Technical Requirements

Canoe Campaign Stewardship SaFI Interface

For Dynamic Ad Insertion

Version 3.0

Document Status

|  |  |
| --- | --- |
|  |  |
| **Document File Name:** | Canoe DAI SaFI Interface Specifications v3.1.doc |
| **Date:** | 2/22/12 |

**Revision History**

| **Revision** | **Date** | **Author(s)** | **Notes** |
| --- | --- | --- | --- |
| WIP 1.5 | 2/9/11 | Canoe | Initial WIP version to be reviewed by the SaFI working group. |
| WIP 1.6 | 4/27/11 | Canoe | WIP version subsequent to SaFI working group reviews on 4/13 & 4/20. |
| WIP 1.7 | 5/31/11 | Canoe | WIP version subsequent to SaFI working group reviews on 4/27 & 5/04. |
| WIP 1.9 | 10/11/11 | Canoe | Update document to align with CIP 3.0 and various other decisions. |
| WIP 2.0 | 10/11/11 | Canoe | Update document to align with multiple assets per placement. |
| WIP 2.1 | 10/21/11 | Canoe | Update document from feedback in project meeting. Also align with changes to sample CIP . |
| WIP 2.2 | 11/03/11 | Canoe | Update document from feedback in project meeting, Augmented Rotation UC, Added User Generated Campaign Updates UC, Added System Generated Campaign Updates UC, Added Invalid Campaign Instructions UC. |
| WIP 2.4 | 11/10/11 | Canoe | Update document from feedback in project meeting, Added Goal Achieved UC, Added FlightEnd date/time reached UC, Added StateFeedback UC, Added Campaign Item information flow UC. |
| WIP 2.5 | 11/18/11 | Canoe | Modified Many UC’s, Updated the Business and Technical Requirements to correspond with changes to UC’s. |
| WIP 2.6 | 11/23/11 | Canoe | Reflect use of phase argument for statefeedback frequency. Add FeedBackNotice req’s and samples. Add UpdateNotice samples. Added SLA section. Corrected Qualifier requirements. |
| WIP 2.7 | 12/05/11 | Canoe | Update Business Requirements, Technical Requirements and Samples in Annex section. |
| WIP 2.8 | 12/09/11 | Canoe | Ad Registration, Update Business Requirements, Technical Requirements and Samples in Annex section. |
| WIP 2.9 | 12/13/11 | Canoe | Corrected Invalid Request UC to incorporate use of StateFeedback Notes element for asynchronous syntax validation reporting. Added related Tech Reqmts. Updated sample xml. Added Technical Definition for specifying “mid-roll by number”. Added requirements for use of Nielsen PCC headers as xpaths for metadata “category” definition. |
| WIP 3.0 | 12/21/2011 | Canoe | Added the detail around SMS requirements, brought the document in sync with current thoughts on subjects like IsFirst, IsLast, use of StateFeedback.Note element for communication of asynchronous errors, manual communication of return data paths, use of a manual process for registration, as well as modifying the metadata qualifier list to include “Program Name”, “Season” and “BroadCast\_Air\_Date”. Documented approach for defining mid-roll by number qualifiers. Change the Single and Multiple samples as well as the productFamily descriptions to remove the Phase for ReturnDataPath. Added 3 UpdateNotice samples to represent three Campaign Items that I used to illustrate the SMS requirements. Defined two SMS data sets that we will need (Campaign Centric Data Set and Ad Centric Data Set). Added samples for each of these using the data from the 3 sample UpdateNotices |
| WIP 3.1 | 2/22/2012 | Canoe |  |

Contents

1 Introduction 1

1.1 Document Purpose 1

1.2 Scope 1

1.2.1 Deliverables 2

1.3 Audience 2

1.4 Technical Terms and Acronyms 2

2 Requirements Overview 4

2.1 Overview 4

2.1.1 Architecture 4

2.2 Roadmap 8

2.3 Risks and Mitigation Strategies 9

2.4 Assumptions 10

2.5 SLA’s 10

2.6 CIP consumer impact 11

2.7 Use Cases 11

2.7.1 Opportunity and Position within Opportunity Targeting 11

2.7.2 Ad Rotation 13

2.7.3 Controlling Usage of Trick Modes in Advertising Content (Phase 3) 15

2.7.4 Controlling Campaign Goal and GoalType 17

2.7.5 Product Category Separation 19

2.7.6 Ad Copy Separation 21

2.7.7 Time of Day and Day of Week Targeting 23

2.7.8 CIP Normative Metadata Targeting 24

2.7.9 User Generated Campaign Updates 26

2.7.10 System Generated Campaign Updates 28

2.7.11 Invalid Campaign Instructions 30

2.7.12 Campaign Item Goal Achieved 32

2.7.13 Campaign Item Flight End Date Reached 35

2.7.14 Campaign Item (CI) Information Communication 36

2.7.15 StateFeedback Communicated from the MSO 43

2.7.16 Detailed Viewership Data 45

2.7.17 Summary Viewership Data 47

2.7.18 Registration 49

3 Business Requirements 50

3.1 General Requirements 50

Annex A The CIP MUST support communication of Position restrictions for a CI. The CIP consumer MUST support the following designations of Position within a Break; First or Last. First and Last position are scoped by Opportunity Type and Break (defined as a group of PO’s with the same PoGroupIndex). First and Last are not scoped by Inventory Ownership. See Annex J - ProductFamily and ProductMember Definitions 50

4 Technical Requirements 62

4.1 General Requirements 62

Annex B See Annex J - ProductFamily and ProductMember Definitions 63

Annex A Metadata Operator Matrix 83

Annex B UpdateNotice Sample of single asset productFamily 84

Annex C UpdateResponse (OK and FAIL) 86

Annex D FeedbackNotice Sample of single asset productFamily 87

Annex E FeedbackResponse (OK and FAIL) 89

Annex F UpdateNotice Sample of multiple asset productFamily 90

Annex G FeedbackNotice Sample of multiple asset productfamily 94

Annex H ProductFamily and ProductMember Definitions 98

Annex I Position Qualifier Sample 99

Annex J Reporting - Definitions 100

Terms and Definitions 100

Data Point Configurations 101

Annex K Reporting – Sample UpdateNotice Messages 103

Annex L Reporting – Sample Campaign Centric Data Set 109

Annex M Reporting – Sample Ad Centric Data Set 110

Annex N Issues 113

I.1 Open Issues 113

I.2 Resolved Issues 113

Annex O Contributors and Acknowledgments 114

# Introduction

## Document Purpose

The purpose of this document is to define the SaFI specification requirements needed to support Canoe’s VOD Dynamic Ad Insertion (DAI) efforts.

These requirements represent both the immediate needs, which Canoe would like to propose be addressed in the next SaFI release (3.0), and future needs that should be addressed in a later version of the specification (3.x or beyond). All of the requirements presented below are labeled as 3.0 or 3.x to reflect this prioritization.

## Scope

These requirements specifically address the following features and capabilities of the SaFI CIP and SMS specifications:

* Communicating instructions which restrict placement of ads to placement opportunities with a specific opportunity type. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which restrict placement of ads to placement opportunities with a specific midroll designation (1st, 2nd, 3rd, 4th, etc). Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which restrict placement of ads to placement opportunities with a specific position (locations within the opportunity). Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which restrict placement of ads to placement opportunities within sessions which are initiated during a specified date and time range (flight window).
* Communicating instructions which restrict placement of ads to placement opportunities within sessions which are initiated on specified days of the week and time ranges within those days. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which restrict placement of ads to placement opportunities within sessions for which the CIP normative metadata relating to the entertainment content meets comparative conditions established for 1 or more attributes of that metadata. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which establish product category separation rules. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which establish ad copy separation rules. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which establish priority for each placement. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which establish the goal and goal type. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions for the rotation of ads. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which establish desired enablement state of trick modes during the play-out of advertising content. Expected behavior for MSO execution of these instructions is also defined.
* Communicating instructions which establish performance feedback standards. Expected behavior for MSO execution of these instructions is also defined.
* Communicating detailed viewership metrics via SMS.
* Communicating summary viewership metrics via SMS.

### Deliverables

Implementation of these requirements shall result in the following outputs:

1. Definition of business functionality that will be supported in each phase of the deployment.
2. Definition of a CIP document which meets the business functionality, can be generated from Canoe’s stewardship platform, and can be ingested and executed by an MSO.
3. Definition of the mechanisms by which campaign instructions will be communicated in the CIP and the related interpretation guidance for those instructions.

## Audience

The requirements in this document are most relevant for people working in the following functional areas:

* Product design
* Engineering and Operations

## Technical Terms and Acronyms

This specification uses the following terms:

| **Technical Term** | **Definition** |
| --- | --- |
| Session | A duration of time (including any time spent in trick modes) that starts when a viewer selects play on an On Demand title and ends when exiting the title, timing out, or the title has been fully played out. |
| Opportunity/Break | A discrete unit of time allocated for advertising within an entertainment viewing session. A session is expected to contain one or more breaks. |
| Position | A specific location within an opportunity where an ad can be shown. An opportunity is expected to contain one or more positions. |
| Entertainment / Title | The non-advertising content that is requested for viewing. The main content asset. |
| Ad / Copy |  |
| Insertion / Scheduled Insertion | The act of placing an advertisement into an available opportunity, regardless of whether or not the ad is actually viewed. |
| View / Ad View | When the first frame of an ad asset is played out in normal play time, it is considered to have been viewed. |
| View Duration | The amount of time an ad asset was played out in normal play time. |
| Impression | An impression is counted when a specific amount or % of time of an ad is viewed in normal play mode. The specific parameters may vary by advertiser, agency, or network. |
| Flight Window | A discrete period of time during which the campaign is active. The flight window begins on the flight start date and ends on the flight end date. Campaigns are not eligible for opportunities outside of their flight window. |
| Trick Mode | Any one of several user features which allow the viewer to manipulate the stream. Examples include Fast Forward, Pause, Rewind, Jump, etc. |

This specification uses the following acronyms:

| **Acronym** | **Definition** |
| --- | --- |
| *FFW* | Fast Forward – a trick mode indicating the video stream is being played at a scale greater than 1, faster than normal viewing speed. |
| *CI* | Campaign Item |

# Requirements Overview

## Overview

This section provides an overview of the requirements needed to support Canoe’s VOD dynamic ad insertion efforts. The requirements are presented in the form of high-level use cases. Each use case covers a related set of requirements, with one or more individual requirements contained in each use case. The next section will provide a detailed breakdown of the individual business and technical requirements needed to support the use cases.

### Architecture

NOTE: A full description of the Canoe VOD Dynamic Ad Insertion architecture can be found in the Technical Definition document. The portions of that document that are most relevant to these requirements is provided here as a quick reference.

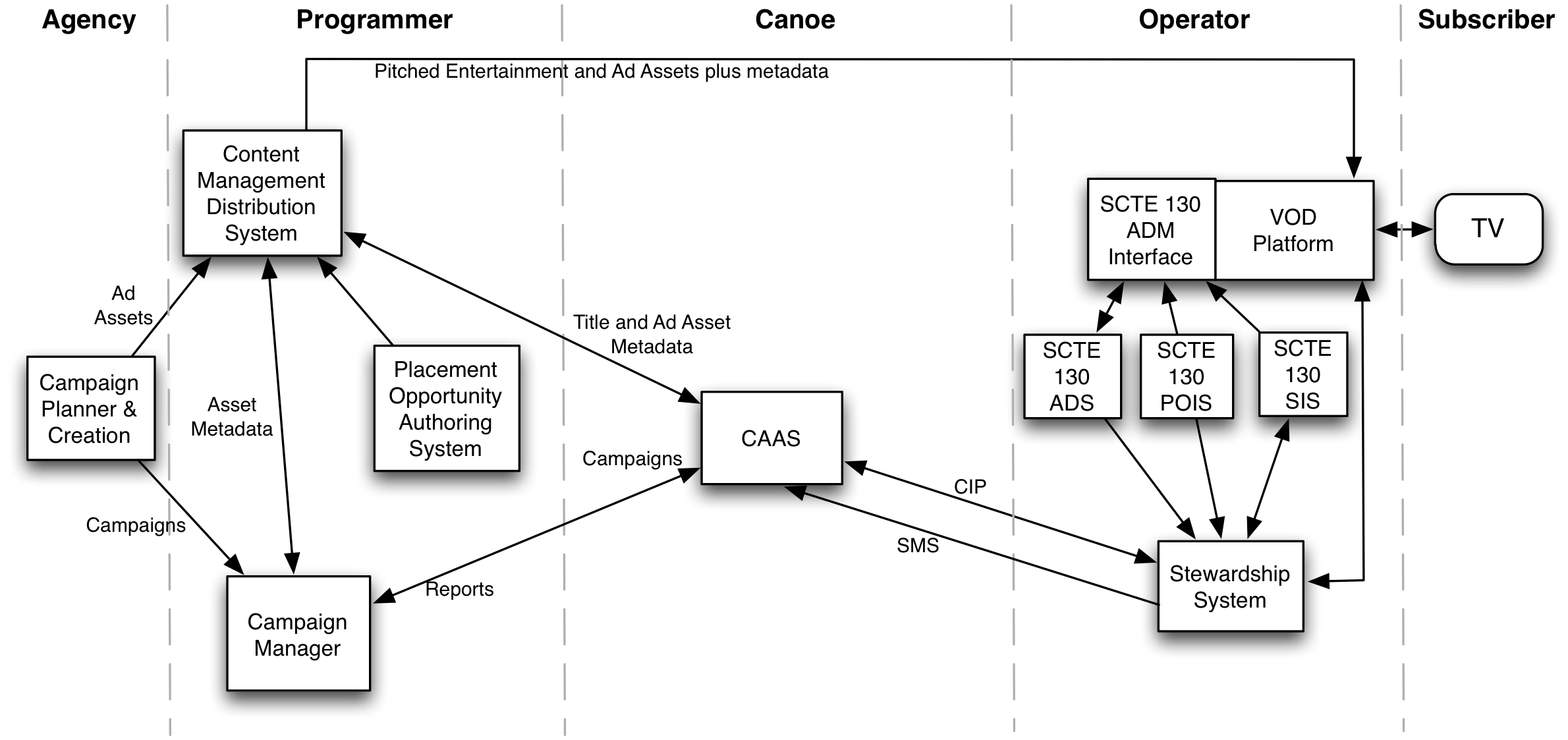


Figure - National On-Demand End-to-end System Context

Figure 1 illustrates the overall system context from the top-level, end-to-end perspective. The system provides a managed pathway for delivery of advertising content to a cable television subscriber. This view is representative of the components typically employed in the various contexts of Agency, Programmer, Canoe, Operator (MSO) and the Subscriber. Actual implementations will certainly vary.

The Canoe National dynamic ad insertion (DAI) platform for on-demand will support three process models as required to accommodate the current states of both Programmer and Operator infrastructure. Each model will include the high-level processes of Campaign Preparation, Run-time Ad Decision, and Performance Reporting. The point at which run-time, dynamic ad placement decisions will be made is the primary distinguishing factor in these models and governs the type and extent of data exchanged at each step in the processes.

The system will be designed such that all three of these execution models may be simultaneously active as required by the infrastructure and policies of the participating MSOs.

#### Campaign Preparation

A Campaign Preparation process must be completed for each campaign to be executed by the Canoe VOD DAI platform. Programmers create campaigns for one or more agencies specifying ad assets to be placed into on-demand views of title assets under specified conditions such as title or genre, time of day, or viewer characteristics. Figure 2 highlights the entities and process steps for campaign preparation.

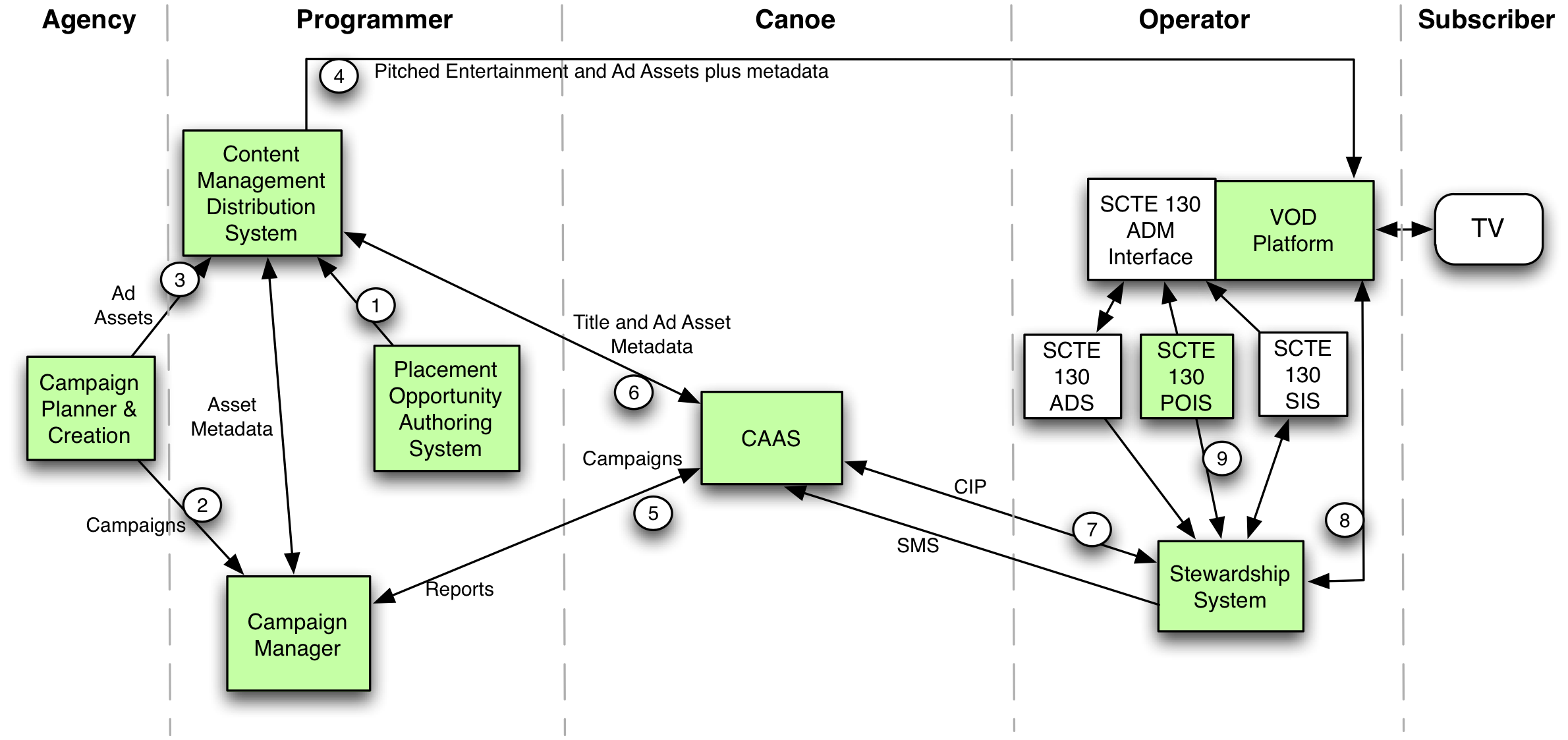


Figure - Campaign Preparation

1. Content Preparation

Title assets must be prepared with either in-band markers (SCTE 35 Segmentation Descriptors) or out-of-band metadata defining ad insertion (placement) opportunities.

1. Campaign Creation

Campaigns are created in the Programmer’s selected Campaign Manager based on the placement requirements expressed by the agency.

1. Campaign Asset Ingest

Ad assets are created by the agency and placed under control of the Programmer’s Content Management System.

1. Content Distribution

The prepared entertainment (title) assets, ad assets and all associated metadata are distributed to the MSOs via satellite pitch or high-speed terrestrial links.

1. Campaign Ingest

The Canoe CAAS will automatically ingest campaign details from the Programmer’s Campaign Manager.

1. Asset Metadatat

The Canoe CAAS will automatically ingest Title asset metadata from the Programmer’s Content Mgmt Distribution System.

1. Campaign Distribution

Campaign instructions will be distributed to the MSOs using SaFI CIP.

1. Operator Campaign Prep

The MSO Stewardship System will prepare the VOD platform to insert the assets as per campaign instructions

1. Operator Placement Opportunity Prep

The MSO Stewardship System will obtain Placement Opportunity descriptions from the SCTE130 POIS.

#### Operator Managed Run-time Decisions

In this model campaign execution parameters will be conveyed to one or more Operator controlled VOD platforms. The Operator’s selected Campaign Manager and associated Ad Decision Service will use these parameters to respond to run-time requests from the operator’s ADM(s). Such systems will execute campaigns autonomously over some defined time period (minutes) and will store and forward to Canoe the details of the resulting insertion at similar time intervals. Based on this feedback loop and aggregated results from across the delivery footprint, Canoe will periodically alter the instructions for executing campaigns to optimize overall delivery and meet objectives.

