

**CableLabs®**  
**Subscription and Fulfillment Interfaces**

**Interactive Application Messaging Specification**

**CL-SP-SaFI-IAMv1.1-100702**

**ISSUED**

**Notice**

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

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

© Copyright 2008-2010 Cable Television Laboratories, Inc.  
All rights reserved.

## Document Status Sheet

<b>Document Control Number:</b>	CL-SP-SaFI-IAMv1.1-100702			
<b>Document Title:</b>	Interactive Application Messaging Specification			
<b>Revision History:</b>	I01 – Released 6/26/09 v1.1 – Released 7/2/10			
<b>Date:</b>	July 2, 2010			
<b>Status:</b>	<del>Work in Progress</del>	<del>Draft</del>	<del>Issued</del>	<del>Closed</del>
<b>Distribution Restrictions:</b>	<del>Author Only</del>	<del>CL/Member</del>	<del>CL/Member/ Vendor</del>	<del>Public</del>

### Key to Document Status Codes:

<b>Work in Progress</b>	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
<b>Draft</b>	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.
<b>Issued</b>	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.
<b>Closed</b>	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

### Trademarks:

CableLabs<sup>®</sup>, DOCSIS<sup>®</sup>, EuroDOCSIS<sup>™</sup>, eDOCSIS<sup>™</sup>, M-CMTS<sup>™</sup>, PacketCable<sup>™</sup>, EuroPacketCable<sup>™</sup>, PCMM<sup>™</sup>, CableHome<sup>®</sup>, CableOffice<sup>™</sup>, OpenCable<sup>™</sup>, OCAP<sup>™</sup>, CableCARD<sup>™</sup>, M-Card<sup>™</sup>, DCAS<sup>™</sup>, tru2way<sup>®</sup>, and Cable PC<sup>™</sup> are trademarks of Cable Television Laboratories, Inc.

# Contents

<b>1</b>	<b>SCOPE</b> .....	<b>1</b>
1.1	Introduction and Purpose.....	1
1.2	Requirements.....	1
<b>2</b>	<b>REFERENCES</b> .....	<b>2</b>
2.1	Normative References.....	2
2.2	Informative References.....	2
2.3	Reference Acquisition.....	2
<b>3</b>	<b>TERMS AND DEFINITIONS</b> .....	<b>3</b>
<b>4</b>	<b>ABBREVIATIONS AND ACRONYMS</b> .....	<b>5</b>
<b>5</b>	<b>OVERVIEW</b> .....	<b>6</b>
5.1	General Context.....	6
5.1.1	<i>Reference Architecture</i> .....	6
5.1.2	<i>Interface Description</i> .....	6
5.2	Transport Protocols.....	7
5.3	Data Model.....	7
5.4	Identifiers.....	7
5.5	Timing Model.....	8
<b>6</b>	<b>INTERACTIVE APPLICATION MESSAGING REQUIREMENTS</b> .....	<b>9</b>
6.1	Requirements on interoperable interactive applications.....	9
6.2	Requirements on compliant network systems.....	9
6.3	Data Model.....	9
6.3.1	<i>Application Information</i> .....	9
6.3.2	<i>ApplicationMessageHeaderGroup</i> .....	9
6.3.3	<i>AppMsg</i> .....	10
6.3.4	<i>AppMsgBlock</i> .....	10
6.3.5	<i>AppReportBlock</i> .....	10
6.3.6	<i>Context Info Type</i> .....	10
6.3.7	<i>Decoder Info Type</i> .....	12
6.3.8	<i>Geographic Code</i> .....	12
<b>7</b>	<b>INTERACTIVE APPLICATION MESSAGING DATA MODEL SCHEMA</b> .....	<b>13</b>
<b>8</b>	<b>HTTP ENCODING OF DATA MODEL</b> .....	<b>14</b>
8.1	'am' Values.....	14
8.1.1	<i>Application Information</i> .....	15
8.1.2	<i>Decoder Info</i> .....	15
8.1.3	<i>Context</i> .....	15
8.2	'amb' Values.....	17
8.3	Examples.....	17
	<b>APPENDIX I XML ENCODING OF DATA MODEL (INFORMATIVE)</b> .....	<b>20</b>
	<b>APPENDIX II REVISION HISTORY</b> .....	<b>21</b>

## Figures

Figure 5–1 - Interactive Application Messaging Reference Architecture Diagram.....6  
Figure 8–1 - am and amb positional referencing .....14  
Figure 8–2 - Application Message Block .....17  
Figure 8–3 - Application Report Block .....17  
Figure 8–4 - Repeating Blocks .....17

## Tables

Table 8–1 - ms and uid values .....14  
Table 8–2 - am and amb values .....15

# 1 SCOPE

## 1.1 Introduction and Purpose

The purpose of this document is to define requirements for an Interactive Application Messaging (IAM) platform. The IAM platform is composed of the content of messages generated by applications on a cable receiver and transported to a network component within an MSO system, a logical architecture for the generation and delivery of such messages, and any other related requirements necessary to implement the platform.

