OpenCable™ Specifications
Stewardship and Fulfillment Interfaces

Common Data Types Specification

OC-SP-SaFI-COMv3.0-120307

ISSUED

Document Status Sheet

Document Control Number:	OC-SP-SaFI-COMv3.0-120307			
Document Title:	Common Data Types Specification			
Revision History:	I01 – Released 6/26/09 v1.1 – Released 7/2/10 v2.0 – Released 1/31/11 v3.0 – Released 3/7/12			
Date:	March 7, 2012			
Status:	~~Work in Progress~~	~~Candidate~~	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

1 SCOPE 1

1.1 Introduction and Purpose 1
1.2 Requirements 1

2 REFERENCES 2

2.1 Normative References 2
2.2 Informative References 2
2.3 Reference Acquisition 2

3 TERMS AND DEFINITIONS 3

4 ABBREVIATIONS AND ACRONYMS 4

5 OVERVIEW 5

5.1 General Context 5
5.1.1 Reference Architecture 5
5.1.2 Interface Descriptions 5
5.2 Specification Components 6

6 SAFI COMMON DATA TYPES 7

6.1 Data Model 7
6.1.1 AcknowledgementMessageType 7
6.1.2 [deleted] 7
6.1.3 DerivativeIdType 7
6.1.4 EpsidType 7
6.1.5 EventIdType 7
6.1.6 ExtType 7
6.1.7 FaultType 7
6.1.8 GeographicCodeType 8
6.1.9 NonEmptyStringType 8
6.1.10 NonNegativeIntType 8
6.1.11 OptionalRelativeTimeAttributeGroup 8
6.1.12 PackageIDGroup 8
6.1.13 ParametersType 8
6.1.14 PeidType 8
6.1.15 [deleted] 8
6.1.16 RelativeTimeAttributeGroup 8
6.1.17 ReportDataType 9
6.1.18 RepositoryAppKeyType 9
6.1.19 Scte35SegmentationUpidType 9
6.1.20 SyscodeType 9
6.1.21 TimeRangeType 9
6.1.22 TimestampType 10
6.1.23 TrackingType 10
6.1.24 UserInputType 10
6.1.25 UuidUrlType 10
6.1.26 VersionType 10
6.1.27 ZipcodeType 11

ANNEX A SAFI COMMON DATA TYPES MODEL SCHEMA (NORMATIVE) 12

APPENDIX I REVISION HISTORY 13

Figures

Figure 5–1 - Advanced Services platform 5

1 SCOPE

1.1 Introduction and Purpose

This document provides an overview of the CableLabs Stewardship and Fulfillment Interfaces (SaFI) and defines the common data types used by SaFI.

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.

[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
[SaFI COM XSD]	OC-SP-SaFI-COM-3.0.0.xsd, March 7, 2012, Cable Television Laboratories, Inc.
[SCTE 35]	ANSI/SCTE 35 2011, Digital Program Insertion Cueing Message for Cable.

2.2 Informative References

This document uses the following informative references.

[CIP]	Campaign Information Package Specification, OC-SP-SaFI-CIPv3.0-120307, March 7, 2012, Cable Television Laboratories, Inc.
[COM HTML]	OC-SP-SaFI-COM-3.0.0.html, March 7, 2012, Cable Television Laboratories, Inc.
[IAF]	Interactive Application Fulfillment Summary Interface Specification, OC-SP-SaFI-IAFv3.0-120307, March 7, 2012, Cable Television Laboratories, Inc.
[IAM]	Interactive Application Messaging Specification, OC-SP-SaFI-IAMv3.0-120307, March 7, 2012, Cable Television Laboratories, Inc.
[SCTE 130-2]	SCTE 130-2 2008a - Digital Program Insertion–Advertising Systems Interfaces Part 2–Core Data Elements
[SMS]	Service Measurement Summary Interface Specification, OC-SP-SaFI-SMSv3.0-120307, March 7, 2012, 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/
- SCTE: Society of Cable Telecommunications Engineers Inc., 140 Philips Road, Exton, PA 19341
  Phone: 610-363-6888 / 800-542-5040; Fax: 610-363-5898; http://www.scte.org/
- CableLabs SaFI 3.0 schemas and WSDLs are available at the following URLs:

