

CableLabs®
SaFI Specifications

Campaign Information Package Specification

CL-SP-SaFI-CIP-I01-090626

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 2008-2009 Cable Television Laboratories, Inc.
All rights reserved.

Document Status Sheet

| | | | | |
|-----------------------------------|--|----------------------|----------------------------------|-------------------|
| Document Control Number: | CL-SP-SaFI-CIP-I01-090626 | | | |
| Document Title: | Campaign Information Package Specification | | | |
| Revision History: | I01 - Released 6/26/09 | | | |
| Date: | June 26, 2009 | | | |
| 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:

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

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..... | 3 |
| 3 | TERMS AND DEFINITIONS..... | 4 |
| 4 | ABBREVIATIONS AND ACRONYMS..... | 5 |
| 5 | OVERVIEW..... | 6 |
| 5.1 | General Context..... | 6 |
| 5.2 | Specification Components..... | 8 |
| 6 | MSO CAMPAIGN INFORMATION PACKAGE INTERFACE REQUIREMENTS..... | 9 |
| 6.1 | Transmission Protocol..... | 9 |
| 6.1.1 | Messages..... | 9 |
| 6.2 | Data Model..... | 10 |
| 6.2.1 | Supported Product Families..... | 10 |
| 6.2.2 | Interactive Products..... | 10 |
| 6.2.3 | Ad Insertion Products..... | 11 |
| 6.2.4 | Identifiers..... | 11 |
| 6.2.5 | General Organization..... | 11 |
| 6.3 | Schema Component Specifications..... | 12 |
| 6.3.1 | Element <i>CampInfoPkg</i> (<i>CampInfoPkg</i>)..... | 13 |
| 6.3.2 | Element <i>Version</i> (<i>common:VersionType</i>)..... | 13 |
| 6.3.3 | Element <i>Gen</i> (<i>CipGeneralType</i>)..... | 13 |
| 6.3.4 | Element <i>Guid</i> (<i>GuidType</i>)..... | 14 |
| 6.3.5 | Element <i>StwShpData</i> (<i>StewardshipDataType</i>)..... | 14 |
| 6.3.6 | Element <i>CipVer</i> (<i>CampaignVersionType</i>)..... | 14 |
| 6.3.7 | Element <i>CipOrdrOwnr</i> (<i>IdentifierOwnerType</i>)..... | 14 |
| 6.3.8 | Element <i>CipOrder</i> (<i>IdentifierType</i>)..... | 14 |
| 6.3.9 | Element <i>CipCreDat</i> (<i>CampaignCreationDateType</i>)..... | 14 |
| 6.3.10 | Element <i>CipRevDat</i> (<i>CampaignRevisionDateType</i>)..... | 14 |
| 6.3.11 | Element <i>Ext</i> (<i>common:ExtType</i>)..... | 14 |
| 6.3.12 | Element <i>Contact</i> (<i>Inline in StewardshipDataType</i>)..... | 15 |
| 6.3.13 | Element <i>ProgEvent</i> (<i>ProgrammedEventType</i>)..... | 15 |
| 6.3.14 | Element <i>Status</i> (<i>RecordStatusType</i>)..... | 15 |
| 6.3.15 | Element <i>Flight</i> (<i>FlightType</i>)..... | 17 |
| 6.3.16 | Attribute Group <i>RelativeTimeAttributeGroup</i> (<i>common:RelativeTimeAttributeGroup</i>)..... | 18 |
| 6.3.17 | Element <i>Targets</i> (<i>TargetsDefinitionType</i>)..... | 18 |
| 6.3.18 | Element <i>ExplicitTarget</i> (<i>ExplicitTargetType</i>)..... | 19 |
| 6.3.19 | Element <i>TargetFilter</i> (<i>TargetFilterType</i>)..... | 19 |
| 6.3.20 | Element <i>FilterRule</i> (<i>FilterRuleType</i>)..... | 20 |
| 6.3.21 | Element <i>ProgramIdent</i> (<i>TargetProgramType</i>)..... | 20 |
| 6.3.22 | Element <i>PoFilter</i> (<i>FilterRuleType</i>)..... | 20 |
| 6.3.23 | Element <i>PlacementDirective</i> (<i>PlacementDirectiveType</i>)..... | 20 |
| 6.3.24 | Element <i>Product</i> (<i>ProductInstanceType</i>)..... | 21 |

6.3.25 *Element ProdFilter (FilterRuleType)* 22

6.3.26 *Element ProdControl (ProductControlType)* 22

6.3.27 *Element Places(PlacementDirectivePlacementType)*..... 22

6.3.28 *Element Time (PlacementGroup.Time)* 22

6.3.29 *Element SegmentationUpid* 22

6.3.30 *Element PlaceFilter (PlacementFilterType)*..... 23

6.3.31 *Element AvAssets (PlacementDirectiveAvAssetType)*..... 23

6.3.32 *Element AssetSelectFilter (FilterRuleType)*..... 24

6.3.33 *Element App (AppType)*..... 24

6.3.34 *Element BizTerms (BusinessTermsType)*..... 24

6.3.35 *Element RetData (ReturnDataType)*..... 25

6.3.36 *Element AppMsgs(Group AppMessageProcessingGroup)*..... 25

6.3.37 *Element SmMsgs (SmMessageProcessingGroup)* 25

6.3.38 *Element Package (PackageType)* 26

6.3.39 *Element PkgElement (PackageElementType)*..... 26

6.3.40 *MsoOrders Element (Inline in CampInfoPkg)*..... 26

6.3.41 *MsoOrderElement (MsoOrderType)*..... 26

6.3.42 *SysOrder Element (SystemOrderType)*..... 27

7 CAMPAIGN INFORMATION PACKAGE DATA MODELS.....28

Figures

FIGURE 5–1 - CONTEXT OF THE CAMPAIGN INFORMATION PACKAGE INTERFACE 7

FIGURE 6–1 - IDENTIFICATION MODELS AND SET-TOP MESSAGE IDENTIFICATION 10

FIGURE 6–2 - MSO CAMPAIGN INFORMATION PACKAGE ORGANIZATION 12

FIGURE 6–3 - RECORD STATE VALID TRANSITION DIAGRAM..... 17

Tables

TABLE 6–1 - RECORD STATE ENUMERATED VALUES 17

1 SCOPE

1.1 Introduction and Purpose

The purpose of this document is to specify the data model and protocols that comprise the MSO Campaign Information Package Interface.

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.

- [ADI2] ADI 2.0 Specification Asset Structure, MD-SP-ADI2.0-AS-I03-070105, January 5, 2007, Cable Television Laboratories, Inc.
- [CIPSCH] CL-SP-SaFI-CIP-I01-schematron.txt, June 26, 2009, Cable Television Laboratories, Inc.
- [CIPXSD] CL-SP-SaFI-CIP-I01.xsd, June 26, 2009, Cable Television Laboratories, Inc.
- [CIPWSDL] CL-SP-SaFI-CIP-I01.wsdl, June 26, 2009, Cable Television Laboratories, Inc.
- [ISO 19757] ISO/IEC 19757-3, Information technology — Document Schema Definition Languages —Part 3: Rule-based validation — Schematron, First edition, 2006-06-01.
- [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
- [RFC 4122] IETF RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace, July 2005.
- [RFC 4648] IETF RFC 4648, The Base16, Base32, and Base64 Data Encodings, October 2006.
- [WSDL] Web Services Description Language (WSDL) Version 2.0 Part 0: Primer W3C Recommendation, 26 June 2007.

Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Recommendation, 26 June 2007.

Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts, W3C Recommendation, 26 June 2007.
- [XML/SCH] XML Schema Part 0: Primer Second Edition, W3C Recommendation, 28 October 2004.

XML Schema Part 1: Structures Second Edition, W3C Recommendation, 28 October 2004.

XML Schema Part 2: Datatypes Second Edition, W3C Recommendation, 28 October 2004.

2.2 Informative References

This document uses the following informative references.

- [IAM] Interactive Application Messaging Specification, CL-SP-SaFI-IAM-I01-090626, June 26, 2009, Cable Television Laboratories, Inc.
- [SMS] Service Measurement Summary Interface Specification, CL-SP-SaFI-SMS-I01-090626, June 26, 2009, Cable Television Laboratories, Inc.
- [IAF] Interactive Application Fulfillment Summary Interface Specification, CL-SP-SaFI-IAF-I01-090626, June 26, 2009, Cable Television Laboratories, Inc.
- [CIP EXMPL] CL-SP-SaFI-CIP-I01-example1.xml, June 26, 2009, Cable Television Laboratories, Inc.
- [CIP HTML] CL-SP-SaFI-CIP-I01.html, June 26, 2009, Cable Television Laboratories, Inc.
- [SCTE 35] ANSI/SCTE 35 2007 - Digital Program Insertion Cueing Message for Cable.

[SCTE 130-2] SCTE 130-2 2008 - Digital Program Insertion–Advertising Systems Interfaces
Part 2–Core Data Elements.