Figure 3 illustrates the active components in the Operator managed run-time.

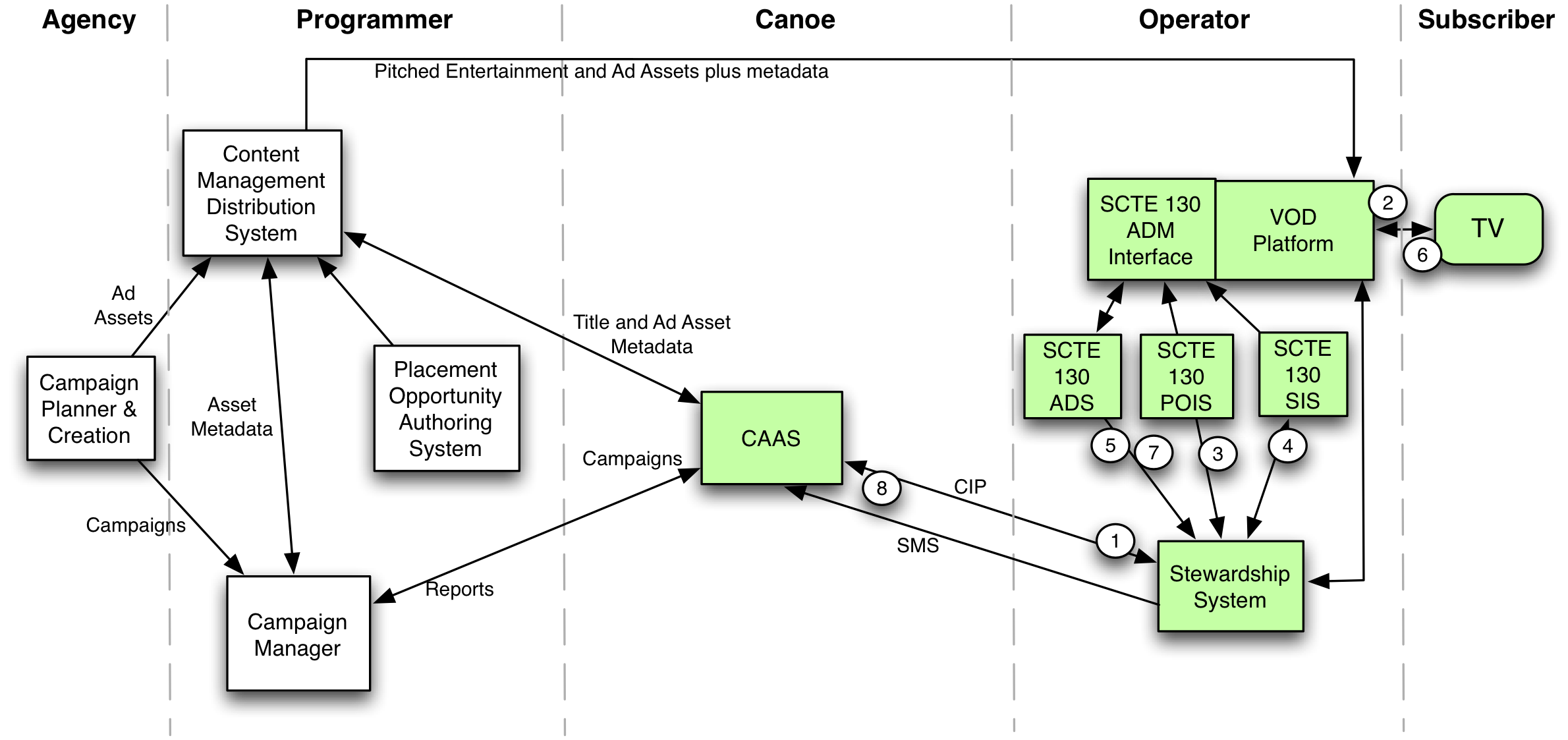


Figure - Operator Managed Run-Time Decisions

1. Campaign Instructions

CAAS generates campaign instructions in the form of a CIP and communicates them to the operator’s stewardship system. The stewardship system provides feedback on campaign execution.

1. Session Start

A subscriber requests a VOD asset by selecting from the guide / catalog on a STB.

1. Placement Opportunity Look-up

The MSO ADM obtains Placement Opportunity descriptions for the requested title asset from the MSO POIS. The POs identify both local and national opportunities.

1. Audience Qualifier Look-up

The MSO ADM obtains Audience Qualifiers for the requesting STB from the MSO SIS

1. Placement Request / Response

The MSO ADM sends a Placement Request to the MSO ADS including both local and national Placement Opportunities and receives a Placement Response specifying decisions for local and national insertions.

1. Playlist Delivery

The VOD platform streams the playlist of the required ad and entertainment assets to the STB

1. Placement Status Notification

The MSO Stewardship platform receives Placement Status Notifications for each national insertion. These are used to generate SMS data products.

1. Service Measurement Summary

The MSO Stewardship platform sends summary data products to CAAS.

#### Performance Reporting

Performance Reporting provides the necessary verification of ad insertion actions as monitored by the Operators’ delivery platforms and Canoe’s CAAS. Performance reports will be created and delivered to the Programmer’s Campaign Manager at scheduled intervals.

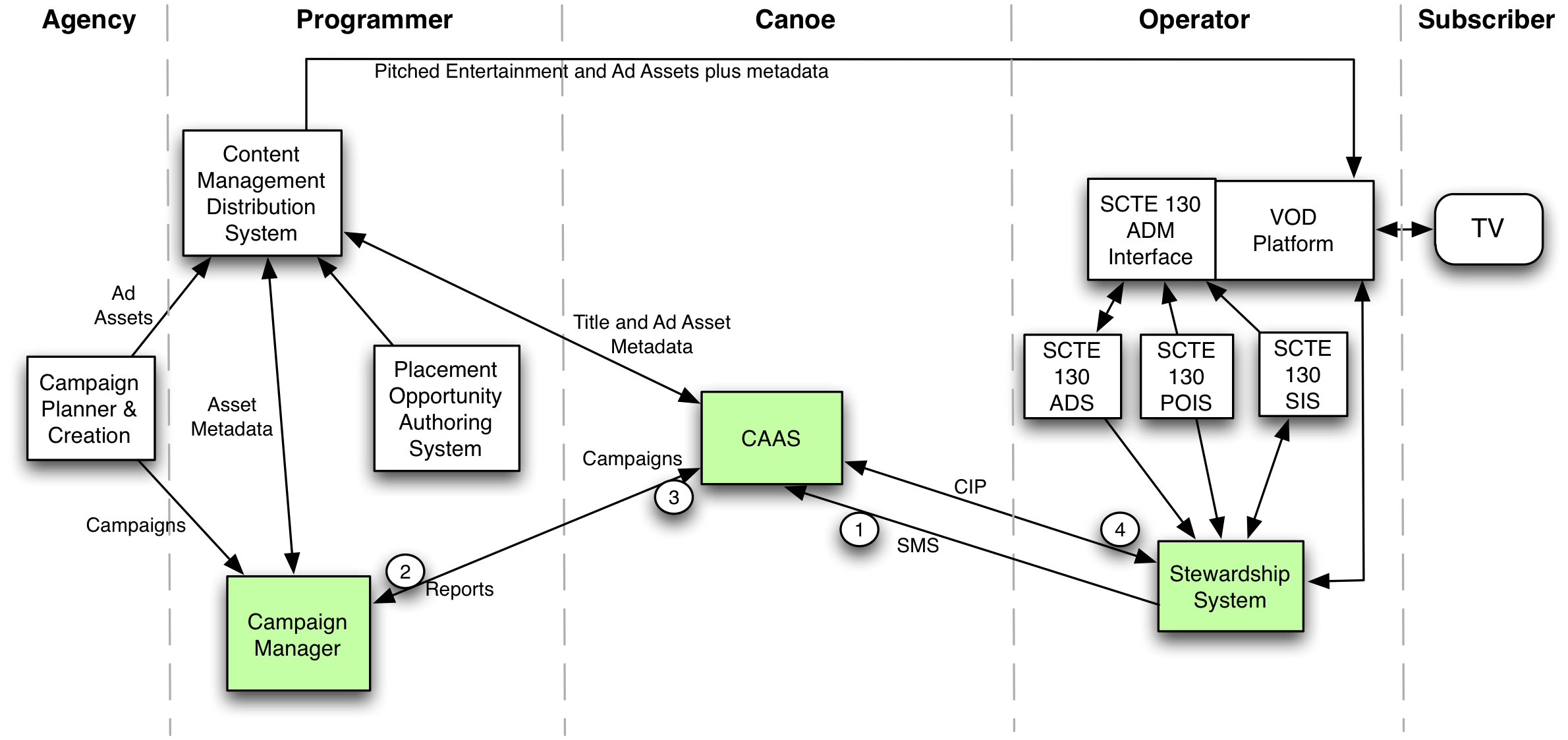


Figure – Reporting and Management

1. MSO reports to Canoe via SMS

The MSO Stewardship System sends Service Measurement Summary messages to the CAAS at scheduled intervals.

1. Canoe reports to Programmer via proprietary interface

The CAAS polls the Campaign Manager for Campaign delivery metrics at scheduled intervals.

1. Programmer may update Campaigns to Canoe

The Programmer sends Campaigns to the CAAS where they are transformed into a CIP

1. Canoe may update CIPs to MSO

The CAAS sends updates to Campaigns to the MSO’s Stewardship System on regular intervals. The MSO Stewardship System sends feedback to the CAAS through the CIP.

## Roadmap

| **Feature** | **Immediate Needs** | **Future Need (SaFI 3.x)** |
| --- | --- | --- |
| 1. Placing ads into specific opportunities and positions within opportunity. | * Opportunity Type control: preroll, postroll, midroll, and midroll by number * Position control: first, last | * Opportunity control: all (exclusive sponsorship) * Position control: bookend |
| 1. Rotation | * Rotation based on a lottery-like model (no cross-session state management per viewer) | * Future asks to include HH and/or STB rotation and frequency capping. Not described within the contents of this document |
| 1. Controlling the use of trick modes on a placement basis during the play-out of advertising content. | * Enabled FFW in ads * Disable FFW in ads | * Future asks include enabling/disabling other trick modes such as Rewind and Pause * Ability to control trick mode at the asset level. |
| 1. Controlling the Goal and specifying the GoalType for a campaign item. | * Goals * Goal Types of “Insertions” or “Views” | * “Impression” based goals |
| 1. Enforcing product category separation within an opportunity, group, or a viewer session. | * Exclusion for Opportunity * Exclusion for group * Exclusion for Session * No Exclusion |  |
| 1. Enforcing ad separation of the same ad content within an opportunity, group, or within a viewing session. | * Ad separation by opportunity, group, or session * Default - Allowing same ad to be placed within the same opportunity but precluding the same ad to be placed tangent to itself within the same opportunity. |  |
| 1. Placing ads at specific times of day and days of the week. | * One or more day of week and time of day conditions |  |
| 1. Placing ads against entertainment content that matches certain conditions when comparing CIP normative metadata to known values. Qualifiers shall be CL 3.0 metadata fields. MSO shall be required to map to 1.1 metadata as necessary. | * Title.TitleBrief * Title.Title * Title.EpisodeName * Title.EpisodeID * Title.Rating * Title.Advisories * Title.Genre * Title.ShowType * Core.Product * Advisory * Movie.Resolution * Movie.BitRate * Movie.ScreenFormat * Title.IsNew   (non CL field, but will leverage StartDateTime and DisplayAsNew)   * Title.IsLastChance   (non CL field, but will leverage EndDateTime and DisplayAsLastChance) | * Title.Year. * Title.Languages * Future asks to include additional or all CableLabs metadata fields |
| 1. Communicating detailed viewership metrics via SMS. | * Title view duration in normal play mode * Ad view duration in normal play mode | * Trick mode events |
| 1. Communicating summary viewership metrics via SMS. | * Title play/request counts * Ad insertion counts * Ad view counts |  |

## Risks and Mitigation Strategies

Not included

## Assumptions

1. Opportunities will accommodate the placement of 1 or more ads.
2. Opportunities will have a defined duration
3. Opportunities may have a defined placement limit. When they do, the execution engine will abide by this limit.
4. Entertainment content contains ad placement opportunities that are owned appropriately and which are defined in the MSO POIS.
5. Time sensitive operational statistics will be available through the CIP Update process
6. Standard business reporting will be available through SMS data feeds.

## SLA’s

1. The Canoe campaign stewardship system communicates preflight campaigns that have matured past Draft (and any updates to those campaigns) every X configurable minutes (initially X=360).
2. The Canoe campaign stewardship system communicates in-flight campaign updates immediately (send the information within 5 seconds of the user saving the campaign in the CM) when any of the following attributes are modified.
   * Flight Start Date
   * Flight End Date
   * Goal
   * Goal Type
   * Add or Remove an ad asset
   * Campaign State – If state is changed from “Active” or state is changed to “Active”  (this will be represented by an appropriate change to recState)
   * Campaign Active Flag (this will be represented by an appropriate change to recState)
   * Campaign Item Active Flag (this will be represented by an appropriate change to recState)
   * If the goal for a campaign item is recognized as having been achieved the system sends an immediate update with a goal of zero (0) and an appropriate recState.
3. The Canoe campaign stewardship system communicates in-flight campaign updates every X configurable minutes (initially X=360) when any of the following attributes are modified.
   * Adding or Removing a Network (Provider)
   * Changing a criteria (time of day, day of week, metadata, opportunity/position conditions)
   * Change Rotation Ratio for an ad Asset
   * Change Product Category Separation Rules
   * Change Ad Copy Separation Rules
   * Change Trick Mode Instructions
   * Priority
   * The Campaign has reached the end of its Flight window (this will be represented by an appropriate change to recState)
   * Campaign State – If state is NOT changed from “Active” or state is NOT changed to “Active”  (this will be represented by an appropriate change to recState)
4. The CIP consumer ingests and implements campaign information within 60 seconds of receipt.  Propagation shall occur in the timeframes of standard system processes (the same as other campaigns managed within the MSO’s systems).  Any time required to perform media or execution readiness is additional.
5. The  CIP consumer MUST provide CIP Feedback every X configurable minutes (initially X=15).
6. The CIP consumer MUST provide SMS viewership data within 15 minutes of session teardown.

## CIP consumer impact

Not included

## Use Cases

### Opportunity and Position within Opportunity Targeting

**Summary**: Create a campaign that directs the placement of ads to specific opportunities and positions within an entertainment viewing session.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign item using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. Each Asset must have a defined Category
   5. The CI must have a priority
3. The Canoe user defines one or more opportunities and position within opportunity conditions where the ads must be placed.
   1. The user can specify the campaign eligible for Pre-roll opportunities.
   2. The user can specify the campaign eligible for Post-roll opportunities.
   3. The user can specify the campaign eligible for mid-roll opportunities.
   4. The user can specify the campaign eligible for only a specific mid-roll opportunity.
   5. The user can specify the campaign eligible for any combination of these.
   6. The user can specify the campaign eligible for any opportunitytype
   7. The user can specify the campaign eligible for the first position (within a given opportunity) (Pase 2)
   8. The user can specify the campaign eligible for the last position (within a given opportunity) eligible. (Phase 2)
   9. The user can specify the campaign eligible for any position (within a given opportunity)
4. The Canoe user saves the campaign information.
5. The Canoe campaign manager transmits the campaign instructions to the MSO.
6. The campaign instructions are made available to the CIP consumer .
7. Pre-flight constraints must be met.
8. A subscriber selects play on an entertainment asset from the VOD menu.
9. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
10. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
    1. Opportunity and Position conditions are taken into account when deciding which campaigns are eligible.
    2. Additional criteria and processes described in other use cases are executed to reach the final selection of the ads to include in the playlist.
11. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
12. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer in the proper sequence and durations.

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Opportunity and Position conditions are considered as part of the ad placement process. Ads which violate the Opportunity and Position conditions are not placed.
2. StateFeedback – separate UC
3. SMS data feed

**Related Use Cases**:

None identified

### Ad Rotation

**Summary**: Create a campaign that rotates the placement of ads across all viewer sessions according to a rotation ratio.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign item using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 2 ad assets
   4. Each ad asset must have a defined Category
   5. Each ad asset must have a defined rotation value.
   6. The CI must have a priority
3. The Canoe user indicates 2 or more ad assets that are to be assigned to the campaign item.
4. The Canoe user specifies a rotation allocation for each ad asset that is assigned to the campaign item.
5. The Canoe user saves the campaign item information.
6. The Canoe campaign manager transmits the campaign instructions to the MSO.
7. The campaign instructions are made available to the CIP consumer .
8. Pre-flight constraints must be met.
9. A subscriber selects play on an entertainment asset from the VOD menu.
10. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
11. The CIP consumer leverages the rotation instructions defined in the campaign item (bundle) to determine which ads to insert into the available opportunities. Ads are selected based on the following rotation methodology.
    1. Rotation is the process of selecting an ad when a campaign item has more than one "eligible" ad. An eligible ad is defined as one which:
       1. Is assigned to a campaign item (represented by an AssetReference in a Placement within a parent Bundle) which has been selected for placement.
       2. Is available (as indicated by the mediaState within StateFeedback)
       3. Does not violate Category Separation constraints
       4. Does not violate Ad Copy Separation constraints
       5. Does not violate the constraints of the remaining duration of the opportunity.
    2. If the campaign item has no eligible ad, no ad is selected for placement.
    3. If the campaign item has only 1 eligible ad, that ad is selected for placement.
    4. If the campaign item has more than 1 eligible ad, rotation rules are used to select the ad to place.  The rotation rules required by Canoe are best described as a random lottery selection, described in detail below.
       1. Each ad has an assigned rotation ratio. The rotation ratio can be any positive integer.
       2. The system must select an ad from the eligible ads in proportions comparable to those calculated using the formula:

(Rotation ratio of an Ad)/(sum of Rotation Ratios for all ads eligible for the placement)

* 1. As an example, assume a campaign item has 5 eligible ads assigned with the following rotation ratios;
     1. Ad #1 = 80
     2. Ad #2 = 25
     3. Ad #3 = 135
     4. Ad #4 = 55
     5. Ad #5 = 40
  2. In our example, Ad #1 gets 80 tickets in the lottery, Ad #2 gets 25 tickets, Ad #3 gets 135 tickets, etc. The system randomly selects from those tickets and the corresponding ad is placed.
  3. In this example, we would expect Ad #1 to be selected 80/335, or roughly 24% of the time.  It's important to note that the requirements do not demand that the system maintain this proportion at all times. Canoe requires that a given set of eligible ads in a rotation, with set rotation allocations for each ad, will provide selection results within 2% of the rotation ratios represented by those allocations over a sample of 10,000 rotations.

1. Taking this example a little further, if Ad #3 violated Category Separation constraints ( and therefore becomes ineligible for this placement) and Ad #5 is found to be unavailable (and therefore not eligible for this placement) we would now expect Ad #1 to be selected 80/160, or roughly 50% of the time. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
2. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer in the proper sequence and durations.

**Alternative course**: The alternative paths may consist of:

1. Only a single ad asset is specified and no rotation allocation is specified
   1. The system assumes a rotation allocation of 1. (The ad must be assumed to be rotated 100% of the time the campaign is selected for placement.)
2. Ad isn’t available – Ad is not included in the rotation group for the placement.
3. Multiple ad assets are present in a group and the rotation ratio is not specified for any of them.
   1. The system must assume a rotation ratio of 1 for every ad asset. (Each ad must be assumed to be rotated evenly with the others in the group.)
4. Multiple ad assets are present in a group and the rotation ratio is specified for 1 or more, but not all assets.
   1. The system must assume a rotation ratio of 0 for those ad assets that don’t have a rotaion ratio defined. (Only ads with a rotation ratio defined will be placed)

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Once a placement is selected, eligible ads are rotated and placed according to their rotation ratios.

**Related Use Cases**:

2.7.5 - Product Category Separation

2.7.6 - Ad Copy Separation

2.7.15 - StateFeedback Communicated from the MSO

### Controlling Usage of Trick Modes in Advertising Content (Phase 3)

**Summary**: Create a campaign that enables or disables the use of the FFW trick mode during the play out of a specific ad.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. Each Asset must have a defined Category
   5. The CI must have a priority
3. The Canoe user specifies whether or not the FFW trick mode will be enabled during the playout of advertising content.
4. The Canoe user saves the campaign information.
5. The Canoe campaign manager transmits the campaign instructions to the MSO.
6. The campaign instructions are made available to the CIP consumer.
7. Pre-flight constraints must be met.
8. A subscriber selects play on an entertainment asset from the VOD menu.
9. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
10. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
11. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
12. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer.
    1. When an ad is played out, FFW is disabled if it was indicated as such in the campaign instructions.
    2. When an ad is played out, FFW is enabled if it was indicated as such in the campaign instructions.