OC-SaFI-COM-3.0.0.xsd	http://www.cablelabs.com/namespaces/safi/xsd/com/v3.0/
OC-SaFI-CIP-3.0.0.xsd OC-SaFI-CIP-.0.0.wsdl	http://www.cablelabs.com/namespaces/safi/xsd/cip/v3.0/
OC-SaFI-IAM-3.0.0.xsd	http://www.cablelabs.com/namespaces/safi/xsd/iam/v3.0/
OC-SaFI-IAF-3.0.0.xsd OC-SaFI-IAF-.0.0.wsdl	http://www.cablelabs.com/namespaces/safi/xsd/ias/v3.0/
OC-SaFI-SMS-3.0.0.xsd OC-SaFI-SMS-3.0.0.wsdl	http://www.cablelabs.com/namespaces/safi/xsd/sms/v3.0/

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Affiliate	An operational entity that performs SaFI operations with one or more MSOs.
Bundle	A set of placements in a specific campaign, at an indicated MSO and syscode. The relationship between the placements that forms the basis of a bundle is beyond the scope of this specification.
Campaign	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.
Enhanced Program Sequence ID	An integer identifying a unique Package or Package Element within a specific Programmed Event.
GeoCode	Geographic Code: the geographic region that this service measurement message represents. The value in this element may indicate a ZIP Code, MSO syscode, or other encoded regional identifier.
MSO Order	The part of a Campaign Information Package (CIP) that falls within a specific MSO's advertising footprint.
Placement	A specific presentation of one or more advanced advertising assets at some advertising placement opportunity. In CIP, a data structure that supplies the definition of the conditions under which a placement may be executed.
Programmed Event ID	A globally-unique identifier for a Programmed Event.
Service Measurement	Information about the reach and usage of a campaign.
Stewardship and Fulfillment Interfaces	A collection of interfaces defined by CableLabs to support advanced services on multiple cable systems.
Syscode	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:

AMB	Application Message Block
ARB	Application Report Block
CAAS	Common Advanced Advertising Systems
CIP	Campaign Information Package
EPSID	Enhanced Program Sequence ID
EpType	Enhancement Package Type
ETV	Enhanced Television
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Sockets Layer (SSL)
PEID	Programmed Event ID
RFI	Request For Information
SaFI	Stewardship and Fulfillment Interfaces
SOAP	Simple Object Access Protocol; as of SOAP 1.2, this no longer represents an acronym
STB	Set-Top Box
WSDL	Web Services Description Language

5 OVERVIEW

5.1 General Context

The CableLabs Stewardship and Fulfillment Interfaces are part of a broader set of interfaces and content formats that support the deployment of advanced services across a national MSO footprint. Advanced services include interactive programming and applications, ad insertion, viewer-selected content (On Demand), and addressable advertising.

5.1.1 Reference Architecture

The following diagram illustrates an overview of the advanced services platform and indicates where the SaFI interfaces fit in:

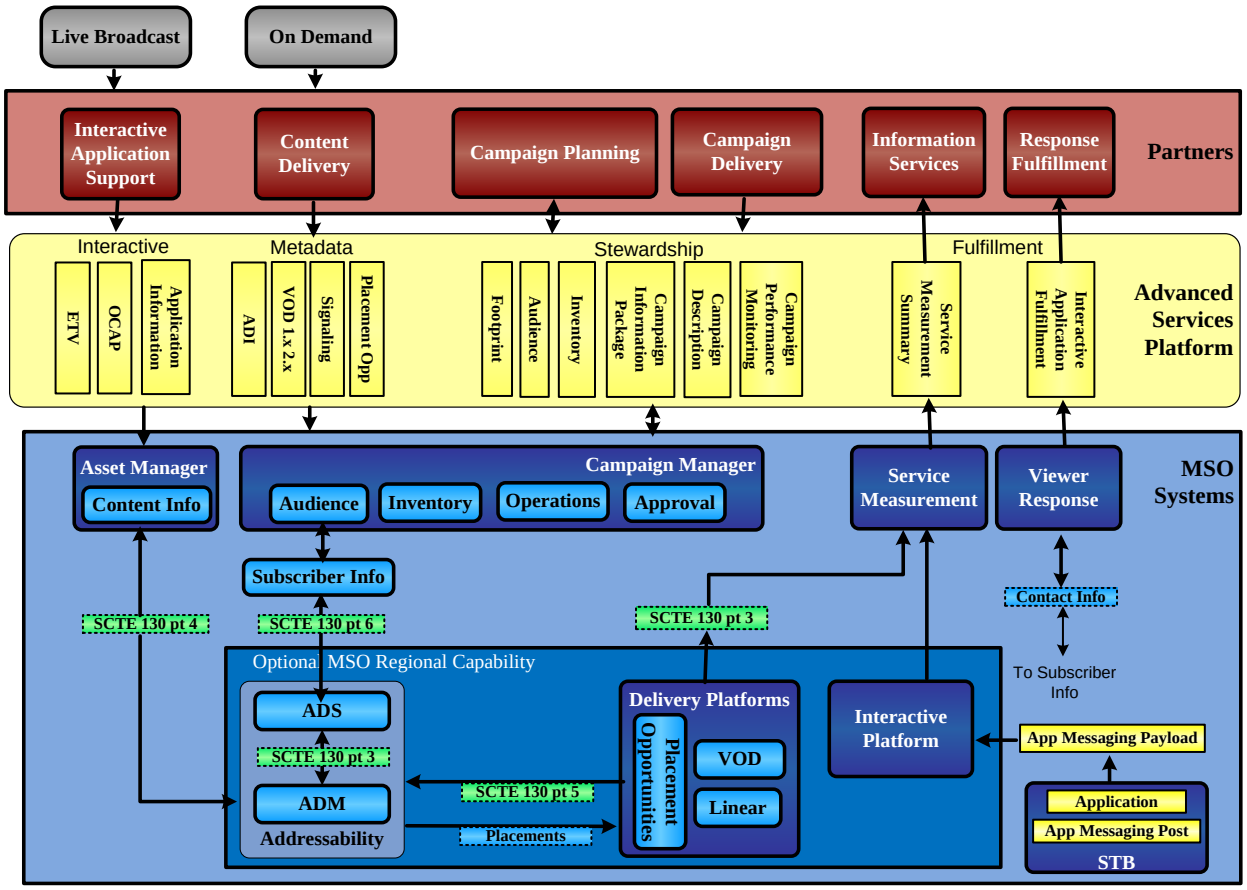

Figure 5–1 - Advanced Services platform

5.1.2 Interface Descriptions

For the purposes of this document, Stewardship and Fulfillment Interfaces are the following elements:

- The Campaign Information Package (CIP) is a web service that defines execution instructions to MSOs for advanced services.
- The Service Measurement Summary Interface (SMSI) is a web service that defines measurement of campaign execution.

- Interactive Application Fulfillment (IAF) is a web service interface that delivers the results of interactive responses from MSOs to partners.
- Interactive Application Messaging (IAM) is a data model definition for compact messaging from interactive applications to MSO systems.

The CIP, SMSI, and IAF interfaces all include this Common Data Type specification as a normative reference.

Other elements shown in Figure 5–1 are either not yet defined, or are illustrated here for informative context.

5.2 Specification Components

This specification consists of the following elements:

- This specification document, which is normative except as otherwise noted.
- The associated XML schema file, [SaFI COM XSD], which is normative.
- The associated XML schema document set contained in OC-SP-SaFI-COM-3.0.0-html.zip, which is informative.

6 SAFI COMMON DATA TYPES

This section defines requirements for metrics and the interface between a metrics engine and a cable headend.

This section outlines key elements of the data model defined in the [SaFI COM XSD] XML schema.

6.1 Data Model

An XML schema is defined in [SaFI COM XSD] to describe shared data-model components, i.e., those that occur in multiple SaFI interfaces.

XML files conformant to this data model SHALL be generated and properly received by the CAAS and MSO systems.

6.1.1 AcknowledgementMessageType

Defines a standard acknowledgement message for the SaFI web services.

- Attribute transmissonId [Required, xs:string] is the identifier of the message being acknowledged.
- Element Ack [Required single occurrence, xs:boolean] is the positive (true) or negative (false) status reported for the message.

6.1.2 [deleted]

6.1.3 DerivativeIdType

Derivative ID that uniquely identifies a document (e.g., IAF or SMSI). This is an xs:string that can be a GUID represented in RFC 4122 Base64-URL form (so 16 bytes convert to a 22-byte string). This might use a more terse form, but some namespace resolution is needed first.

6.1.4 EpsidType

Enhancement sequence number within a PEID. This is an xs:int with a domain of zero to 65,535 (two binary bytes). These sequence numbers are assigned to message-addressable nodes in the campaign description associated with UUIDs and ETypes. The UUID+EPSID can be used to locate the node, and its EType allows further navigation. EpsidType is limited to two binary bytes.

6.1.5 EventIdType

These numbers are processing identifiers assigned to message-addressable nodes in the campaign description, i.e., containing a PEID+EPSID. The PEID+EPSID can be used to locate the node, and its EType identifies specific message processing associated with that node. EventIdType is an xs:int with a domain of zero to 99.

