trigger could be a total duration as defined by the @end attribute, an absolute datetime defined by the @expires attribute or the occurrence of an in-band signal as defined by the MatchSignalEnd element. If more than one of these values is provided then the first condition met shall cause the repeat to end.

Table 17 - Semantics of the Template element

Element or Attribute Name	Use	Description
`@interval`	O	The regular interval for which the instruction *should* be repeated starting with the offset
`@end`	O	When repeat is given, this value it specifies the end duration.
`@expires`	O	An absolute date and time of when the context of the template shall be removed from scope.
`MatchSignalStart`	O	
`MatchSignalEnd`	O	
`MatchWatermarkStart`	O	
`MatchWatermarkEnd`	O	
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

7.8. AcquisitionSystem

An AcquisitionSystem element represents a unique instance of a Signal Acquisition System registered with the Signal Decision System. The AcquisitionSystem element *shall* be one of the following types:

Table 18 – AcquisitionSystem Types

Encoder	encoder or transcoder
Packager	packager

Switcher	linear stream switcher

Table 19 - Semantics of an AcquisitionSystem element

Element or Attribute Name	Use	Description
`@id`	M	This value *shall* identify the Acquisition System.
`Endpoint`	0..1	The URI where the SDS can POST instructions to
`pid`	0..N	SCTE 35 PID number configured for this stream on Encoder. Need to provide all SCTE 35 output PID’s for this stream during registration.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

Example **`AcquisitionSystem`** (namespace declarations omitted for readability):

```
<Encoder id=" esam/media/xyz/encoder/enc1">
    <Endpoint>http://10.10.10.10:1010/enc1<Endpoint/>
</Encoder>
```

7.9. Response

A *TemplateResponse* ***may*** be used to configure a Packager. A *TemplateResponse* ***shall*** be able to use macro substitutions in the *ManifestResponse* which instruct the packager about how to condition the manifests using the SCTE 35 message as a source. These macros are delimited by the '$' character in the notification response. Here is a list of the possible macros along with the origin of the data for that macro. If a specified macro element is not present in the SCTE 35 message, that element of the notification *should* be ignored.

Table 20 – Standard template parameters

acquisitionPointIdentity	Configured on the packager
acquisitionSignalID	Randomly generated per signal
segmentationEventId	Extracted from segmentation_event_id in the segmentation descriptor
segmentationTypeId	Extracted from segmentation_type_id in the segmentation descriptor
segmentationUpid	Extracted from segmentation_upid() name in the segmentation descriptor
duration	Extracted from segmentation_duration in the segmentation descriptor
hdsDuration	Duration expressed as fractional seconds
availNum	Extracted from splice insert command or segment num in the segmentation descriptor
availExpected	Extracted from splice insert command or segments expected in the segmentation descriptor

subSegmentNum	Extracted from the sub_segment_num in the segmentation descriptor
subSegmentsExpected	Extracted from the sub_segments_expected in the segmentation descriptor
utcPoint	Expected wall clock time of signal point in UTC timestamp (XML dateTime)
ptsTime	Expected PTS value of signal point
smoothTime	Expected Smooth timestamp of signal point
hdsTime	Expected signal point time expressed as fractional seconds
binarySignal	Base64 encoded representation of the SCTE 35 signal

7.10. Status

The Status element is provided wherever structured status information is needed.

Table 21 - Semantics of the Status element

Element or Attribute Name	Use	Description
`Error`	0..N	This element shall be present when an error occurred during processing. An error is any situation that prevents an instruction from being executed. The element shall always contain a text description of the error.
`Warning`	0..N	This element shall be present when a warning occurred during processing. A warning is any situation where the instruction was executed, but with special circumstances. The element shall always contain a text description of the warning.
`Information`	0..N	This element shall be present when additional information is needed to qualify the status. The element shall always contain a text description of the information.
Conventions used in the tables: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>..<maxOccurs>, where N=unbounded		

8. Service Interface

The signal processing services ***shall*** be exposed as a single HTTP RESTful endpoint. Each resource within the service is defined by a path consisting of keywords in this standard along with resource identifiers. When structured data payloads are defined, they ***shall*** be represented as either XML or JSON and the client ***shall*** specify its preference using the request message's HTTP Accept header, setting the value to "application/json" or "application/xml".