[WSCT] CL-SP-SaFI-WSCT, 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>
- 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/>
- W3C, <http://www.w3.org>

3 TERMS AND DEFINITIONS

This specification uses the following terms:

| | |
|-------------------------|---|
| Affiliate | From the viewpoint of the CIP, an operational entity that performs SaFI operations with one or more MSOs. |
| Campaign | From an MSO execution perspective, a Campaign provides a set of delivery plans and/or placement directions for one or more MSOs, specific systems within the MSOs footprint, and finally, 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. |
| Package | A Package 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. A Programmed Event may include one or more Packages. |
| Package Element | An Package Element 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. |
| Package Type | 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) will identify 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. |
| MSO Order | An MSO Order is the part of a Campaign Information Package (CIP) that falls within a specific MSO's advertising footprint. |
| Programmed Event | A Programmed Event (e.g., a program, network spot, time shifted asset, guide page, an advertisement, etc.) represents a well defined and constrained subscriber experience that is identified for the purpose of including an enhanced experience. Each Programmed Event will have a Programmed Event Identifier (PEID). A primary function of the Programmed Event Identifier is to uniquely specify the context of application lifecycle events and act as a container for Packages to be delivered in the Programmed Event. |
| Schematron | A rule-based validation language for making assertions about the presence or absence of specific patterns in xml trees. |
| Syscode | An element whose text is a required four-character, pre-defined code that represents a specific zone-level cable plant. |
| System Order | A System Order is that part of an MSO Order that falls within a single zone-specific syscode. In simple cases, all the Programmed Events, Enhanced Packages and Enhanced 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:

| | |
|---------------|--|
| CIP | Campaign Information Package |
| EPSID | Enhanced Program Sequence ID, a small integer identifying a unique Enhanced Package or Enhanced Package Element within a specific Programmed Event |
| EpType | Enhancement Package Type |
| ETV | Enhanced Television |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol over Secure Socket Layer |
| PEID | Programmed Event ID, a globally-unique identifier for a Programmed Event |
| RFI | Request For Information |
| SaFI | Stewardship and Fulfillment Interfaces. A collection of interfaces defined by CableLabs to support advanced services on multiple cable systems |
| SOAP | Simple Object Access Protocol; as of SOAP 1.2, it no longer represents an acronym |
| WSDL | Web Services Description Protocol |

5 OVERVIEW

5.1 General Context

The MSO Campaign Information Package Interface informs all MSO systems that have a direct role in the preparation, delivery, execution, or reporting of the interactive campaign elements. The role of a document conforming to this interface is to provide a single, self-consistent description to an MSO of the actions they are expected to perform for the campaign and to enable complete characterizations of the resources required for those actions. In all cases, the actual performance of actions described are solely within the domain of the MSO. In a general sense, the primary MSO functions that will utilize the CIP are campaign management, which will use the CIP and to managing the MSO resources in accordance with those directions, and campaign reporting, which will use the CIP to characterize the reporting required for the campaign and set the reporting destination or destinations.

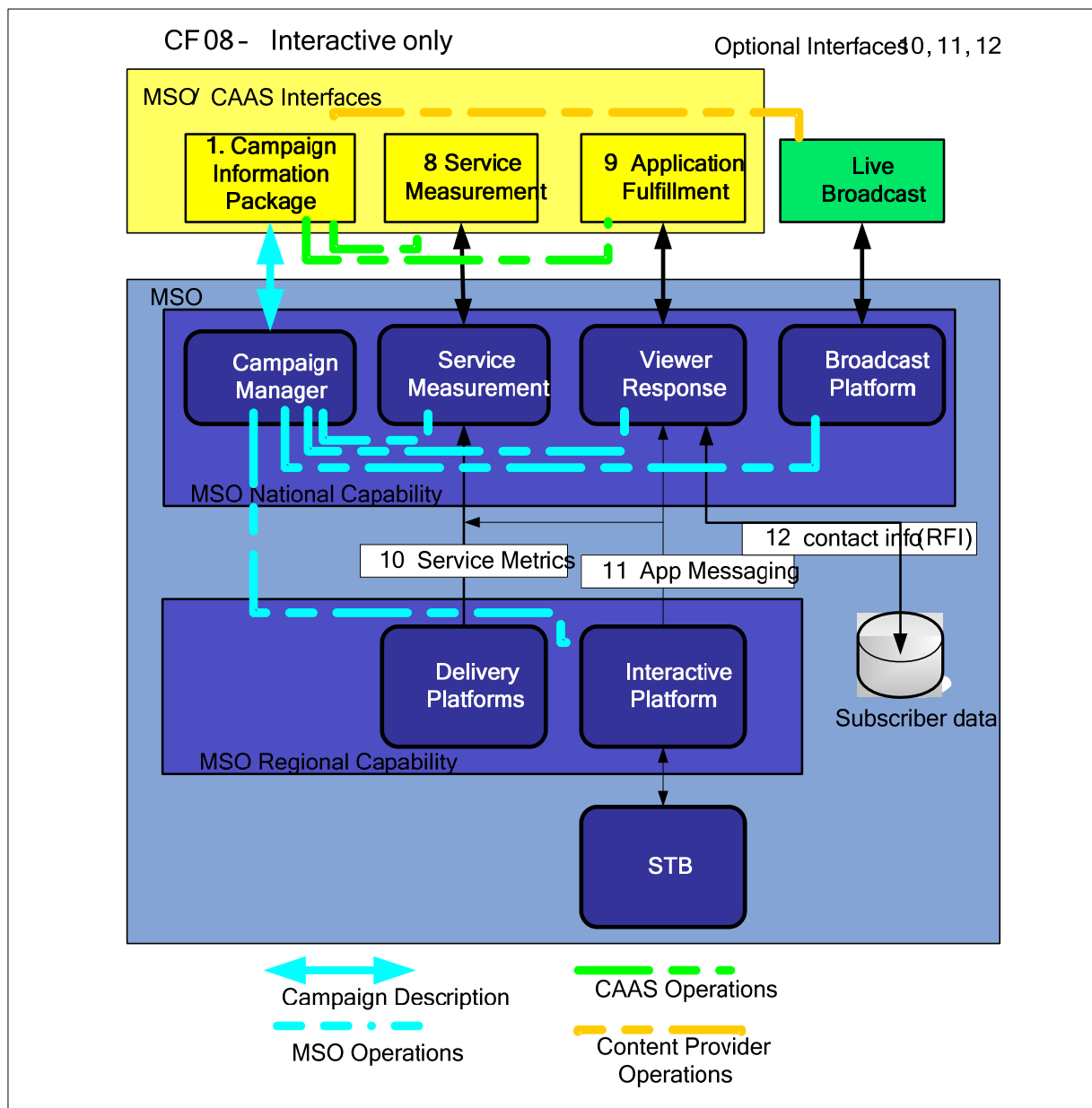


Figure 5-1 - Context of the Campaign Information Package Interface

The MSO Campaign Information Package describes the container element for the delivery of the Campaign details to the MSO over a defined interface. This specification provides the operational characterization of a Campaign that informs the MSO system elements responsible for application filtering, measurement and fulfillment. An MSO system or systems will only receive or process portions of a campaign relevant to its operation. The MSO Campaign Information Package identifies the MSOs and the syscodes within that MSO that are to participate in a campaign. The MSO Campaign Information Package covers the Affiliate’s instructions for content and application placement directives within Programmed Events, Packages within the Programmed Events, and Package Elements within the Packages. It is a subset of the broader campaign information managed between the Affiliate and the MSO’s systems, which also includes the operational characterization for the Affiliate and interactions with the agency, advertising customer, programmer, and any national third parties that may be involved.

NOTE: The I01 release also contains a set of elements relating to placed assets and placements that provided experimental support for a limited set of VOD ad insertion use cases. This facility is not intended for field use, and there should be no expectation for compatibility with future versions.

5.2 Specification Components

This specification consists of the following elements

1. This specification document, which is normative except as otherwise noted.
2. The associated XML schema file, CLP-SP-AA-CIP-D03-090505.xsd, which is normative.
3. The associated WSDL FILE, CLP-SP-AA-CIP-D03-090505.wsdl, which is normative.
4. The associated Xml Schema document set contained in CLP-SP-AA-CIP-D03-090505-html.zip, which is informative.
5. The associated CIP example file, CLP-SP-AA-CIP-D03-090505-example1.xml, which is informative.

6 MSO CAMPAIGN INFORMATION PACKAGE INTERFACE REQUIREMENTS

6.1 Transmission Protocol

This and other CableLabs web service definitions are developed under the guidelines issued in the CableLabs Web Services Transport specification [WSCT].

The CIP coordination protocol uses a subscription-based consumer/producer model for CIP documents. In this model, there are one or more CIP producer sites and one or more CIP consumer sites. Site is used here as an address providing the defined set of web services. Both the producer and consumer advertise web services in support of the CIP coordination protocol. The physical or logical location of these sites is not defined.