## 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.

- [IAM XSD] CL-SaFI-IAM-1.1.0.xsd, July 2, 2010, Cable Television Laboratories, Inc., <http://www.cablelabs.com/namespaces/safi/xsd/iam/v1.1/CL-SaFI-IAM-1.1.0.xsd>
- [MHP 1.1.2] DVB Multimedia Home Platform (MHP) Specification 1.1.2. [http://www.mhp.org/mhp\\_technology/mhp\\_1\\_1/mhp\\_a0068r1.zip](http://www.mhp.org/mhp_technology/mhp_1_1/mhp_a0068r1.zip)
- [RFC 3986] IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, URN Namespace.
- [SaFI COM XSD] CL-SaFI-COM-1.1.0.xsd, July 2, 2010, Cable Television Laboratories, Inc., <http://www.cablelabs.com/namespaces/safi/xsd/com/v1.1/CL-SaFI-COM-1.1.0.xsd>

### 2.2 Informative References

This document uses the following informative references.

- [CIP] Campaign Information Package Specification, CL-SP-SaFI-CIPv1.1-100702, July 2, 2010, Cable Television Laboratories, Inc.
- [ETV] Enhanced Television (ETV) Binary Interchange Format 1.0, OC-SP-ETV-BIF1.0-I05-091125, November 25, 2009, Cable Television Laboratories, Inc.
- [IAF] Interactive Application Fulfillment Summary Interface Specification, CL-SP-SaFI-IAFv1.1-100702, July 2, 2010, Cable Television Laboratories, Inc.
- [IAM EXMPL] CL-SaFI-IAM-I.0.0-example1.xml, July 2, 2010, Cable Television Laboratories, Inc.
- [SMS] Service Measurement Summary Interface Specification, CL-SP-SaFI-SMSv1.1-100702, July 2, 2010, Cable Television Laboratories, Inc.

### 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) Secretariat, 46000 Center Oak Plaza, Sterling, VA 20166, Phone +1-571-434-3500, Fax +1-571-434-3535, <http://www.ietf.org>
- W3C, <http://www.w3.org>

### 3 TERMS AND DEFINITIONS

This specification uses the following terms:

<b>Affiliate</b>	An operational entity that performs SaFI operations with one or more MSOs.
<b>Campaign</b>	Provides a set of delivery plans and/or placement directions for one or more MSOs, specific systems within an MSO's footprint, as well as a set of Programmed Events within a system. A Campaign is negotiated, purchased, and managed as a single entity via campaign planning and management tools that are not in scope for the MSO interfaces. Within a Programmed Event, one or more products from predefined product families can be defined for placement by MSO delivery and/or processing.
<b>Enhanced Program Sequence ID (EPSID)</b>	Identifies a unique Enhanced Package or Enhanced Package Element within a specific Programmed Event.
<b>GeoCode</b>	Geographic Code: the geographic region which this service measurement message represents. The value in this element may indicate a ZIP Code, MSO syscode, or other encoded regional identifier.
<b>MSO Order</b>	The part of a Campaign Information Package (CIP) that falls within a specific MSO's advertising footprint.
<b>Package</b>	Provides identifiers and parameters for signaling and messaging of placement directives associated directly with the package, and also a container for one or more Package Elements. The Package creates an implicit relationship between the elements regardless of the specific content they are delivered in or when they are actually delivered to the subscriber. For example, a Package might be the complete set of enhanced elements a subscriber would encounter in a single RFI product, or a single Vote/Poll product. This is typically, but not necessarily, expressed in a single ETV or OCAP application. A Package does not necessarily need to include any interactive elements or on-display presentations at all.
<b>Package Element</b>	Represents a single element of an enhanced experience that can be delivered to a subscriber. It provides identifiers and parameters for any signaling and messaging of placement directives associated directly with the element.
<b>Package Type</b>	An identifier that selects a message set used by an application in some specific enhancement context. The EpType, qualified by the application (orgId, appID, version), identifies an external set of messages, which is generally further divided by EventID. The external form of the message-set definition is not within the scope of this document.
<b>Programmed Event</b>	A Programmed Event (such as a program, network spot, VOD asset, time-shifted asset, guide page, or an advertisement) represents a constrained subscriber experience that includes an enhanced experience. Each Programmed Event has a Programmed Event Identifier (PEID), which uniquely specifies the context of application lifecycle events as well as qualifies any underlying measurement or fulfillment messages generated by the enhancements delivered during the lifetime of the Programmed Event. A Programmed Event may include one or more Packages.
<b>Schematron</b>	A rule-based validation language for making assertions about the presence or absence of specific patterns in XML trees.
<b>Service Measurement</b>	Information about the reach and usage of a campaign.
<b>Syscode</b>	A four-character, predefined code that represents a specific zone-level cable plant.

**System Order**

The part of an MSO Order that falls within a single zone-specific syscode. In simple cases, all the Programmed Events, Packages, and Package Elements of the Campaign will appear within each System Order; however, this may not be true due to site capabilities, or when targeting is applied.