The server ***shall*** return a valid HTTP status code indicating the result of the message exchange and application processing. For application processing errors, the HTTP payload ***shall*** contain additional information describing the nature of the error.

System to system communications are considered to be in a trusted environment; therefore, security-related concerns are currently outside the scope of this specification. Further, systems *may* experience communications failures, and it is left to the implementer to devise the best messaging resiliency and timeout tactics to meet specific customer needs. It is suggested the implementer provide configurations around potential retries in the event synchronous notifications are not received, as well as configurations around what behavior would be expected in such cases. For example, one might provide a default permissive configuration in the event a notification is not received: the signaled event is allowed to pass. Another *may* take the same behavior but also retry multiple times on some interval. Another *may* take a more restrictive default approach where it is not allowed to pass if a timeout or ambiguous error occurs. All of these approaches are valid and the selection will depend on the needs and characteristics of the environment. It is therefore strongly suggested that implementations allow for default behavior to be configurable.

8.1. Time Synchronization

It is expected that time synchronization with multiple high accuracy sources ***shall*** be maintained by all components of the systems. Protocols such as NTP or PTP allow time synchronization in the sub-millisecond on LANs and up to a few milliseconds on WANs.

8.2. Configuration

The SAS ***shall*** be configured with an endpoint for each SDS with which it will communicate. Redundancy may be accomplished through DNS, load balancing or the SAS *may* be given multiple SDS endpoints. When the SAS is configured with multiple SDS endpoints, a policy ***shall*** also be provided for timing of when to contact each endpoint and how to handle redundancy.

8.3. Discovery

An SDS ***shall*** provide some useful documentation at the base endpoint, which includes the various resources supported by the service. The documentation *may* be provided in any format, but HTML is recommended.

```
GET http://sds
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>ESAM Service</title>
    </head>
    <body>
        <h1>ESAM Service</h1>
        <h2>Resources</h2>
        <ul>
            <li><a href="media">Media</a></li>
        </ul>
    </body>
</html>
```

An SDS ***shall*** at minimum, provide a Media resource, which when retrieved provides a list of linear streams that the SDS can support (i.e. provide instructions to an SAS). Note that an SDS *may* also support VOD assets, but these are not listed nor individually eligible for registration.

Example (namespace declarations omitted for readability):

```
GET http://sds/media
```

```
<Response>
    <Media id="media/xyz" description="XYZ Linear Network"/>
    <Media id="media/denhd" description="Denver HD Linear Network"/>
</Response>
```

Within each individual media resource, the SDS ***shall*** implicitly include all of the SAS device types that it can support.

Table 22 – SAS Types

Encoder	encoder or transcoder
Packager	packager
Switcher	linear stream switcher

Requesting a specific media linear stream from the SDS ***shall*** provide a listing of all of the registered SAS components for that stream.

Example (namespace declarations omitted for readability):

```
GET http://sds/media/xyz
```

```
<Media id="media/xyz" description="XYZ Linear Network">
    <Encoder id="enc1"/>
    <Encoder id="enc2"/>
    <Packager id="pkg1"/>
    <Packager id="pkg2"/>
    <Switcher id="lss1"/>
    <Switcher id="lss2"/>
</Media>
```

8.4. Registration

An SDS ***shall*** provide the ability for an SAS to register itself as an acquisition system for linear streams. Upon discovering the Media resources for which an SDS can make decisions, the SAS ***shall*** make itself known to the SDS by PUTting its own representation (i.e. endpoint). When a previously registered SAS is removed from service, it should remove itself by DELETEing its identity from the SDS.

8.4.1. Use Case: Register a new SAS

The following example registers a new encoder to receive instructions for linear network XYZ. This same request could be used to update the endpoint if the registration already exists. This request would have no effect if the registration had previously existed and the endpoint was unchanged. An organization ***shall*** be responsible to maintain the uniqueness of SAS identifiers. The SDS *should* respond with an instruction for linear network XYZ otherwise just empty media in the response.

If MatchTime contains past time (which means the event is past) then execute instruction immediately.

Pid is optional and can be used during Encoder registration. This contains a numerical value which is a pid number. All SCTE 35 output pid numbers can be defined here so that SDS will use it in the future.