It is anticipated that there will be multiple CIP producer sites. CIP documents are created and modified only at the producer sites. Each Producer site receives and retains subscriptions, issue notices of availability against those subscriptions, and responds to requests for its CIP documents.

It is anticipated that each MSO will have one or more CIP consumer sites. Each is responsible for CIP processing on behalf of one or more syscodes, and a CIP consumer site subscribes on behalf of its syscodes. Following registration, the site is notified by any Producer site whenever a document that interacts with one or more of its subscribed syscodes is created or changed. Either in response to the notification, or for its own independent reasons, a consumer site can request a copy of a CIP document.

6.1.1 Messages

The web service implementation between CIP producers and consumers utilizes the following message set, defined in the schema.

6.1.1.1 *Element SetRegistrationRequest, SetRegistrationResponse*

The SetRegistration message SHALL be issued by a consumer site to a publication site to add or remove that consumer site from the distribution site's subscription list, or to maintain the set of syscodes the consumer site presents. The subscription list holds the current set of consumer sites and their subscribed syscodes.

If a SetRegistration is received with a syscode that is currently subscribed elsewhere, the response will indicate failure and include the list of syscodes that had such a conflict.

6.1.1.2 *NotifyRequest, NotifyResponse*

The Notify message is issued by a publication site to indicate to subscribers that there has been distributable change in a specific CIP document that is uniquely identified in the Notify message. Whenever a distributable change occurs in a CIP document at a publication site, the site SHALL issue a Notify message to each consumer site on the subscription list that is representing a syscode affected by the change.

6.1.1.3 *ReadRequest, ReadResponse*

The Read message SHALL be issued by a consumer site to obtain the current copy of a CIP document. The consumer site does not have to be on the producer site's subscription list in order to issue a Read. The Read request can identify a specific document or supply parameters to identify a set of documents. A parameterized query will return the first within the parameter set using a pre-defined ordering. A subsequent request can then be issued for the remainder of the set excluding the document just received.

A valid Read request receives a response that includes the selected CIP document.

6.2 Data Model

The MSO Campaign Information Package data model is defined by the normative XML schema with Schematron constraints that is contained in [CIPSCH]. Additional informative material for various data elements is presented in this section.

6.2.1 Supported Product Families

The data model of this CIP is intended to provide support for certain products within the following product families.

1. Vote/Poll products
2. Request for Information (RFI) products
3. OnDemand Advertising Insertion products (experimental)

6.2.2 Interactive Products

In Figure 6–1, a timeline is shown for a typical sports event with a total of three enhancement products applied during the event. They are Active Voting Package 1 and 2, and RFI Package. This diagram will be used to illustrate examples in this section.

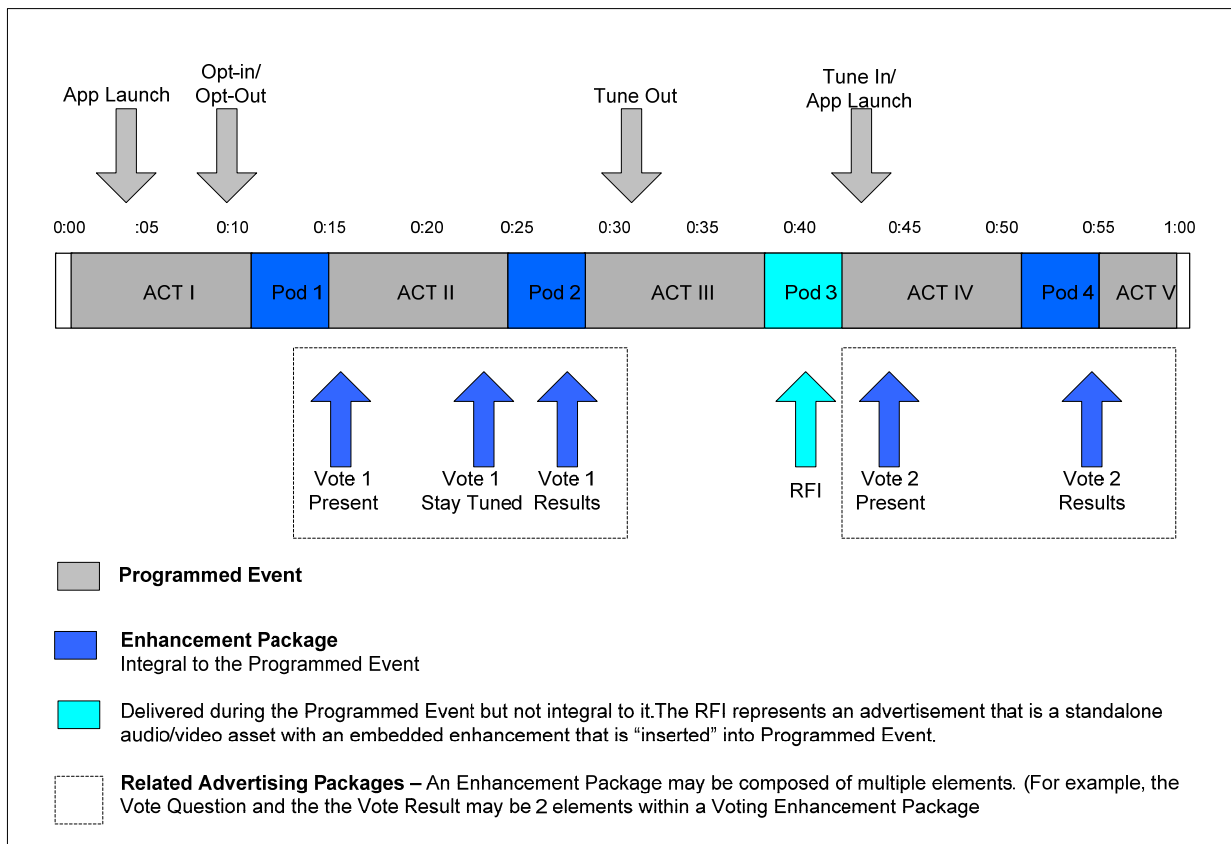


Figure 6–1 - Identification Models and Set-top Message Identification

6.2.3 Ad Insertion Products

This version of the CIP data model defines partial and preliminary support for the insertion of a second content file into a server-based offering of an initial content file. The CIP data model represents various elements using SCTE-130 compatible concepts and definitions. Although the I01 version is neither intended nor suitable for field use, this experimental facility is ultimately intended to fully support implementation in an SCTE-130 environment behind an ingesting campaign manager and SCTE-130 ADS.

6.2.4 Identifiers

This section describes the primary identifiers and their relation to return message processing. The host will be able to supply a device-specific identifier, but all other context for processing return messages must be supplied within the application. An arbitrary message must be placed in the correct context when many similar messages are arriving from the same or different applications in different programming or repeats of the same application in one program. To provide deterministic mapping of every message to its parsing and processing instructions and business context in the campaign, two identifiers are used. The first is the Programmed Event ID (PEID). As defined above, this is a universally-unique identifier for a specific programmed event. Within that programmed event, the enhancement components are identified with a local index that is unique only within the context of that specific PEID. This index is named the Enhancement Package Sequence ID (EPSID). There are three levels of enhancement components that can be identified by an EPSID. One is the Programmed Element itself, which is the context associated with application lifecycle events. The second is a Package – some specific set of functionality implemented by the application and identified as a package. The third is a Package Element, which is a portion of an enhancement that can also have an independent schedule and message processing configuration. Any time the application changes, a Package or Element container must be used to define the new application's messaging structure.

There is at least one PEID that is valid at every instant of the programmed event. That is the value associated with the programmed event itself. Messages, e.g., application lifecycle messages, can always reference that PEID with an appropriate, predefined EPSID. In addition, during the period that a Package is active, there may be a separate PEID and/or EPSID associated with the package. If so, they are present in the Product element of the package's PlacementDirective, and messages will reference that pair. Similarly, when a Package Element is active, it may have a separate PEID and/or EPSID under its PlacementDirective's Product element, and if so, messages will reference that pair.

6.2.5 General Organization

This section addresses the characteristics of the individual data elements that participate in the data model. The general organization of the MSO Campaign Information Package data is shown in Figure 6–2. This shows a generally hierarchical model of a Campaign Information Package containing one or more MSO orders, each containing one or more System Orders, each containing one or more Programmed Events, each with its operational data.

However, Programmed Event data are permitted directly within the Campaign Information Package, within an MSO order, and within a System Order. The higher level representations of these data supply default values for the lower levels, one XML element at a time. The Programmed Event data are actually interpreted only for each System Order, with the applicable data being the set of XML elements specifically found or defaulted at that System Order element. A higher level default can be nullified at a lower level with an appropriate empty element or an element with appropriate empty attribute values. Note that for any data element that is controlled by a Status element (Section 6.3.14), only the currently valid copy of the data element may be used for any purpose, including defaulting.