## 4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

<b>AMB</b>	Application Message Block
<b>ARB</b>	Application Report Block
<b>CAAS</b>	Common Advanced Advertising Systems
<b>CIP</b>	Campaign Information Package
<b>EPSID</b>	Enhanced Program Sequence ID: an integer identifying a unique Package or Package Element within a specific Programmed Event
<b>EpType</b>	Enhancement Package Type
<b>ETV</b>	Enhanced Television
<b>HTTP</b>	Hypertext Transfer Protocol
<b>HTTPS</b>	Hypertext Transfer Protocol over Secure Sockets Layer (SSL)
<b>PEID</b>	Programmed Event ID: a globally-unique identifier for a Programmed Event
<b>RFI</b>	Request For Information
<b>SaFI</b>	Stewardship and Fulfillment Interfaces: a collection of interfaces defined by CableLabs to support advanced services on multiple cable systems
<b>SOAP</b>	Simple Object Access Protocol; as of SOAP 1.2, this no longer represents an acronym
<b>STB</b>	Set-Top Box
<b>WSDL</b>	Web Services Description Protocol

## 5 OVERVIEW

### 5.1 General Context

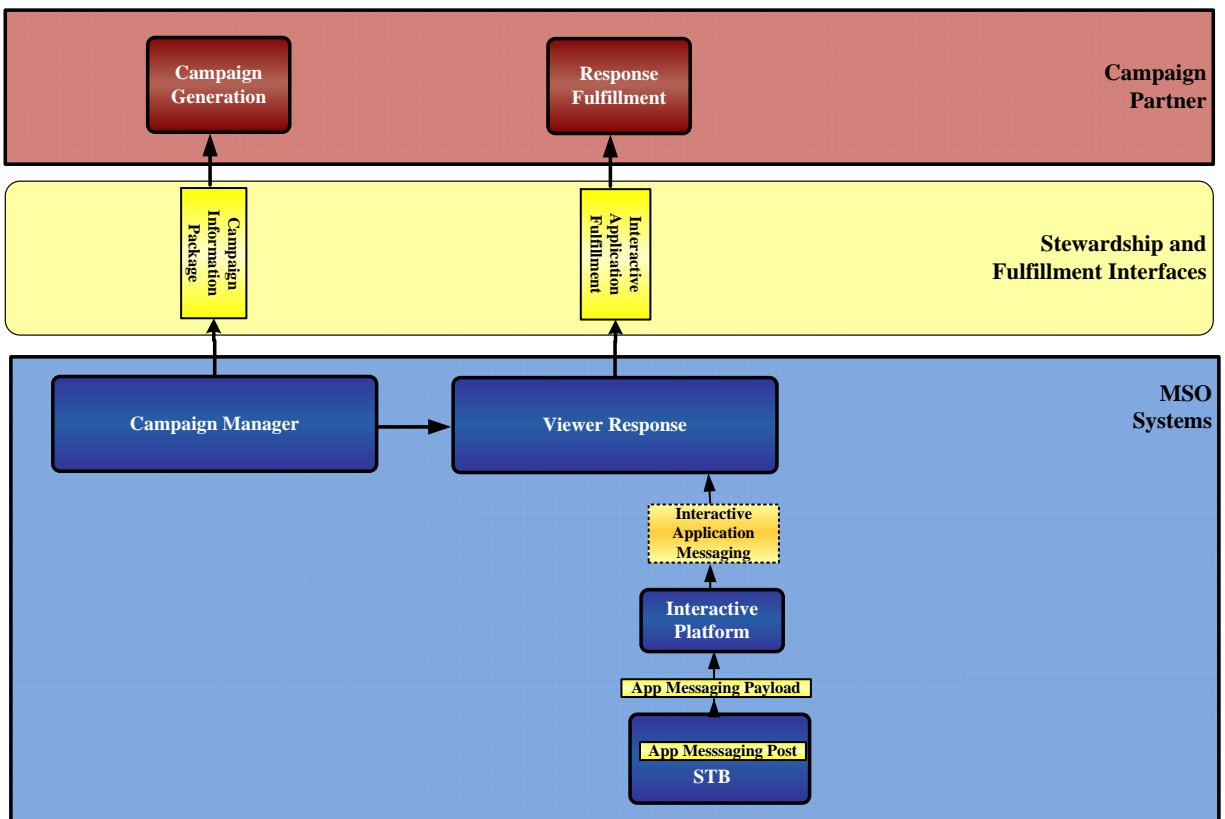
The IAM platform provides a critical interface between interoperable applications and MSO systems. Interoperable applications are applications that are distributed to more than one MSO. Messages instantiated by interoperable applications must therefore take the same form, regardless of the MSO system in which it is executing. This specification defines what form these messages may take and describes how MSO system components decode them.

