OpenCable™ Specifications
Alternate Content

Event Scheduling and Notification Interface

OC-SP-ESNI-I02-131001

ISSUED

Notice

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-ESNI-I02-131001			
Document Title:	Event Scheduling and Notification Interface			
Revision History:	I01 – Released 11/16/12 I02 – Released 10/1/13			
Date:	October 1, 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

This page left blank intentionally.

1 SCOPE

1.1 Introduction and Purpose

Video distribution is transitioning from Quadrature Amplitude Modulation (QAM) to delivery over Internet Protocol (IP) networks. These IP networks are delivering video content to subscribers via many IP-connected devices, including mobile phones, tablets and game consoles. Video content includes both Video on Demand (VOD) and linear, live content. Traditionally, many systems were developed to enable programmers to inform and affect the content delivered to subscribers. For example, during a regional sports blackout, programmers may use an Integrated Receiver and Decoder (IRD) to provide alternate content to a select set of subscribers constrained by a headend servicing their unique geographical area. As we migrate to IP-delivered content, systems must be created to replicate the functional systems in order to create a contiguous service capability between traditional and IP video delivery.

This document defines a functional method for allowing programmers to notify cable operators of upcoming regional blackouts, which will require a subscriber's device to access alternate content during a specific time frame. The Event Scheduling and Notification Interface (ESNI) is a web interface facilitating the notification of these events in a similar way they traditionally used an IRD. Instead of, for example, traditional manual change of the content via the IRD, the cable operator will be notified through an in-band MPEG2-TS signal, SCTE-35, and use programmer-provided information via the ESNI, such as time frame, alternate content, and geographical location to change the content the subscriber is watching.

This interface is part of the overall alternate content cable operator ecosystem, which comprises an Event Signaling and Management (ESAM) API [ESAM] and the ESNI. This ecosystem is represented in Figure 1. The ESAM API uses in-band signaling received via SCTE-35 in the MPEG2-TS or by inserting one via an out-of-band method. This SCTE-35 signal [SCTE 35] contains information regarding the blackout, which the ESAM API communicates to the Alternate Event Service (AES). The AES receives additional information about the blackout from the Programmer via the ESNI. This interface provides additional information, such as time frame, alternate content, and location.