The instructions controlling an advanced advertising product are actually contained within Placement Directives. These may exist directly under a Programmed Event, a Package, or a Package Element. Each of these elements is permitted to have only one Placement Directive (despite the intervening Placement Directives element), so if

multiple directives are required within one Programmed Event, they must be referenced through lists of Packages or Package Elements. These intervening elements can also serve to meaningfully structure the Placement Directives.

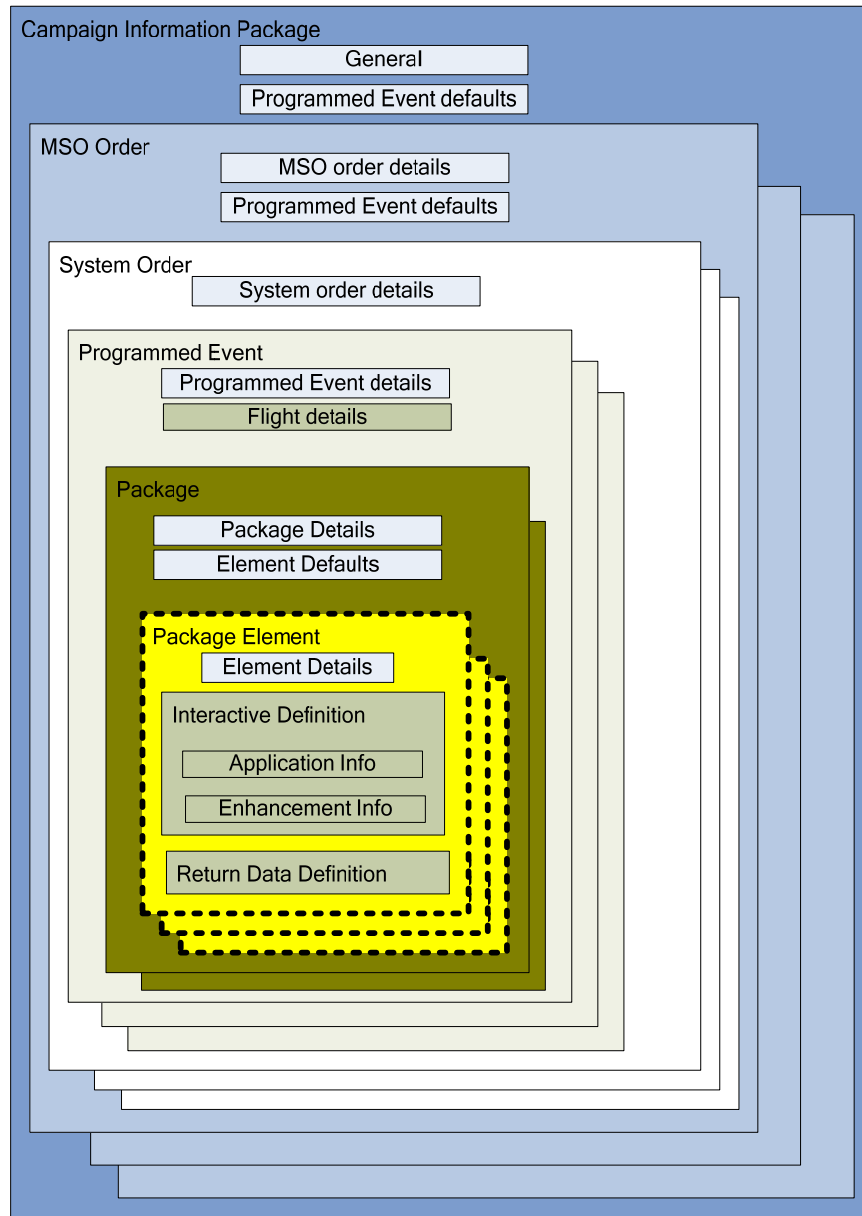


Figure 6–2 - MSO Campaign Information Package Organization

6.3 Schema Component Specifications

This section contains informational descriptions and normative relational and content definitions for individual schema components. Each section name contains the schema type and schema name that appear in the XML, followed by the parenthesized type or element name to reference in the documentation file index. References without a namespace prefix are within the CIP schema.

6.3.1 Element CampInfoPkg (CampInfoPkg)

This is the root document for describing an MSO Campaign Information Package.

It contains the following data units.

- Element Version. See 6.3.2.
- Element Gen., the general data element – This contains identification, version, and reference linkages for the Campaign. See 0.
- Element ProgEvents (Programmed EventsType) – an optional element pointing to a set of ProgEvent (ProgrammedEventType) defaults for all MSO orders. See 6.3.13.
- Element MsoOrders – A required element containing a sequence of at least one MSO Order (see 6.3.40).
- Element Ext – An optional extension element. See 6.3.11.

6.3.2 Element Version (common:VersionType)

This is a schema version control element, not a document instance version element. It contains the following attributes:

- MajorVersion (common:positiveIntType) – An attribute holding the CIP schema major version number under which this document was prepared.. Different major versions are generally not backward-compatible, although transforms to bring a lower version forward to a later version will generally exist.
- MinorVersion (common:nonNegativeIntType) – An attribute holding the CIP schema minor version number under which this document was prepared. All minor versions of the schema are backward-compatible within a major version.
- MinSchVer (common:positiveIntType) represents the minimum version of the CIP schema that will validate this particular document. For the I01 release, the minimum schema version is one. In later versions, however, it may be possible for a document to using avoid non-compatible constructs and so provide specific compatibility with a generally incompatible prior version.

6.3.3 Element Gen (CipGeneralType)

The Campaign General Element holds a single type which contains most of the scalar data for the MSO Campaign Information Package.

The General element contains the following units:

- Element Guid, a campaign-specific UUID. See 6.3.4.
- Element StwShpData – Stewardship Data element, which is a set of data identifying this specific campaign information in the context of the controlling stewardship entity. See 6.3.5.
- Element CipVer, CIP document version, see 6.3.6,
- Element CipOrdrOwnr, the owner of the order this CIP represents, see 6.3.7.
- Element CipOrder, the owner's ID assigned to the order this CIP represents. See 6.3.8.
- Element CipCreDat, creation date of the CIP. See 6.3.9.
- Element CipRevDat, date this version of the CIP was created. See 6.3.10.
- Element Ext, extension element. See 6.3.11.

6.3.4 Element Guid (GuidType)

The Guid element is empty and contains two attributes.

- Attribute guid (xs:string). This SHALL be a non-empty string that is UUID as defined by [RFC 4122], which defines a 16-byte integer. In CIP documents, it is encoded using RFC 4122 Base64-URL form, so encodes to $\text{ceiling}(\text{bytes}/3)*4 = 24$ Base64 bytes less padding, so 22 total. A new UUID SHALL be assigned to each CampInfoPkg. The UUID is not changed when the CampInfoPkg is updated.
- Attribute gname (xs:string), which is a Human-readable name associated with every UUID to allow meaningful communication. It SHOULD be chosen to have a very low probability of duplication, but uniqueness is neither required nor testable.

6.3.5 Element StwShpData (StewardshipDataType)

A Stewardship Data element holds the association between the Campaign Information Package and the underlying stewardship data from which it is derived, and generated this CIP.

The StwShpData element consists of the following data units:

- Attribute IdOwner (xs:anyURI). The required Id owner SHALL be a non-empty string set to a registered Internet domain name or a qualified domain name that is rooted in a registered internet domain name belonging to the provider of the identifier.
- Attribute id (xs:string). The required id attribute identifies this specific campaign in the context of the IdOwner. The identifier SHALL be a non-empty string unique within the namespace of its IdOwner.
- A series of one or more Contact elements identifying contacts at the stewardship organization. See 6.3.12.

6.3.6 Element CipVer (CampaignVersionType)

A non-negative int element identifying the Cip owner's version assigned to the specific CIP document. The version SHALL be increased each time a duplicate CIP (as identified by GUID) is distributed that contains any content difference.

6.3.7 Element CipOrdrOwnr (IdentifierOwnerType)

A string element identifying the entity that owns the order from which this CIP is derived. All identifier owners SHALL be a non-empty string set to a registered Internet domain name or a qualified domain name that is rooted in a registered internet domain name belonging to the provider of the identifier.

6.3.8 Element CipOrder (IdentifierType)

Cip order is a generic identifier of the order from which this CIP is derived, assigned by the order owner. The identifier SHALL be a non-empty string that is unique within the namespace of the Identifier Owner.

6.3.9 Element CipCreDat (CampaignCreationDateType)

An XML date-time indicating the time the initial version of the Campaign Information Package was created.

6.3.10 Element CipRevDat (CampaignRevisionDateType)

An XML date-time indicating the time this version of the Campaign Information Package was revised.

6.3.11 Element Ext (common:ExtType)

The general extensibility element. It permits elements from any namespace and any attributes.

6.3.12 Element Contact (Inline in StewardshipDataType)

A contact element is empty with a set of attributes as follows.