Messages support two primary MSO-CAAS interfaces: the Service Measurement Summary [SMS] and Interactive Application Fulfillment [IAF] Interfaces. Service Measurement contains information about the execution of a campaign, including viewership of applications and individual overlays, while Application Fulfillment carries the results of voting/polling and RFI overlays. Both of these higher-level interfaces are supplied with specific application event and overlay data through the Interactive Application Messaging interface.

#### 5.1.1 Reference Architecture

The following diagram illustrates a systems view of the IAM platform. This diagram is derived from work produced by the Advanced Advertising Interfaces team, and represents an advertising-centric view of the platform. The IAM platform will be generalized such that applications unrelated to advertising can use the same platform.

The IAM platform is represented by the blocks labeled "App Messaging Payload" and "App Messaging Post" in Figure 5-1.



*Figure 5-1 - Interactive Application Messaging Reference Architecture Diagram*

#### 5.1.2 Interface Description

The Interactive Application Messaging platform includes two components: the App Messaging Post and the App Messaging Payload indicated in Figure 5-1.

The messaging payload is formally defined by an XML schema and conveys information from an STB to MSO network components. The encoding of the payload may be defined by the MSO, using any protocol supported by the STB and other network components, but the data model nevertheless conforms to this specification, and the URL address used by the application must reference an MSO or CAAS component and will be the same irrespective of the MSO system within which the application executes. It is the responsibility of the MSO to ensure this URL resolves to the proper system address for its respective network.

The format of a message expressed as an HTTP POST is described herein. The specific API calls necessary to construct and initiate a message are not defined. Both ETV and OCAP applications MAY generate messages in this format.

## 5.2 Transport Protocols

This messaging specification may be supported by ETV, OCAP, and other application formats. For ETV, MSO ETV User Agents and associated MSO network components define the transport protocol used to transmit messages from an STB to MSO network components. On ETV implementations, return channel messaging is modeled on HTTP, but for performance reasons, an MSO MAY choose to implement another protocol. All OCAP-enabled networks support HTTP; therefore, OCAP platforms that make use of this specification SHALL use HTTP as the transport.

Other MSO network components MAY adhere to this specification; for instance, a proprietary or HTTP form of a message MAY be transformed into an XML representation of the data model.

## 5.3 Data Model

An XML schema is defined that formally describes the data that may be generated by interoperable interactive applications and transmitted to MSO system components. An HTTP encoding of the data model is described for use by applications.

Based upon the Schema defined in Section 7, MSO system components MAY easily transform the HTTP format into XML or some other useful format.

## 5.4 Identifiers

The data model contains several identifiers. These identifiers may be resolved to campaign elements described in the campaign description associated with the application, or otherwise used to disambiguate message elements. The identifiers are:

**UnitID** – This value may be retrieved by an application through a well-known API. It indicates a unique device on an MSO's network and can be used to calculate unique responses. The semantics of the value are defined by the MSO; it may represent a MAC address, or an obfuscated value set by the MSO.

**ProgrammedEventID (PEID)** – This value is accessed by an application through a well-known resource transmitted with the application. The value is assigned during the stewardship process and is included as an element within the Campaign Description associated with the application instance. The PEID uniquely identifies the programmed event associated with the application instance, e.g., the Gators/Razorbacks football game broadcast at 4 pm ET, October 17, 2008.

**EnhancementPackageSequenceID (EPSID)** – This value is accessed by an application through a well-known resource transmitted with the application. The value is assigned during the stewardship process and is included as an element within the Campaign Description associated with the application instance. The EPSID identifies a "package" within an application. A package is a group of User Interface "overlays" or other application components. For instance, a sequence of overlays presenting an atomic vote/response group might be identified with an EPSID distinct from other overlays presented by the application.

**EventID** – An EventID indicates a discreet component or event within the context of a "package" identified by an EPSID.

## 5.5 Timing Model

The data model supports a timing model for accurately communicating the real-time value at which certain events transpire.

Messages that contain instrumentation information MAY include both the display time and the termination time of the overlay. Termination may occur when a viewer selects the "exit" key, chooses a response from an interactive overlay, or the overlay times out. The termination time allows calculation of "dwell time" on a given UI component. Each message contains an absolute time value in the Timestamp element of the ApplicationMessageHeaderGroup element. Subsequent time values are relative to the Timestamp. The duration timestamp attribute of Offset is offset in seconds from Timestamp. The Interval attribute is the duration in seconds after the Offset.

Messages that contain lifecycle and response information include a single Timestamp at which the event occurred.

## 6 INTERACTIVE APPLICATION MESSAGING REQUIREMENTS

This section defines requirements for the Interactive Application Messaging platform.

### 6.1 Requirements on interoperable interactive applications

Interoperable interactive applications SHALL comply with this specification by instantiating HTTP commands that conform to the data model and encoding rules defined herein. The data model is formally defined in Section 7, and encoding rules for HTTP transport are defined in Section 8.

### 6.2 Requirements on compliant network systems

Since interoperable interactive applications execute unchanged on all MSO systems, URLs utilized by interoperable interactive applications to post messages conformant to this specification SHALL be supported on all MSOs systems. Compliant URL forms are defined in Section 8.