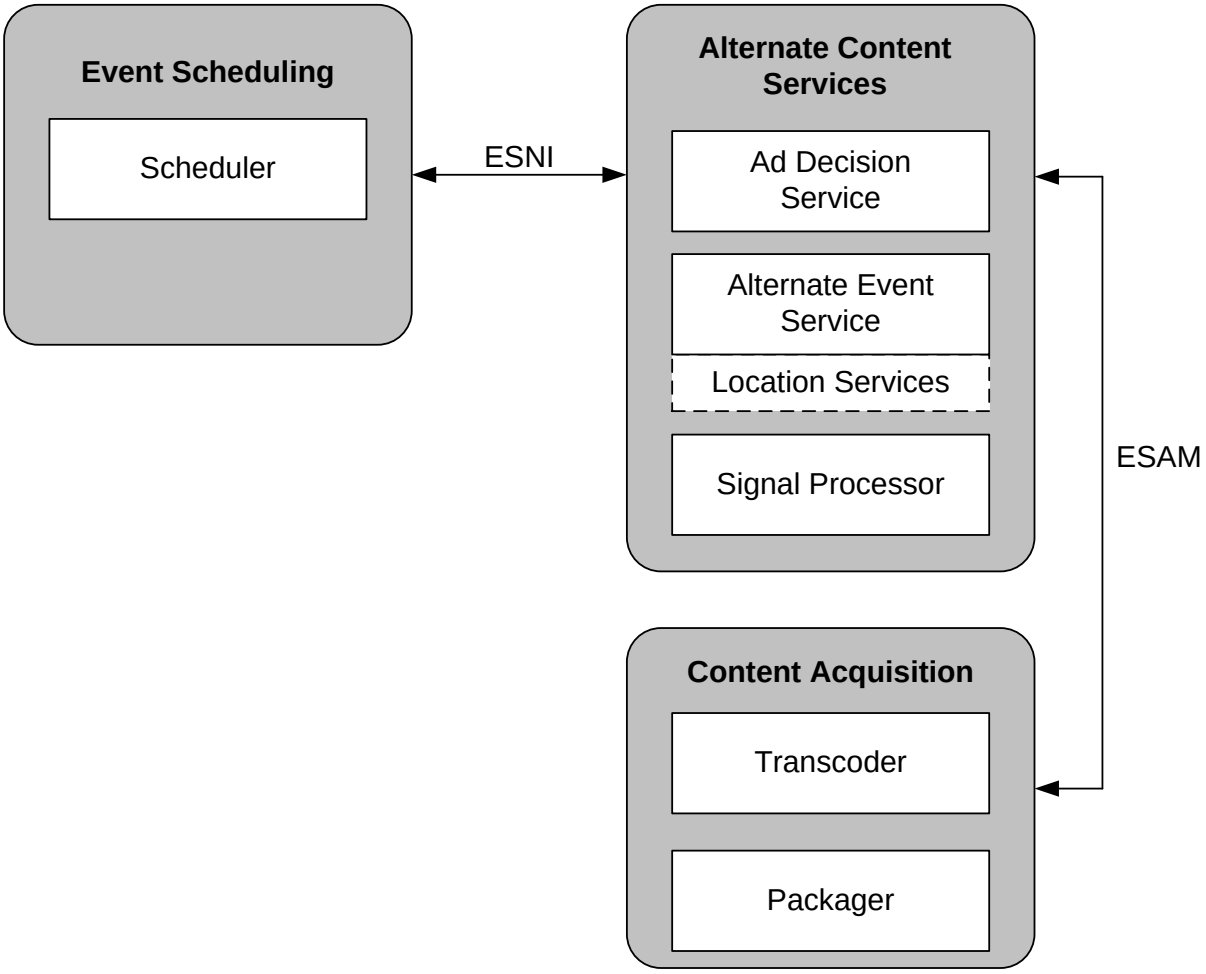

Figure 1 - Cable Alternate Content Ecosystem

Table 1 - Alternate Content Element Descriptions

Alternate Content Elements	Description
Scheduler	Maintains the metadata associated with alternate content events, such as program information, duration, and affected geo-location
Ad Decision Service	Verifies Ad placement timing and consults other systems in order to respond with the correct ad
Alternate Event Service	Verifies alternate content timing and programmer-provided metadata received via the ESNI
Signal Processor	Receives ESAM messaging and acts as an intermediary between the appropriate decision service and content acquisition systems
Transcoder	Sends and receives ESAM-related data and affects the video stream, if necessary, for downstream processing by the Packager
Packager	Sends and receives ESAM-related data for modifying client video source information files, such as an HLS manifest or Smooth sparse track

1.2 Requirements

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

"SHALL"	This word means that the item is an absolute requirement of this specification.
"SHALL 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.

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

[RFC 2119] IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. March 1997.

[RFC 2616] IETF RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June 1999.

2.2 Informative References

This specification uses the following informative references.

[ESAM] Real-time Event Signaling and Management API, OC-SP-ESAM-API-I02-131001, October 1, 2013, Cable Television Laboratories, Inc.

[SCTE 35] ANSI/SCTE 35 2012, Digital Program Insertion Cueing Message for Cable.

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
- Internet Engineering Task Force (IETF), Internet: http://www.ietf.org/
- ISO, www.iso.org
- SCTE, Society of Telecommunication Engineers, http://www.scte.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Alternate Event Service	Receives content via the ESNI and verifies alternate content timing and programmer-provided metadata.
Scheduler	Also referred to as the Alternate Content Scheduling System, it maintains the metadata associated with alternate content events, such as program information, duration, and affected geo-location.
Virtual IRD	A logical grouping of subscribers loosely associated with their geographic location and IRD.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

ABR	Adaptive Bitrate
AES	Alternate Event Service
EIDR	Entertainment Identifier Registry
ESAM	Event Signaling and Management
HTTP	Hypertext Transfer (or Transport) Protocol
IRD	Integrated Receiver/Decoder
QAM	Quadrature Amplitude Modulation
TMS	Tribune Media Services
UTC	Coordinated Universal Time
VIRD	Virtual IRD
XML	Extensible Markup Language

5 ARCHITECTURE

There are two architectures that support varying levels of functionality with regards to the Event Scheduling and Notification Interface. The essential difference between the architectures is the granularity of geographical association of the subscriber to a blackout area. In the first architecture, the geographical area is associated with the traditional IRD and headend association of the subscriber. In the second architecture, the use of a location services function allows the subscriber's device to communicate its geo-location, which is correlated to the blackout-affected area. The architecture implementation requirement may vary based on programmer. At least one of the architectures SHALL be implemented in order to support the ESNI.

5.1 Virtual IRD Architecture

The first is the Virtual IRD (VIRD) proposal that ties customers to a fixed location and virtual IRD. The VIRD architecture requires an authorization and registration process to assign a geographic serving area to a particular VIRD. The setup follows the current methodology that is used today for QAM-delivered content. The largest drawback of this approach would make roaming video use difficult or impossible. The architecture is very useful for serving in-home content only.