6.1.6 ExtType

A standardized extensibility element. It contains

- Attribute ##any, i.e., from any namespace and using lax processing.
- Element of zero or more children of ##any, i.e., from any namespace and using lax processing.

6.1.7 FaultType

Defines a standard fault model for the SaFI web services.

- Attribute code [Optional, xs:int] is an error code unique to the specific fault.
- Element Reason [Required, xs:string] is a description of the fault.

6.1.8 GeographicCodeType

GeographicCode is an identifier used to denote the location of the device/unit. It is an xs:choice of Zipcode or Syscode.

6.1.9 NonEmptyStringType

A derivative of xs:string that is required to have content. Using this in place of xs:string precludes required attribute or element content being left empty. However, it does not preclude only whitespace content.

6.1.10 NonNegativeIntType

A derivative of xs:int that permits only zero and positive values.

Note: As opposed to xs:nonNegativeInteger, which derives from Integer, NonNegativeIntType derives from xs:int, which has a much smaller domain.

6.1.11 OptionalRelativeTimeAttributeGroup

Attribute set that may define some instant, or interval starting at some instant, but may also be empty. The instant is offset from some event defined by application convention (as xs:duration), and the interval is a duration after the offset.

- Attribute offset [Required, xs:duration] is the specified instant. This is a positive or negative offset from some predefined event, usually the start of flight.
- Attribute interval [Optional, xs:duration] is optional, and specifies a duration before or after the offset.

6.1.12 PackageIDGroup

Contains identifiers used to unambiguously identify a stewarded package. PackageIDGroup includes the following data objects:

- Element PEID [Required, PeidType] is the Programmed Event ID.
- EPSID [Required, EpsidType] is the Enhanced Package Sequence ID.
- EventID [Optional, EventIdType] is an optional event ID associated with the PEID and EPSID.

6.1.13 ParametersType

An optional unconstrained xs:string used to pass other relevant data.

6.1.14 PeidType

The identifier of a CIP document message-addressable node. This identifier is unique over all time. For all interoperable use, this is an xs:string of 22 characters which contains a GUID represented in RFC 4122 Base64-URL form (so 16 bytes convert to a 22-byte string). For private use the PEID may be under namespace control, so the length may be less than 22, but non-zero.

6.1.15 [deleted]

6.1.16 RelativeTimeAttributeGroup

An attribute set that may define some instant, or interval starting at that instant, but may also be empty.

- Attribute offset [Optional, xs:duration] is the specified instant. This is a positive or negative offset from some predefined event, usually the start of flight.
- Attribute interval [Optional, xs:duration] is optional, and specifies a duration before or after the offset.

6.1.17 ReportDataType

Carries a name-value pair to convey custom data.

6.1.18 RepositoryAppKeyType

The RepositoryAppKeyType provides a URL pointing to application data in a yet-to-be-defined repository format. This data will, at a minimum, identify the location of the application source code and define mappings from application-specific Event IDs to globally-recognized EPTYPES.

- Attribute AppDataReference (xs:string) contains the URL.

6.1.19 Scte35SegmentationUpidType

Describes a data model to facilitate in-band content asset identification. It is copied from [SCTE 130-2] into the CableLabs namespace.

The SegmentationUpid element's value is of type xsd:hexBinary and contains the [SCTE 35] segmentation_upid bit field. The value SHOULD NOT be empty. The value is specific to the @type attribute and SHALL meet the requirements as specified in SCTE 35. See [SCTE 35] for additional information.

Scte35SegmentationType includes the following attributes:

- Attribute type [Required, xsd:unsignedByte] is any valid value from the [SCTE 35] Table 8-6 Type column where the attribute maps to the SCTE 35 segmentation_upid_type bit field. See [SCTE 35] for additional information.
- Attribute length [Optional, xsd:unsignedByte] is any valid value from the [SCTE 35] Table 8-6 Length Bytes column. The length attribute's value is the binary data length. The length value is dependent upon the type value and maps to the SCTE 35 segmentation_upid_length bit field. See [SCTE 35] for additional information.
- Attribute eventID [Optional, xsd:unsignedInteger] is the [SCTE 35] segmentation_event_id bit field. See [SCTE 35] for additional information.
- Attribute typeID [Optional, xsd:unsignedByte] is any valid value from [SCTE 35] Table 8-7 that maps to the segmentation_type_id bit field. See [SCTE 35] for additional information.
- Attribute segmentNum [Optional, xsd:unsignedByte] is an attribute conformant to the [SCTE 35] segment_num bit field description. See [SCTE 35] for additional information.
- Attribute segmentsExpected [Optional, xsd:unsignedByte] is an attribute conformant to the [SCTE 35] segments_expected bit field description. See [SCTE 35] for additional information.
- Attribute referenceDateTime [Optional, xsd:dateTime] is the date and time providing contextual reference.
- Attribute ##any [Optional] is any additional attribute from any namespace.