Example (namespace declarations omitted for readability):

```
PUT http://sds/media/xyz/enc/enc1
```

```
<Encoder id="enc1">
    <Endpoint>http://enc1/media/xyz</Endpoint>
    <pid>111</pid>
</Encoder>
```

Response (with instruction):

```
<Media id="media/xyz" description="XYZ Linear Network">
    <MediaPoint>
```

```
        <MatchTime>2021-08-30T10:00:00Z</MatchTime>
        <Content offset=”PT0S” zone=”zone1”>SLATE_ID1</Content>
        <Tracking>http://sds/media/xyz/enc/enc1/signal/1483023A3992B</Tracking>
    </MediaPoint>
</Media>
```

Response (without instruction):

```
<Media id="media/xyz" description="XYZ Linear Network">
</Media>
```

The following example registers a new packager to receive instructions for linear network XYZ

```
PUT http://sds/media/xyz/pkg/pkg1
```

```
<Packager id="pkg1">
    <Endpoint>http://pkg1/media/xyz</Endpoint>
</Packager>
```

Response (with instruction):

```
<Media description="XYZ Linear Network" id="media/xyz">
  <MediaPoint>
    <MatchTime>2021-08-30T10:00:00Z</MatchTime>
    <DateRange id=”1001001” class=”Program” plannedDuration=”7200.00”
scte35Cmd=”/DAwAAAAAAAA///wBQb+AAAAAAAaAhhDVUVJQAAAAn+HAQkxMTAwMDAwMzUQAABFh8Zt” />
    <dash:Event duration="7200" id="1001001" presentationTime="0"
xmlns:dash="urn:mpeg:dash:schema:mpd:2011">/DAwAAAAAAAA///wBQb+AAAAAAAaAhhDVUVJQAAAAn+HAQkxMTAw
MDAwMzUQAABFh8Zt</dash:Event>
    <SparseTrack name="Program">…</SparseTrack>
    <SecurityMetadata>z4KP…</SecurityMetadata>
  </MediaPoint>
</Media>
```

The following example registers a new switcher to receive instructions for linear network XYZ
Example (namespace declarations omitted for readability):

```
PUT http://sds/media/xyz/acd/acd1
```

```
<Switcher id="acd1">
    <Endpoint>http://acd1/media/xyz/acd/</Endpoint>
</Switcher>
```

Response:

```
<Media description=”ESPN Linear Network” id=“media/ESPN”>
</Media>
```

8.4.2. Use Case: Deregister an existing SAS

An existing registration *may* be removed or deregistered by simply sending a DELETE verb with the HTTP request.

```
DELETE http://sds/media/xyz/enc/enc1
```

8.5. Request Instructions

An SAS ***shall*** request instructions from an SDS by getting the instruction resource. The instruction resource ***shall*** be composed of the following components:

[base-endpoint]/media/[media-identifier]/[SAS-type]/[SAS-identifier]/instruction

The request *may* include the following optional parameters:

Table 23 – Request parameters

signal	A base-64 encoded version of the SCTE-35 signal for which instructions are being requested. The SDS ***shall*** return all known active signals when this parameter is omitted. When the signal is provided within a URL, it shall be either URL encoded or Base64url encoded.
pid	When specified, indicates the MPEG-2 Transport Stream PID number on which the associated signal was found. Because this value is associated with an in-band signal, it should only be provided along with the signal parameter.
pts	The SAS *should* interpret the target PTS time from the SCTE-35 signal and pass it as a separate parameter.
start	Constrains the response to MediaPoints, which were encountered on or after the date/time provided. The format of the date/time ***shall*** follow ISO 8601.
zone	The zone identity for which instructions should apply.
expand	A Boolean expression, which when assigned any non-empty value shall be taken to mean that the SDS ***shall*** expand all SCTE-35 signals using the corresponding XML Schema. This feature may be ***should*** be used primarily for testing and diagnostic purposes.
watermark	A base-64 encoded version of the watermark for which instructions are being requested.
contentType	Stream type which can be HLS or Dash or HSS
zip	5 digit zip code which is geo location of the device
deviceType	The device model which can be PHONE or TABLET or PC etc.

8.5.1. Use Case: SpliceInsert to signal opportunity

A Content provider inserts an SCTE 35 splice insert (splice_command=0x05) marker into the stream to signal a placement opportunity for a content distributor to replace a national ad with a local ad. The SAS extracts the signal and includes it in a request to the SDS. The SDS confirms the signal validity and responds to the SAS with instructions to condition the stream.