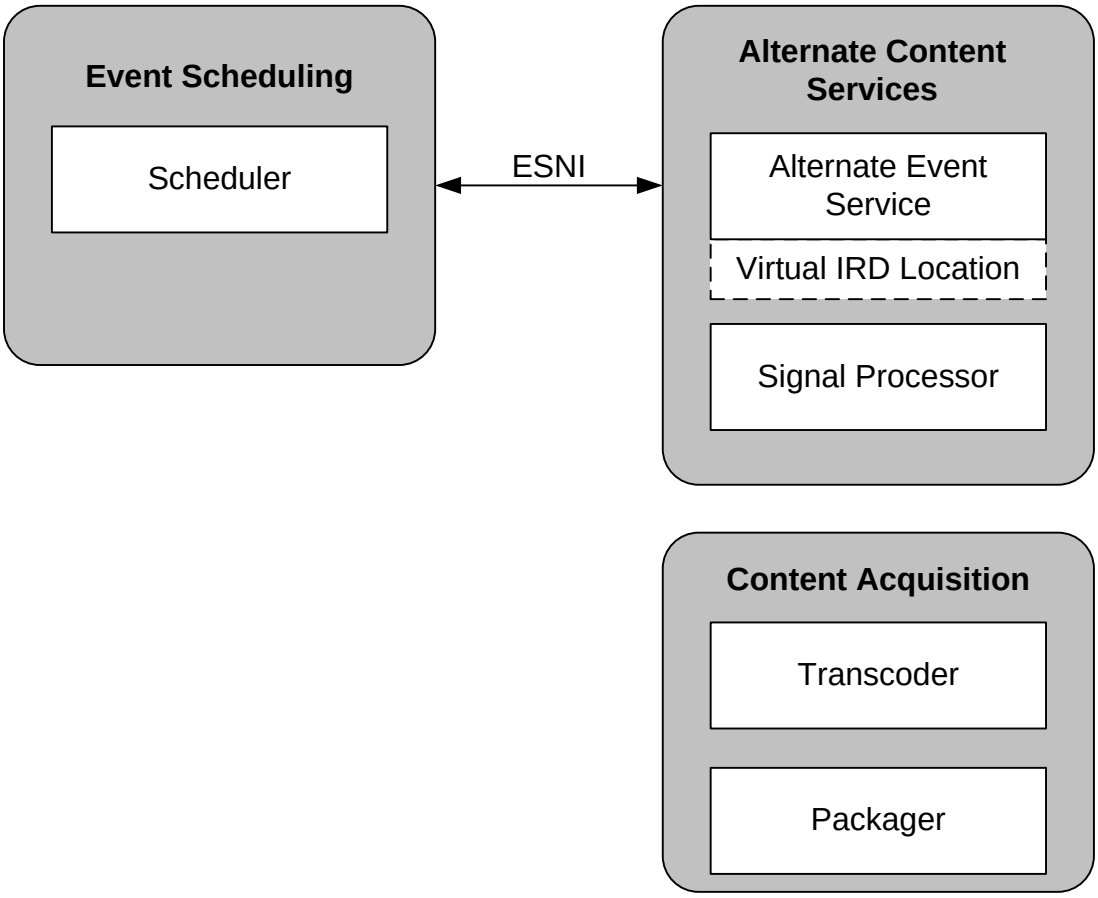

Figure 2 - Virtual IRD Architecture

5.2 Geo-Location Architecture

The second architecture uses the subscriber's device geo-location to determine the associated blackout policy. It would use a very detailed scheduling interface to communicate blackout regions/zones, start/stop times, etc. The

advantage of this method is that it would allow for roaming and more accurately expressing the proper event for customers outside the home. Also, this architecture would be much more reliant on a service to determine customer location during content playback time.