**Alternative course**: The alternative paths may consist of:

1. The MSO VOD system is unable to modify the FFW trick mode from its current setting within the entertainment content.
   1. When an ad is played out, the FFW trick mode remains the same as the entertainment content.
2. The MSO VOD system is unable to specify individual FFW trick mode settings for specific ad assets in a VOD session.
   1. When an ad is played out, the FFW trick mode remains the MSO default setting for ads.

**Exceptions**: The following exceptions may occur

1. Trick mode restrictions may come from the attributes of the Placement Opportunity. These restrictions would override campaign based trick mode instructions.

**Postconditions**: The following post conditions are expected:

1. Ads are played out with the FFW trick mode enabled or disabled according to the campaign instructions provided.

**Related Use Cases**:

None identified

### Controlling Campaign Goal and GoalType

**Summary**: Create a campaign with a specific performance GoalType and a performance goal designated for each campaignItem.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. The CI must have a priority
   5. Each Asset must have a defined Category
3. The Canoe user specifies a campaign performance goal type for the campaign. This goal type is inherited by all campaign items within the campaign.
   1. The goal type can either be based off of the delivery of insertions (phase 2) or views (see section 1.4).
4. The Canoe user specifies a campaign performance goal for each campaign item.
   1. The goal amount must not be negative.
   2. The absence of a goal amount indicates that the campaignItem can be placed without limit until its flight ends.
5. The Canoe user saves the campaign information.
6. The Canoe campaign manager transmits the campaign instructions to the MSO.
7. The campaign instructions are made available to the CIP consumer.
8. Pre-flight constraints must be met.
9. A subscriber selects play on an entertainment asset from the VOD menu.
10. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
11. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
    1. Campaigns with a goal type of “Insertions” continue to place until the insertion goal is reached (Phase 2).
    2. Campaigns with a goal type of “Views” continue to place until the view goal is reached.
    3. Additional criteria and processes described in other use cases are executed to reach the final selection of the ads to include in the playlist.
12. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
13. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer.

**Alternative course**: The alternative paths may consist of:

1. A campaign item goal is not set.
   1. The campaign item must be selected for placement any time it otherwise qualifies.
2. A campaign item goal is changed.
   1. If the new goal has not yet been met, the campaign item continues (or resumes) to place in the standard methodology used by the CIP consumer to allocate goals to subsystems and fill those goals until the goal is met.
   2. If the new goal has been met, the campaign item will no longer be placed.
3. A campaign item goal type is changed (Phase 2).
   1. The system must support moving from one goal type to another and have available campaign item life to date statistics for both Insertions and Views.
   2. If the new campaign item goal (given the new goaltype) has not yet been met, the campaign item continues (or resumes) to place in a in the standard methodology used by the CIP consumer to allocate goals to subsystems and fill those goals until the goal is met.
   3. If the new campaign item goal (given the new goaltype) has been met, the campaign item will no longer be placed.

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Placement performance goals are taken into consideration when placing ads.
2. Placements that have reached their goal are no longer eligible for execution.

**Related Use Cases**:

2.7.10 - System Generated Campaign Updates2.7.12 - Campaign Item Goal Achieved

2.7.14 - Campaign Item (CI) Information Communication

### Product Category Separation

**Summary**: Create a campaign that enforces product category separation.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.
4. Ad content in all campaigns has been categorized according to a standard list of product categories specified in the asset metadata.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. Each Asset must have a defined Category
   5. The CI must have a priority
3. The Canoe user specifies the product category separation rules:
   1. By indicating the name of the category per asset from a pre-defined list. This “category” may be from any level of the PCC Exclusion Category structure (Industry Group, PCC Group, PCC Subgroup, or Product Category). This will be defined within an AssetMetadata element and only 1 category can be defined for each asset. There is no derived exclusion enforced between the levels of the PCC Exclusion structure. The user may define a category of “DRUGS & REMEDIES” (Industry Group) and this does not exclude assets with a defined category of “RX-HUMAN-INFERTILITY” which is a Product Category within DRUGS & REMEDIES. Canoe will provide the fully qualified hierarchy path as the category. As an example;
      1. If the category for the campaign item is “CIGARETTES” from the Product Category, Canoe will communicate the category as “CIGARETTES, TOBACCO & ACCESS\CIGARETTES & TOBACCO\CIGARETTES\CIGARETTES”
      2. If the category for the campaign item is “CIGARETTES” from the PCC subgroup, Canoe will communicate the category as “CIGARETTES, TOBACCO & ACCESS\CIGARETTES & TOBACCO\CIGARETTES”.
   2. By indicating whether or not to enforce separation from other campaigns (and the associated ads). **(This is defined in the CategoryExclusion element. Values can be “opportunity” or “Session”. An opportunity can be standalone (cableLabs metadata 3.0) or multiple PO’s linked together with a POIndex. If absent, no exclusion exists for the placement.)**
   3. By indicating the scope of the separation as either within the same opportunity, within the same group of opportunities (defined by a common PoGroupIndex) or within the same viewing session. This is defined by the scope attribute within the CategoryExclusion element.
4. The Canoe user saves the campaign information.
5. The Canoe campaign manager transmits the campaign instructions to the MSO.
6. The campaign instructions are made available to the CIP consumer.
7. Pre-flight constraints must be met.
8. A subscriber selects play on an entertainment asset from the VOD menu.
9. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
10. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
    1. Additional criteria and processes described in other use cases are executed to reach the final selection of the ads to include in the playlist.
    2. If two ads are part of the same opportunity-exclusive product category and both are selected to be placed into the same ad opportunity, only one can be placed in the opportunity.
    3. If two ads are part of the same POGroup-exclusive product category and both are selected to be placed into the same POGroup, only one can be placed in the POGroup.
    4. If two ads are part of the same session-exclusive product category and both are selected to be placed into the same session, only one can be placed in the session.
    5. Exclusion shall be executed on a “First Placed Stays” basis. If an ad is placed that has a Product Category Separation rule, no other ad can be placed if it violates that rule. An ad can only be placed if it does not violate the Product Category Separation rules of any previously placed ads.
11. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
12. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer.

**Alternative course**: The alternative paths may consist of:

1. When two ads are part of the same non-exclusive category, both may be selected for placement.
2. When two ads are part of different exclusive or non-exclusive categories, both may be selected for placement.

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Product Category Separation rules are enforced when placing ads. Ads which violate the Product Category Separation rules are not placed.

**Related Use Cases**:

2.7.2 - Ad Rotation

### Ad Copy Separation (phase 2)

**Summary**: Create a campaign that enforces ad copy separation by break or session.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. Each Asset must have a defined Category
   5. The CI must have a priority
3. The Canoe user specifies the ad separation rules for each ad, indicating if:
   1. The ad should not be shown more than once in the same opportunity
   2. The ad should not be shown more than once in the same POGroup
   3. The ad should not be shown more than once in the same viewer session
   4. The ad should abide by a default restriction which is to preclude an ad from being placed adjacent to itself within the same POGroup. Two ads are considered the same if they have the same ProviderID/AssetID combination.
4. The Canoe user saves the campaign information.
5. The Canoe campaign manager transmits the campaign instructions to the MSO.
6. The campaign instructions are made available to the CIP consumer.
7. Pre-flight constraints must be met.
8. A subscriber selects play on an entertainment asset from the VOD menu.
9. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
10. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
    1. Additional criteria and processes described in other use cases are executed to reach the final selection of the ads to include in the playlist.
    2. If an opportunity-restricted ad is placed in an opportunity and subsequently selected for placement again in the same opportunity, the ad must not be placed the second time in that opportunity.
    3. If a POGroup-restricted ad is placed in a POGroup and subsequently selected for placement again in the same POGroup, the ad must not be placed the second time in that POGroup.
    4. If a session-restricted ad is placed in a session and subsequently selected for placement again in the same session, the ad must not be placed the second time.
    5. If an ad has no defined copy separation restrictions, placement must conform to the default restriction (preclude an ad from being placed adjacent to itself within the same POGroup).
11. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
12. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer.

**Alternative course**: The alternative paths may consist of:

1. If the same session-restricted ad from the same campaign is to be selected for placement more than once into the same session, the ad must only be placed once.
2. If an ad has no Ad Copy Separation rules, the ad can be placed multiple times in the session, in a POGroup, in an opportunity, but cannot be placed adjacent to itself in the same POGroup.

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Ad copy separation rules are enforced when placing ads. Ads which violate the Ad Copy Separation rules are not placed.

**Related Use Cases**:

2.7.2 - Ad Rotation

### Time of Day and Day of Week Targeting

**Summary**: Create a campaign that places ads at specific times of day and/or days of the week.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. Each Asset must have a defined Category
   5. The CI must have a priority
3. The Canoe user specifies the time of day and day of week targeting conditions as a pair by:
   1. Indicating one or more days to be eligible for ad placement
   2. Indicating an accompanying start and end time to be eligible for ad placement
4. The Canoe user saves the campaign item information.
5. The Canoe campaign manager transmits the campaign item instructions to the MSO.
6. The campaign item instructions are made available to the CIP consumer.
7. Pre-flight constraints must be met.
8. A subscriber selects play on an entertainment asset from the VOD menu.
9. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
10. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
    1. Additional criteria and processes described in other use cases are executed to reach the final selection of the ads to include in the playlist.
    2. If the current day and time on which the viewer is requesting the entertainment content (local to the viewer’s time) falls within the conditions specified for the CI, the CI must be made eligible for selection (pending other conditions).
11. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
12. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer.

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Day of week and time of day conditions are considered as part of the ad placement process. Ads which violate the Day of Week/Time of day constraints are not placed.

**Related Use Cases**:

2.7.9 - User Generated Campaign Updates

### CIP Normative Metadata Targeting

**Summary**: Create a campaign that places ads according to conditions on CIP normative metadata associated with entertainment content being viewed.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. Each Asset must have a defined Category
   5. The CI must have a priority
3. The Canoe user specifies one or more metadata conditions against a known set of CIP normative 3.0 metadata fields.
4. The Canoe user saves the campaign information.
5. The Canoe campaign manager transmits the campaign instructions to the MSO.
6. The campaign instructions are made available to the CIP consumer.
7. Pre-flight constraints must be met.
8. A subscriber selects play on an entertainment asset from the VOD menu.
9. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
10. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
    1. Additional criteria and processes described in other use cases are executed to reach the final selection of the ads to include in the playlist.
    2. If the metadata of the entertainment asset being viewed fall within the conditions specified in the campaign, the campaign must be made eligible for selection (pending other conditions).
11. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
12. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer.

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Metadata conditions are considered as part of the ad placement process. Ads which violate the metadata qualifications are not placed.

**Related Use Cases**:

2.7.9 - User Generated Campaign Updates

### User Generated Campaign Updates

**Summary**: Update one or many of the following aspects of a campaign:

1. Flight Start Date
2. Flight End Date
3. Goal
4. Goal Type
5. Adding or Removing a Network (Provider)
6. Changing a criteria (time of day, day of week, metadata, opportunity/position conditions)
7. Add or Remove an ad asset
8. Change Rotation Ratio for an ad Asset
9. Change Product Category Separation Rules
10. Change Ad Copy Separation Rules
11. Change Trick Mode Instructions
12. Priority
13. Campaign State
14. Campaign Active Flag
15. Campaign Item Active Flag

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. A Canoe user updates an existing campaign using the Canoe campaign manager.
2. The Canoe user saves the campaign information.
3. The Canoe campaign manager transmits the updated campaign instructions to the CIP consumer. The CIP consumer implements the updates.
4. The system shall communicate updates to the following attributes in an immediate manner. Immediate is defined as communicating the update within 5 seconds of the Canoe user saving the updates to the campaign information.
   1. Flight Start Date
   2. Flight End Date
   3. Goal
   4. Goal Type
   5. Add or Remove an ad asset
   6. Campaign State – If state is changed from “Active” or state is changed to “Active” (this will be represented by an appropriate change to recState)
   7. Campaign Active Flag (this will be represented by an appropriate change to recState)
   8. Campaign Item Active Flag (this will be represented by an appropriate change to recState)
5. The system shall communicate updates to the all other attributes of a campaign item in a batch manner. Batch communications of User Generated campaign updates must occur every X configurable minutes (initially X=360). The campaign instructions are made available to the CIP consumer . The CIP consumer shall ingest updates within 60 seconds of receiving the updates from the Campaign Manager.
6. Pre-flight constraints must be met.
7. A subscriber selects play on an entertainment asset from the VOD menu.
8. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
9. The CIP consumer leverages the **updated** campaign instructions to determine which ads to insert into the available opportunities.
10. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
11. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer.

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Updates to campaigns are implemented resulting in placement decisions in accordance with the updated characteristics of the campaign.

**Related Use Cases**:

2.7.10 - System Generated Campaign Updates

2.7.11 - Invalid Campaign Instructions

### System Generated Campaign Updates

**Summary**: The system performs certain tasks that impact the CIP. These tasks are listed below.

1. The system re-ranks Campaign Items.
2. The system ingests StateFeedback information.
3. The system recognizes when a Campaign Item has reached the end of its flight window.
4. The system will recognize that a Campaign Item has been in the recState of “Windup” for an appropriate period of time.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.

**Basic course**: The nominal path consists of:

1. The Canoe campaign manager encounters one of the following 4 events.
   1. The system re-ranks campaign items which modifies priority. The re-ranking of campaigns occurs every x configurable minutes and the new priority is communicated in the next batch update. Initially X=360.
   2. The system ingests performance information which may result in recognizing a campaign as having achieved its’ goal.
      1. If the goal for a campaign item is recognized as having been achieved the system sends an immediate update with a goal of zero (0) and an appropriate recState (pse).
      2. If the goal for a campaign item is not yet achieved the system uses the performance information accumulated to calculate a new goal for the campaign item. The generation of campaign item goals occurs every x configurable minutes. The new goal is communicated in the next batch update. Initially X=360.
   3. The system recognizes when a campaign item has reached the end of its flight window which will impact recState. The communication of the new recState (pse) occurs in the next batch update. Canoe expects that the CIP consumer will cease execution of the campaign item when the endDateTime of the flight window is reached and will continue processing reporting for the CI.
   4. The system recognizes when a campaign item has been reached the end of its flight window and been in the recState of “Pause” for X configurable hours (initially X=72). The system sends an update with the next batch changing the recState to “Windup”. Canoe expects that the CIP consumer will continue processing reporting for the CI.
   5. The system will recognize that a campaign item has been in the recState of “Windup” for X configurable hours. Initially X=72. The system sends an update with the next batch, changing the recState from “Windup” to “Closed”. Canoe expects that the CIP consumer will discontinue processing reporting for the CI.
2. The system shall communicate immediate updates within 5 seconds of the triggering event. The MSO shall implement immediate updates within 60 seconds of receiving the updates from the Campaign Manager.
3. The MSO has received the Campaign Updates. The MSO must implement those updates within 60 seconds of receipt.
4. Pre-flight constraints must be met.
5. A subscriber selects play on an entertainment asset from the VOD menu.
6. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
7. The CIP consumer leverages the updated campaign instructions to determine which ads to insert into the available opportunities.
8. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
9. The MSO VOD system plays out the entertainment and the appropriate ads to the viewer.

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Updates to campaigns are implemented resulting in placement decisions in accordance with the updated characteristics of the campaign.

**Related Use Cases**:

2.7.9 - User Generated Campaign Updates

2.7.11 - Invalid Campaign Instructions

### Invalid Campaign Instructions

**Summary**: The Canoe Campaign Manager communicates a campaign which does not pass established validations. The MSO rejects the CIP and responds.

**Actors**: The following actors are engaged:

1. CIP consumer
2. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. CIP validations have been established.

**Basic course**: The nominal path consists of:

1. A Canoe user creates a campaign using the Canoe campaign manager.
2. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. Each Asset must have a defined Category
   5. The CI must have a priority
3. The Canoe campaign manager transmits the campaign instructions to the MSO.
4. The CIP consumer validates the message against the CIP schema.
   1. If the message is valid, the CIP consumer generates an UpdateRespons. The UpdateResponse must contain a msgResult=“OK”
   2. If the message is not valid, the CIP consumer generates an UpdateRespons. The UpdateResponse must contain a msgResult=“FAIL” and contain a Notes element containing specific details of all failures. The Canoe campaign manager must log the Acknowledgement as unsuccessful. Canoe Operations will monitor the log and facilitate resolution. MSO operations must also log, monitor and facilitate resolution of all “failed” UpdateResponse messages. There is no further processing of the contents of the CIP.
5. If the message is successfully validated against the schema, the CIP consumer can then validate the syntax of the campaign instructions based on the productFamily definition and productMember definition (see Annex J - ProductFamily and ProductMember Definitions). These validations would include those listed below.
   1. Parent Bundle – All source, operation, xpath or namespace of a Term element must be properly formed.
   2. Parent Bundle – Must contain at least 1 Flight element.
   3. Parent Bundle - Must contain at least 1 CommonQualifierGroup element containing at least 1 Qualifier element containing at least 1 Term element.
   4. Parent Bundle – Must contain a StateFeedback element. This StateFeedback element must contain the Statistics element(s) which must contain a Statistic element to track insertion performance and a Statistic element to track view performance.
   5. Parent Bundle – Must contain a Phase with a ProcessRuleSelector@ruleId="SetGoal"
   6. Parent Bundle - Must contain a Phase with a ProcessRuleSelector@ruleId= "SetRetDataPath"
   7. Inner Bundle – Must contain 1 Placement element
   8. Inner Bundle - The @priority of all inner bundles Must be the same as the parent bundle.
   9. AssetReference – uriID is malformed or not present.
   10. AssetReference – AssetMetadata describing product category  is not provided
   11. ProcessRuleSelector - ruleID is malformed
   12. ProcessRuleSelector - ruleRepository is malformed
   13. StateFeedback.Statistics.Statistic - datamap is not recognized, provided or is malformed.
   14. StateFeedback.Statistics.Statistic - variable is not recognized or provided.
   15. StateFeedback - @reportAsOf is not provided or malformed
6. The CIP consumer should perform all syntax validations and communicate any and all validation errors in the Notes element of a StateFeedback element in the next FeedbackResponse generated. The CIP consumer shall insert a StateFeedback element in the Bundle, Placement, AssetReference or Phase immediately containing the validation error. The StateFeedback element MUST have a revision = 0 and a Notes element. The Notes element shall contain the details of any validation errors identied in the containing element. The CIP consumer should not implement the Bundle if syntax validation errors are detected.
7. The CIP publisher must recognize syntax validation errors communicated in the Notes element of the Statefeedback element. These errors are logged by the CIP publisher and appropriate notifications generated to prompt investigation and correction of the errors.

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Only validated Bundles are implemented by the CIP consumers.

**Related Use Cases**:

2.7.9 - User Generated Campaign Updates

2.7.10 - System Generated Campaign Updates

### Campaign Item Goal Achieved

**Summary**: The campaign item goals are established, progress towards the goal is tracked and execution completes when the goal is achieved. This use case describes the flow for either GoalType, insertions or views.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.
4. The StateFeedback reportInterval is set to X configurable minutes, initially set to 15 minutes.
5. reportAsOf attribute is set to currentDateTime of communication to MSO
6. The flight end date is not reached during the span of this use case.