Example (namespace declarations omitted for readability):

```
GET http://sds/media/xyz/enc/enc1/instruction?signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl
```

```
<Media id="media/xyz" description="XYZ Linear Network">
    <MediaPoint>
        <ReferenceSignal
remove="true">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=</ReferenceS
ignal>
        <Signal
offset="PT0.000S">/DBLAAAAAAAAA///wBQb+AAAAAAA1AjNDVUVJYgAFin//AABSZcAJH1NJR05BTDpEUjIxWjA3WlQ4
YThhc25pdVVoZWlBPT00AADz3GdX</Signal>
        <Signal
offset="PT1M0.000000011111S">/DBEAAAAAAAAA///wBQb+AFJlwAAuAixDVUVJYgAFin+/CR1TSUdOQUw6My1zUTROZ
0ZUME9qUHNHNFdxVVFvdzUAAEukzlg=</Signal>
        <Condition offset="PT0.000S" direction="OUT"/>
        <Condition offset="PT1M0.000S" direction="IN"/>
        <Tracking>http://sds/media/xyz/enc/enc1/signal/1483023A3992B</Tracking>
    </MediaPoint>
</Media>
```

8.5.2. Use Case: TimeSignal with SegmentationDescriptor to signal opportunity

A content provider inserts an SCTE 35 time signal (splice_command=0x06) marker into the stream to signal a placement opportunity for a content distributor to replace a national ad with a local ad. The SAS extracts the signal and includes it in a request to the SDS. The SDS confirms the validity of the signal and responds to the SAS with instructions to condition the stream.

Example (namespace declarations omitted for readability):

```
GET http://sds/media/xyz/enc/enc1/instruction?signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl
```

```
<Media id="media/xyz" description="XYZ Linear Network">
    <MediaPoint>
        <ReferenceSignal
remove="true">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=</ReferenceS
ignal>
        <Signal
offset="PT0.000S">/DBLAAAAAAAAA///wBQb+AAAAAAA1AjNDVUVJYgAFin//AABSZcAJH1NJR05BTDpEUjIxWjA3WlQ4
YThhc25pdVVoZWlBPT00AADz3GdX</Signal>
        <Signal
offset="PT1M0.000000011111S">/DBEAAAAAAAAA///wBQb+AFJlwAAuAixDVUVJYgAFin+/CR1TSUdOQUw6My1zUTROZ
0ZUME9qUHNHNFdxVVFvdzUAAEukzlg=</Signal>
        <Condition offset="PT0.000S" direction="OUT"/>
        <Condition offset="PT1M0.000S" direction="IN"/>
        <Tracking>http://sds/media/xyz/enc/enc1/signal/1483023A3992B</Tracking>
    </MediaPoint>
</Media>
```

8.5.3. Use Case: TimeSignal with SegmentationDescriptor to signal blackout

A content provider inserts an SCTE 35 time signal (splice_command=0x06) marker into the stream to signal the beginning of a region that *may* be subject to restrictions (e.g. blackout). The SAS extracts the signal and includes it in a request to the SDS. The SDS confirms the validity of the signal and responds to the SAS with instructions to condition the stream and to repeat the signal for the duration of the restriction.

Example (namespace declarations omitted for readability):

```
GET http://sds/media/xyz/enc/enc1/instruction?signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl
```

```
<Media id="media/xyz" description="XYZ Linear Network">
    <MediaPoint>
        <ReferenceSignal
remove="true">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=</ReferenceS
ignal>
        <Signal
offset="PT0.000S">/DBLAAAAAAAA///wBQb+AAAAAAA1AjNDVUVJYgAFin//AABSZcAJH1NJR05BTDpEUjIxWjA3WlQ4
YThhc25pdVVoZWlBPT00AADz3GdX</Signal>
        <Signal
offset="PT1M0.000000011111S">/DBEAAAAAAAA///wBQb+AFJlwAAuAixDVUVJYgAFin+/CR1TSUdOQUw6My1zUTROZ
0ZUME9qUHNHNFdxVVFvdzUAAEukzlg=</Signal>
        <Signal offset="PT5.000S" interval="PT5.000S"
end="PT50.000S">/DBEAAAAAAAA///wBQb+AFJlwAAuAixDVUVJYgAFin+/CR1TSUdOQUw6My1zUTROZ0ZUME9qUHNHNF
dxVVFvdzUAAEukzlg=</Signal>
        <Condition offset="PT0.000S" direction="OUT"/>
        <Condition offset="PT1M0.000S" direction="IN"/>
        <Tracking>http://sds/media/xyz/enc/enc1/signal/1483023A3992B</Tracking>
    </MediaPoint>
</Media>
```

8.5.4. Use Case: Normalize signal format

In order to provide a uniform signal to the downstream application, the SPS *may* be used to replace (or modify elements of) the existing signal sent in the request event.