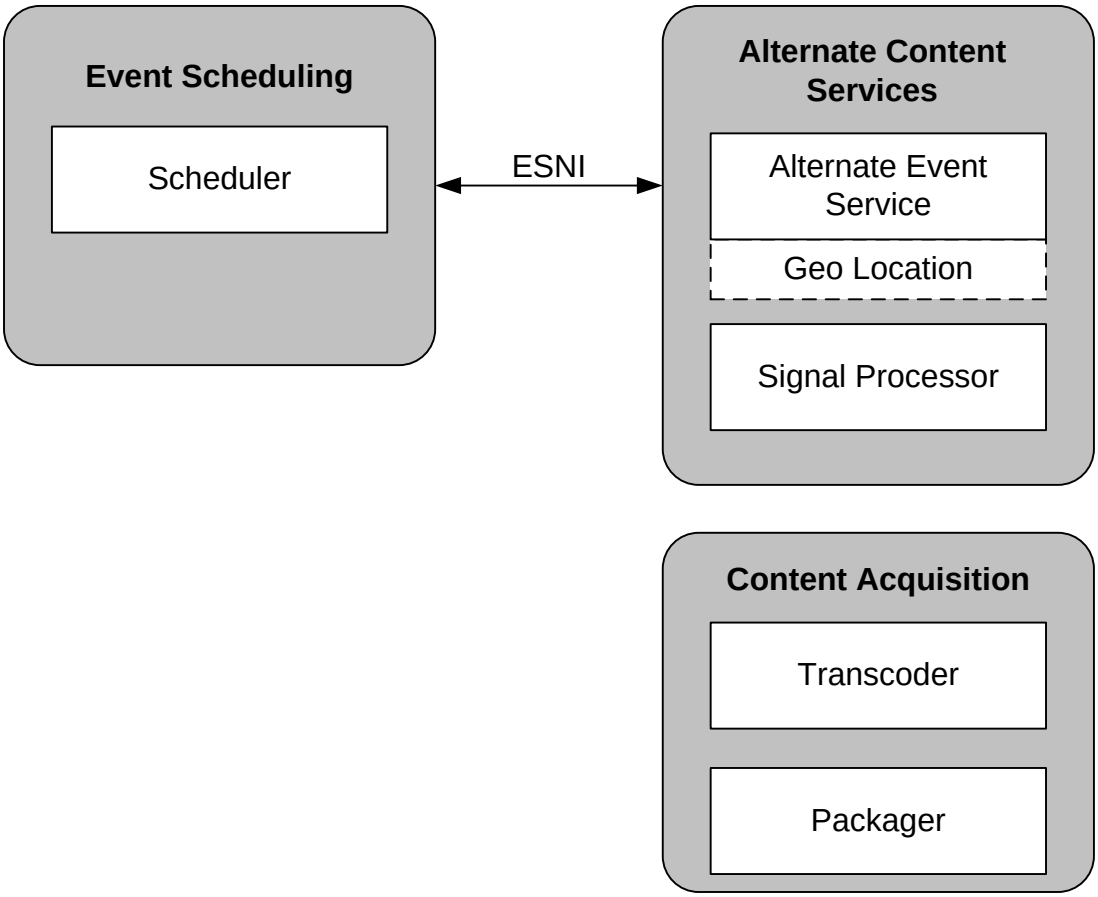

Figure 3 - Alternate Architecture Allowing Roaming

6 USE CASES AND REQUIREMENTS

6.1 Alternate Event

The current requirement is to be able to provide alternate event content for instances when viewers are permitted to view certain events in their area. The use of this system could cover blackouts and other programming change needs.

6.2 Alternate Event Requirements

- Timing for alternate events SHALL be synchronized in some way such that local player clock drift or changes to the local clock do not influence when the player performs a blackout.
- For blackouts without alternate content, the player SHOULD be given sufficient information to provide a video display to the user without having to tune to an alternate stream (e.g., if the blackout event should result in a black screen with some basic text, the player should be directed to create this screen rather than tune to a stream which has the black screen and text pre-encoded).

7 ALTERNATE EVENT SCHEDULING INTERFACE

This interface is used to provide blackout-scheduling information in advance of an actual blackout event. For information regarding the schema for this interface, see Annex A.

7.1 HTTP Interface Operation

7.1.1 Virtual IRD (VIRD)

Request Direction: Event Scheduler to Alternate Event Service (AES)

Method: HTTP GET

Description: The Event Scheduler issues a request to the Alternate Event Service to discover existing pre-defined locations (Virtual IRDs).

Location: http://<AES_IP>:<AES_PORT>/altcontentsvc/virds

Response: *Element:* **vIRDs**

Children	Use	Data type	Description
vIRD	1...unbounded	VIRD	

Element: **vIRD**

Children	Use	Data type	Description
locationId	1…unbounded	LocationID	A location identifier expressed as a string. Contains a mandatory "type" attribute with one of the following: • VIRD • ZIP Code • ZIP Code + 4 • Mileage • Latitude/Longitude • Any value beginning with "private" It is expected that the first locationId will define the enclosing virtual IRD and subsequent locationIds will provide geographic locations that make up the virtual IRD.

7.1.2 Scheduling an Event (ScheduleCreate)

Request Direction: Event Scheduler to Alternate Event Service

Method: HTTP POST

Description: The Event Scheduler issues a ScheduleCreate to the Alternate Event Service to inform the AES of a newly scheduled event.