**Basic course**: The nominal path consists of:

1. The user sets up a campaign using the Canoe campaign manager.
2. The campaign contains a single campaign item.
3. The Canoe user indicates the following for the campaign item.
   1. The state of the campaign in the Canoe campaign manager is “Active”
   2. The CI Flight Window has been established
   3. At least 1 ad asset
   4. Each Asset must have a defined Category
   5. The CI must have a priority
4. The goal for the campaign item is set at 1100 views. The CIP consumer allocates goals to subsystems and fills those goals in their standard methodology.
5. The campaign item is communicated to the CIP consumer for pre-flight readiness.
6. The campaign item reaches the start of its flight window. The current revision is “1” for all Bundles, Placements, Phases and StateFeedback elements.
7. A subscriber selects play on an entertainment asset from the VOD menu.
8. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
9. The CIP consumer leverages the campaign instructions and determines that an ad from this campaign item should be placed.
10. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
11. The placement cycle (Steps 6-9) occurs 350 times over the next 15 minutes.
12. The CIP consumer receives no viewership records for any of these 350 placements during these 15 minutes.
13. The CIP consumer provides a FeedbackNotice containing a performance summary for the last 15 minutes.
    1. The performance summary for revision 1 shows 350 insertions, 0 views.
14. The Canoe campaign manager acknowledges the FeedbackNotice and persists the counts.
15. The CIP consumer continues to make placements for the campaign item. Over the next 5 minutes the placement cycle occurs another 250 times (revision 1 executions).
16. The Canoe campaign manager sends an update of the campaign item to the CIP consumer. The update changes the priority. Revision of Bundles, Placements, Phases and StateFeedback elements are incremented. The current revision in the UpdateNotice is “2”. The campaign item will include a StateFeedback element only for revision “2”.
17. The CIP consumer ingests and implements the campaign update.
18. The placement cycle occurs 500 times over the next 10 minutes (revision 2 executions).
19. The CIP consumer receives 200 viewership records.
20. The CIP consumer provides a FeedbackNotice for revision 2. This FeedbackNotice communicates the cumulative counts.
    1. The performance summary in the revision 2 FeedbackNotice shows 1100 insertions, 200 views
21. The Canoe campaign manager acknowledges the FeedbackNotice and persists the counts.
22. Total views are 200 for the campaign item. The CIP consumer continues to make placements for the campaign item. The placement cycle occurs 700 times over the next 5 minutes (revision 2 executions)
23. The Canoe campaign manager sends an update of the campaign item to the CIP consumer. The update changes the priority. Revision of Bundles, Placements, Phases and StateFeedback elements are incremented. The current revision in the UpdateNotice is “3”. The campaign item will include a StateFeedback element only for revision “3”.
24. The CIP consumer ingests and implements the campaign update.
25. The placement cycle occurs 400 times over the next 10 minutes (revision 3 executions).
26. The CIP consumer receives 300 viewership records.
27. The CIP consumer provides a FeedbackNotice for revision 3. This FeedbackNotice communicates the cumulative counts.
    1. The performance summary in the revision 3 FeedbackNotice shows 2200 insertions, 500 views
28. The Canoe campaign manager acknowledges the FeedbackNotice and persists the counts.
29. Total views are 500 for the campaign item. The CIP consumer continues to make placements for the campaign item. The placement cycle occurs 300 times over the next 5 minutes (revision 3 executions)
30. The Canoe campaign manager sends an update of the campaign item to the CIP consumer. The update changes the priority. Revision of Bundles, Placements, Phases and StateFeedback elements are incremented. The current revision in the UpdateNotice is “4”. The campaign item will include a StateFeedback element only for revision “4”.
31. The CIP consumer ingests and implements the campaign update.
32. The placement cycle occurs 550 times over the next 10 minutes (revision 4 executions).
33. The CIP consumer receives 600 viewership records.
34. The CIP consumer provides a FeedbackNotice for revision 4. This FeedbackNotice communicates the cumulative counts.
    1. The performance summary in the revision 4 FeedbackNotice shows 2950 insertions, 1100 views
35. The Canoe campaign manager acknowledges the FeedbackNotice and persists the counts.
36. Total views are 1100 for the campaign item. The CIP consumer STOPS execution of the campaign item (ceases making placements for the campaign item). This occurs when the 1100th execution occurs, not when the FeedbackNotice is sent to the Canoe campaign manager.
37. The CIP consumer receives 250 viewership records.
38. The Canoe campaign manager sends an update of the campaign item to the CIP consumer. Having now been informed by the CIP consumer that the campaign item has reached its’ goal, the update changes recState of all bundles and placements in the campaign item to “pse” (pause). Nothing else changes. Revision of Bundles, Placements, Phases and StateFeedback elements are incremented. The current revision in the UpdateNotice is “5”. The campaign item will include a StateFeedback element only for revision “5”.
39. The CIP consumer ingests and implements the campaign updates.
40. The CIP consumer receives 500 viewership records.
41. The CIP consumer provides a FeedbackNotice for revision 5. This FeedbackNotice communicates the cumulative counts.
    1. The performance summary in the revision 5 FeedbackNotice shows 2950 insertions, 1850 views
42. The Canoe campaign manager does not include a campaign item in the Re-Ranking or Goal Setting processes when the recState is “pse”. This eliminates the updates that would normally occur for a campaign item from these processes.
43. The CIP consumer continues to provide FeedbackNotices with updated performance metrics.
44. The user does not change the goal in the canoe campaign manager during the allowed time period.
45. The Canoe campaign manager sends an update moving the recState of all bundles and placements in the campaign item to “wnd” (windup). Nothing else changes. Revision of Bundles, Placements, Phases and StateFeedback elements are incremented. The current revision in the UpdateNotice is “6”. The campaign item will include a StateFeedback element only for revision “6”.
46. The user is not allowed to update a campaign item in the Canoe campaign manager once the recState=”wnd”.
47. The CIP consumer continues to provide FeedbackNotices with updated performance metrics.
48. The Canoe campaign manager does not include a campaign item in the Re-Ranking or Goal Setting processes when the recState is “wnd”. This eliminates the updates that would normally occur for a campaign item from these processes. After the configurable windup period has expired, the Canoe campaign manager sends an update moving the recState of all bundles and placements in the campaign item to “cls” (closed). Nothing else changes. Revision of Bundles, Placements, Phases and StateFeedback elements are incremented. The current revision in the UpdateNotice is “7”. The campaign item will include a StateFeedback element only for revision “7”.
49. This is the final UpdateNotice for the CI from the Canoe campaign manager.
50. The CIP consumer ingests and impliments the campaign update and discontinues all FeedbackNotice messages for this campaign item.

**Alternative course**: The alternative paths may consist of:

1. A campaign item goal is changed.
   1. If the new goal has not yet been met, the placement continues (or resumes) to place in the standard methodology used by the CIP consumer to allocate goals to subsystems and fill those goals until the goal is met.
   2. If the new goal has been met, the placement will no longer be placed.
2. A campaign item goal type is changed (Phase 2 scenario).
   1. The system must support moving from one goal type to another and have available campaign item life to date statistics for both Insertions and Views.
   2. If the new campaign item goal (given the new goaltype) has not yet been met, the campaign item continues (or resumes) to place in the standard methodology used by the CIP consumer to allocate goals to subsystems and fill those goals until the goal is met.
   3. If the new campaign item goal (given the new goaltype) has been met, the campaign item will no longer be placed.

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. The campaign item has achieved its’ Goal and further execution of the campaign item has been halted. Notice that this UC demonstrates a campaign item may exceed a view goal even though executions are halted when the goal is reached.

**Related Use Cases**:

2.7.4 - Controlling Campaign Goal and GoalType

2.7.15 - StateFeedback Communicated from the MSO

### Campaign Item Flight End Date Reached

**Summary**: The campaign item completes execution by reaching the end of its flight window.

**Actors**: The following actors are engaged:

1. CIP consumer
2. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. The campaign item has an established Flight Window.
2. The established flight end date/time is reached.

**Basic course**: The nominal path consists of:

1. The CIP consumer recognizes that the campaign item flight end date/time has been reached.
2. The CIP consumer must prevent further placements.
3. Further execution for this campaign item is halted.
4. The Canoe campaign manager will update the campaign item to a recState=”pse” in the next batch update.
5. Subsequent updates of the campaign by the Canoe campaign manager will be 1 of these two options:
   1. If the Canoe user assogns a new Flight endDate (one which is greater than the current date/time), the campaign item is updated with a new flight end date/time and recState=”act” OR
   2. If no change is made to flight window endDate, the campaign item is moved to recState=”wnd” after some configurable number of hours (initially set to 72).

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. The campaign item has achieved its’ Goal and further execution of the campaign item has benn temporarily halted.

**Related Use Cases**:

None identified

### Campaign Item (CI) Information Communication

**Summary**: The campaign item is created by the Canoe campaign manager and shared between Canoe and the CIP consumer. Campaign Item related data is communicated at various stages including:

1. When the Pre-Flight stage is reached (the campaign item is sufficiently defined to support readiness checks)
2. When the Flight stage is reached (a configurable amount of time prior to the flight startDate)
3. When the campaign item is in Pre-Flight or Flight and a change is made to the campaign item information
4. When StateFeedback information is available

**Actors**: The following actors are engaged:

1. CIP consumer
2. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. The Canoe campaign manager has been configured to send the initial CIP as soon as it meets required completeness to support asset availability checking.
2. The Canoe campaign manager has been configured to send each campaign item in its own UpdateNotice.
3. The Canoe campaign manager has been configured to send the “active” CIP 1 hour prior to the flight start date/time.
4. The CIP consumer has been configured to send each campaign item in its own FeedbackNotice.

**Basic course**: The nominal path consists of:

1. The User creates a campaign containing 1 or more CI’s using the Canoe campaign manager. The Canoe campaign manager does not communicate a CI to the MSO until the following readiness information requirements are met for the CI:
   1. The state of the campaign in the Canoe campaign manager is past “Draft”
   2. The CI Flight Window has been established
   3. 1 or more ad assets have been assigned to the CI
2. The Canoe campaign manager communicates the CI with a recState=”com” (as a CIP in an UpdateNotice) as part of a standard scheduled CI update to the MSO. The revision of all Bundles, Placements, Phases and StateFeedback elements will be “1”.
3. The CIP consumer must achnowledge the UpdateNotice with an UpdateResponse.
   1. If the UpdateNotice is accepted, the campaign item is implemented within the CIP consumer and the UpdateResponse must contain a msgResult=”OK”. The Canoe campaign manager must log the Acknowledgement as successful.
   2. If the UpdateNotice is not accepted, the CIP consumer do not implement the campaign item and the UpdateResponse must contain a msgResult=“FAIL” and contain a Notes element containing specific details of the failure. The Canoe campaign manager must log the Acknowledgement as unsuccessful. Canoe Operations will monitor the log and facilitate resolution. MSO operations must also log, monitor and facilitate resolution of all failed FeedbackResponses.
4. The CIP consumer must begin checking asset availability (mediaState) based on the date/time represented in the beginRequired attribute of the ContentMgmt element. The beginRequired date/time will be the current date/time when the Canoe campaign manager created the UpdateNotice. The endRequired attribute will equal the flight endDate.
5. The CIP consumer generates a FeedbackNotice for each campaign item. It is communicated with other FeedbackNotices in a batch communication. The FeedbackNotice(s) must:
   1. Report mediaState for each asset in the campaign item.
   2. Report executionState for the campaign item.
   3. Occur no less than every X configurable minutes. Initially X=15 minutes.
6. The Canoe campaign manager must acknowledge each FeedbackNotice with a FeedbackResponse.
   1. If the FeedbackNotice is accepted, the FeedbackResponse must have a msgResult=”OK”
   2. If the UpdateNotice is not accepted, the CIP consumer do not implement the campaign item and the UpdateResponse must contain a msgResult=“FAIL” and contain a Notes element containing specific details of the failure. The Canoe campaign manager must log the Acknowledgement as unsuccessful. Canoe Operations will monitor the log and facilitate resolution. MSO operations must also log, monitor and facilitate resolution of all failed FeedbackResponses.
7. The Canoe campaign manager may communicate pre-flight updates to the CIP. Any pre-flight updates to the CIP must be transmitted in an UpdateNotice with the batch update. Revision of Bundles, Placements, Phases and StateFeedback elements are incremented with each UpdateNotice.
8. The CIP consumer must acknowledge these UpdateNotices with an UpdateResponse in the manner described in step 3.
   1. If an UpdateNotice is not accepted the CIP consumer must continue to execute readiness checks using the currently implemented CI information.
9. The Canoe campaign manager must communicate the CI to the CIP consumer X configurable duration prior to the flight start date/time (referred to as the “Activate Duration”). In this UC, Activate Duration=1 hour. The CI is communicated within a CIP within an UpdateNotice. The CI is communicated with a recState=”act”. Revision of Bundles, Placements, Phases and StateFeedback elements are incremented. This update only occurs when the CI contains the following execution information incremental to the readiness requirements defined in step 1.
   1. The CI must have a priority
   2. Each Asset must have a defined Category
10. The CIP consumer must acknowledge each UpdateNotice with an UpdateResponse in the manner described in step 8.
11. The CIP consumer generates a FeedbackNotice for each campaign item. It is communicated with other FeedbackNotices in a batch communication. The FeedbackNotice(s) must:
    1. Report current mediaState for all assets in the campaign item.
    2. Report current executionState for the campaign item.
    3. Report cumulative occurences of insertions for each ad asset.
    4. Report cumulative occurences of views for each ad asset.
    5. Occur no less than every X configurable minutes. Initially X=15 minutes.
12. The Canoe campaign manager must acknowledge each FeedbackNotice with a FeedbackResponse.
    1. If the FeedbackNotice is accepted, the FeedbackResponse must have a msgResult=”OK”
    2. If the FeedbackNotice is not accepted, the FeedbackResponse must have a msgResult=”FAIL”. The Canoe campaign manager must log the Acknowledgement as unsuccessful. Canoe Operations will monitor the log and facilitate resolution. MSO operations must also log, monitor and facilitate resolution of all non-“OK” FeedbackResponses.
13. The Canoe campaign manager will communicate changes to the CIP. Certain changes to the CIP must be transmitted in an immediate manner (see 2.7.9 & 2.7.10). Other changes are communicated with the batch update. All changes are transmitted in an UpdateNotice. Every UpdateNotice will result in incrementing the revision of Bundles, Placements, Phases and StateFeedback elements.
14. The CIP consumer must acknowledge each UpdateNotice with an UpdateResponse in the manner described in step 3.
    1. If an UpdateNotice is not accepted the CIP consumer must continue to execute using the currently implemented CI informationCIP consumer .

**Alternative course**: recState Transition below demonstrates the recState flows that occur in the normal path as well as the alternate paths described below.



Figure - recState Transition

The alternative paths may consist of:

A campaign item is created within the Activate Duration or later

1. The User creates a CI during the flight window or within the Activate Duration of the flight.window.
2. If both the readiness information requirements (see step 1 in the main flow) and execution information requirements (se step 9 in the main flow) have been met, the Canoe campaign manager sends an immediate update for the CI. The update must communicate a recState of “act” (rather than “com” as would be the normal case when a CI is created).
3. The CIP consumer responds to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

An active campaign is paused

1. The User changes the state of a campaign within the Canoe campaign manager from “Active” to “Paused”. This will pause every campaign item in the campaign. A user can also have the same impact by changing the “Active Flag” for the campaign from “true” to “false”.
2. The Canoe campaign manager sends an immediate update for every campaign item in the paused campaign to the CIP consumer. The update changes the recState of each bundle and placement element from “act” to “pse”
3. The CIP consumer respond to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

A paused campaign is activated

1. The User changes the state of a campaign within the Canoe campaign manager from “Paused” to “Active”. This will activate every campaign item in the campaign. A user can also have the same impact by changing the “Active Flag” for the campaign from “false” to “true”.
2. The Canoe campaign manager sends an immediate update for every campaign item in the activated campaign to the CIP consumer. The update changes the recState of each bundle and placement element from “pse” to “act”
3. The CIP consumer respond to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

The campaign item “Active Flag” is changed from “true” to “false”

1. The User changes the campaign item “Active Flag” from “true” to “false” within the Canoe campaign manager. This will pause the campaign item.
2. The Canoe campaign manager sends an immediate update for the campaign item to the CIP consumer. The update changes the recState of each bundle and placement element from “act” to “pse”
3. The CIP consumer respond to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

The campaign item “Active Flag” is changed from “false” to “true”

1. The User changes the campaign item “Active Flag” from “false” to “true” within the Canoe campaign manager. This will activate the campaign item.
2. The Canoe campaign manager sends an immediate update for the campaign item to the CIP consumer. The update changes the recState of each bundle and placement element from “pse” to “act”
3. The CIP consumer respond to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

A campaign is moved from Draft

1. The User can change the state of a campaign within the Canoe campaign manager from “Draft” to more mature states like “Requires Approval” or “Active”. This will impact every campaign item in the campaign.
2. For each CI that meets readiness information requirements, the Canoe campaign manager communicates the CI to the CIP consumer with a recStae=”com” .
3. The CIP consumer respond to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

An active campaign is disapproved

1. The User can change the state of a campaign within the Canoe campaign manager from “Active” to “Requires Approval”. This will impact every campaign item in the campaign.
2. If the current recState of the CI is “com”, no update is sent to the MSO.
3. If the current recState of the CI is “act”, the Canoe campaign manager sends an immediate update to the CIP consumer. The update changes the recState of each bundle and placement element from “act” to “pse”
4. The CIP consumer responds to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

A campaign is cancelled

1. The User can change the state of a campaign within the Canoe campaign manager from any state to “Cancelled”. This will de-activate every campaign item in the campaign.
2. If an UpdateNotice has never been sent for the campaign, no UpdateNotice will be generated. Otherwise the Canoe campaign manager sends an immediate update for every CI in the cancelled campaign to the CIP consumer. The update changes the recState of each bundle and placement element to “wnd”.
3. The CIP consumer respond to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

A campaign item is completed

1. The Canoe campaign manager can recognize the completion of a campaign in two ways:
   1. The flight end date/time is reached
   2. The goal is achieved
2. If either of these occurs the Canoe campaign manager sends an immediate update for the campaign item to the CIP consumer . The update changes the recState of each bundle and placement element to “pse”.
3. The CIP consumer responds to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.
4. The Canoe campaign manager allows the user to extend the flight window after a campaign has completed. The user is allowed to perform this action Z configurable days after the end of the flight window. If the flight end date/time and/or goal (whichever prompted the completion of the campaign item) is not modified in Z days, the Canoe campaign manager sends an update for the campaign item to the CIP consumer. The update changes the recState of each bundle and placement element to “act”.
5. The CIP consumer responds to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

A campaign item flight window is extended after completion

1. The Canoe campaign manager allows the user to extend the flight window after a campaign has completed. The user is allowed to perform this action Z configurable days after the end of the flight window.
2. When the flight window of a completed campaign (completed for reaching its flight end date/time) is extended, the Canoe campaign manager sends an immediate update for the campaign item to the CIP consumer . The update changes the flight window and also changes the recState of each bundle and placement element to “act”.
3. The CIP consumer responds to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

A campaign item goal is raised after completion

1. The Canoe campaign manager allows the user to raise the goal after a campaign has completed. The user is allowed to perform this action Z configurable days after the existing goal had been reached. The goal can be raised in any of three ways:
   1. The quantity can be increased, leaving the goal type unchanged. This will always result in an update to the CIP consumer .
   2. The quantity can be left as is, and the goal type changed. In some cases this may not result in an Update to the CIP consumer .
   3. The quantity and goal type can be changed. In some cases this may not result in an Update to the CIP consumer .
2. When the goal of a completed campaign (completed for reaching its goal) is raised, the Canoe campaign manager sends an immediate update for the campaign item to the CIP consumer . The update changes the goal and/or goal type as appropriate and also changes the recState of each bundle and placement element to “act”.
3. The CIP consumer responds to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

A campaign item reaches windup duration

1. The Canoe campaign manager allows for a campaign item to remain in the recState of “wnd” (Windup) for W configurable duration. The measurement of the duration begins when the Canoe campaign manager sends the UpdateNotice that changes the recState to “wnd”. The Canoe campaign manager recognizes when a campaign item has reached this windup duration.
2. When the windup duration is reached for a campaign item the Canoe campaign manager sends an update to the CIP consumer . The update is sent with the batch. The update changes the recState of each bundle and placement element to “cls”.
3. The CIP consumer responds to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

Ad Assets are changed for a campaign item

1. The User changes the ad assets assigned to a campaign item using the Canoe campaign manager.
2. The Canoe campaign manager sends an immediate update to the CIP consumer .
3. The update changes the recState of the child bundle and placements that contain a “removed” asset (recState changed to “wnd”).
4. The update adds a child bundle and placement for each “added” asset (recState=“act”). **Newly added assets may not be available. If so, they won’t be included in the rotation pool (see 2.7.2 – Ad** **Rotation)**.
5. The CIP consumer responds to each UpdateNotice with an UpdateResponse in the same manner as described in step 13.

The CIP consumer prepares for a planned outage or experience an unplanned outage

1. The CIP consumer may require suspension of campaign item execution to facilitate planned maintenance outages of unplanned outages.
2. To the extent that campaign item execution cannot be halted through the suspension of Decision Engine processing, the CIP consumer may suspend execution
   1. By notifying Canoe Operations 7 days in advance for planned outages
   2. By halting the CIP consumer’s ingestion of UpdateNotifications (which should yield “unsuccessful aknowledgements”)
   3. By changing the recState to “pse” for any CI’s with a recState of “act” as required. CI’s in recStates other than “act” should not be changed. CIP consumer should never change recState to anything other than “pse”
   4. By producing a FeedbackNotice to inform Canoe that the recState has been changed.
   5. By enabling the CIP consumer’s ingestion of UpdateNotifications when the maintenance or downtime has been resolved (this would allow the recStates to be reset to the correct current state).