8.5.5. Use Case: Out-of-band notification of blackout

A content provider notifies the content distributor of a content restriction (e.g. blackout) without an SCTE 35 in-band signal using a suitable out-of-band method (e.g. SCTE 224). The SDS sends a set of instructions directly to the SAS.

Example (namespace declarations omitted for readability):

```
<Media description="XYZ Linear Network" id="media/xyz">
  <MediaPoint>
    <MatchTime>2021-08-30T10:00:00Z</MatchTime>
    <Signal>/DAwAAAAAAAA///wBQb+AAAAAAAaAhhDVUVJQAAAAn+HAQkxMTAwMDAwMzUQAABFh8Zt</Signal>
    <Condition offset="PT0.000S" direction="OUT"/>
    <Tracking>http://sds/media/xyz/enc/enc1/signal/1483023A3992B</Tracking>
  </MediaPoint>
</Media>
```

8.5.6. Use Case: Regions defined using signal metadata

A content provider notifies the content distributor of regions of interest (e.g. ad insertion opportunities, chapters) for a content asset. The SPS ***shall*** provide instructions to insert in-band signals and condition the stream for all of the applicable points in the asset. This message exchange *may* be initiated by the SAS that is processing the asset (via request to the SDS) or by the SDS by notifying the SAS.

8.5.7. Use Case: Packager conditioning for ad insertion

The packager discovers a signal in the stream and includes it in a request to the SDS. The SDS responds with instructions to insert HLS tag, Dash Event, sparse track and security metadata into the respective manifest along with the payload. If the request doesn't have a parameter contentType then the response instruction contains all content type tags otherwise instruction contains only tags specific to the content type.

Example without contentType(namespace declarations omitted for readability):

```
GET
http://esam/media/xyz/packager/pkg1/instruction?signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl
```

```
<Media description="XYZ Linear Network" id="media/xyz">
  <MediaPoint>
    <ReferenceSignal duration="PT1M0S"
remove="true">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=</ReferenceS
ignal>
    <DateRange id=”2044334492” class=”Distributor opportunity” duration=”60”
scte35Cmd=”/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=”/>
    <dash:Event duration="60" id="0" presentationTime="0"
xmlns:dash="urn:mpeg:dash:schema:mpd:2011">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAh
DVUVJADgyMWLvc/g=</dash:Event>
    <SparseTrack name="ad_marker">…</SparseTrack>
    <SecurityMetadata>z4KP…</SecurityMetadata>
  </MediaPoint>
</Media>
```

Example with contentType(namespace declarations omitted for readability):

```
GET
http://esam/media/xyz/packager/pkg1/instruction?signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl&c
onteType=dash
```

```
<Media description="XYZ Linear Network" id="media/xyz">
  <MediaPoint>
    <ReferenceSignal duration="PT1M0S"
remove="true">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAhDVUVJADgyMWLvc/g=</ReferenceS
ignal>
    <dash:Event duration="60" id="0" presentationTime="0"
xmlns:dash="urn:mpeg:dash:schema:mpd:2011">/DAvAAAAAAAAAP///wViAAWKf+//CXVCAv4AUmXAAzUAAAAKAAh
DVUVJADgyMWLvc/g=</dash:Event>
  </MediaPoint>
</Media>
```

8.5.8. Use Case: Packager conditioning for blackout

The packager discovers a signal in the stream and includes it in a request to the SDS. The SDS responds with instructions to insert HLS tag, Dash Event, sparse track and security metadata into the respective manifest along with the payload.