When scheduling a blackout event that does not have alternate content, the following attributes SHALL be set as follows:

- restriction/alternateContent.altContentType of FORCE_DESCRIPTION
- restriction/alternateContent.blacklist set to TRUE
- restriction/@id set to 0 (this indicates that the blackout event has no locational restrictions and all locations are subject to the blackout event)
- restriction description should contain a textual description of the event that will be displayed to the user during the blackout event

Location: http://<AES_IP>:<AES_PORT>/altcontentsvc/events

Request: *Element:* **event**

The event element is of type Event (defined below).

startSignalId and endSignalId are optional IDs that uniquely identify the start and end signals sent by the provider and can be used to match up the signal throughout the alternate content system.

ComplexType: **Event**

Children	Use	Data type	Description
type (attribute)	Required	xs:token	Restricted string. Currently the only defined value is ALT_CONTENT.
id (attribute)	Required	xs:token	A unique event ID assigned to this scheduled event. This ID cannot be reused for the life of this event as it is used to delete the event as well as update the event. This is also the key used to correlate Player requests to scheduled events.
action (attribute)	Required	AlternateAction	
providerId	Required	xs:token	Globally unique Id that identifies the content provider who submitted the scheduled event. This is typically the providers registered domain name (e.g., espn.com).
primarySrc	Required	Source	A Source identifier expressed as a string. Contains a mandatory “type” attribute with one of the following: • EIDR • TMS • ROVI • Any value beginning with “private”
utcStartTime	Conditionally Required	xs:dateTime	The start time of the blackout event in UTC time ([ISO 8601]).
utcStopTime	Conditionally Required	xs:dateTime	The stop time of the blackout events in UTC time.

startSignalId	Optional	xs:token	Optional signal ID from the provider. Used to track the signal through the alternate content system.
endSignalId	Optional	xs:token	Optional signal ID from the provider. Used to track the signal through the alternate content system.
restriction	1..unbounded	Restriction	
ext	Optional	Ext	Allows custom extensibility

ComplexType: **Restriction**

Children	Use	Data type	Description
type (attribute)	Required	xs:token	Restricted string. • VIRD • ZIP Code • ZIP Code + 4 • Mileage • Latitude/Longitude • Any value beginning with "private"
id (attribute)	Required	xs:token	A unique restriction ID assigned to this particular restriction. The type of this id is defined by the type attribute above.
blacklist	Required	xs:boolean	TRUE indicates the provided location data is blacklisted from viewing the indicated content. FALSE indicates the location data is whitelisted and are the only locations allowed to view the content. All other locations are required to view alternate content.
altContentType	Required	xs:token	Restricted string. Current defined values are FORCE_ALTERNATE, FORCE_DESCRIPTION, and USER_CHOICE.
altSource	Conditionally Required	Source	A Source identifier expressed as a string. Contains a mandatory "type" attribute with one of the following: • EIDR • TMS • ROVI • Any value beginning with "private" Required for altContentType of USER_CHOICE or FORCE_ALTERNATE
description	Conditionally Required	xs:string	A description of the scheduled event. Required for altContentType of FORCE_DESCRIPTION. In this case, the Player will display the text contained within this element.
confirmation	Optional	Confirmation	Presence of this element is used to report confirmation of an alternate content event
ext	Optional	Ext	Allows custom extensibility

Either altSource or description SHALL be provided depending on the value in altContentType.

ComplexType: **Confirmation**

Children	Use	Data type	Description
success (attribute)	Required	xs:boolean	Indicator true if this restriction was correctly processed for this restriction
exception (attribute)	Optional	xs:string	Explanation for why a restriction might not have been processed as expected – provided if success is false
actualStart (attribute)	Optional	xs:dateTime	Actual time when this event restriction was initiated
actualEnd (attribute)	Optional	xs:dateTime	Actual time when this event restriction was terminated – if known
ext (attribute)	Optional	Ext	Allows custom extensibility

Request Example:

```
POST /altcontentsvc/events HTTP/1.1

<event id="42983" type="ALT_CONTENT" action="CREATE"
xmlns="urn:cablelabs:iptvservices:esni:xsd:2">
   <providerId>HBO.COM</providerId>
   <primarySrc type="EIDR">10.5240/5F4F-D28A-0E6A-7DE3-FAE2-Q</primarySrc>
   <utcStartTime>2011-09-13T13:21:22.858-06:00</utcStartTime>
   <utcStopTime>2011-09-13T13:21:52.858-06:00</utcStopTime>
   <restriction id="0" type="private:global">
      <blacklist>true</blacklist>
      <altContentType>FORCE_DESCRIPTION</altContentType>
      <altSource type="EIDR">10.5240/AB70-729D-BEB3-F185-7D31-E</altSource>
      <description>The program will resume shortly after the break...</description>
   </restriction>
   <restriction id="0" type="private:global">
      <blacklist>true</blacklist>
      <altContentType>FORCE_DESCRIPTION</altContentType>
      <altSource type="TMS">10002</altSource>
      <description>The program will resume shortly after the break...</description>
   </restriction>
</event>
```

Response: *HTTP Status Code*:

201	Created
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error
503	Service Unavailable

No XML data is returned on successful (201 Created) or failed response.