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. The Canoe campaign manager can monitor asset availability and execution readiness for each campaign item. The Canoe campaign manager has very current statistics on insertions and views for each campaign item.

**Related Use Cases**:

2.7.9 - User Generated Campaign Updates

2.7.10 - System Generated Campaign Updates

2.7.11 - Invalid Campaign Instructions

2.7.12 - Campaign Item Goal Achieved

2.7.13 - Campaign Item Flight End Date Reached

**Additional Information**:

Figure 6 - Campaign Item Information Flow describes the major components of campaign item communication.



Figure - Campaign Item Information Flow

### StateFeedback Communicated from the MSO

**Summary**: The CIP consumer provides the following information through the FeedbackNotice. The FeedbackNotice will be communicated every X configurable minutes. Initially x = 15.

1. The mediaState of an asset
2. The mediaState of the parent Bundle
3. The executionState of the parent Bundle
4. The cumulative totals of insertions for the CI
5. The cumulative totals of views for the CI
6. Notification when the recState of an element containing a StateFeedback element changes

**Actors**: The following actors are engaged:

1. CIP consumer
2. Canoe campaign manager

**Preconditions**: The following pre conditions are expected:

1. The campaign item has been communicated by the Canoe campaign manager and ingested and implemented by the CIP consumer.
2. The campaign item has generated placements.

**Basic course**: The nominal path consists of:

1. The CIP consumer ingests and implements the campaign item.
2. The CIP consumer begins checking asset availability (mediaState) based on the date/time represented in the beginRequired attribute of the ContentMgmt element. The beginRequired date/time will be set to the current date/time that the UpdateNotice was generated by the Canoe campaign manager. The endRequired attribute will = the Flight end date.
3. Availability is reported when the reportAsOf date/time is reached.
4. Availability is reported when the mediaState of an asset changes.
5. The CIP consumer must report mediaState for every asset.
6. The CIP consumer must report executionState for the campaign item when the reportAsOf date/time.
7. When executionState changes the CIP consumer must report executionStae with the next batch update.
8. The CIP consumer accumulates an insertion occurrence each time an ad is placed. Insertion occurrences are reported regardless of the campaign item’s goalType. Insertion occurrences will be aggregated at the parent Bundle and communicated in the batch updates.
9. The CIP consumer accumulates a view occurrence each time it receives viewer metrics indicating an ad was viewed. View occurences are reported regardless of the campaign item’s goalType. View occurences will be aggregated at the parent Bundle and communicated in the batch updates.
10. Each StateFeedback element in the FeedbackNotice shall include the current recState of the element it supports. The CIP consumer shall insure that a FeedbackNotice is generated whenever the CIP consumer initiates a change to an element’s recState if that element contains a StateFeedback element.
11. The Canoe campaign manager receives the feedback and responds with a FeedbackResponse:
    1. If the FeedbackNotice was successfully received, the FeedbackResponse contains a msgResult attribute = “OK”
    2. If the FeedbackNotice was unsuccessful, the FeedbackResponse contains a msgResult attribute = “FAIL” and contain a Notes element containing specific details of the failure. The Canoe campaign manager must log the Acknowledgement as unsuccessful. Canoe Operations will monitor the log and facilitate resolution. MSO operations must also log, monitor and facilitate resolution of all failed FeedbackResponses.

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. The Canoe campaign manager can monitor asset availability and execution readiness for each campaign item. The Canoe campaign manager has very current statistics on insertions and views for each campaign item.

**Related Use Cases**:

2.7.2 - Ad Rotation

2.7.10 - System Generated Campaign Updates

### Detailed Viewership Data

**Summary**: Communicate detailed ad and entertainment viewership metrics to Canoe via SMS.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe data collection endpoint

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.
4. Canoe campaigns have been communicated and made available to the CIP consumer.
5. Pre-flight constraints must be met.
6. FFW is not enabled for any of the ad content.
7. FFW is enabled in the entertainment content.
8. Pause, rewind, and stop are available at any time.

**Basic course**: The nominal path consists of:

1. A subscriber selects play on an entertainment asset from the VOD menu.
2. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
3. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
4. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
5. The viewing session begins (see Figure 7 for a visual representation).
   1. Session begins and subscriber views National pre-roll #1 in 1x (FFW is not enabled in this ad).
   2. Subscriber selects FFW once the entertainment content starts (they have already watched the first half of this show), skipping over National mid-roll # and Local mid-roll #1 (FFW is enabled for both ads)
   3. Subscriber then selects play once the entertainment content starts and continues to watch the programming until half-way through the second segment of the entertainment content.
   4. Subscriber selects RW and returns all the way back to the half-way point of National mid-roll #1.
   5. Subscriber selects play and watches half of National mid-roll #1, all of Local mid-roll #1, and then continues into the entertainment content.
   6. Subscriber selects pause (10 minutes).
   7. Subscriber selects play and watches the remainder of the entertainment content, plus a portion of National post-roll #1 (FFW is not enabled for this ad).
   8. Half-way through National post-roll #1, the subscriber ends their VOD session.
6. Detailed title and ad viewership information is collected and sent as SMS data to the Canoe data collection endpoint on a regular basis.

**Alternative course**: The alternative paths may consist of:

1. A Viewer leaves the ad asset entirely (by rewinding to a point in the stream prior to the ad assets start binding) and returns to the first frame presented at NPT scale of 1 in the forward direction. This must be interpreted as an additional view, separate from any other views recorded for this ad asset during this session.
2. A Viewer plays the first frame of an ad asset at NPT scale of 1 in the forward direction. This represents a view. Without leaving the ad assets binding, the viewer rewinds to the first frame of the ad asset, then plays it at NPT scale of 1 in the forward direction. This is not interpreted as a second view of the ad asset.

**Exceptions**: The following exceptions may occur

Not included

**Postconditions**: The following post conditions are expected:

1. Canoe receives detailed title and ad viewership metrics for each viewer session on a regular basis.

**Related Use Cases**:

None identified

**Additional Information**:

The following visual representation of the viewing session portions use case is offered as additional detail.



Figure - Subscriber VOD Session Example

### Summary Viewership Data

**Summary**: Communicate aggregated ad and entertainment viewership metrics to Canoe via SMS.

**Actors**: The following actors are engaged:

1. MSO client device
2. MSO VOD system
3. CIP consumer
4. Canoe data collection endpoint

**Preconditions**: The following pre conditions are expected:

1. Entertainment content available for play out on the VOD systems.
2. Entertainment content contains ad placement opportunities.
3. Ad content available for play out on the VOD systems.
4. Canoe campaigns have been communicated and made available to the CIP consumer.
5. Pre-flight constraints must be met.

**Basic course**: The nominal path consists of:

1. A number of subscriber viewing session are initiated (over a period of time)
2. The MSO VOD system recognizes that ads need to be dynamically inserted and communicates with the CIP consumer to determine which ads to insert.
3. The CIP consumer leverages the campaign instructions to determine which ads to insert into the available opportunities.
4. The CIP consumer communicates to the VOD system which ads should be inserted, and when.
5. Detailed title and ad viewership information is collected by the MSO VOD system after each viewing session is complete or times out.
6. A view is determined to have occurred when the first frame of an ad is played in NPT with scale=+1.
7. A view
8. On a regular basis the ad viewership information is aggregated for a given period of time on each VOD system.
   1. The number of insertions (see section 1.4) for each ad asset over this time range is calculated.
   2. The number of views (see section 1.4) for each ad asset over this time range is calculated.
   3. Play Time (average) for various scopes.
   4. Run Time (average) for various scopes.
   5. Unique Household count.
9. The data is formatted as SMS data and communicated to the Canoe data collection endpoint on a regular basis.

**Alternative course**: The alternative paths may consist of:

None identified

**Exceptions**: The following exceptions may occur

Title viewership information is out of scope for Phase 1.

**Postconditions**: The following post conditions are expected:

1. A summary of the entertainment titles requested and the ads inserted/viewed is generated in an SMS format and communicated to Canoe on a regular basis.

**Related Use Cases**:

None identified

### Registration

**Summary**: The CIP consumer must register with the CIP publisher to receive CIPs.

**Actors**: The following actors are engaged:

1. CIP consumer
2. CIP publisher (Canoe campaign manager)

**Preconditions**: The following pre conditions are expected:

1. The CIP consumer has the endpoint information required to send a SetRegistrationRequest message.
2. The CIP Publisher has the subscriber endpoint.

**Basic course**: The nominal path consists of:

1. The CIP consumer configures systems to accept UpdateNotice, FeedbackResponse, ReadResponse and InvalidRequest messages from the CIP publisher and send UpdateResponse, FeedbackNotice, and ReadRequest messages to the CIP publisher.
2. The CIP publisher configures systems to send UpdateNotice, FeedbackResponse, ReadResponse and InvalidRequest messages to the CIP consumer and accept UpdateResponse, FeedbackNotice, and ReadRequest messages from the CIP consumer.

**Alternative course**:  N/A

**Exceptions**:

None identified

**Postconditions**:

The Cip consumer is configured for VOD messages in the CIP publisher systems. The CIP Publisher is configured for VOD messages in the CIP consumer systems.

**Related Use Cases**:

None identified

### Operational Performance Monitoring

**Summary**: The CIP consumer must provide data which allows the CIP publisher to verify that VOD systems and processes are operational for logical segments of the footprint (markets, syscodes or some similar segmentation).

**Actors**: The following actors are engaged:

1. CIP consumer
2. CIP publisher (Canoe campaign manager)

**Preconditions**: The following pre conditions are expected:

1. The CIP consumer has established logical segmentation of their footprint and communicated that segmentation information to the CIP Publisher.

**Basic course**: The nominal path consists of:

1. The CIP consumer makes segmentation performance information (insertions and views) available to the CIP publisher.
2. The CIP publisher evaluates the segmentation performance information for indications of operational issues within specific segments of the footprint.
3. The CIP publisher contacts the CIP consumer when segmentation performance information indicates an operational issue exists.

**Alternative course**:  N/A

**Exceptions**:

None identified

**Postconditions**:

The Cip consumer is configured for VOD messages in the CIP publisher systems. The CIP Publisher is configured for VOD messages in the CIP consumer systems.

**Related Use Cases**:

None identified

# Business Requirements

This section defines the business instructions that must be communicated within the Campaign Information Package. This section also defines the business information that must be communicated within the SMS message. This section also defines certain interpretation standards for the instructions contained within these packages.