Example (namespace declarations omitted for readability):

```
GET
http://esam/media/xyz/packager/pkg1/instruction?signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl
```

```
<Media description="XYZ Linear Network" id="media/xyz">
  <MediaPoint>
    <ReferenceSignal
remove="true">/DAwAAAAAAAA///wBQb+AAAAAAAaAhhDVUVJQAAAAn+HAQkxMTAwMDAwMzUQAABFh8Zt</ReferenceSi
gnal>
    <DateRange id=”1001001” class=”Program” plannedDuration=”7200.00”
scte35Cmd=”/DAwAAAAAAAA///wBQb+AAAAAAAaAhhDVUVJQAAAAn+HAQkxMTAwMDAwMzUQAABFh8Zt”/>
    <dash:Event duration="7200" id="1001001" presentationTime="0"
xmlns:dash="urn:mpeg:dash:schema:mpd:2011">/DAwAAAAAAAA///wBQb+AAAAAAAaAhhDVUVJQAAAAn+HAQkxMTAw
MDAwMzUQAABFh8Zt</dash:Event>
    <SparseTrack name="Program">…</SparseTrack>
    <SecurityMetadata>z4KP…</SecurityMetadata>
  </MediaPoint>
</Media>
```

8.5.9. Use Case: Packager conditioning when no ad insertion or blackout

The packager discovers a signal in the stream and includes it in a request to the SDS. The SDS responds with no instructions.

Example (namespace declarations omitted for readability):

```
GET
http://esam/media/xyz/packager/pkg1/instruction?signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl
```

```
<Media description="XYZ Linear Network" id="media/xyz">
  <MediaPoint>
    <ReferenceSignal remove="true">/DAwAAAAAAAA///wBQb+AAAAAAAaAhhDVUVJQAAAAn+HAQkxMTAwMDAwMzUQAABFh8Zt</Refere
  </MediaPoint>
</Media>
```

8.5.10. Use Case: Support VOD processing

The encoder *may* request all instructions for a VOD asset. The SDS responds with instructions for all of the points within the asset.

Example (namespace declarations omitted for readability):

```
GET http://sds/media/xyz.com/ASST0000000000000000/enc/enc1/instruction
```

```
<Media id="media/xyz.com/ASST0000000000000000" description="A VOD Asset" batch="17671c91-9178-
42ca-bfca-e5ad977d499f">
    <Signal
offset="P0Y0M0DT0H0M0.000S">/DBLAAAAAAAAA///wBQb+AAAAAAA1AjNDVUVJAAAAAX//AAATPnIJH1NJR05BTDphZ1
dVMy9MMVRxYWFRVTVyOGRwMStBPT00AAAxB+aV</Signal>
    <Signal
offset="P0Y0M0DT0H0M14.013S">/DBGAAAAAAAAA///wBQb+AAAAAAAwAi5DVUVJAAAAAX+/CR9TSUdOQUw6YWdXVTMvT
DFUcWFhUVU1cjhkcDErQT09NQAAObhR8g==</Signal>
    <Signal
offset="P0Y0M0DT0H0M14.013S">/DBLAAAAAAAAA///wBQb+AAAAAAA1AjNDVUVJAAAAAn//AABnGY4JH1NJR05BTDpua
0ZERXN6b1JTKzNiUldubDY2dWJnPT00AAAHeJqd</Signal>
    <Signal
offset="P0Y0M0DT0H1M29.088S">/DBGAAAAAAAAA///wBQb+AAAAAAAwAi5DVUVJAAAAAn+/CR9TSUdOQUw6bmtGREVze
m9SUyszYlJXbmw2NnViZz09NQAALsRkRA==</Signal>
    <Signal
offset="P0Y0M0DT0H10M31.129S">/DBLAAAAAAAAA///wBQb+AAAAAAA1AjNDVUVJAAAAA3//AAApPWwJH1NJR05BTDpl
WlUvTlFrZFJ3VytjTmR1Uzh4M1NRPT00AAA9Mycl</Signal>
    <Signal
offset="P0Y0M0DT0H11M01.159S">/DBGAAAAAAAAA///wBQb+AAAAAAAwAi5DVUVJAAAAA3+/CR9TSUdOQUw6ZVpVL05R
a2RSd1crY05kdVM4eDNTUT09NQAAcLXWPA==</Signal>
    <Signal
offset="P0Y0M0DT0H16M18.476S">/DBLAAAAAAAAA///wBQb+AAAAAAA1AjNDVUVJAAAABH//AAAwHbIJH1NJR05BTDp1
blFyR0hFY1R5MkpYYlduUCtLZnZnPT00AAD6gKwZ</Signal>
    <Signal
```

```
offset="P0Y0M0DT0H16M53.513S">/DBGAAAAAAAA///wBQb+AAAAAAAwAi5DVUVJAAAABH+/CR9TSUdOQUw6dW5RckdI
RWNUeTJKWGJXblArS2Z2Zz09NQAA7ztIVQ==</Signal>
    <Signal
offset="P0Y0M0DT0H23M34.913S">/DBLAAAAAAAA///wBQb+AAAAAAA1AjNDVUVJAAAABX//AAABX+oJH1NJR05BTDpG
dFgxZkVjTlJJRzdBcDI3QkNiWkpnPT00AAAkibx4</Signal>
    <Signal
offset="P0Y0M0DT0H23M35.911S">/DBGAAAAAAAA///wBQb+AAAAAAAwAi5DVUVJAAAABX+/CR9TSUdOQUw6RnRYMWZF
Y05SSUc3QXAyN0JDYlpKZz09NQAAjxVpoA==</Signal>
    <Condition offset="P0Y0M0DT0H0M0.000S" direction="OUT"/>
    <Condition offset="P0Y0M0DT0H0M14.013S" direction="IN"/>
    <Condition offset="P0Y0M0DT0H0M14.013S" direction="OUT"/>
    <Condition offset="P0Y0M0DT0H1M29.088S" direction="IN"/>
    <Condition offset="P0Y0M0DT0H10M31.129S" direction="OUT"/>
    <Condition offset="P0Y0M0DT0H11M01.159S" direction="IN"/>
    <Condition offset="P0Y0M0DT0H16M18.476S" direction="OUT"/>
    <Condition offset="P0Y0M0DT0H16M53.513S" direction="IN"/>
    <Condition offset="P0Y0M0DT0H23M34.913S" direction="OUT"/>
    <Condition offset="P0Y0M0DT0H23M35.911S" direction="IN"/>
</Media>
```