- Attribute name (xs:string) is a required non-empty string and SHALL be the name of a person or group that can respond to issues appropriate to the related role.
- Attribute role (xs:string) is a required non-empty string and SHALL identify the area of responsibility of the person or group identified by the related name.
- Attribute phone (xs:string, max length 25) is optional, and is a contact phone number for name.
- Attribute email (xs:string, max length 100) is optional, and is a contact email address for name.

While phone and email are both optional, at least one of them SHALL be supplied for each contact.

6.3.13 Element ProgEvent (ProgrammedEventType)

A Programmed Event represents the portion of the Campaign order at one operational location (syscode) falling within one defined set of programming events (see "Programmed Event" definition in Section 3). The ProgEvent element provides the operational characterization of the campaign placements, as well as the identity of the programmed event(s) in which they are embedded. The components are:

- Element Status (RecordStatusType), a required element which defines the ProgEvent as an updatable section. See 6.3.14 for details.
- Element Peid (common:UuidUrlType), the identifier for this ProgEvent.
- Element Flight (FlightType), which defines the temporal domain in which this ProgEvent may be applied. See 6.3.15.
- Element Targets (TargetsDefinitionType), which identifies the programming to which this ProgEvent applies, either as explicit program identifiers or as a set of attribute filters, or both. See 6.3.17.
- Element PlacementDirectives (PlacementDirectivesType), which points to a set of elements of PlacementDirective (PlacementDirectiveType), each of which specifies the information to defining one placement and control all of its placement instances (see 6.3.23). These directives may be operational as part of the ProgEvent, or may exist only to provide defaults for child Packages (see 6.3.38).

NOTE: In version I01, each PlacementDirectives element SHALL have one and only one PlacementDirective element as a child.

- Element Packages (ProgrammedEventType.Packages), an optional sequence of one or more Package (PackageType) elements that are interactive components of the ProgEvent (see 6.3.38).
- Ext (common:ExtType), the general extensibility element (see 6.3.11).

6.3.14 Element Status (RecordStatusType)

The Status element is the only member of a RecordStatusGroup. This element supplies information for interpreting its parent element instance. Status child elements provide update control at several hierarchical levels of a CIP document; MsoOrder, SysOrder, ProgrammedEvent, and PlacementDirective. Each occurrence of one of these elements and its descendants is defined as an "updatable section", and the parent always contains a unique section identifier which correlates the instances of that section across different updated document versions. These are, respectively, the Mso and Syscode identifiers, the ProgrammedEvent PEID, and the PlacementDirective PEID/EPSID.

If an updatable section has any modification after it is created, the original section is retained in the CIP document and a new copy of the section is appended immediately after, containing the modified records. These copies are

differentiated by the contents of their Status element, especially the revision numbers and the recRevoked flag. An updatable section instance is initially issued as revision one with its recRevoked flag set to false. That specific copy of the updatable section, as identified by its revision number, will not be modified in any way other than 1) changing the Status.recState and/or 2) setting Status.recRevoked to true when the section is superseded by a higher revision copy or deleted, and in either case setting the status RevDate to the time of the update. Changes are contained in the consecutive copies of the updatable section differentiated by sequential revision numbers. The highest revision numbered copy of an updatable section will either have its recRevoked set to false and be the effective section or have its recRevoked set to true because the section has been deleted. However, the processing appropriate to a copy of an updatable section is determined by the RecState, and independent of the RecRevoked flag. Only when the RecState is closed and the RecRevoked flag true is the copy in an immutably terminal state.

The copy and status control of an updatable section at a given hierarchical level does not propagate to any higher level or to any lower level. For example, a new revision of an instance at the ProgrammedEvent level will copy all descendent data, but only data that is within the ProgrammedEvent but not within any PlacementDirective need be inspected for updates. If any data within a PlacementDirective is also changed in the same CIP version, the new ProgrammedEvent revision will contain a separate copy and control of each changed PlacementDirective updatable section. In the same way, if the changes were fully contained within one or more PlacementDirective elements, it would not be necessary or desirable to issue a new revision of the ProgrammedEvent. Although the same policy applies to the MsoOrder and SysOrder levels, updates at those levels have no utility in I01 since all their contained data is within independent updatable sections.

In conformance with the above philosophy, the top level elements Version, Gen, and its descendants are considered an updatable section whose revision is the Gen@cipVer and whose section identifier is the GUID@guid. Any change in the cipVer requires that this section be inspected for updates. Since cipVer is incremented on any change in the CIP, it is not necessary that any changes will be present. All other changes will fall under the explicit control of some other lower level updatable section.

When a version of the CIP does not contain any reference to an updatable section (i.e., has no reference to a specific updatable section identifier), that section is unaffected by the CIP version. This policy permits a CIP to be issued incrementally, for example as a “sliding window” across the duration of the campaign. Updatable sections that were defined prior to those referenced in a particular CIP remain in the state that was last communicated, and those not yet addressed by a particular CIP will appear in a future version.

Taken as a whole, this update policy permits the history of an element to be retained in each issued MSO Campaign Information Package version (i.e., sequentially numbered document versions with a common GUID), so a document consumer can compare the current state to the prior state, or any older state that might be represented in some dependent systems. Note, however, the adoption of a revision is neither instantaneous nor simultaneous at all operational sites. Consequently, any CIP update that can affect reporting should be expected to result in the receipt of concurrent reporting messages derived from multiple versions of the CIP. Since the reporting formats in I01 do not include a CIP version, these cases can only be disambiguated by changes in the reporting identifiers: the PEIDs and EPSIDs. This requires that any CIP revision incorporating updates that can change the interpretation of reported data must terminate (Windup or Close) any ProgrammedEvent or PlacementDirectives containing those changes and create new updatable sections at revision one and with distinct identifiers. The RecRevoked flag will be set in the prior Revision of the section, and the chosen RecState will (eventually) control the permitted activity with respect to that revision.

Status is an empty element that takes the following attributes:

Attribute RecState (RecordStatusType) is a required string with enumerated values, defined in Table 6–1 below. This attribute describes the current campaign owner's direction with respect to the updatable section. RecState is not intended to reflect real-time changes in campaign execution, but rather the owner's current intent. A newly created updatable section may have any initial value; however, subsequent valid transitions between these states are shown in Figure 6–3.

Table 6–1 - Record State Enumerated Values

| Enumerated Value | Meaning |
|------------------|--|
| pnd (Pending) | Data is present, but may not be an operationally complete or consistent set. |
| com (Committed) | Data is present that is an operationally complete and consistent set (although not necessarily final). Data may be modified, even when active. |
| act (Active) | The data present is intended to be operational and the element should be operationally implemented according to the contained parameters. The actual time for various executions will vary by operational role (e.g., inserters before presentations). |
| wnd (Windup) | All executions related to the record should have ceased, but post-processing functions may continue. This includes upstream messaging within any pre-defined window. |
| cls (Closed) | All activity with respect to the record should be terminated. No additional revisions of the element may be provided after it is closed. |

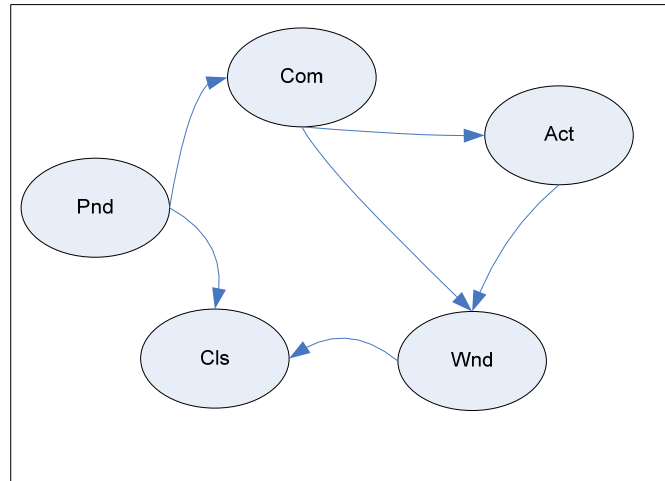


Figure 6–3 - Record State Valid Transition Diagram

- Attribute RecRevoked (RecordRevokedFlagType) is a required Boolean, set to false initially and modified to true if the updatable section is either deleted or superseded.
- Attribute Revision (ElementRevisionNumberType) is a required non-negative int supplying the revision number of this particular instance of the updatable section. The revision is incremented each time any component of the section with this peid is modified in any way, excluding changes to the value of this Status element.
- Attribute RevDat (ElementRevisionDateType) is the required date and time the updatable section was last modified, including changes to the status element only.