## General Requirements

|  |
| --- |
| **Requirement/Title/Phase/Description** |
| 1. **/ Phase 1 / Targeting – Opportunity**   The CIP MUST support communication of the Opportunity Type restrictions for a CI. The CIP MUST support the following Opportunity Type designations; preroll, postroll, midroll. The CIP consumer MUST qualify placement candidates based on these Opportunity Type restrictions. As an example, placements restricted to “preroll” will only be eligible for placement opportunities contained within opportunities that have an Opportunity Type of “preroll”. |
| 1. **/ Phase 1 / Targeting – Opportunity**   The CIP consumer MUST interpret the absence of Opportunity Type restrictions to mean that the CI can qualify for any opportunity regardless of OpportunityType, subject to the other constraints of the placement. |
| 1. **/ Phase 1 / Targeting – Opportunity**   The CIP MUST support designation of a specific midroll opportunity for a CI. This designation will be communicated as an integer from 1 thru 99. The CIP consumer MUST qualify placement candidates based on this midroll designation. As an example, when the designation is “1” the CIP consumer considers the placement eligible only for placement opportunities that are within the first midroll opportunity. |
| 1. **/ Phase 1 / Targeting – Opportunity**   The CIP consumer MUST interpret the absence of a specific midroll number designation to mean that the placement can qualify for any midroll opportunity, subject to the other constraints of the placement. |
| 1. **/ Phase 2 / Targeting – Position within Opportunity**   The CIP MUST support communication of Position restrictions for a CI. The CIP consumer MUST support the following designations of Position within a Break; First or Last. First and Last position are scoped by Opportunity Type and Break (defined as a group of PO’s with the same PoGroupIndex). First and Last are not scoped by Inventory Ownership. See Annex I - Position Qualifier Sample. |
| 1. **/ Phase 1 / Targeting – Position within Opportunity**   The CIP consumer MUST interpret the absence of Position restrictions to mean that the placement can qualify for ANY opportunity regardless of position, subject to the other constraints of the placement. |
| 1. **/ Phase 1 / Targeting – Flight Window**   The CIP MUST support communication of the Flight Window of a placement. The CIP consumer MUST restrict placements to placement opportunities from sessions that were requested during the Flight Window (based on local time of the user requesting the VOD session). |
| 1. **/ Phase 1 / Targeting – Day of Week and Time of Day**   The CIP MUST support communication of the Day of Week and Time of Day restrictions for a placement. The CIP MUST support the Day of Week designations sun, mon, tue, wed, thu, fri, sat or any combination of those designations. The CIP consumer MUST interpret Day of Week designations to mean that the placement qualifies for opportunities that are requested (based on local date/time of the user requesting the VOD session) within those designated day(s) of the week. The CIP consumer MUST interpret Time of Day information as a range of time. The CI will be eligible for placement opportunities that stem from sessions requested during this time range (based on local time of the user requesting the VOD session). |
| 1. **/ Phase 1 / Targeting – Day of Week and Time of Day**   The CIP must clearly communicate the association between a Day of Week restriction set and a single Time of Day restriction. These associated restrictions are logically AND’d. Multiple sets are allowed per placement. The CIP consumer must logically OR these sets. |
| 1. **/ Phase 1 / Targeting – Content Metadata**   The CIP MUST support communication of Entertainment Content Metadata qualifiers of a CI. The CIP consumer MUST interpret entertainment content metadata qualifiers to mean that the placement qualifies for opportunities within entertainment content that shares the designated metadata value(s). |
| 1. **/ Phase 1 / Targeting – Content Metadata**   The CIP MUST support communication of the following CableLabs 3.0 Content Metadata elements as qualifiers: Core.Product, Core.AlternateId (provider ID and asset ID), Title.TitleBrief, Title.Title, Title.TitleMedium, Title.TitleLong, Title.EpisodeName, Title.EpisodeID, Title.Rating, Title.Advisory, Title.ShowType, Title.Genre, Content.Resolution, Content.BitRate, Content.ScreenFormat and ProgramName. The CIP consumer MUST be capable of evaluating these entertainment content metadata qualifiers. |
| 1. **/ Phase 1/2 / Targeting – Content Metadata**   The CIP MUST support communication of CableLabs 3.0 Content Metadata qualifiers that use the “contains” operator in addition to the standard eq, ne, lt, gt, ge, le. The CIP consumer MUST be capable of evaluating metadata qualifiers that use these operators. See Annex A – Metadata Operator Matrix. Only the ne and eq operators are in scope for phase 1. |
| 1. **/ Phase 2 / Targeting – Content Metadata**   The CIP MUST support communication of the IsNew metadata condition. The valid values are true/false. The CIP consumer MUST be capable of evaluating the IsNew condition through a comparison of the current date/time against the date range derived from DisplayAsNew. |
| 1. **/ Phase 2 / Targeting – Content Metadata**   The CIP MUST support communication of the IsLastChance metdata condition. The valid values are true/false. The CIP consumer MUST be capable of evaluating the IsLastChance condition through a comparison of the current date/time against the date range derived from DisplayAsLastChance. |
| 1. **/ Phase 1 / Targeting – Content Metadata**   The CIP consumer MUST be capable of transforming from 1.1 Metadata to 3.0 Metadata if metadata restrictions are described using 1.1 attributes. |
| 1. **/ Phase 2 / Targeting – Content Metadata**   As part of a Phase 2 delivery, the CIP should also support communication of the following Cablelabs 3.0 Content Metadata elements as qualifiers: Title.Year, and Content.Language. The CIP consumer MUST be capable of evaluating these entertainment content metadata qualifiers. |
| 1. **/ Phase 2 / Product Category Separation**   The CIP MUST support communication of product category separation restrictions that clearly define a CI’s exclusion category and the scope of the exclusions. The CIP consumer MUST be capable of evaluating these product category separation restrictions and insuring that placements do not violate these restrictions. |
| 1. **/ Phase 2 / Product Category Separation**   Every placed asset must have an assigned product category (category name) using the Nielson PCC string values. |
| 1. **/ Phase 2 / Product Category Separation**   The CIP MUST support communication of a category name for each asset, an indication of the desire for exclusivity and the scope of the exclusion. The scope of exclusion can be opportunity, group, or session. The CIP consumer MUST interpret the scope designation of “opportunity” to mean the ad cannot be placed in the same Opportunity more than once. The CIP consumer MUST interpret the scope designation of “group” to mean the ad cannot be placed twice in opportunities which share the same ownership and PoGroupIndex. The CIP consumer MUST interpret the scope designation of “session” to mean the ad cannot be placed in the same VOD session more than once. |
| 1. **/ Phase 2 / Product Category Separation**   The CIP consumer MUST interpret the presence of exclusion scope for a placement to mean that a product category separation restriction exists. The CIP consumer MUST interpret the absence of exclusion scope for a CI to mean that a product category separation restriction does not exist. |
| 1. **/ Phase 2 / Product Category Separation**   Once placements with restrictions are made the CIP consumer must avoid violating those restrictions with subsequent placements. |
| 1. **/ Phase 2 / Ad Copy Separation**   The CIP MUST support communication of an Ad Copy Separation restriction for each ad. The CIP MUST support the restriction designations of opportunity, group, or session. The CIP consumer MUST interpret the restriction designation of “opportunity” to mean the ad cannot be placed in the same Opportunity more than once. The CIP consumer MUST interpret the restriction designation of “group” to mean the ad cannot be placed twice in opportunities which share the same ownership and PoGroupIndex. The CIP consumer MUST interpret the restriction designation of “session” to mean the ad cannot be placed in the same VOD session more than once. |
| 1. **/ Phase 2 / Ad Copy Separation**   The CIP consumer MUST interpret the absence of an Ad Copy Separation restriction for an ad to mean that the ad can be placed multiple times in opportunities, group, or sessions but cannot be placed adjacent to themselves within opportunities of shared ownership with the same PoGroupIndex. |
| 1. **/ Phase 1 / Ad Rotation**   The CIP MUST support communication of the rotation value for an ad. |
| 1. **/ Phase 1 / Ad Rotation**   The CIP MUST support communication of a grouping of ads that MUST be rotated for a CI. |
| 1. **/ Phase 1 / Ad Rotation**   The CIP consumer MUST identify eligible ads for each placement, interpret the Rotation information and alternate the ads in the proper ratio. |
| 1. **/ Phase 1 / Ad Rotation**   If all eligible ads for a placement lack rotation values, the CIP consumer MUST rotate the ads as if they had the same rotation values. |
| 1. **/ Phase 1 / Ad Rotation**   The CIP consumer MUST rotate each eligible ad based on the ratio from the following calculation:  rotation allocation for ad / sum of the rotation allocations for all eligible ads within a Placement. Canoe requires that a given set of eligible ads in a rotation, with set rotation allocations for each ad, will provide selection results within 2% of the rotation ratios represented by their allocations, over a sample of 10,000 rotations. |
| 1. **/ Phase 1 / Ad Rotation**   If an ad asset is not available to be placed (as indicated by the mediaState within StateFeedback), the CIP consumer MUST not place the ad. |
| 1. **/ Phase 1 / Ad Rotation**   If an ad asset is not assigned to a campaign item (represented by an AssetReference in a Placement within a parent Bundle) the CIP consumer MUST not place the ad. |
| 1. **/ Phase 1 / Ad Rotation**   If an ad asset violates Category Separation constraints, the CIP consumer MUST not place the ad. |
| 1. **/ Phase 1 / Ad Rotation**   If an ad asset violates Ad Copy Separation constraints, the CIP consumer MUST not place the ad. |
| 1. **/ Phase 1 / Ad Rotation**   If an ad asset violates the constraints of the remaining duration of the opportunity, the CIP consumer MUST not place the ad. |
| 1. **/ Phase 1 / Priority**   The CIP MUST support communication of the defined Priority of a CI. |
| 1. **/ Phase 1 / Performance Goals**   The CIP MUST support communication of a Goal for each CI. A goal can be null (no goal), or any non-negative integer. |
| 1. **/ Phase 1 / Performance Goals**   The CIP MUST support communication of a goal type for each CI. For Phase 1, the valid value for goal type is “views”. |
| 1. **/ Phase 1 / Performance Goals**   Campaign items with a goal type of “Views” continue to place until the goal is met. A View occurs when the viewer who requested a vod session “sees” an ad from a placement. |
| 1. **/ Phase 2 / Performance Goals**   For Phase 2, the CIP MUST also support “insertions” as a valid goal type. |
| 1. **/ Phase 2 / Performance Goals**   Campaign items with a goal type of “Insertions” continue to place until the goal is met. An Insertion occurs when an ad from a campaign item is selected for placement in a session. |
| 1. **/ Phase 1 / Performance Goals**   The CIP consumer MUST interpret the absence of a goal to mean that the campaign item will execute until the end of the flight window is reached. |
| 1. **/ Phase 2 / Performance Goals**   The CIP consumer MUST support transitioning between Goal Types at any point during a campaign item’s flight window. The system must support moving from one Goal Type to another with reporting that reflects campaign item life to date statistics as if the campaign item had been that goal type since inception. |
| 1. **/ Phase 1 / Performance Goals**   The CIP consumer MUST support modifications of goal amounts at any point during a campaign item’s flight window. The system must support moving from one Goal to another, with execution reflecting the current status to the new goal. |
| 1. **/ Phase 3 / Trick Mode Control**   The CIP MUST support communication of the desired Trick Mode behavior. The two designations that must be supported are FFW Enabled and FFW Disabled. The CIP consumer MUST interpret FFW Enabled to mean that the viewer is allowed to fast-forward through ads. The CIP consumer MUST interpret FFW Disabled to mean that the viewer is not allowed to fast-forward through ads. |
| 1. **/ Phase 3 / Trick Mode Control**   The Trick Mode Control MUST NOT impact the trick mode behavior allowed within the entertainment content. |
| 1. **/ Phase 3 / Trick Mode Control**   The Trick Mode Control MUST be subservient to any Trick Mode Control rules that the MSO otherwise has in place. |
| 1. **/ Phase 3 / Trick Mode Control**   As part of future releases, the CIP should accommodate communication of additional trick modes beyond FFW. |
| 1. **/ Phase 1 / Performance Feedback**   The CIP MUST support communication of the desired Performance Feedback Standards. The performance feedback information that must be provided is Asset Status, Execution Readiness, View Count, and Insertion Count. |
| 1. **/ Phase 1 / Performance Feedback**   Asset Status feedback must be provided to Canoe when the reportAsOf attribute date/time has been reached and/or the availability of the asset changes. The feedback must provide an AssetID, ProviderID, Status (unknown|processing|ready|private:.+), and a time stamp for the Status Change. |
| 1. **/ Phase 1 / Performance Feedback**   Execution readiness status must be provided to Canoe when the reportAsOf attribute date/time has been reached and/or the exectution status has changed. The feedback must provide a Status (notReady|processing|ready|private:.+) and a time stamp for the Status Change. |
| 1. **/ Phase 1 / Performance Feedback**   View Count statistics must be provided to Canoe whenever an ad is viewed. An ad is said to be viewed if the first frame of the ad is played in Normal Play Time. View count statistics should be provided to Canoe within 15 minutes of the teardown of the session that the view occurred for. View count statistics will be provided as cumulative totals. |
| 1. **/ Phase 1 / Performance Feedback**   Insertion Count statistics must be provided to Canoe whenever an ad is placed. An ad is said to be inserted when the ADS sends the ad to the ADM for insertion in the playlist. No validation of playout is required. Insertion count statistics should be provided to Canoe at least every X configurable minutes (initially set to 15 minutes). Insertion count statistics will be provided as cumulative totals. |
| 1. **/ Phase 1 / Campaign Item Updates - Informative**   The CIP consumer should expect and accommodate changes to the any aspect of the CIP instructions representing a campaign item except for peid’s, guids and epsids. |
| 1. **/ Phase 1 / Campaign Item Updates - Informative**   The Canoe campaign stewardship system communicates in-flight campaign updates immediately (Does Not wait for the next batch update) when certain attributes are modified. |
| 1. **/ Phase 1 / Campaign Item Updates**   The CIP consumer MUST ingest and implement campaign information within 60 seconds of receipt.  Propagation shall occur in the timeframes of standard system processes (the same as other campaigns managed within the CIP consumer’s systems).  Any time required for performance of checks for media or execution readiness is additional. |
| 1. **/ Phase 1 / Campaign Item Updates**   The CIP consumer should expect batch updates to in-flight campaign items as often as every 15 minutes but in no case less frequently than every 24 hours. Initially this interval will be every 6 hours. Changes too many campaign elements and attributes are held for communication in the next batch update. |
| 1. **/ Phase 1 / Resolution Compatibility**   The CIP consumer must insure that the resolution of placed ads is compatible with the resolution of the entertainment asset. The CIP publisher is not required to communicate any specific qualification for resolution. The CIP Publisher may communicate a resolution qualifier which restricts placement to entertainment assets based on resolution, but this restriction does not relinquish the CIP consumer’s responsibility to insure resolution compatibility. The CIP consumer should insure that opportunities do not remain unfilled when qualifying campaigns with resolution compatible ad assets exist. |
| 1. **/ Phase 1 / Decision**   The CIP consumer must utilize opportunities for placement to the fullest extent possible. This means ads will continue to be inserted into an opportunity until insertion of any of the remaining qualified ads would cause the aggregate duration of all ads placed (in the opportunity) to exceed the established duration of that opportunity. |
| 1. **/ Phase 1 / Decision**   Canoe expects only those placements which match the characteristics of the placement opportunity will be considered for execution. These placement opportunity characteristics are;   * Opportunity Type of the opportunity containing the placement opportunity * Midroll designation of the opportunity containing the placement opportunity * Position in Opportunity of placement opportunity * CurrentDateTime the session containing the placement opportunity was requested (matched to flight window, Day of Week and Time of Day restrictions) * CL 3.0 Metadata relating to the entertainment asset of the session containing the placement opportunity |
| 1. **/ Phase 1 / Decision**   Canoe expects that only those campaign items in the proper state (recState=’act”) will be considered for placement. |
| 1. **/ Phase 1 / Summary Viewership Data**   The CIP consumers must provide SMS viewership data within 15 minutes of session teardown. |
| 1. **/ Phase 1 / Summary Viewership Data**   The CIP consumer MUST provide the summary data defined by Annex J- Reporting - Definitions/Data Point Configurations. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST contain the data necessary to generate the following ad viewership summary report (for multiple ads):  Filter by:   * Network (All / Selected) * Date Range (All / Custom Start & End) * title asset ID * campaign identifier/name * Operator * VOD system * DMA * zip code   Fields:   * Total Gross Views * Average Gross Views per HH * Unique HHs Viewing * Total Run Time * Average run time per view * Total scheduled insertions * Total views   For each ad: asset ID, name (title), gross views, views/hh, unique hh, total runtime |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST contain the data necessary to generate the following ad viewership details report (for a single ad):  Filter by:   * Ad asset ID * Network * MSO * VOD system * DMA * zip code   Fields:   * Network * Asset ID * Asset name (title) * Total gross views * Average gross views per HH * Unique HH’s viewing * Total run time * Average run time per view * Average run time per HH * Total scheduled insertions * Total views * Number of insertions for each opportunity and position pairing * For each title inserted against: title asset ID and number of insertions per title inserted against |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST contain the data necessary to generate the following ad viewership log report (for a single ad):  Filter by:   * Ad asset ID * Date Range (All / Custom Start & End) * Network * MSO * VOD system * DMA * zip code   Fields:   * Unique identifier for the playout of the ad * Date/time and timezone of playout * Duration of ad (formatted as MM:SS) * % of the ad that was viewed (duration spent in 1x / duration of the ad) * Trick mode [Enabled/Disabled] * Break number placed into * Position number within break ad was placed into * Campaign identifier/name * Asset ID of title inserted into |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST contain the data necessary to generate the following title viewership summary report (for multiple titles):  Filter by:   * Programming Network (All/Selected) * Date Range (All/Custom Start & End) * MSO * DMA * VOD system * Zip code * Ad asset ID   Fields:   * Total Gross Views * Average Gross Views per HH * Unique HHs Viewing * Total Run Time * Average Run Time per View * Average Run Time per HH * For each title: Title name, title asset ID, gross views, views/hh, unique HH viewing, total runtime |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST contain the data necessary to generate the following title viewership details report (for a single title):  Filter by:   * Title asset ID * Network * MSO * VOD system * DMA * zip code   Fields:   * Network * Report date range * Title * Asset ID * Total gross views * Average gross views per HH * Unique HH’s viewing * Total run time * Average run time per view * Average run time per HH |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST contain the data necessary to generate the following title viewership log report (for a single title):  Filter by:   * Title asset ID * Network * MSO * VOD system * DMA * zip code   Fields:   * Unique identifier for the playout of the entertainment content * Date/time and timezone of play * Duration of the title asset * % of the title that was viewed (duration spent in 1x / duration of the title asset) * Trick mode [Enabled/Disabled] * The ability to link to view all ads inserted in the session * The ability to link to view all ads viewed within the session |
| 1. **/ Phase 1 / Detailed Viewership Data**   Total Gross Views is defined as the gross count of playouts for a given title or ad asset in 1X mode |
| 1. **/ Phase 1 / Detailed Viewership Data**   Average Gross Views per HH is defined as the Total Gross Views / Total Unique HH’s viewing |
| 1. **/ Phase 1 / Detailed Viewership Data**   Total Unique HHs Viewing is defined as the total unique count of playouts for a given title or ad asset in 1X mode |
| 1. **/ Phase 1 / Detailed Viewership Data**   Total Run Time is defined as the total sum of run times for a given title or ad asset regardless of any trick modes that may have been initiated. |
| 1. **/ Phase 1 / Detailed Viewership Data**   Average Run Time per View is defined as the Total Run Time / Total Gross Views (Initiated) for a given title or ad asset. |
| 1. **/ Phase 1 / Detailed Viewership Data**   Average Run Time per HH is defined as the Total Run Time / Total Unique HHs Viewing (Initiated) for a given title or ad asset. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST include the date, time, and time zone associated with the start and end of the viewer’s session. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST accommodate communication of the MSO and VOD system that the data is originating from. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST include a unique HH and unique device identifier for each set of viewer session data. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST provide a distinct data set for each portion of normal playback for each asset (both ads and titles). |
| 1. **/ Phase 3 / Detailed Viewership Data**   The detailed SMS message MUST provide a distinct data set for each portion of trick mode playback for each asset (both ads and titles). |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST group or provide a way to link all playback data sets for a single viewer session together. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST indicate the asset ID of the asset being represented within each playback data set. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST indicate the start and end date/time/time zone of each playback data set. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST indicate the normal play time location at the start and end for each playback data set. |
| 1. **/ Phase 3 / Detailed Viewership Data**   The detailed SMS message MUST indicate the trick mode used and the speed (where applicable) for each playback data set. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST include the Asset ID (as originally provided by the Programmer) for each asset (both title and ads) being reported on. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST include insertion data on all ad placements regardless of whether or not the ad was actually played out to the viewer (see section 1.4 for the definition of Scheduled Insertions). |
| 1. **/ Phase 3 / Detailed Viewership Data**   The detailed SMS message MUST accommodate the communication of whether or not the FFW trick mode was enabled for a given ad placement. |
| 1. **/ Phase 3 / Detailed Viewership Data**   The detailed SMS message MUST accommodate the communication of whether or not the FFW trick mode was enabled for a given ad placement. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The detailed SMS message MUST accommodate the communication of the break number and spot position number that a given ad was placed into. |
| 1. **/ Phase 1 / Detailed Viewership Data**   The SMS summary data MUST provide an aggregate view of the following metrics for a specific date/time range:   1. Total scheduled ad insertions (ads inserted into a playlist, regardless of whether or not they were watched by the viewer) 2. Total ad views (at least 1 frame of the ad is viewed in normal play mode) 3. Total title views (the entertainment title is requested for viewing) |

# Technical Requirements

This section defines low-level technical requirements for the Campaign Information Package and SMS message as well as certain interpretation standards for the instructions contained within these packages. Where appropriate, the business requirements defined in section 3 will have a corresponding set of technical requirements defined in this section.