### 6.3 Data Model

An XML schema is defined in Appendix I to fully describe the data elements that will be transmitted between client device and the MSO. This section is provided as an Informative reference. The root of the data model is an AppMsg (Application Message).

#### 6.3.1 Application Information

This information identifies the specific application that created the message.

- orgId – Decimal organization identifier in the format defined in [MHP 1.1.2], section 10.5.1.
- appId – Decimal application Id in the format defined in [MHP 1.1.2], section 10.5.1.
- appVerMajor – Decimal major version number identifying the specific release of the application.
- appVerMinor – Decimal version number identifying the specific release of the application.

#### 6.3.2 ApplicationMessageHeaderGroup

The Application Message Header contains common data elements to all Application Message Blocks being reported within a single Application Message.

It contains the following data units.

- MinSchemaVersion is the lowest compatible schema version that will validate this message. The minimum schema version of 1.1 must be used if any of the options added in this release are present. A minimum version of 1.0 remains valid if no 1.1 options are present. For backward compatibility, a minimum version of 1 with no minor version is to be interpreted as 1.0. The *ms* key field is limited to the following values:  
ms=1&  
ms=1.0&  
ms=1.1&  
As of this version 1.1, no other values are valid under the specification, and an implementation conforming to this and to no higher version SHALL NOT process a message containing any other values.
- Unit ID – Unique identifier for client device.
- Geographic Code – One of either zip code or system code used to report location information on client/client device.
- PEID – Programming Event Identifier used to uniquely identify a scheduled program.

- Timestamp – Reference timestamp used by all Application Report Blocks. Each individual Event will report its time as an offset of this timestamp.
- Decoder Info (DecoderInfoType) – Optional Hardware and Software information. See 6.3.7.

All following elements of the data model are optional to provide backward compatibility with version I01 (or 1.0). If any of the following optional elements or attributes appear, the MinSchemaVersion MUST be 1.1.

- ApplicationInfo – A data element whose attributes describe the application sending this message. Both the element and individual attributes are optional. See 6.3.1.
- ContextInfo (ContextInfoType) – The context in which this Application Message is being generated. See 6.3.6.

### 6.3.3 AppMsg

The Application Message is the root element in the Interactive Application Message data model. It provides a container for all data being transferred.

It contains the following data units.

- Application Message Header – Contains common addressing elements to all messages.
- Application Message Block – Repeating wrapper element for individual messages.

### 6.3.4 AppMsgBlock

The Application Message Block is a repeating container used to encapsulate messages associated with a common EPSID. It can contain multiple sub-blocks called Application Report Blocks.

It contains the following data units.

- EPSID – Enhanced Program Sequence ID. This is a unique value within the scope of an associated PEID.
- Application Report Block – Contains data relating to a specific Event.

### 6.3.5 AppReportBlock

The Application Report Block is the lowest-level container used to report on individual Events. It contains the following data units.

- Event ID – The ID used to uniquely identify a reportable Event within an EPSID. Events are defined by each application.
- User Input – Specific client response being reported.
- Parameters – Optional data field used to transmit additional information. Contents of this field and parsing rules defined by each application. The Parameter data field MAY use the standard data element delimiter character "-" or another valid character as a delimiter.
- Relative Time Attribute – The Relative Time Attribute is a common element used to represent time as an offset and optionally length as a duration of a common time. In this case, the common time is carried in the Application Message Header as the Timestamp element.

### 6.3.6 Context Info Type

The optional Context Info Type contains context information for an application message.

- ContextType – Required attribute identifying the type of context information that follows:
  - Linear – media ids of category service and program allowed (applies to both live or TSB. TSB applies to time shifted content delivered from local memory, local DVR, network DVR or other storage location.)
  - DVR – media ids of category service and program allowed (applies to local DVR, network DVR or other persistent storage location.)
  - VOD – media ids of media only

- Description – Optional attribute containing a descriptive string supplementing the other information supplied.
- ContextDesignators – Optional element containing one or more attribute pairs identifying underlying content context for the message.

contextDesignatorType – The type for the associated value.

- 1 – VCN
- 2 – universal source id
- 3 – tms source id
- 4 – rovi source id
- 5 – local source id
- 6 – short name
- 7 – rovi network id
- 8 – tms network id
- 9 – universal network id
- 10 – session id

contextDesignatorValue – A value within the designator type.

- ProgramDesignators - Optional element containing one or more attribute pairs identifying underlying programming.

programType – The type of the associated value.

- 1 – universal program id
- 2 – rovi series id
- 3 – tms series id
- 4 – universal series id
- 5 – local program id
- 6 – local series id

programValue – A value within the designator type.

- MediaDesignator – Optional element containing an attribute pair identifying underlying media.

mediaDesignatorType – The type of the associated value.

- 1 – CableLabs Metadata ProviderID/AssetID pair

mediaDesignatorValue – The syntax of the value specified for mediaDesignatorValue is based on the mediaDescriptorType.

mediaDescriptorType value	mediaDesignatorValue syntax	Notes
1	<i>providerID"/"assetID</i>	The "/" is the separator character between the <i>providerID</i> and <i>assetID</i> and SHALL be supplied.