On all failure responses, the HTTP status code and status text specify the error. Optionally, extended error information can be returned in a plain text body.

7.1.3 Updating an Event (ScheduleUpdate)

Request Direction: Event Scheduler to Alternate Event Service

Method: HTTP PUT

Description: The Event Scheduler issues a ScheduleUpdate to the Alternate Event Service to modify a previously scheduled event stored in the AES.

Events for which the utc (or pts) end times have passed cannot be modified. Attempts to modify an expired event will result in an error.

When modifying an existing event, the Event Scheduler SHALL include the eventId which was used to create the scheduled event. An update SHALL include all relevant attributes and elements, as an update completely replaces an existing event. Partial updates to an existing event are not allowed.

Location: http://<AES_IP>:<AES_PORT>/altcontentsvc/events/<eventId>

Request: See ScheduleCreate

Response: *HTTP Status Code*:

200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error
503	Service Unavailable

No XML data is returned on successful (200 OK) or failed response.

On all failure responses, the HTTP status code and status text specify the error. Optionally, extended error information can be returned in a plain text body.

7.1.4 Deleting an Event (ScheduleDelete)

Request Direction: Event Scheduler to Alternate Event Service

Method: HTTP DELETE

Description: The Event Scheduler issues a ScheduleDelete to the Alternate Event Service to cancel a previously scheduled event from the Event Scheduler.

The request includes the providerId and eventId used to create the scheduled event.

Location: http://<AES_IP>:<AES_PORT>/altcontentsvc/events?providerId=<providerId>&eventId=<eventId>

Request: DELETE /altcontentsvc/events?providerId=hbo.com&eventId=42983

There is no data contained within the ScheduleDelete Request.

Response:

HTTP Status Code:	200	OK
	400	Bad Request
	401	Unauthorized
	404	Not Found
	500	Internal Server Error
	503	Service Unavailable

No XML data is returned on successful (200 OK) or failed response.

On all failure responses, the HTTP status code and status text specify the error. Optionally, extended error information can be returned in a plain text body.

7.1.5 Querying for an Event (ScheduleQuery)

Request Direction: Either Event Scheduler to AES or AES to Event Scheduler

Method: HTTP GET

Description: The Event Scheduler MAY issue a ScheduleQuery request to the AES to retrieve a previously scheduled event or group of events. The AES MAY issue a ScheduleQuery request periodically to the Event Scheduler to retrieve new and updated events (pull model).

Location: AES as the server:

http://<AES_IP>:<AES_PORT>/altcontentsvc/events?providerId=<providerId>&utcStartTime=<utcStopTime>&utcStopTime=<utcStopTime>

Event Scheduler as server:

http://<ES_IP>:<ES_PORT>/<path>/events?utcStartTime=<utcStopTime>&utcStopTime=<utcStopTime> (where the <path> is determined by the host of the Event Scheduler)

Request: Schedule Queries can be performed at multiple levels:

- providerId - represents the provider id (required when requesting from the AES)
- utcStartTime - returns all resources that have a utcStartTime greater than this value.
- utcStopTime - returns all resources that have a utcStopTime less than this value.

Request Example:

`GET /altcontentsvc/events?providerId=1234`	Returns all known events for provider 1234
`GET /altcontentsvc/events?providerId=1234&utcStartTime=1316018176`	Returns all known events for provider 1234 with a utcStartTime > 1316018176
`GET /altcontentsvc/events?providerId=1234&utcStartTime=1316018176&utcStopTime=1319000000`	Returns all known events for provider 1234 in utc range [1316018176, 1319000000]
`GET /eventscheduler/events?utcStartTime=1316018176&utcStopTime=1319000000`	Returns all known events from the event scheduler in utc range [1316018176, 1319000000]

Response: *HTTP Status Code*:

200	OK
400	Bad Request
401	Unauthorized
404	Not Found
500	Internal Server Error
503	Service Unavailable

On a successful response (200 OK), the AES returns the XML data used to create the scheduled event. The XML response consists of the event root element and the associated child elements as specified in the ScheduleCreate.