## General Requirements

|  |
| --- |
| **Requirement/Title/Phase/Description** |
| 1. **/ Phase 1 / CIP Structure - informative**   Canoe will publish productFamily and productMember definitions. These definitions will be appropriately versioned to facilitate change management. The CIP consumer may use these definitions within their execution processes. See Annex H - ProductFamily and ProductMember Definitions. |
| 1. **/ Phase 1 / CIP Structure**   Canoe may include elements and attributes in a CIP which are not used universally by all CIP consumers. The CIP consumer MUST ignore any elements and attributes within a CIP which are extraneous to their implementation. |
| 1. **/ Phase 1 / Targeting – Opportunity Type**   Opportunity type restrictions will be defined through qualifiers in the parent Bundle. |
| 1. **/ Phase 1 / Targeting – Opportunity Type**   Opportunity attributes must be available as targets through a designated namespace. The current need is for targeting opportunity types of slate, pre-roll, mid-roll, post-roll, interstitial as well as mid-roll by number. |
| 1. **/ Phase 1 / Targeting – Opportunity Type**   The CIP Consumer must recognize targeting for any combination of the opportunity types of slate, pre-roll, mid-roll, post-roll, and interstitial. |
| 1. **/ Phase 1 / Targeting – Opportunity Type**   Opportunity is currently defined within a Term element with source=”ent”, operation=”eq” or “ne”, xpath = "typeOfPlacementOpportunity" and namespace =<http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1> with the value set to the qualifying OpportunityType.  Valid values for "typeOfPlacementOpportunity" are pre-roll|mid-roll|post-roll|interstitial|slate|private:.+  As an example, a term specifying midroll would look like:  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">mid-roll</Term> |
| 1. **/ Phase 1 / Targeting – Mid-roll by number**   Opportunity sequence is currently defined within a Term element with source=”ent”, operation=”eq” or “ne”, xpath = "poGroupIndex" and  namespace=<http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1> with the value set to the qualifying poGroupIndex. The use of the “poGroupIndex” xpath indicates that a specific mid-roll PoGroupIndex is being targeted.  There are specific rules for PlacementOpportunity metadata.   * If a PlacementOpportunity contains typeOfPlacementOpportunity=”pre-roll”, the poGroupIndex must always = ”1”. * If a PlacementOpportunity contains typeOfPlacementOpportunity=”mid-roll”, the poGroupIndex must always be between 2 and 98.  They always start at 2, even if there is no pre-roll. * If a PlacementOpportunity contains typeOfPlacementOpportunity=”post-roll”, the poGroupIndex must always = ”99”. * If a PlacementOpportunity contains typeOfPlacementOpportunity=(”slate” or “interstitial”), the poGroupIndex is not constrained.   The constraints of these rules allow us to identify mid-roll break X as poGroupIndex X+1.  As an example, a term specifying the 1st mid-roll break is:  <Term source="ent" operation="eq" xpath="poGroupIndex" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">2</Term> |
| 1. **/ Phase 1 / Targeting – Opportunity Type**   The execution engine MUST interpret the absence of an Opportunity qualifier to mean that the placement can qualify for ANY opportunity regardless of opportunity type. |
| 1. **/ Phase 1 / Targeting – Position**   Position restrictions will be defined through qualifiers in the parent Bundle. |
| 1. **/ Phase 2 / Targeting – Position**   Position attributes must be available as targets through a designated namespace. The current need is for First and Last designations. |
| 1. **/ Phase 2 / Targeting – Position – IsFirst/IsLast**   IsFirst and IsLast are special cases of “order”. They are scoped by opportunityType within a set of opportunities with a common PoGroupindex.  As an example, a term specifying the IsFirst position would look like:  <Term source="ent" operation="eq" xpath="IsFirst" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">True  </Term> and target the first placement of the first opportunity in the PoGroupIndex.  A term specifying the IsLast position would look like:  <Term source="ent" operation="eq" xpath="IsLast" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">True  </Term> and target the last placement of the last opportunity in the PoGroupIndex.  This is a distinctly different qualifier than:  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">1</Term> which targets any placement within the first opportunity in a PoGroupIndex.  OR  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">3  </Term> which targets any placement in the third opportunity in a PoGroupIndex. |
| 1. **/ Phase 1 / Targeting – Position**   The CIP consumer MUST interpret the absence of a Position qualifier (no “IsFirst” or “IsLast” qualifier) to mean that the placement can qualify for ANY placement in the PO, regardless of position. |
| 1. **/ Phase 1 / Targeting – Flight Window**   The CIP consumer MUST interpret the flight window to start on the earliest date/time that a placement can execute and to end on the latest date/time that a placement can execute. The Flight Window is defined in the Bundle with the Flight element.  As an example, the Flight element below describes a flight window that starts on 12/1 @ 9:00 am and ends on 12/31 @ 1:00 pm.  <Flight startDateTime="2011-12-01T09:00:00" endDateTime="2011-12-31T13:00:00" startTime="09:00:00" duration="PT04H00M00S" /> |
| 1. **/ Phase 1 / Targeting – Day of Week/Time of Day**   Day of Week and Time of Day targeting will be defined through the Flight element. The MSO can expect that Canoe will define Flight at the Bundle. The Flight will be defined using all attributes of the EffectiveDatesDayTimeType.  *<xs:element name="Flight" type="cip:EffectiveDatesDayTimeType" minOccurs="0" maxOccurs="unbounded"/>*   * startDateTime - provides the earliest date and time on that date of execution for this CI and will contain an offset. * endDateTime - provides the latest date and time on that date of execution for this placement and will contain an offset * DayofWeek Booleans - designate which days the placement is active (true) or inactive (false). If a DayofWeek is not designated, it should be assumed to be active (true). Each set of DayofWeek Booleans is paired with a time frame defined by the combination of the startTime and duration attributes. * startTime – Start time of day for being active. For example, 09:00:00 represents 9:00am local time;. Canoe will express startTimes as local times. If startTime is omitted, the placement can execute for the entirety of the days listed based on local time. * Duration – Effective duration of time from the start time. If duration is supplied and startTime is omitted, duration MUST be ignored. Note that duration can extend into the next day. If an end date/time is specified, it MUST be observed when calculating the actual end time based on startTime and duration, and supersedes the duration attribute.   As an example, the Flight element below describes a flight window that is only active on weekdays, from 9:00 am thru 1:00 pm.  <Flight startDateTime="2011-12-01T09:00:00-05:00" endDateTime="2011-12-31T13:00:00-05:00" startTime="09:00:00" duration="PT04H00M00S" sat="false" sun="false"/>  There will always be at least one Flight element. **The MSO should expect multiple Flight elements.** |
| 1. **/ Phase 1 / Targeting – Content Metadata**   Cablelabs Metadata 3.0 attributes must be available as targets through a designated namespace. The Metadata terms required by Canoe (defined in Annex A – Metadata Operator Matrix) are currently defined through the following namespaces:   * "cablelabs.com/namespaces/metadata/xsd/title/1" * "cablelabs.com/namespaces/metadata/xsd/core/1" * "cablelabs.com/namespaces/metadata/xsd/offer/1" * "cablelabs.com/namespaces/metadata/xsd/content/1"   Series, Season and Episode Information must also be available (current suggestion is to obtain this through EIDR)   * “eidr.org/schema/1.0/SeriesInfo" (For Series ResourceName) * "eidr.org/schema/1.0/SeasonInfo" (For ReleaseDate of Season) * "eidr.org/schema/1.0/EpisodeInfo" (For Episode ResourceName)   As examples:  <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">The Office</Term>  <Term source="ent" operation="eq" xpath="providerID" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">NBC.COM</Term> |
| 1. **/ Phase 1 / Targeting – Content Metadata**   All metadata target criteria will be defined using the Metadata 3.0 specification. The CIP consumer MUST be capable of transforming from 3.0 Metadata to 1.1 Metadata when the metadata referenced by the MSO is provided in 1.1 specifications. |
| 1. **/ Phase 1 / Targeting – Content Metadata**   All Metadata qualifiers will be included in the CommonQualifierGroup elements within the parent Bundle. |
| 1. **/ Phase 1 / Targeting – Content Metadata**   In cases where Content MetaData is received and subsequently manipulated by the CIP consumer, the manipulated data will be used for execution of the campaign item. |
| 1. **/ Phase 2 / Targeting – Content Metadata**   Canoe expects the CIP consumer to recognize the IsNew Metadata qualifier. As an example;  <Qualifier>  <Term source="ent" operation="eq" xpath="IsNew" namespace="cablelabs.com/namespaces/metadata/xsd/offer/1"> true</Term>  </Qualifier>  When IsNew = true the CIP consumer MUST interpret this to mean that the request date/time of the vod session must be within the IsNew date/time range.  For 3.0 metadata the IsNew range is determined as follows; Start = Offer.startDateTime, End = (Offer.startDateTime + Presentation.displayAsNew) |
| 1. **/ Phase 2 / Targeting – Content Metadata**   The CIP consumer MUST recognize the IsLastChance Metadata qualifier. As an example;  <Qualifier>  <Term source="ent" operation="eq" xpath="IsLastChance” namespace="cablelabs.com/namespaces/metadata/xsd/offer/1"> true</Term>  </Qualifier>  When IsLastChance = true the CIP consumer MUST interpret this to mean that the request date/time of the vod session must be within the IsLastChance date/time range.  For 3.0 metadata the IsLastChance range is determined as follows; Start = (Offer.endDateTime - Presentation.DisplayAsLastChance), End = Offer.endDateTime. |
| 1. **/ Phase 1 / Targeting – parent Bundle Qualifiers - Informative**   The list of CommonQualifierGroup elements defines qualifier expressions that MUST be satisfied for any placement that is a child of the parent bundle. Multiple CommonQualifierGroups are always ANDed, which means that child placements only execute if every group is individually true.  The CommonQualifierGroup will contain a Qualifier for every permutation of the Terms representing a campaign item’s:   * Break/Position Condition Sets * Metadata Groups * Networks Selected   Sample:   * Break/Position Sets   + <! – Break-Position Set #1 -->     - <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>     - <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>   + <! – Break-Position Set #2 -->     - <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term> * Networks Selected (3 selected for this campaign item) – >   + <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TCM.COM</Term>   + <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TNT.COM</Term>   + <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TBS.COM</Term> * Metadata Groups   + <!-- Metadata Group #1 -->     - <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Cheers</Term>     - <Term source="ent" operation="eq" xpath="DisplayAsNew" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>   + <!-- Metadata Group #2 -->     - <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Friends</Term>     - <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>   + <!-- Metadata Group #3 -->     - <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Sienfeld</Term>     - <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>   In this sample we will have 18 Qualifier elements (2 break/position conditions x 3 networks selected x 3 metadata groups)  <CommonQualifierGroup condition="or">  <Qualifier>  <! -- Break-Position Set #1, TCM, Metadata Group #1 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TCM.COM</Term>  <!-- Metadata Group #1 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Cheers</Term>  <Term source="ent" operation="eq" xpath="DisplayAsNew" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #1, TNT, Metadata Group #1 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TNT.COM</Term>  <!-- Metadata Group #1 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Cheers</Term>  <Term source="ent" operation="eq" xpath="DisplayAsNew" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #1, TBS, Metadata Group #1 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TBS.COM</Term>  <!-- Metadata Group #1 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Cheers</Term>  <Term source="ent" operation="eq" xpath="DisplayAsNew" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #1, TCM, Metadata Group #2 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TCM.COM</Term>  <!-- Metadata Group #2 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Friends</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #1, TNT, Metadata Group #2 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TNT.COM</Term>  <!-- Metadata Group #2 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Friends</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #1, TBS, Metadata Group #2 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TBS.COM</Term>  <!-- Metadata Group #2 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Friends</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #1, TCM, Metadata Group #3 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TCM.COM</Term>  <!-- Metadata Group #3 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Sienfeld</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <! -- Break-Position Set #1, TNT, Metadata Group #3 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TNT.COM</Term>  <!-- Metadata Group #3 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Sienfeld</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #1, TBS, Metadata Group #3 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">Last</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TBS.COM</Term>  <!-- Metadata Group #3 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Sienfeld</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #2, TCM, Metadata Group #1 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TCM.COM</Term>  <!-- Metadata Group #1 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Cheers</Term>  <Term source="ent" operation="eq" xpath="DisplayAsNew" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #2, TNT, Metadata Group #1 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TNT.COM</Term>  <!-- Metadata Group #1 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Cheers</Term>  <Term source="ent" operation="eq" xpath="DisplayAsNew" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #2, TBS, Metadata Group #1 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TBS.COM</Term>  <!-- Metadata Group #1 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Cheers</Term>  <Term source="ent" operation="eq" xpath="DisplayAsNew" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #2, TCM, Metadata Group #2 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TCM.COM</Term>  <!-- Metadata Group #2 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Friends</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #2, TNT, Metadata Group #2 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TNT.COM</Term>  <!-- Metadata Group #2 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Friends</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #2, TBS, Metadata Group #2 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TBS.COM</Term>  <!-- Metadata Group #2 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Friends</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #2, TCM, Metadata Group #3 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TCM.COM</Term>  <!-- Metadata Group #3 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Sienfeld</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <! -- Break-Position Set #2, TNT, Metadata Group #3 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TNT.COM</Term>  <!-- Metadata Group #3 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Sienfeld</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  <Qualifier>  <! -- Break-Position Set #2, TBS, Metadata Group #3 -->  <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">TBS.COM</Term>  <!-- Metadata Group #3 -->  <Term source="ent" operation="contains" xpath="Title.Title" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">Sienfeld</Term>  <Term source="ent" operation="eq" xpath="DisplayAsLastChance" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">true</Term>  </Qualifier>  </CommonQualifierGroup> |
| 1. **/ Phase 2 / Targeting – Product Category Separation**   Product Category Separation (AKA Category Exclusion) is defined using the CategoryExclusion element in the child bundle. The CategoryExclusion element is a control for enforcing product category separation within opportunities, groups of opportunities with the same PoGroupIndex, and viewer session. The scope attribute indicates how broad the exclusion test needs to be and can have the values “opportunity”, “group” or “session”.  Category Schema  *<xs:complexType name="CategoryExclusionType">  <xs:attribute name="scope" type="cip:ExclusionScopeType" use="required"/> </xs:complexType>*  As an example;  <CategoryExclusion scope=”opportunity”/> |
| 1. **/ Phase 2 / Targeting – Product Category Separation - opportunity**   When scope=”opportunity” the CIP consumer must insure;   1. that no placements occur if a pre-existing placement of the same category exists in the opportunity. 2. that once placed, no other placement with the same category can be placed in the that opportunity. |
| 1. **/ Phase 2 / Targeting – Product Category Separation - group**   When scope=”group” the CIP consumer must insure;   1. that no placements occur if a pre-existing placement of the same category exists in an opportunity with the same owner and PoGroupIndex. 2. that once placed, no other placement with the same category can be placed into opportunities within the same owner and PoGroupIndex. |
| 1. **/ Phase 2 / Targeting – Product Category Separation - session**   When scope=”session” the CIP consumer must insure;   1. that no placements occur if a pre-existing placement of the same category exists in the session 2. that once placed, no other placement with the same category can be placed in the session. |
| 1. **/ Phase 2 / Targeting – Product Category Separation**   The absence of CategoryExclusion MUST be interpreted by the CIP consumer to mean that no category exclusions are present for the placement. |
| 1. **/ Phase 2 / Targeting – Product Category Separation**   The “category” is defined using 1 or more AssetMetadata elements in the AssetReference. Values must be defined using the namespace=<http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1> for 1 or more of the following xpaths: IndustryGroup, MajorGroup, PCCDescription, ProductCategory.  As an example;  <AssetReference epsid="0" peid="/ld7ekaAEd6ukwAf4hSEQQ" apply="valAndPlace" assetType="media">  <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">CSMK1600002900000000</AssetMetadata>  <AssetMetadata xpath="IndustryGroup" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS, OFF. EQUIP. &amp; STATIONERY</AssetMetadata>  <AssetMetadata xpath="MajorGroup" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS &amp; DATA PROCESSING EQUIPMENT</AssetMetadata>  <AssetMetadata xpath="PCCDescription" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS,COMPONENTS &amp; ACCESS</AssetMetadata>  <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">OFFICE AUTOMATION SYS</AssetMetadata>  <ContentMgmt beginRequired="2011-09-29T00:00:00.000-05:00" endRequired="2011-10-04T09:00:00-05:00"/>  <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback> </AssetReference> |
| 1. **/ Phase 2 / Targeting – Product Category Separation – Default Restriction**   If “category” is not defined within an AssetMetadata element in the AssetReference, the campaign item will be rejected by the CIP consumer. |
| 1. **/ Phase 2 / Targeting – Ad Copy Separation**   The CIP MUST support communication of an Ad Copy Separation constraint. Ad Copy Separation constraints will be communicated using the separationConstraint attribute of the child bundle. All assets within a child bundle must therefore adhere to the same separationConstraint. |
| 1. **/ Phase 2 / Targeting – Ad Copy Separation**   The Ad Copy Separation constraint in the CIP must support the separation of an ad from itself (from being shown more than once) within a single opportunity (separationConstraint=”opportunity”), within opportunities of common ownership sharing a PoGroupIndex (separationConstraint=“group”), within a single viewer session (separationConstraint=”session”), or restricted only to the prevention of adjacent placements within opportunities sharing common ownership and a PoGroupIndex (when separationConstraint not defined). |
| 1. **/ Phase 2 / Targeting – Ad Copy Separation - opportunity**   When separationConstraint=”opportunity” the CIP consumer must insure;   1. that no placements occur if the asset has previously been placed in the same opportunity 2. that once placed, no other placement of the same asset can be made in that opportunity |
| 1. **/ Phase 2 / Targeting – Ad Copy Separation - group**   When separationConstraint=”group” the CIP consumer must insure;   1. that no placements occur if the asset has previously been placed in opportunities with the same owner and PoGroupIndex. 2. that once placed, no other placement of the same asset can be made in opportunities with the same owner and PoGroupIndex. |
| 1. **/ Phase 2 / Targeting – Ad Copy Separation - session**   When separationConstraint=”session” the CIP consumer must insure;   1. that no placements occur if the asset has previously been placed in the same session. 2. that once placed, no other placement of the same asset can be made in the session. |
| 1. **/ Phase 2 / Targeting – Ad Copy Separation – Default restriction**   When separationConstraint=”” the CIP consumer must prevent adjacent placements within opportunities sharing common ownership and a PoGroupIndex |
| 1. **/ Phase 1 / Priority**   The CIP MUST accommodate communication of the defined Priority of a campaign item. Canoe will communicate the priority of a placement using the priority attribute in the Bundle. The Priority value will fall within the range 1-400, |
| 1. **/ Phase 1 / Priority**   The CIP consumer must interpret a priority of “1” to be “better” than a priority of “2” (the lower the number, the “better” the priority).  The CIP consumer will give execution preference to the campaing item with the best priority. |
| 1. **/ Phase 1 / Priority**   When 2 or more campaign items have the same priority and otherwise qualify for an opportunity, the CIP consumer (Execution Engine) MUST use a random selection method (giving each campaign item equal chance of selection) to determine the placement at decision time. |
| 1. **/ Phase 1 / Performance Goals**   Canoe will communicate a goal and goalType for each campaign item. Canoe will provide the goal and goalType using a Phase and rule set in the parent bundle. As an example;  <Phase epsid="0" peid="1111ZM5AEd6ukwAf4hSEQQ" invoked="defined">  <ProcessRuleSelector ruleId="SetGoal" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  <Arg variable="goal">100000</Arg>  <Arg variable="goalType">views</Arg>  </ProcessRuleSelector> </Phase> |
| 1. **/ Phase 1 / Performance Goals**   Canoe has designated the “SetGoal” rule to be invoked on “defined” so the goal and goalType are only communicated once each time the campaign item is updated. |
| 1. **/ Phase 1 / Performance Goals**   The goalType can have a valid value of “views” |
| 1. **/ Phase 2 / Performance Goals**   The goalType can have a value of “insertions” or “views” |
| 1. **/ Phase 2 / Performance Goals**   Canoe expects the CIP consumer to identify an “insertion” as having occurred for a campaign item when an ad within the scope of the associated parent Bundle is designated for inclusion in a playlist (included in a placementResponse) by the CIP consumer. |
| 1. **/ Phase 2 / Performance Goals**   The CIP consumer is expected to report insertions in regularly scheduled FeedbackNotice messages. The FeedbackNotice includes a cumulative count of the insertions for an ad. The FeedbackNotice is generated at the time specified in the reportAsOf attribute of the StateFeedback element and then every X duration after that as indicated by the reportInterval attribute until the reportEndTime is reached. |
| 1. **/ Phase 1 / Performance Goals**   Canoe expects the CIP consumer to identify a “view” as having occurred for a campaign item when the first frame from an ad in the scope of the associated parent Bundle is played in NPT with a scale of 1. |
| 1. **/ Phase 1 / Performance Goals**   The CIP consumer is expected to report views in regularly scheduled FeedbackNotice messages. The FeedbackNotice includes a cumulative count of the views for an ad. The FeedbackNotice is generated at the time specified in the reportAsOf attribute of the StateFeedback element and then every X duration after that as indicated by the reportInterval attribute until the reportEndTime is reached. |
| 1. **/ Phase 1 / Performance Goals**   Canoe expects the CIP consumer will continue to execute to fill the goal until the goal is met, regardless of the goalType. |
| 1. **/ Phase 1 / FeedBack – Asset & Execution Readiness**   Canoe expects that Asset Availability information will be communicated by the MSO to Canoe through the StateFeedback element during the ContentMgmt window. Canoe’s published CIPs will specify a Statefeedback element within the AssetReference. Canoe expects the MSO to provide Availability feedback through the MediaState attribute of the StateFeedbackElement.  As an example of the published CIP;  <ContentMgmt beginRequired="2011-09-29T00:00:00.000-07:00" endRequired="2011-11-03T00:00:00.000-07:00"/>  <StateFeedback revision="1" reportAsOf=""2011-10-02T00:00:00-06:00"</StateFeedback>  The FeedbackNotice would contain the following related StateFeedback element  <StateFeedback revision="3" mediaState="unknown" executionState="notReady" recState="act"></StateFeedback> |
| 1. **/ Phase 1 /FeedBack – Statistics (views and insertions)**   Canoe expects The CIP consumer to maintain the performance counts for both insertions and views for the entire flight window of the campaign item. |
| 1. **/ Phase 1 / FeedBack – View Counts**   Canoe expects that View Count statistics will be communicated by the MSO to Canoe through the StateFeedback element. Canoe’s published CIPs will specify a Statefeedback element within the parent Bundle element. This StateFeedback element will contain a Statistics Element. As an example;  <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S" reportEndTime="2011-10-07T09:00:00-05:00">  <Statistics>  <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="View"/>  </Statistics> </StateFeedback>  Canoe expects the MSO to provide the View Count statistics for the Placement in a CIP Response based on the reportAsOf, reportinterval, and reportEndTime attributes.  The FeedbackNotice would contain the following related StateFeedback element  <StateFeedback revision="3" mediaState="unknown" executionState="notReady" recState="act">  <Statistics>  <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="Views">164</Statistic>  </Statistics> </StateFeedback> |
| 1. **/ Phase 1 / Feedback – Insertion Counts**   Canoe expects that Insertion Count statistics will be communicated by the MSO to Canoe during the Flight Window through the StateFeedback element. Canoe’s published CIPs will specify a Statefeedback element within the parent Bundle element. This StateFeedback element will contain a Statistics Element. As an example;  <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S" reportEndTime="2011-10-07T09:00:00-05:00">  <Statistics>  <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions"/>  </Statistics> </StateFeedback>  Canoe expects the MSO to provide the Insertion Count statistics for the Placement in a CIP Response based on the reportAsOf, reportinterval, and reportEndTime attributes.  The FeedbackNotice would contain the following related StateFeedback element  <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S" reportEndTime="2011-10-07T09:00:00-05:00" mediaState="unknown"  executionState="notReady" recState="act">  <Statistics>  <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions">218</Statistic>  </Statistics> </StateFeedback> |
| 1. **/ Phase 1 / Ad Rotation**   Canoe will communicate the rotationValue for each ad asset through a phase/rule set. This phase/rule set will be implemented within the child Bundle element. As an example;  <Phase epsid="0" peid="1111dc5AEd6ukwAf4hSEQQ" event="defined">  <ProcessRuleSelector ruleId="assignAssetRotationValue"  ruleRepository="urn:canoe:safi:rulesRepo:1.0">  <Arg name="rotationValue">50</Arg>  </ProcessRuleSelector> </Phase> |
| 1. **/ Phase 1 / Asset Duration**   Canoe will communicate the asset duration for each ad asset through an AssetMetadata element. |
| 1. **/ Phase 1 / Asset Duration**   The CIP Consumer must support assets of varying durations assigned to a single campaing item.  Canoe will communicate the asset duration for each ad asset through an AssetMetadata element. Canoe This phase/rule set will be implemented within the child Bundle element. As an example;  <Phase epsid="0" peid="1111dc5AEd6ukwAf4hSEQQ" event="defined">  <ProcessRuleSelector ruleId="assignAssetRotationValue"  ruleRepository="urn:canoe:safi:rulesRepo:1.0">  <Arg name="rotationValue">50</Arg>  </ProcessRuleSelector> </Phase> |
| 1. **/ Phase 1 / Ad Rotation**   Canoe has designated the “assignAssetRotationValue” to be invoked on “defined” so the rotation value is only communicated once each time the bundle revision is first recognized. |
| 1. **/ Phase 3 / Trick Mode Restriction**   TrickMode will be defined using the TrickModeExclusion element in the Placement. It defines type and viewing rate scale restrictions.  The following designates disabling of FastForward;  *<TrickModeRestriction>  <TrickModeExclusion type="trick" scale="0.0,100"/> </TrickModeRestriction>*  Complex TrickModeExclusionType Schema  *<xs:complexType name="TrickModeExclusionType">  <xs:annotation>  <xs:documentation>  <p>Defines one particular trick mode exclusion composed of a type and, for type="trick", a speed scaling factor or range of factors. The scale factor is the ratio of the play speed to normal play speed; forward speeds are positive, and backward speeds are negative.</p>  <p>Normal is forward at an implied scale of 1.0, and that combination is always explicit (i.e., normal play is never an implied part of a trick range).</p>  <p>Trick is a multiple of normal play speed, signed for forward or reverse. Trick 0.0 describes no play (i.e., pause).</p>  <p>Jump is a navigational notation used for transitions such as "digital rewind" or "go to chapter". It does not have a speed.</p>  <p>All is all trick modes, but excludes normal play.</p>  </xs:documentation>  </xs:annotation>  <xs:attribute name="type" type="cip:TrickModeType" use="required"/>  <xs:attribute name="scale" type="cip:SpeedScaleType"/> </xs:complexType>* |
| 1. **/ Phase 3 / Trick Mode Restriction**   In the absence of restrictions, all stream controls are permitted. |
| 1. **/ Phase 3 / Trick Mode Restriction**   When TrickModeExclusion restrictions are defined at the Placement the restrictions apply to all assets in the placement, unless the asset has a distinct restriction of its own. For a given asset, TrickModeExclusion restrictions defined within the AssetReference element have precedence over TrickModeExclusion restrictions defined at the Placement level. |
| 1. **/ Phase 3 / Trick Mode Restriction**   Support for control of additional Trick Modes should be provided in the future. |
| 1. **/ Phase 1 / Return Data Paths**   The CIP publisher shall communicate return data paths through an undefined out-of-band method. |
| 1. **/ Phase 1 / Campaign Update – recState change**   Canoe may change the recState of any element in a CIP. The CIP consumer MUST recognize and ingest these changes and adjust execution processing appropriately. |
| 1. **/ Phase 1 / Campaign Update – parent Bundle recState**   The CIP consumer must suspend execution of a campaign item if the recState of the parent Bundle is anything other than “act”. The CIP consumer must continue to process reporting (StateFeedback and SMS) when the state of the parent Bundle is anything other than “cls”. |
| 1. **/ Phase 1 / Campaign Update – child Bundle recState**   The CIP consumer must suspend execution of a child Bundle (descendent assets are ineligible) if the recState is anything other than “act”. The CIP consumer must continue to process reporting (StateFeedback and SMS) when the recState of the parent Bundle is anything other than “cls”. |
| 1. **/ Phase 1 / Campaign Update – Placement recState**   The CIP consumer must suspend execution of a Placement (descendent assets are ineligible) if the recState is anything other than “act”. The CIP consumer must continue to process reporting (StateFeedback and SMS) when the recState of the parent Bundle is anything other than “cls”. |
| 1. **/ Phase 1 / Campaign Update – Phase recState**   Canoe will maintain consistency between the Phase@recState and the recState of it’s containing element. The CIP consumer must suspend execution of a campaign item if the recState of a Phase within the parent Bundle is anything other than “act”. The CIP consumer must continue to process reporting (StateFeedback and SMS) when the state of the parent Bundle is anything other than “cls”.  As an example, if a parent Bundle had the following attributes:  <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a"  decisionOwner="NBC.COM" revision="3" recRevoked="false" **recState="act"**   revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00" gname="IBM\_Market\_Awareness"   guid="ntDbws5AEd6ukwAf4hSEQQ" epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ"   priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  And contained a Phase element with the following attributes:  <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" **recState="pse"**   revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  The CIP Consumer would suspend execution because the recState of the Bundle does not match the recState of the descendent Phase element. Reporting would continue because the recState of the parent Bundle is not “cls”.  As another example, if a child Bundle had the following attributes:  <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a"  decisionOwner="NBC.COM" revision="3" recRevoked="false" **recState="act"**   revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00" gname="IBM\_Market\_Awareness"   guid="ntDbws5AEd6ukwAf4hSEQQ" epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ"   priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  And contained a Phase element with the following attributes:  <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" **recState="pse"**   revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  The CIP Consumer would suspend execution only of the child Bundle and related Placement/AssetReference. Execution for any other child Bundles could continue. Reporting would continue because the recState of the child Bundle is not “cls”. |
| 1. **/ Phase 1 / Campaign Update – revision**   The revision attribute of each Bundle, Placement, and Phase will be changed with each UpdateNotice. The CIP consumer must persist the new revision to trigger appropriate Phase execution. For a new campaign item, revision=”1”. |
| 1. **/ Phase 1 / Campaign Update - Informative**   The following date/time attributes will be updated with each UpdateNotice. Each of these will be set to the Date/Time the UpdateNotice is created.   * Bundle@revDat * Placement@revDat * StateFeedback@reportAsOf * Phase@revDat |
| 1. **/ Phase 1 / Campaign Update – ContentMgmt - Informative**   The ContentMgmt@beginRequired attribute is populated by Canoe with the creation date/time of the current UpdateNotice. It is of the format “YYYY-MM-DDT00:00:00.000-00:00" |
| 1. **/ Phase 1 / Campaign Update – ContentMgmt - Informative**   The ContentMgmt@endRequired attribute is populated by Canoe with the Flight endDate/time. It is not modified in subsequent UpdateNotices unless the flight endDateTime is changed. It is of the format “YYYY-MM-DDT00:00:00.000-00:00" |
| 1. **/ Phase 1 / Campaign Update – campOrderOwner - Informative**   The campOrderOwner is always populated by Canoe with "canoe-ventures.com". |
| 1. **/ Phase 1 / Campaign Update**   Canoe will change the revision and revDat attributes of every Bundle, Placement and Phase each time an UpdateNotice is published. |
| 1. **/ Phase 1 / Campaign Update – Syncronous Validation**   The CIP consumer must validate each UpdateNotice message against the schema version specified in the message. The CIP consumer must respond with an UpdateResponse (see Annex C- UpdateResponse (OK and FAIL)) with a msgResult=”OK” if the UpdateNotice validates with the schema. The CIP consumer must respond with an UpdateResponse with a msgResult=”FAIL” if the UpdateNotice fails to validate with the schema. If msgResult=”FAIL” the Notes element contains the details of the validation failure(s).  The CIP Publisher must log all UpdateResponse messages as successes when msgResult=”OK”.  The CIP Publisher must log all UpdateResponse messages as failures when msgResult=”FAIL”.  The CIP Publisher will alert and take action to correct the validation errors prior to re-submitting a failed UpdateNotice. |
| 1. **/ Phase 1 / FeedbackNotice – Asyncronous Validation**   The CIP consumer validates each UpdateNotice based on the rules/constraints defined in the productFamily and productMember definitions (see Annex J - ProductFamily and ProductMember Definitions).  The “urn:canoe:safi:pfamily:vod:1.0” productFamily dictates the following validations:   1. Orders Element – Only 1 Orders element exists. 2. Orders Element – Only 1 Parent Bundle exists in the Orders element. 3. Parent Bundle – Establishes priority. All inner bundles and placements must replicate the priority of the Parent Bundle. 4. Parent Bundle - Must contain at least 1 CommonQualifierGroup element containing at least 1 Qualifier element containing at least 1 Term element. 5. Parent Bundle - CommonQualifierGroup@condition must = “or”. 6. Parent Bundle – All source, operation, xpath or namespace of a Term element must be properly formed. 7. Parent Bundle – Must contain at least 1 Flight element. 8. Parent Bundle – Must contain a StateFeedback element. This StateFeedback element must contain the Statistics element(s) which must contain a Statistic element to track insertion performance and a Statistic element to track view performance. 9. Parent Bundle – Must contain a Phase with a ProcessRuleSelector@ruleId="SetGoal" and a ProcessRuleSelector@ruleRepository="urn:canoe:safi:rulesRepo:1.0" 10. The inner Bundle productFamily must be “urn:canoe:safi:pfamily:single:1.0” or “urn:canoe:safi:pfamily:multiple:1.0”   The “urn:canoe:safi:pfamily:single:1.0” productFamily dictates the following validations:   1. The Parent Bundle productFamily Must be “urn:canoe:safi:pfamily:vod:1.0” 2. Inner Bundle – Must contain exactly 1 Placement element 3. Inner Bundle - The @priority of all inner bundles Must be the same as the parent bundle. 4. The CategoryExclusion element, when defined (it is Optional), will be contained in the inner Bundle. CategoryExclusion will not be defined in any other element. 5. A StateFeedback element must be defined in the inner Bundle. This StateFeedback element will be used to report syntax validation errors. 6. The Placement productMember must be "urn:canoe:safi:pmember:vodplacement:1.0"   The “urn:canoe:safi:pfamily:multiple:1.0” productFamily dictates the following validations:   1. The Parent Bundle productFamily Must be “urn:canoe:safi:pfamily:vod:1.0” 2. Inner Bundle – Must contain more than 1 Placement element 3. Inner Bundle - The @priority of all inner bundles Must be the same as the parent bundle. 4. Inner Bundle – Must contain a Phase with a ProcessRuleSelector@ruleId="assignAssetRotationValue" and a ProcessRuleSelector@ruleRepository="urn:canoe:safi:rulesRepo:1.0" 5. The CategoryExclusion element, when defined (it is Optional), will be contained in the inner Bundle. CategoryExclusion will not be defined in any other element. 6. A StateFeedback element must be defined in the inner Bundle. This StateFeedback element will be used to report syntax validation errors. 7. The Placement productMember must be "urn:canoe:safi:pmember:vodplacement:1.0".   The “urn:canoe:safi:pmember:vodplacement:1.0” productMember dictates the following validations:   1. The productFamily of the Containing bundle must be "urn:canoe:safi:pfamily:single:1.0" or "urn:canoe:safi:pfamily:multiple:1.0" 2. AssetReference – The AssetReference must contain an AssetMetadata element with xpath=”uriID” to define the assetID. The uriID value must be properly formed. 3. AssetReference – The AssetReference must contain at least 1 AssetMetadata element with one of the following xpath values to define the product category of the asset; IndustryGroup|MajorGroup|PCCDescription|ProductCategory. 4. StateFeedback.Statistics.Statistic - datamap must be recognized, provided and properly formed. 5. StateFeedback.Statistics.Statistic - variable is recognized and provided. 6. StateFeedback - @reportAsOf is provided and properly formed. |
| 1. **/ Phase 1 / FeedbackNotice – Asyncronous Validations**   When syntax validation errors are identified, the CIP publisher MUST insert a StateFeedback element in the containing Bundle, Placement, AssetReference, or Phase. This StateFeedback element MUST have a revision=”0” and contain a Notes element with descriptive details of the error(s). All errors within a Bundle, Placement, AssetReference, or Phase will be reported in the same StateFeedback element (within the same Notes element). Errors will not be repetitively reported in ancestor elements. |
| 1. **/ Phase 1 / Campaign Update – reportEndTime - Informative**   Canoe will publish the flightEndDateTime plus a configurable number of hours as the StateFeedback@reportEndTime. |
| 1. **/ Phase 1 / Registration**   The GetRegistrationRequest and SetRegistrationRequest message structures will not be supported by CIP consumer or CIP Publisher. Registration is a manual process. The CIP consumer and CIP publisher exchange endpoint information and make any required configurations manually.  The Comcast endpoint is [https://slapp.westchester.pa.bo.comcast.net:7021/cipUpdate/svCIPClientService](https://caas.canoe-ventures.com/Canoe/msoEndpoint/show/326)  The Canoe endpoint is [https://slapp.westchester.pa.bo.comcast.net:7021/cipUpdate/svCIPClientService](https://caas.canoe-ventures.com/Canoe/msoEndpoint/show/326) |
| 1. **/ Phase 1 / ReadRequest**   The ReadRequest message defines selection criteria for Bundle data to be returned in a ReadResponse. The ReadResponse returns zero or more top-level Bundles that satisfy all the selection criteria present in the ReadRequest. Omitted criteria do not restrict the returned set, except as specifically noted. All child components are included for each top-level Bundle selected. A specific campaign can be selected by guid or PEID+EPSID. revision and certain other attributes of the BundleSelectorGroup are not supported (revision, gname, campOrder, campOrderOwner) and must always be omitted. Only the current revision of a bundle is returned. revDateFrom and revDateTo select a window of revision dates for the Bundle. Only current revisions with revDat in the window will be returned.  As an example: <ReadRequest xmlns="urn:cablelabs:safi:xsd:cip:3.0"  xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" guid="ntDbws5AEd6ukwAf4hSEQQ" peid="sOHzM85AEd6ukwAf4hSEQQ" epsid="0" revDateFrom="2011-10-01T00:00:00-05:00" revDateTo="2011-10-05T00:00:00-05:00"> </ReadRequest> |
| 1. **/ Phase 1 / ReadResponse**   The ReadResponse message mimics the UpdateNotice message except ReadResponse replaces UpdateNotice as the root element of the message. A ReadResponse can contain 1 or many top level Bundles. The ReadResponse element also contains a msgResult attribute which should have the value “OK”. |
| 1. **/ Phase 1 / InvalidRequest**   An InvaldRequest is sent in response to an errored Read Request. If a ReadRequest from the Cip consumer returns no results, the CIP publisher MUST publish an InvalidRequest message.  As an example:  <!-- Example InvalidRequest SaFI Interface Specifications v2.8 --> <InvalidRequest xmlns="urn:cablelabs:safi:xsd:cip:3.0"  xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >  <!-- reason must be 1 of "From-ToNotPaired|IdOwnerNotPaired|NoSelection|NoneFound|private:.+" -->  <Reason>NoSelection</Reason>  <Values guid="ntDbws5AEd6ukwAf4hSEQQ" peid="sOHzM85AEd6ukwAf4hSEQQ"  epsid="0" revDateFrom="2011-10-01T00:00:00-05:00" revDateTo="2011-10-05T00:00:00-05:00"/> </InvalidRequest> |
| 1. **/ Phase 1 / Resolution Compatibility**   The CIP consumer must insure that the resolution of placed ads is compatible with the resolution of the entertainment asset. The CIP consumer can accomplish this through referencing assets of appropriate resolution, by upconverting as required, as when necessary filtering from placement those CI’s that only contain assets of resolution incompatible with the entertainment asset. The CIP consumer should insure that opportunities do not remain unfilled when qualifying campaigns with resolution compatible ad assets exist. |
| 1. **/ N/A / Detail Viewership Requirements**   NOTE: A full technical recommendation to support the business requirements for detailed SMS data is not included within this document. Only the need for specific client identifiers is addressed below. Canoe will collaborate with the working group to determine the best overall solution. |
| 1. **/ Phase 1 / Detail Viewership Requirements**   To enable the creation of the household-based metrics, an additional element must be added to *ServiceMeasurement*. This element, *Client*, shall be a container for unique household and client identifiers. The *Client* element is directly based off of the SCTE 130-3 Client element, with only the *TerminalAddress* represented. It may be possible to combine this new element with the existing *GeoCode* element, creating an overall container for all attributes meant to identify the location of the measurement data. The proposed schema is below:  *<xsd:complexType name="ClientType">*  *<xsd:sequence>*  *<xsd:element ref="TerminalAddress" minOccurs="0" maxOccurs="unbounded" />*  *</xsd:sequence>*  *</xsd:complexType>*  *<xsd:element name="Client" type="ClientType"/>*  *<xsd:complexType name="TerminalAddressType"><xsd:attribute name="type" type="sms:AddressType" use="required"/>*  *</xsd:complexType>*  *<xs:simpleType name=”AddressType”>*  *<xs:restriction base=”xs:string”>*  *<xs:enumeration value=”householdId”/>*  *<xs:enumeration value=”clientId”/>*  *</xs:restriction>*  *</xs:simpleType>*  *<xsd:element name="TerminalAddress" type="TerminalAddressType"/>* |
| 1. **/ N/A / Summary Viewership Data**   NOTE: A full technical recommendation to support the communication of summary SMS data is not included within this document. Canoe will collaborate with the working group to determine the best overall solution.  The CIP consumer must deliver two distinct data sets to the CIP publisher.   1. Campaign Item centric data set – used to produce the Summary Report, Campaign Summary reports, and Campaign Proof of Performance reports. 2. Ad centric data set – used to produce the Ad Viewership Summary reports and a summary version of the Ad Viewership logs |
| 1. **/ N/A / Summary Viewership Data – Campaign Centric Data Set**   The CIP consumer must deliver a Campaign Centric Data Set to the CIP publisher. This data set contains the following Data elements. The Campaign Centric Data Set includes all campaigns that are not in a reqState of “cls”. The Data is cumulative. The dataset is time stamped by the StartDateTime and the EndDateTime.   1. Campaign Number – which is defined by the campOrder attribute of the Bundle 2. Campaign Item ID - which is defined by the parent Bundle PEID 3. Ad Asset ID – which is defined through the uriID Asset Metadata element within the AssetReferences at decision time 4. Program Identifier – which is defined at decision time 5. Network – which is defined at decision time 6. OpportunityType - which is defined at decision time 7. poGroupIndex - which is defined at decision time 8. position - which is defined at decision time 9. StartDateTime of the data set 10. EndDateTime of the data set 11. Insertion Count 12. Views Count 13. Play Time 14. Run Time 15. Unique HH’s   See Annex J - Reporting - Definitions and Annex K - Reporting – Sample UpdateNotice Messages for additional information.  The Campaign Centric Data Set is sorted as follows   1. Campaign Number - Ascending 2. Campaign Item ID - Ascending 3. Ad Asset ID – Ascending 4. Program Identifier – Ascending 5. Network – Ascending 6. OpportunityType – pre-roll, mid-roll, post-roll, interstitial, slate 7. poGroupIndex – Ascending 8. position - Ascending   The Campaign Centric Data Set is delivered with break totals at the following levels.   1. Campaign Item ID - Ascending 2. Ad Asset ID – Ascending 3. Program Identifier – Ascending 4. Network – Ascending 5. OpportunityType – pre-roll, mid-roll, post-roll, interstitial, slate 6. poGroupIndex – Ascending 7. position - Ascending   See Annex N - Reporting – Sample Campaign Centric Data Set – evaluation of this sample and the samples in Annex M - Reporting – Sample UpdateNotice Messages clearly demonstrates the transition of campaign item information into the reporting dataset. |
| 1. **/ N/A / Summary Viewership Data – Ad Centric Data Set**   The CIP consumer must deliver an Ad Centric Data Set to the CIP publisher. The Ad Centric Data Set includes all assets associates with any campaign that is not in a reqState of “cls”. The Data is cumulative. The dataset is time stamped by the StartDateTime and the EndDateTime. This data set contains the following Data elements.   1. Ad Asset ID – which is defined through the uriID Asset Metadata element within the AssetReferences at decision time 2. Program Identifier – which is defined at decision time 3. Network – which is defined at decision time 4. OpportunityType - which is defined at decision time 5. poGroupIndex - which is defined at decision time 6. position - which is defined at decision time 7. StartDateTime of the data set 8. EndDateTime of the data set 9. Insertion Count 10. Views Count 11. Play Time 12. Run Time 13. Unique HH’s   See Annex L - Reporting - Definitions and Annex M - Reporting – Sample UpdateNotice Messages for additional information.  The Ad Centric Data Set is sorted as follows   1. Ad Asset ID – Ascending 2. Program Identifier – Ascending 3. Network – Ascending 4. OpportunityType – pre-roll, mid-roll, post-roll, interstitial, slate 5. poGroupIndex – Ascending 6. position - Ascending   The Ad Centric Data Set is delivered with break totals at the following levels.   1. Ad Asset ID – Ascending 2. Program Identifier – Ascending 3. Network – Ascending 4. OpportunityType – pre-roll, mid-roll, post-roll, interstitial, slate 5. poGroupIndex – Ascending 6. position - Ascending   See Annex M - Reporting – Sample Ad Centric Data Set – evaluation of this sample and the samples in Annex K - Reporting – Sample UpdateNotice Messages clearly demonstrates the transition of campaign item information into the reporting dataset. |
| 1. **/ N/A / Summary Viewership Data – Campaign Centric Data Set**   The CIP consumer must deliver both Campaign and Ad Centric Data Sets to the CIP publisher every X configurable minutes. Initially X = 15. |