8.5.11. Use Case: Support Alternate Content Decisioning through Manifest Manipulator

The switcher *may* request all instructions for the alternate content decisioning. The SDS responds with instructions to switch to alternate content URL or stay tune to live.

Example with Slate URL(namespace declarations omitted for readability):

```
GET http://sds/media/xyz/acd/acd1/instruction? signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl
&contentType=hls&zip=30266&deviceType=PHONE&no_regional_blackout_flag=0&device_restrictions=3&web
_delivery_allowed_flag=0&session=dde112a7-ba25-48d8-932c-fe0db25c387b
```

```
<Media description=”ESPN Linear Network” id=“media/ESPN”>
             <MediaPoint>
               <ReferenceSignal
remove=“true”>U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl</ReferenceSignal>
               <Content>http://cdn.com/hls/ESPN/SLATE_STREAM/index.m3u8</Content>
               <Tracking>http://sds/media/ESPN/enc/enc1/signal/1483023A3992B</Tracking>
             </MediaPoint>
</Media>
```

Example with Stay Tuned(namespace declarations omitted for readability):

```
GET http://sds/media/xyz/acd/acd1/instruction? signal=U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl
&contentTypel=dash&zip=80112&deviceType=PHONE&no_regional_blackout_flag=0&device_restrictions=3&w
eb_delivery_allowed_flag=0&session=dde112a7-ba25-48d8-932c-fe0db25c387b
```

```
<Media description=“ESPN Linear Network” id=“media/ESPN”>
             <MediaPoint>
               <ReferenceSignal
remove=“true”>U29tZSBiYXNlNjQgZW5jb2RlZCBzY3RlMzUgY3Vl</ReferenceSignal>
             </MediaPoint>
</Media>
```

8.6. Providing Status

The SDS *should* provide a Tracking element within each Media and MediaPoint element. The content of this element ***shall*** specify an endpoint for sending status information from the SAS. The SAS ***shall*** POST an empty body to the resource specified by the endpoint to indicate successful application of the associated instructions. It is important that the POST body be completely empty and not contain an innocuous message confirming that the status was a success. When an exception occurs in processing the instructions, the body of the POST shall contain a message describing the exception with pertinent details.

There is no explicit format for the body and is left to the implementer to decide an appropriate type. The exception could be described as XML, JSON or just plain text.

8.7. Service Check

The SAS may periodically check availability of the SDS by performing a GET request to its corresponding resource that was created upon registration. An explicit endpoint for service check shall not be established. The SAS may establish any reasonable service check interval.