- PositionalDesignator – Optional element containing positional information.

TemporalPositionOffset – To be interpreted as follows

Linear (ContextType = 1) – absolute # of seconds offset from live

DVR (ContextType = 2) – # of seconds offset from the start of the recording

VOD (Context = 3) – # of seconds offset from the start of the session

An optional DirectionSpeedDesignator element containing 2 attributes

Direction

F – forward

R – rewind

P – Pause

Speed

1 – normal speed

2 – 2X

Etc.

### 6.3.7 Decoder Info Type

The Decoder Info Type contains information about the decoder the application that sourced the message is running on. The information is about both the hardware and the software. All attributes are optional and are typically acquired by an ETV application via the Platform Information Item by specifying the appropriate value for pvr4Item (see [ETV]).

- HWManufacturer
- HWModel
- HWVersionMajor
- HWVersionMinor
- SWManufacturer
- SWModel
- SWVersionMajor
- SWVersionMinor

### 6.3.8 Geographic Code

The Geographic Code contains client device location information in the form of either a zip code or a system code.

It contains the following data units.

- System Code – A syscode is a geographic identifier allocated by agreement of MSOs.
- Zip Code – Zip code for client and/or client device

## 7 INTERACTIVE APPLICATION MESSAGING DATA MODEL SCHEMA

The formal data definition is found in [IAM XSD].

## 8 HTTP ENCODING OF DATA MODEL

Applications may generate messages that adhere to this specification by instantiating HTTP POST commands that encode elements as key/value pairs. The HTTP POST SHALL be sent to the URL `ara.cablelabs.com/am`.

**Note** The host `ara.cablelabs.com` is not a public URL. It MUST resolve to the local server that processes the IAM posts. Normally, this is accomplished through the use of a CNAME or an A record in the local DNS.

The HTTP POST is a sequence of records of the form `'key=valueString&'`, where the encoding scheme defines four keys, each with its own unique value string. The `'&'` is the standard value string terminator. The first two keys are contractions of their XML counterpart and are used to carry a single value, as shown below. Each may occur only once in a message.

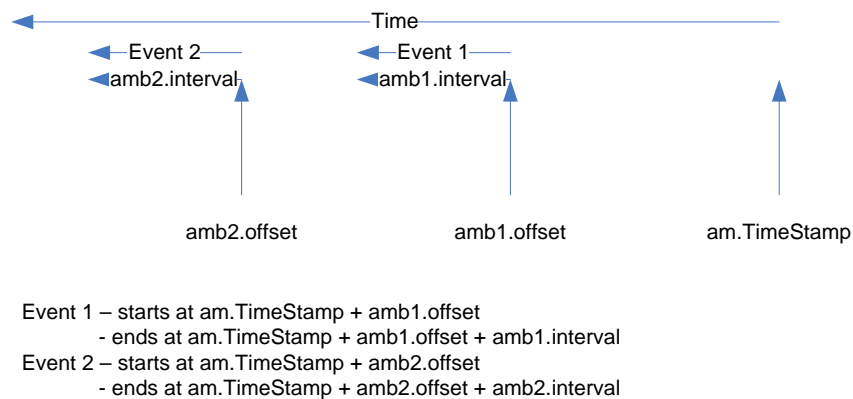
**Table 8-1 - ms and uid values**

Key	Value (Full Name)
Ms	MinSchemaVersion
Uid	UnitID

The next two keys have compound value strings, formed of a number of fields (see Table 8-2). All but the first of these fields is introduced by the standard delimiter, `'-'`. Table 8-2 provides the appropriate mapping and field values.

### 8.1 'am' Values

The `am` header occurs exactly once per message, and supplies information that applies to all subsequent data. The values have semantics as defined in Section 6.3.2. Note that the `Timestamp` value is a Unix time (seconds elapsed since 1970-01-01 0:00:00 UTC) representation of UTC. The duration timestamp of `Offset` is an offset in seconds from `Timestamp`, and `Interval` is the duration in seconds after `Offset`, as illustrated in the following diagram:



**Figure 8-1 - am and amb positional referencing**

All fields following the `Timestamp` are optional. If none are present, the form is compatible with this document version and a `MinSchemaVersion` of 1.1 is not required. For each of these optional fields, the value MAY be omitted. However, for any optional field whose value is not omitted, the standard delimiter for all prior fields

MUST be present. The standard delimiter MAY be omitted for all fields following the last optional field with a value present.

**Table 8-2 - am and amb values**

Key	Full Name	Value
Am	Application Message	GeoID-PEID-Timestamp-AppInfo-DecoderInfo-Context
Amb	Application Message Block	EPSID-offset-interval-EventID-UserInput-Parameters

The AppInfo, DecoderInfo, and Context components of an am block are described in the following sections.