6.3.15 Element Flight (FlightType)

The Flight element defines the validity of the temporal domain of a ProgrammedEventType. No user-visible effects derived from a specific ProgrammedEventType record SHALL be visible outside of the time window defined by the Flight. All times XML date times in a Flight element SHALL include a timezone. The timezone MAY be one of “Z” when directly in UTC, the Eastern Time offset (Standard or Daylight as appropriate), or the offset of the

timezone of the program origin (also Standard or Daylight as appropriate). The Flight element has a number of attributes and children that define that domain. They are:

- Attribute number (FlightNumberType) is a campaign-assigned non-negative int identifying this flight.
- presBegin (FlightBeginType). A required presentation begin time is an XML date time indicating when a broadcast Programming Event is scheduled to start. For stored products, this is the time they become available to viewers.
- presEnd (FlightEndType). A required presentation end time is an XML date time indicating when a broadcast Programming Event is scheduled to end. For stored products, this is the time they cease to be available to viewers.
- rptBegin (FlightBeginType). An optional report begin time is an XML date time indicating the earliest XML date time that reporting will be accepted for the flight. The opening of the reporting window is defined as the presBegin time when this attribute is omitted.
- rptEnd (FlightEndType). An optional report end time is an XML date time indicating the latest time that reporting will be accepted for the flight. The closing of the reporting window is indeterminate when this attribute is omitted.
- TzCode (TimezoneCodeType) is an optional enumerated string containing a code for the timezone plan of the national feed. The interpretation of Flight times are to be in accordance with this overall timezone plan. Values are:
 - NATL - Aired simultaneously in all timezones from national feed,
 - EMP - Aired with no delay in Eastern, local delay in Mountain, and Pacific zones, Central shares Eastern,
 - LCL - Aired with local delay in all time zones.
 All times are to be considered absolute when this attribute is omitted. This is equivalent to the NATL option.
- Element PresStartAdj (common:RelativeTimeAttributeGroup) is an optional adjustment that applies only to broadcast programming and defines an interval in which the presBegin date time may be adjusted for programming schedule changes. There is no adjustment permitted when this element is omitted. The origin of the interval (offset) may fall before or on the presBegin time. The interval may extend to or after the presBegin time, i.e., the adjustment may not exclude the default presBegin time. See 6.3.16.
- Element PresEndAdj (common:RelativeTimeAttributeGroup) is an optional adjustment that applies only to broadcast programming and defines an interval in which the presEnd date time may be adjusted for programming schedule changes. There is no adjustment permitted when this element is omitted. The origin of the interval (offset) may fall before or on the presEnd time. The interval may extend to or after the presEnd time (i.e., the adjustment may not exclude the default presEnd time). See 6.3.16.

6.3.16 Attribute Group RelativeTimeAttributeGroup (common:RelativeTimeAttributeGroup)

This is a pair of attributes that defines an interval as an offset from a base time and an interval extending from the offset. Both attributes are of type XML duration and use that notation. Note that durations may be negative, so both the offset from the base time and the interval extending from the offset could be negative as well as positive. The base time must be specified by the context in which this attribute group is applied.

- Attribute offset (xs:duration) is a required XML duration defining a point in time with respect to a reference time.
- Attribute interval (xs:duration) is an optional XML duration defining an interval by specifying a second point in time with respect to the corresponding offset time.

6.3.17 Element Targets (TargetsDefinitionType)

This is a container for information defining or qualifying a set of target assets that make up a ProgrammedEvent. It may select targets by qualification rules, specific asset identifications, or both. Qualifications and explicit

identifications are individually optional since either can occur alone. However, at least one target selection criterion is required. When both explicit targets and filter rules exist, they SHALL be effectively logically OR'ed by the implementation.

In addition, the Targets element MAY list specific placement opportunity filters that are to be applied on a target-independent basis. (There may also be additional filters that may be target-specific or filter-specific).

Targets are composed of:

- Element TargetFilters (TargetFiltersType) is an optional element that holds a sequence of TargetFilter (TargetFilter Type) elements (see 6.3.19).
- Element ExplicitTargets (ExplicitTargetsType) is an optional element that holds a sequence of ExplicitTarget (ExplicitTargetType) elements (see 6.3.18).

6.3.18 ExplicitTarget (ExplicitTargetType)

Element ExplicitTarget (ExplicitTargetType) is an optional element that represents qualifications for one explicit target. It holds a single ProgramIdent (TargetProgramType) identifying a specific target asset. An ExplicitTarget may also have a set of PoFilters that apply to placement opportunities only when that identified target selection is true.

- Element ProgramedIdent (ProgramIdentType) identifies one permitted specific target (see 6.3.21).
- Element PoFilters (PoFiltersType) – an optional element that holds a sequence of PoFilter (PoFiltersType) elements. These filters are specific to the ExplicitTarget, and so are considered to be active for a proposed target if and only if the ProgramIdent is true for that target (see 6.3.22).

Each of the PoFilter elements that is directly or indirectly under an ExplicitTarget defines a boolean predicate filter that may select or reject each placement opportunity associated with that proposed target. That set of PoFilter elements for any target defines the placement opportunity predicates that are required to exist in the specific explicit target asset in order to apply the placement directives of this Programmed Event. Consequently, each PoFilter element is logically applied to each placement opportunity of the target, and the PoFilter returns the logical OR of the predicate across all those opportunities. When there are multiple active PoFilter instances, the predicates returned by each are logically AND'ed, and the proposed target may be selected if and only if the resulting predicate is true.

6.3.19 Element TargetFilter (TargetFilterType)

A TargetFilter is a container for a set of target FilterRule expressions and an optional set of related placement opportunity filter expressions. All target expressions are logically OR'ed, and the target filter may be true if and only if all those expressions evaluate to true.

Each of the PoFilter elements that are directly or indirectly under a TargetFilter defines a boolean predicate filter that may select or reject each placement opportunity associated with that proposed target. That set of PoFilter elements for any target defines the placement opportunity predicates that are required to exist in a proposed target asset in order to apply the placement directives of this Programmed Event. Consequently, each PoFilter element is logically applied to each placement opportunity of the target, and the PoFilter returns the logical or of the predicate across all those opportunities. When there are multiple active PoFilter instances, the predicates returned by each are logically AND'ed, and the filter may be true if and only if all the predicates are true.

The targetFilter contains:

- Element FilterRules (TargetFiltersType.FilterRules) is a required child that holds a set of FilterRule elements, each with a target filter expression (see 6.3.20).

- Element PoFilters (PoFiltersType) is an optional element that holds a sequence of PoFilter (PoFiltersType) elements. These filters are target filter specific, and so are considered to be active for a proposed target, if and only if the target FilterRule composite predicate is true for that target (see 6.3.22).

6.3.20 FilterRule (FilterRuleType)

A FilterRule uses the FilterRuleType which defines an enumerated attribute of RuleEngine and a body that holds a predicate expression that is valid in the rule engine and references only data externally defined as accessible in a TargetFilter filter rule. It is anticipated that such data may come from a PlacementRequest for a target, or the asset metadata associated with the asset identified by that PlacementRequest.

- Attribute RuleEngine (RuleProcessingEngineType) is an enumerated string containing XQuery or Undef. This identifies the expression engine that will be responsible for processing the contained expression. (This was named MsgProcessingEngineType and specific to message processing in D01.) The Undef type generally cannot be processed in the absence of proprietary extensions to this specification.
- Text(xs:string). The body of the TargetFilter element is a non-empty string of the predicate expression that is to be evaluated for each proposed target.

6.3.21 Element ProgramIdent (TargetProgramType)

A ProgramIdent element uses some attributes of the [SCTE 130-2] Program Element to identify a specific program. The ProgramIdent is considered true if a proposed target is an exact match to a contained identifier in any supported identifier space.

- Attribute uniqueProgramID (xs:nonNegativeInteger) is an optional attribute uniquely identifying the program.
- Attribute referenceDateTime [xs:dateTime] is an optional attribute identifying when the @uniqueProgramID attribute was established (i.e., contextual reference). This attribute should only be used when the @uniqueProgramID attribute is present.
- Text. The body text of type XML string may further identify the program, or may be empty.

6.3.22 Element PoFilter (FilterRuleType)

A PoFilter uses the FilterRuleType that defines an enumerated attribute of RuleEngine and a body that holds a predicate expression that is valid in the rule engine and references only data externally defined as accessible in a PoFilter. It is anticipated that such data may come from the PlacementOpportunities of a PlacementRequest for a target, or the placement metadata obtained from a POIS using the placement identification from the PlacementRequest.

- Attribute RuleEngine (RuleProcessingEngineType) is a required enumerated string containing XQuery or Undef. This identifies the expression engine that will be responsible for processing the contained expression. (This was named MsgProcessingEngineType and specific to message processing in D01.) The Undef type generally cannot be processed in the absence of proprietary extensions to this specification.
- Text. The body of the PoFilter element is the predicate expression that is to be evaluated for each proposed placement opportunity.

6.3.23 Element PlacementDirective (PlacementDirectiveType)

A PlacementDirective defines all information required to implement a placement. In any specific case, it must be populated as required by the definition of the specific product family of the placement. At execution time, the placement will be made in a target that is currently selected by the ProgEvent.Targets element, and the current time will be within the window(s) defined by the ProgEvent.Flight.