On all failure responses, no XML data is returned. The HTTP status code and status text specify the error. Optionally, extended error information can be returned in a plain text body.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 527

<event id="42983" type="ALT_CONTENT" action="GET"
xmlns="urn:cablelabs:iptvservices:esni:xsd:2">
   <providerId>HBO.COM</providerId>
   <primarySrc type="EIDR">10.5240/5F4F-D28A-0E6A-7DE3-FAE2-Q</primarySrc>
   <utcStartTime>2011-09-13T13:21:22.858-06:00</utcStartTime>
   <utcStopTime>2011-09-13T13:21:52.858-06:00</utcStopTime>
   <restriction id="0" type="private:global">
          <blacklist>true</blacklist>
          <altContentType>FORCE_DESCRIPTION</altContentType>
          <altSource type="EIDR">10.5240/AB70-729D-BEB3-F185-7D31-E</altSource>
          <description>The program will resume shortly after the
break...</description>
          <confirmation success="true" actualStart="2011-09-13T13:21:22.858-06:00"
actualEnd="2011-09-13T13:21:52.858-06:00"/>
   </restriction>
   <restriction id="0" type="private:global">
          <blacklist>true</blacklist>
          <altContentType>FORCE_DESCRIPTION</altContentType>
          <altSource type="TMS">10002</altSource>
          <description>The program will resume shortly after the
break...</description>
          <confirmation success="false" exception="Redundant restriction"/>
   </restriction>
</event>
```

8 HTTP PROTOCOL

The HTTP Protocol [RFC 2616] is used for exchanging scheduling data between the content provider and the content distributor (e.g., Ad Spots, Blackout Events, etc.).

8.1 Connection

The Content Provider's Alternate Content Scheduling System (Scheduler) initiates the HTTP/1.1 connection to the Alternate Event Service. In HTTP/1.1, connections are persistent by default. The connection remains open unless either the client or server explicitly indicates the connection should close. The HTTP Response must include the header *Connection: close* in order to indicate that the connection will be closed.

During periods when no messages are being exchanged, the client or server MAY close the connection to conserve resources. If the Event Scheduler does not transmit a message for an extended period of time (for example, 30 seconds or more), then the Event Scheduler SHALL close the connection.

8.2 HTTP Headers

8.2.1 Request Messages

Regardless of request type, the same HTTP request header format will be used. Basic request requirements include:

- Creates must be sent via POST
- Updates must be sent via PUT
- Deletes must be sent via DELETE
- Requests (Queries) must be sent via GET
- URI specifies the application name and service name
- HTTP Version must be 1.1
- Entity-bodies will be used to convey XML data

Required entity headers:

- content-length: bytes (length of xml data)
- content-type: text/xml

Other entity headers may optionally be included. Non-required entity headers MAY be ignored by the server.

HTTP Request example header:

```
POST /AES/Schedule HTTP/1.1
Content-Type: text/xml
Content-Length: 1039
```

8.2.2 Response Messages

The Status Code and Status Text in the response indicate either the server successfully processed the request or failed to process the request. The failure may have been the result of bad client data or server side processing issue. Some HTTP response messages (typically failures) consist of only the HTTP header, while others (typically successes) consist of both HTTP header and XML body.

Required entity headers (with XML message body):

- content-length: bytes
- content-type: text/xml

Other entity headers may optionally be included. Non-required entity headers MAY be ignored by the client when processing the response.

The status codes and status text are HTTP response specific. HTTP response codes are provided within each sub-section defining the Scheduling operation.

HTTP Response example with no XML body:

```
HTTP/1.1 400 Bad Request
```

HTTP Response example with XML body (body not shown):

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 402
```

8.3 Message Flow

8.3.1 Request Response

The client connects to the server, sends an HTTP Request, and waits for an HTTP Response. The client MAY pipeline requests. That is, the client may send multiple requests on the same connection before receiving a response to the first request. The server must process all requests and return responses in the order they were sent.

Alternatively, the client MAY open multiple connections to the server to issue multiple requests in parallel. The server processes the request and sends an HTTP Response to the client on the same connection on which it received the request.

Based on the HTTP headers, the client or server MAY close the connection or the connection may be left open for subsequent requests from the client.

The client MAY have several open connections to the server at any given time.

The figure below shows the message flow sequence when the connection is closed after a single request-response pair.