### 8.1.1 Application Information

The ApplicationInformation group is a sequence of optional fields whose semantics are identical to those of Section 6.3.1. The fields with values present in the value string SHALL be in the same order and SHALL be separated by standard delimiters. Also see the information on optional fields in Section 8.1.

### 8.1.2 Decoder Info

The Decoder Info is a sequence of optional fields whose semantics are identical to those of Section 6.3.7. The fields with values present in the value string SHALL be in the same order and SHALL be separated by standard delimiters. Also see the information on optional fields in Section 8.1.

### 8.1.3 Context

The optional context information SHALL be encoded as the last field of the Application Message block if present. See Section 6.3.6 for the valid values for use within each context information field. Each context information subfield SHALL be separated by a semi-colon (;). The context information field SHALL be terminated by the Application Message block terminator or key value string terminator. If present, the context information field SHALL be coded as follows:

- Context type SHALL be specified once and occupy the first character of the context information field. The context type SHALL be introduced by the standard delimiter ("-") and followed by a semi-colon (;) if additional context information field values are coded.

The following context information field values MAY be provided in any order following the context type field. Each field supplied, with the exception of the last coded value, SHALL be terminated by a semi-colon (;).

- Zero or more context designators MAY be specified in the context information field. The context designator is encoded as follows:

`c/<context designator type>/<context designator value>`

Where:

`c` – lower case character "c" that introduces the context designator field in the context information field

`/` - separator character

`<context designator type>` - a valid context designator type value.

`/` - separator character

`<context designator value>` - a valid context designator value value.

- Zero or more program designators MAY be specified in the context information field. The program designator is encoded as follows:

`p/<program type>/<program value>`

Where:

p – lower case character "p" that introduces the program designator field in the context information field.

/ - separator character

<program type> - a valid program type value

/ - separator character

<program value> - a valid program value value

- Zero or one media designators MAY be specified in the context information field. The media designator is encoded as follows:

m/<media designator type>/<media designator value>

Where:

m – lower case character "m" that introduces the media designator field in the context information field

/ - separator character

<media designator type> - a valid media designator type value

/ - separator character

<media designator value> - a valid media designator value value. This field may contain additional valid separator character (/ in the case of a VOD asset identifier. As specified above, the field is terminated by a semi-colon or end of application message character.

- Zero or one temporal positional designators MAY be specified in the context information field. The positional designator is encoded as follows:

t/<temporal position offset>/<direction-speed designator>

Where:

t – lower case character "t" that introduces the positional designator field in the context information field.

/ - separator character

<temporal position offset> - a valid temporal position offset value

/ - separator character – only included if the direction-speed designator field is included.

<direction-speed designator> - a value specifying valid direction and speed. Character position one is the direction and the additional characters specify the speed. If the stream is paused (first character is a "P"), no speed SHALL be specified.

- Zero or one descriptive text for the context information field. The descriptive text is encoded as follows:

d/<descriptive text>

Where:

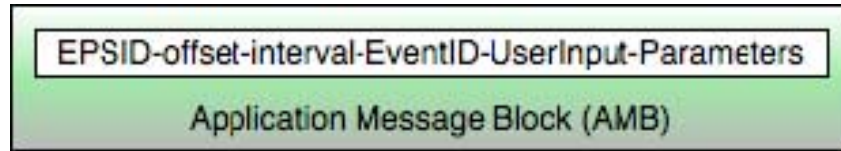
d – lower case character "d" that introduces the descriptive text in the context information field

/ - separator character

<descriptive text> - additional text specific to the context information field. The content of this field SHALL NOT use semi-colon, dash, or the application message termination character.

## 8.2 'amb' Values

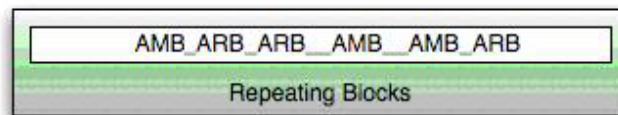
As shown in Table 8–2, the amb value string begins with an EPSID. The following elements, i.e., offset through parameters, are called an ApplicationReportBlock (ARB). One ARB is required with each amb, so no delimiter is necessary to identify the initial copy. However, the ARB can be repeated as necessary within an amb. If more than one ARB is present, a single underscore ("\_") SHALL introduce subsequent ARBs. Additionally, the value string can contain multiple repetitions of the entire described value sequence, beginning with EPSID. These repetitions are logically equivalent to a series of amb key-value pairs, but since only one amb key is permitted, they are instead appended to its value string. If more than one such sequence is present, a double underscore ("\_\_") SHALL introduce subsequent sequences. For detailed illustration, see example and diagrams below.



**Figure 8–2 - Application Message Block**



**Figure 8–3 - Application Report Block**



**Figure 8–4 - Repeating Blocks**

It is important to note that the '-' and '\_' are special characters used in the message encoding. Any application leveraging this interface should take special care to ensure these characters either are not present in the data fields being transmitted over the interface or that they are properly encoded if they are being used in anyway except as field and message delimiters. The proper encoding should follow [RFC 3986] encoding schema, where the character '-' is encoded as '%2d' and '\_' is encoded as '%5f'.