- Element Status (RecordStatusType) is a required element that defines the PlacementDirective as an updatable section. See 6.3.14 for details.
- Element Product (ProductInstanceType) is a required element that identifies the specific product family and instance of that family being delivered by the placement directive (see 6.3.24).
- Element Places(PlacementDirectivePlacementType). The required Places element identifies one or more specific placement filter expressions that apply to this particular placement directive. These will typically be a subset of the selected target's placement filters. See 6.3.27.
- Element AvAssets (PlacementDirectiveAvAssetType). The optional AvAssets element identifies in a set of one or more AvAssetIdent (AvAssetIdentType) specific audio-visual assets that apply to this placement directive. The AvAssets element is missing for product families that do not utilize audio-visual assets. When more than one asset is defined, the assets can also include expressions to select among the possible assets. See 6.3.31.
- Element App (AppType) is an optional element that identifies the application that is to be executed as part of this placement directive, provided an application is part of the product family. If no application is required, the App element is missing. See 6.3.33.
- Element BizTerms (BusinessTermsType) is an optional element that provides data for an SCTE-130 ADS or similar decision engine to choose between conflicting placement requires. See 6.3.34.
- Element RetData (ReturnDataType) is a required element that specifies the message identification, parsing, and forwarding for supporting messages identified by this NodeRef's PEID/EPSID. See 6.3.35.

6.3.24 Element Product (ProductInstanceType)

A Product element identifies the product family of a placement directive and supplies identifiers that tie this product instance to the particular product request from the initial order. The placement directive may provide for one or many execution occurrences of this specific product instance. The Product may also have children that filter the launching of occurrences by product-specific criteria, or that define specific controls to apply to occurrences that are launched. The Product element labels an MSO Campaign Information node to permit direct references to that node in return messages and supplies the identifiers that will implement such references.

A Product element takes the following attributes.

- Attribute peid (Programmed Event Identifier) is a required attribute that SHALL be a unique value conforming to the definition of a GUID as described in Section 6.3.4, and serves to identify one or a set of labeled Product nodes. Although PEIDs SHALL be unique across the Programmed events of all campaigns, the same PEID can be used to label multiple notes of one Programmed Event when that is appropriate. In the case that nodes of a Programmed Event share duplicate PEIDs, they SHALL have distinct EPSIDs.
- Attribute gname (xs:string) is an optional human readable supplement to the peid. It is desirable that the gname value be both meaningful and unique.
- Attribute epsid (common:EpsidType) is a required index assigned within the ProgrammedEvent for identifying functional groups of enhancements. An EPSID is scoped by its PEID, and must be distinct in Product nodes of one Programmed Event that share the same PEID.
- Attribute ProductFamily (ProductFamilyType) is an optional enumerated string attribute that must be one of a predefined set of known and approved enhanced product families. The family is used to direct the interpretation of the CIP data model for a specific placement directive. The values defined in this schema are "VotePoll", "RFI", and "OnDemandInsertion".
- Element ProdFilter (FilterRuleType) is an optional set of child elements that can define product-specific filters to control occurrences of this placement directive. See 6.3.25.

- Element ProdControl (ProductControlType) is an optional child element that can carry product-specific controls to be applied to occurrences of the placement directive. See 6.3.26.

6.3.25 Element ProdFilter (FilterRuleType)

A ProdFilter uses the FilterRuleType, which defines an enumerated attribute of RuleEngine and a body that holds a predicate expression that is valid in the rule engine and references only data externally defined as accessible in a ProdFilter. It is anticipated that appropriate data will be defined for each product family. For example, in an OnDemandInsertion product, the data set might provide total impressions, impression frequency, or other system context references. The system providing the rule engine will supply the values for these fields as part of the expression engine environment.

- Attribute RuleEngine (RuleProcessingEngineType) is a required enumerated string containing XQuery or Undef. This identifies the expression engine that will be responsible for processing the contained expression. (This was named MsgProcessingEngineType and specific to message processing in D01.) The Undef type generally cannot be processed in the absence of proprietary extensions to this specification.
- Text. The body of the ProdFilter element is the predicate expression that is to be evaluated for each proposed target.

6.3.26 Element ProdControl (ProductControlType)

A ProdControl element supplies values on a product-family basis that control run-time aspects of the expression of that family. The only current datum is the specification of trick mode controls for OnDemandInsertion products.

Attribute trickMode (TrickModeControlType) is an optional enumerated string of trick mode control values to be applied during this placement.. The enumeration is currently empty.

6.3.27 Element Places(PlacementDirectivePlacementType)

The Places element may contain an optional Time element, an optional SegmentationUpid, or a set of zero or more specific placement filter expressions in PlaceFilter elements that apply to this particular placement directive. These will typically be a subset of the selected target's placement filters. Specifically, a Places element contains at least one element from the set of:

- Element PlaceFilter (PlacementFilterType). A filter expression that selects the placement opportunity associated with this particular Placement Directive. If there are multiple PlaceFilters, they are evaluated individually and their predicate values logically OR'ed. For any given placement opportunity, if the single or collective value is false, the placement directive cannot have a launched occurrence. See 6.3.28.
- Element Time.
- Element SegmentationUpid.

6.3.28 Element Time (PlacementGroup.Time)

The time element is an empty element that accepts two attributes:

- Offset is a required value specifying the relative time with respect to the Programmed Event actual start time at which this Time element takes effect, as an XML duration.
- Interval is an optional value specifying the duration of the interval this Time element covers with respect to the offset. A missing interval attribute specifies an instantaneous event. The interval is an XML duration.

6.3.29 Element SegmentationUpid

This element contains:

- Attribute type [Required, xsd:unsignedByte]. Any valid value from [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]. Any valid value from [SCTE 35] Table 8-6 Length Bytes column and 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]. The [SCTE 35] segmentation_event_id bit field. See [SCTE 35] for additional information.
- Attribute typeID [Optional, xsd:unsignedByte]. 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]. An attribute conformant to the [SCTE 35] segment_num bit field description. See [SCTE 35] for additional information.
- Attribute segmentsExpected [Optional, xsd:unsignedByte]. An attribute conformant to the [SCTE 35] segments_expected bit field description. See [SCTE 35] for additional information.
- Attribute referenceDateTime [Optional, xsd:dateTime]. The date and time providing contextual reference.
- Attribute ##any [Optional]. Any additional attribute from any 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.

6.3.30 Element PlaceFilter (PlacementFilterType)

A PlaceFilter uses the FilterRuleType, which defines an enumerated attribute of RuleEngine and a body that holds a predicate expression that is valid in the rule engine and references only data externally defined as accessible in a PoFilter. It is anticipated that such data may come from the PlacementOpportunities of a PlacementRequest for a target, or the placement metadata obtained from a POIS using the placement identification from the PlacementRequest. It is expected that a PlaceFilter will generally be a subset of the active Target PoFilters.

- Attribute RuleEngine (RuleProcessingEngineType) is a required enumerated string containing XQuery or Undef. This identifies the expression engine that will be responsible for processing the contained expression. (This was named MsgProcessingEngineType and specific to message processing in D01.) The Undef type generally cannot be processed in the absence of proprietary extensions to this specification.
- Text. The body of the PlaceFilter element is the predicate expression that is to be evaluated for each proposed placement opportunity.

6.3.31 Element AvAssets (PlacementDirectiveAvAssetType)

This is a container for information identifying a set of one or more placement assets that may be used by the parent placement directive. The children are AvAssetIdent elements (AvAssetIdentType), which have asset identifiers, and, optionally, selection filters.. Each AvAssetIdent contains identifiers for a specific asset. These identifiers may come from a number of name domains. However, the only identifiers currently supported are the CableLabs ADI identifiers.

- Attribute providerID (xs:string) is a non-empty string that is the placement asset's providerID formatted as defined in [ADI2].
- Attribute assetID (xs:string) is a non-empty string that is the placement asset's assetID formatted as defined in [ADI2].

- Element `AssetSelectFilter` (`FilterRuleType`) is a set of zero or more selection filters to apply to this asset. If there is a single `AvAssetIdent` element, it is always used for placement. When there is more than one `AvAssetIdent` element, the asset selection filters are evaluated to determine which one to apply to a given placement. If there are multiple `AssetSelectFilter` elements for one `AvAssetIdent`, the predicates of the expressions are logically OR'ed. Any `AvAssetIdent` with a true predicate may be selected for a given placement. If more than one has a true predicate, the choice among them is implementation-specific, as the order of evaluation is undefined.

6.3.32 Element `AssetSelectFilter` (`FilterRuleType`)

An `AssetSelectFilter` uses the `FilterRuleType` which defines an enumerated attribute of `RuleEngine` and a body that holds a predicate expression that is valid in the rule engine and references only data externally defined as accessible in a `ProdFilter`. It references a specific set of externally defined data elements created for this purpose. It includes elements such as impression counts for the asset for rotation or episode-based selection from a set of assets. The system providing the rule engine will supply the values for these fields as part of the expression engine environment.