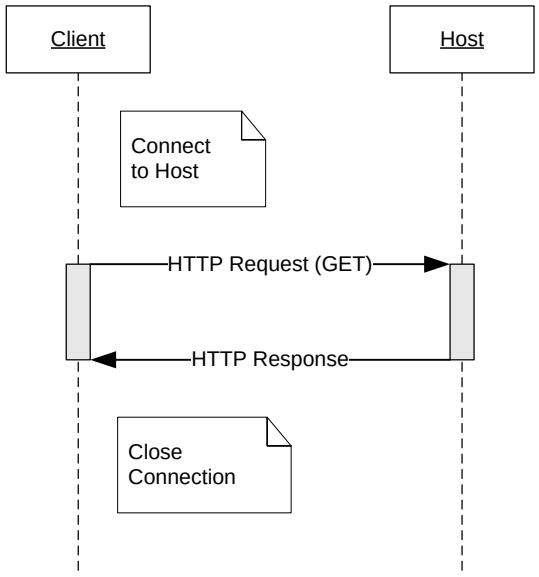

Figure 4 - Message Flow - Semi-Persistent Connection

The figure below shows the message flow sequence when the connection remains open between request-response pairs.

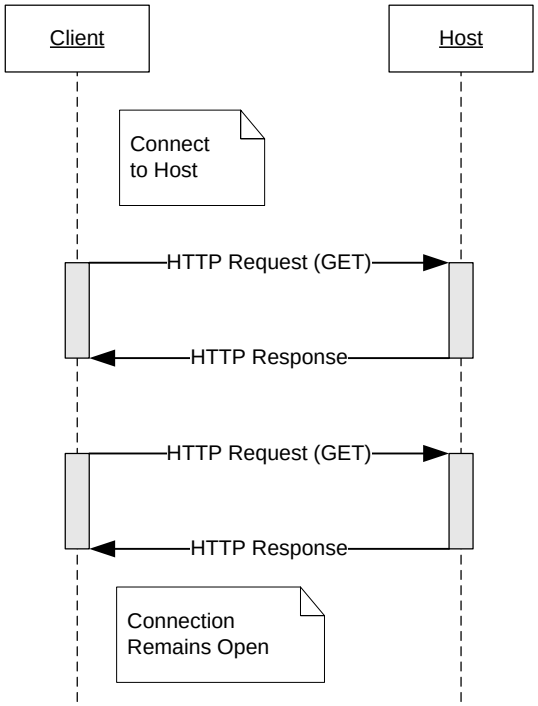

Figure 5 - Message Flow - Persistent Connection

8.3.2 Connection Lost

During the HTTP transaction, if the connection to the server is lost at any time prior to receiving a valid HTTP Response message, the client SHALL consider the request as failed. The client SHOULD immediately retry the request. If the retry fails, the client MAY discard the request or retry the request at a later time. The method of handling the failure is implementation-specific, and may vary by importance of the request type.

In most cases, the server will have started processing the request prior to connection loss. The server SHOULD finish processing the request. The server SHALL NOT establish a connection with the client in order to return the response. It is up to the client to reconnect and retransmit the request.

8.3.3 Request Timeout

The client sets a timer when it sends an HTTP Request. This timer represents the maximum amount of time that the client will wait for a response from the server. The maximum timeout period SHOULD be five (5) seconds.

If the client fails to receive an HTTP Response after the timer expires, the client SHALL consider the request as failed. The client MAY discard the request, immediately retry the request, or retry the request at a later time. The method of handling the failure is implementation-specific, and may vary by importance of the request type.

Annex A ESNI Schema

The messages, elements, and attributes that make up this specification are not explicitly defined in this document. The XML schemas, provided separately, are a normative part of this specification and SHALL be adhered to.

No messages representing the interfaces defined in the schema are considered conformant unless they are valid according to the schema documents. In the case where the written normative specification and the normative schema document conflict, the specification SHALL take precedence. Furthermore, there are additional requirements described in the schema annotations that are not enforceable by schema validation mechanisms. These requirements SHALL also be strictly observed.

The inclusion of a normative XML schema document does not require or imply the specific use of the schema nor a requirement that a message be validated.

The normative version of the ESNI schema file is published as a part of this release with the name OC-SP-ESNI-I02-131001.xsd.

The file is located here: http://www.cablelabs.com/downloads/iptv/OC-SP-ESNI-I02-131001.xsd

Appendix I Acknowledgements

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

Time Warner Cable: Chuck Hasek

Comcast: Allen Broome, Francesco Dorigo, Kevin Flanagan and Walt Michel

Appendix II Revision History

The following ECN was incorporated into version I02 of this specification:

EC Identifier	Author	Title of EC	Date Accepted
ESNI-N-13.1836-2	Batmanglidj	Alter Schema Definition for ESNI	8/29/2013