## 8.3 Examples

1) An example message is provided using the XML sample included in Appendix I. This example has no optional am fields, but two amb value sequences, and the first amb has two arb sequences. "^1" and "^2" mark the start of the two amb, and "^3" marks the start of the second arb. In this example the second arb appears to use "," as an

application level internal delimiter; however, a "-" would have been valid here as introducing another parameter value. It has also encoded the "-" in "M-XL" as "%2d".

**Note** HTTP header information is included in the example, but may be dropped depending on the actual transport used between the set-top box and the headend network.

```
POST /am HTTP/1.1
Host: ara.cablelabs.com
User-Agent: MSO-UserAgent/1.0
Content-Length: 114
Content-Type: application/x-www-form-urlencoded
ms=1&
uid=0123456789ab&
am=80027-48taPS4lQVu7q6RjEA40kg-1207276200&
amb=1-30--1-A-^1_60--2-Y-30519-RED,M%2dXL^2_2-30-5-3-A-&
                ^3
```

2) The next example shows only the am block portion with all of the am optional attributes fully populated. Note that there is no line break in the actual am value string, although one must appear in this document.

```
am=80027-48taPS4lQVu7q6RjEA40kg==1207276200-72-12-1-0-Man-Mod-22-0-TVW-SWMod-5-0-1;c/1/37;c/4/2872;p/2/38756;t/5/F1;d/a sample context message&
```

The application information is the first four fields:

- 72 – organization ID
- 12 – application ID
- 1 – version major
- 0 – version minor

The decoder information is the next eight fields:

- Man - HWManufacturer
- Mod - HWModel
- 22 - HWVersionMajor
- 0 - HWVersionMinor
- TVW - SWManufacturer
- SWMod - SWModel
- 5 - SWVersionMajor
- 0 - SWVersionMinor

The context information is the last field (at '-1;'), and it contains the following sub-fields:

- 1 – Linear
- c/1/37 – VCN of 37
- c/4/2872 – Rovi Source ID of 2872
- p/2/38756 – Rovi Service ID of 38756
- t/5/F1 – Position is 5 seconds from live playing forward at normal speed
- d/a sample context message – descriptive text about this context entry

3) This section provides some additional examples of am header information. These are all subsets of example (2).

*Context Only – while application and decoder information is not included the separators are required*

- `am=80027-48taPS4IQVu7q6RjEA40kg-1207276200-----1;c/1/37;c/4/2872;p/2/38756;t/5/F1;d/a sample context message&`

*Application Info Only – since the decoder and context info is omitted no additional separators are required.*

- `am=80027-48taPS4IQVu7q6RjEA40kg-1207276200-72-12-1-0&`

*Decoder Info Only – while application information is not included the separators for the fields are required but nothing is required for context that is not specified*

- `am=80027-48taPS4IQVu7q6RjEA40kg-1207276200-----Man-Mod-22-0-TVW-SWMod-5-0&`

4) This section has two examples, one of a message valid only under MinSchemaVersion of 1 (or 1.0) and the second of a message valid only under MinSchemaVersion of 1.1. The first message contains no amb options following the am timestamp field, while the second does.

Version 1.0 spec ◇ "ms=1"

`ms=1&`

`uid=0123456789ab&`

`am=80027-48taPS4IQVu7q6RjEA40kg-1207276200&`

`amb=1-30--1-A-_60--2-Y-RED,XL__2-30-5—A-&`

Version 1.1 spec ◇ "ms=1.1"

`ms=1.1&`

`uid=0123456789ab&`

`am=80027-48taPS4IQVu7q6RjEA40kg-1207276200-72-12-1-0-Man-Mod-22-0-TVW-SWMod-5-0-1;c/1/37;c/4/2872;p/2/38756;t/5/F1;d/a sample context message&`

`amb=1-30--1-A-_60--2-Y-RED,XL__2-30-5—A-&`

## **Appendix I      XML Encoding of Data Model (Informative)**

Examples of IAM data expressions can be found in [IAM EXMPL].

## Appendix II      Revision History

The following ECNs are incorporated into CL-SP-SaFI-IAMv1.1-100702:

<b>EC Identifier</b>	<b>Date Accepted</b>	<b>Title of EC</b>
SaFI-IAM-N-10.0067-1	7/2/10	Correct definition of interval in IAM message
SaFI-IAM-N-10.0068-2	7/2/10	Add decoder, hardware, and software information to the application message header
SaFI-IAM-N-10.0069-2	7/2/10	Add application information to the application message header
SaFI-IAM-N-10.0070-2	7/2/10	Add ability to append multiple parameters to Parameters field
SaFI-IAM-N-10.0071-2	7/2/10	Mandatory updates to IAM for version change
SaFI-IAM-N-10.0077-3	7/2/10	Add STB presentation contest
SaFI-IAM-N-10.0081-2	7/2/10	Editorial and technical updates for consistency across SaFI specs