- Attribute `RuleEngine` (`RuleProcessingEngineType`) is a required enumerated string containing `XQuery` or `Undef`. This identifies the expression engine that will be responsible for processing the contained expression. (This was named `MsgProcessingEngineType` and specific to message processing in D01.) The `Undef` type generally cannot be processed in the absence of proprietary extensions to this specification.
- Text. The body of the `AssetSelectFilter` element is the predicate expression that is to be evaluated for each proposed target.

6.3.33 Element `App` (`AppType`)

The optional `App` element supplies sufficient data to launch an application as part of a placement directive and also identifiers to locate the context for processing returned data from that application. If the product family of a given placement directive does not include launching an application, the `app` element is omitted. When present, the `App` element contains:

- Attribute `orgId` is an organization identifier in the format defined in [MHP 1.1.2], section 10.5.1.
- Attribute `appId` is an application Id in the format defined in [MHP 1.1.2], section 10.5.1.
- Attribute `appVer` (`xs:unsignedInt`) is an integer version number identifying the specific release of the application.
- Attribute `AppDataReference` (`RepositoryAppKeyType`) is a URL pointing to application data in a yet-to-be-defined repository format. This data will, as a minimum, identify the location of the application source code and define mappings from application specific `EventIds` to globally recognized `EPTYPES`.
- Attribute `epTypeReference` (`AppMessageTemplateType`) is a URL pointing to `EPTYPE` data in a yet-to-be-defined repository format. The `EPTYPE` data is expected to provide processing rules to be applied in converting [IAM] messages to [IAF] and [SMS] messages.
- Attribute `epType` identifies a specific active message set within the application. The `epType` attribute must be either actually present or defaulted for any `PlacementDirective` element appearing in a node at or below the System Order level.

6.3.34 Element `BizTerms` (`BusinessTermsType`)

The `BizTerms` element holds data to permit an SCTE 130 ADS or equivalent system to choose between conflicting placements as a business decision. The element is currently a placeholder with a single element representing some future composite valuation. It consists of:

- Attribute Priority (PriorityType) is a placeholder for business terms used by SCTE ADS or equivalent to choose among conflicting potential placements. Priority takes an integer value from zero to one hundred, inclusive. Higher values of priority represent more desirable placements.

6.3.35 Element RetData (ReturnDataType)

The Return data element characterizes the process for processing and forwarding returned data.

NOTE: EventIDs and related parsing rules described in this section are application-specific and out of scope of this specification. See the application-specific documentation for more information.

RetData is an empty element with three components:

- The PriUrl element (MessagePrimaryUrlType) contains the URL to which application messages are directed for primary processing. This URL typically represents a local MSO collector. The common URL supplied here is required to resolve to the appropriate local collector at every syscode of the Programmed Event.
- An AppMsgs element (inline in RetData) element is a container for a series of elements that characterize the messages and processing appropriate to application messages issued by the application in the context of this interaction node. See 6.3.36.
- An SmMsgs element (inline in RetData) is a container for a series of elements that characterize the messages and processing appropriate to service metrics messages issued by the application in the context of this interaction node. See 6.3.37.

6.3.36 Element AppMsgs(Group AppMessageProcessingGroup)

The AppMsgs element contains:

- Element ProcRules (inline in AppMessageProcessingGroup). An optional element that can define processing of upstream application messages issued within this interaction. The text body contains a processing script, and the appEngType attribute selects a processing engine that interprets the script. The script transforms the received host messages into the appropriate upstream format (probably utilizing the message EventID).
- Element AppSecUrl (inline in AppMessageProcessingGroup). An element whose body is the URL to which secondary messages, i.e., those emitted from the primary processor, are directed.
- Ext (common:ExtType). The general extensibility element. See 6.3.11.

6.3.37 Element SmMsgs (SmMessageProcessingGroup)

The SmMsgs element contains:

- Element ProcRules (inline in SmMessageProcessingGroup). An optional element that can define processing of upstream application messages issued within this interaction. The text body contains a processing script, and the smEngType attribute selects a processing engine that interprets the script. The script transforms the received host messages into the appropriate upstream format (probably utilizing the message EventID).
- Element SmSecUrl (inline in SmMessageProcessingGroup). An element whose body is the URL to which secondary messages, i.e., those emitted from the primary processor, are directed.
- Ext (common:ExtType). The general extensibility element. See 6.3.11.

6.3.38 Element Package (PackageType)

A Package element represents one or more instances of a predefined Advanced Advertising enhancement implemented in one or more interactive applications. A Package with no child Package Elements SHALL either define placements or inherit default placements from the ProgEvent. A Package that does contain child Package Elements MAY provide placements that serve as defaults for those children.

A Package Element is of PackageType, and consists of the following data units.

- Element PlacementDirectives (PlacementDirectivesType). An optional container for a set of one or more PlacementDirective elements. These directives may be operational as part of the Package, or may exist only to provide defaults for child PackageElements. To the extent that any PlacementDirective at this level is missing a required component, it SHALL be considered to include the corresponding component from the corresponding PlacementDirective at the ProgEvent level (see 6.3.13). If no such PlacementDirective or component exists, the local PlacementDirective SHALL be rejected as malformed.
- Element PkgElements (inline in PackageType). An optional sequence of one or more PkgElement that are interactive components of the Package (see 6.3.39).
- Ext (common:ExtType). The general extensibility element. See 6.3.11.

6.3.39 Element PkgElement (PackageElementType)

A PackageElement element represents one or more instances of a predefined Advanced Advertising enhancement implemented in one or more interactive applications. A PackageElement SHALL provide placements.

A PackageElement Element is of PackageElementType, and consists of the following data units.

- Element PlacementDirectives (PlacementDirectivesType). An optional container for a set of one or more PlacementDirective (PlacementDirectiveType) elements. To the extent that any PlacementDirective at this level is missing a required component, it SHALL be considered to include the corresponding component from the corresponding PlacementDirective at the Package level (see 6.3.38) or ProgEvent level (see 6.3.13). If no such PlacementDirective or component exists, the local PlacementDirective SHALL be rejected as malformed.
- Ext (common:ExtType). The general extensibility element. See 6.3.11.

6.3.40 MsoOrders Element (Inline in CampInfoPkg)

An MsoOrders element is an empty element that serves as a container for a set of one or more MsoOrder (MsoOrderType) elements. Each MsoOrder element holds all the information related to the Campaign for a single MSO (see 6.3.41).

6.3.41 MsoOrderElement (MsoOrderType)

The MsoOrder contains the following data units.

- Attribute name (MsoNameType). This required non-empty attribute is the name of the MSO who operates the included syscodes.

NOTE: MSO name elements must be globally unique. When a Syscode level element is updated, the value of the syscode is the identifier that correlates the set of records with different status values. If the syscode is not unique, it creates ambiguity in the membership of these record sets.

- Element Status (RecordStatusType), a required element that defines the MsoOrder as an updatable section. See 6.3.14 for details.

- Element ProgEvents (ProgrammedEventsType). An optional element pointing to a set of ProgEvent (ProgrammedEventType) defaults for all SysOrders of the MsoOrder (see 6.3.13). If any ProgEvent exists in the parent CampInfoPkg.ProgEvents with the same ProgEvent.Guid as a ProgEvent defined here, the local definition replaces the parent definition. All other ProgEvent elements defined for the MsoOrder augment the set defined for the CampInfoPkg.
- Element SysOrders (inline in MsoOrder) – A required container holding a set of at least one SysOrder (see 6.3.42).
- Ext (common:ExtType). The general extensibility element. See 6.3.11.

6.3.42 SysOrder Element (SystemOrderType)

A SysOrder element holds the information related to the Campaign in a single zone syscode. It consists of a SysOrder element containing only a SystemOrderGroup.

- Element Syscode (SyscodeType) is an element whose text is a required four-character, predefined syscode that represents a specific zone-level cable plant.

NOTE: Syscode elements must be globally unique. When a Syscode level element is updated, the value of the syscode is the identifier that correlates the set of records with different status values. If the syscode is not unique, it creates ambiguity in the membership of these record sets.

- Element Status (RecordStatusType) is a required element that defines the SysOrder as an updatable section. See Section 6.3.14 for details.
- Ext (common:ExtType). The general extensibility element. See 6.3.11.
- Element ProgEvents (ProgrammedEventsType) is an optional element pointing to a set of ProgEvent (ProgrammedEventType) defaults for this Syscode. If any ProgEvent exists in the parent MsoOrders.ProgEvents with the same ProgEvent.Guid as a ProgEvent defined here, the local definition replaces the parent definition. All other ProgEvent elements defined for the Syscode augment the set defined for the MsoOrder.

7 CAMPAIGN INFORMATION PACKAGE DATA MODELS

The following data model documents are normative components of this specification.

The formal XML schema data definition is found in [CIPXSD].

The formal web service interface data definition is found in [CIPWSDL].

A Schematron schema conforming to [ISO 19757] provides additional validation related to CIP documents, particularly of intra-field relationships. The Schematron schema is found in [CIPSCH].