6.1.20 SyscodeType

A syscode is a unique human-readable identifier that is an xs:string of length 4. The identifier tends to represent a network insertion point (e.g., a splicer). These system-level syscodes are at the MSO level within a given market. Larger network areas are represented by syscodes that are themselves composed of other syscodes.

6.1.21 TimeRangeType

Defines an absolute time range. Includes the following data objects:

- Attribute starttime [Required, xs:dateTime] is the point in time marking the beginning of the range.
- Attribute endtime [Required, xs:dateTime] is the point in time marking the end of the range.

6.1.22 TimestampType

Indicates a point in time and an interval around that time. It includes the following data objects:

- RelativeTimeAttributeGroup. See definition in Section 6.1.16.
- Element StartTime [Required xs:dateTime] identifies a point in time.

6.1.23 TrackingType

The Tracking element is copied from [SCTE 130-2] into the CableLabs namespace. It provides carriage for an identifier composed of privately-defined attributes and data that SHALL be included in returned messages where required. When returned, it SHALL be an exact copy of the received original (i.e., the element is echoed back). The element's usage and return requirements are defined explicitly by the including specification. (For example, see SCTE 130-3.) The internal element information is opaque to all other logical services, as the data is implementation-specific. Typically, the value is assigned by an ADS to track a specific ad-asset instance.

The Tracking element's value is of type xsd:string and SHOULD NOT be empty (but may be if all the data is provided as attributes).

TrackingType includes the following attribute:

- Attribute ##any [Optional] is any additional attribute from any namespace.

6.1.24 UserInputType

UserInput is an xs:string used to store a user's selection to a poll, RFI, etc. This element is optional, as it will not be included in messages representing Service Metric updates.

6.1.25 UuidUrlType

UUID is a 16-byte integer, encoded using RFC 4122 Base64-URL form. This encodes to

```
ceiling((bytes)/3)*4 = 24
```

Base64 bytes less padding, making this type an xs:string of length 22.

6.1.26 VersionType

A Version element that provides programmatically-accessible values for the major, minor, and micro version of the SaFI schema under which the document was created. Major and minor version are fixed at three and zero, respectively, in 3.0 releases. The micro version contains consecutive negative integers for Release Candidate versions, and consecutive non-negative numbers, starting with zero, in Release versions. The micro version is optional, and should be assumed to be zero when missing.

- Attribute majorVersion [Fixed at 3, PositiveIntType] is the major version of the interface schema.
- Attribute minorVersion [Fixed at 0, xs:int] is the minor version of the interface schema.
- Attribute microVersion [Optional, xs:int] is the micro version of the interface schema. Assumed to be zero if omitted.
- Attribute minSchemaVersion [Defaulted to 3, PositiveIntType] is a non-negative int representing the lowest compatible schema version number. A document with a minSchemaVersion of 3 is compatible with any processor implementing schema version 3 or greater, but not with a processor implementing schema version 2.
- Attribute minSchemaMinorVersion [xs:int] is an int representing the lowest compatible schema minor version number. A document with a minSchemaVersion of 1 and minSchemaMinorVersion of 1 is compatible with any processor implementing schema version 1.1 or greater, but not with a processor implementing schema version 1.0. If omitted, all minor versions are assumed to be compatible.

6.1.27 ZipcodeType

Zipcode is an xs:string of five decimal digits, or five digits followed by "-" and four additional digits. The contents are a US Postal Service ZIP code.

Annex A SaFI Common Data Types Model Schema (Normative)

The formal data definition is found in [SaFI COM XSD].

Appendix I Revision History

COM Version 3.0 is a major version, and as such does not require a deterministic EC history from the prior release. The following identifies the substantive changes from the last release, v2.0, January 31, 2011:

Revision	Content
1.	Update release version number from 2.0 to 3.0 in documents, schemas, namespaces, and examples.
2.	Update references to current revisions.
3.	Renamed SchemaVersionType to VersionType, and updated its documentation.
4.	Editorial changes throughout.