1. Metadata Operator Matrix



1. UpdateNotice Sample of single asset productFamily

<!-- Example parent productFamily="vod", inner productFamily="Single" - Canoe SaFI Interface  
 Specifications v2.9 -->  
<UpdateNotice xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <CampInfoPkg>  
 <Version minSchemaVersion="2" minSchemaMinorVersion="1"/>  
 <MsoOrder name="<MSO\_NAME>">  
 <!-- this order is for all syscodes for the MSO -->  
 <Orders>  
 <!-- parent Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a"  
 decisionOwner="NBC.COM" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00" gname="IBM\_Market\_Awareness"   
 guid="ntDbws5AEd6ukwAf4hSEQQ" epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ"   
 priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- All Qualifiers appear in the parent Bundle -->  
 <CommonQualifierGroup condition="or">  
 <Qualifier>  
 <!-- Target preroll opportunities on the Office -->  
 <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">The Office</Term>  
 <Term source="ent" operation="eq" xpath="uriID" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">NBC.COM</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target midroll opportunities when they are the first opportunity -->  
 <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  
 <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">1</Term>  
 <Term source="ent" operation="eq" xpath="uriID" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">NBC.COM</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target preroll opportunities on the Office -->  
 <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">The Office</Term>  
 <Term source="ent" operation="eq" xpath="uriID" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">BRAVO.COM</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target midroll opportunities when they are the first opportunity -->  
 <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  
 <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">1</Term>  
 <Term source="ent" operation="eq" xpath="uriID" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">BRAVO.COM</Term>  
 </Qualifier>  
 </CommonQualifierGroup>  
 <!-- child Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_1" guid="ntDbws5AEd6ukwAf4hSEQR" epsid="0" peid="sOHzM85AEd6ukwAf4hSEQL"   
 priority="3" productFamily="urn:canoe:safi:pfamily:single:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement - creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00"  
 creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="+cYTZM5AEd6ukwAf4hSEQQ" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="/ld7ekaAEd6ukwAf4hSEQQ" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">CSMK1600002900000000</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS, OFF. EQUIP. &amp; STATIONERY</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS &amp; DATA PROCESSING EQUIPMENT</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS,COMPONENTS &amp; ACCESS</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">OFFICE AUTOMATION SYS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-29T00:00:00.000-05:00" endRequired="2011-10-04T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Phase 3 - Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </Placement>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the opportunity -->  
 <CategoryExclusion scope="opportunity"/>  
 </Bundle>  
 <!-- Set window 10/1 - 10/31, M-F, from 9:00 am to 1:00 pm -->  
 <!-- Flight is established in the parent Bundle -->  
 <Flight startDateTime="2011-10-01T09:00:00-05:00" endDateTime="2011-10-04T09:00:00-05:00" startTime="09:00:00" duration="PT04H00M00S"  
 sat="false" sun="false"/>  
 <!-- Statefeedback mediaState and executionState and well as statistics are summarized in the  
 parent Bundle -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S"  
 reportEndTime="2011-10-07T09:00:00-05:00">  
 <Statistics>  
 <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions"/>  
 <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="Views"/>  
 </Statistics>  
 </StateFeedback>  
 <!-- When bundle is defined, goal is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetGoal" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="goal">100000</Arg>  
 <Arg variable="goalType">views</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 </Bundle>  
 </Orders>  
 </MsoOrder>  
 </CampInfoPkg>  
</UpdateNotice>

1. UpdateResponse (OK and FAIL)

<!-- Example UpdateResponse for "OK" Canoe SaFI Interface Specifications v2.8 -->  
<UpdateResponse xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" msgResult="OK">  
 <Notes></Notes>  
</UpdateResponse>

<!-- Example UpdateResponse for "FAIL" Canoe SaFI Interface Specifications v2.8 -->  
<UpdateResponse xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" msgResult="FAIL">  
 <Notes> Parent Bundle Malformed; priority missing; @reportAsOf missing; </Notes>  
</UpdateResponse>

1. FeedbackNotice Sample of single asset productFamily

<!-- Example parent productFamily="vod", inner productFamily="Single" - Canoe SaFI Interface  
 Specifications v2.9 -->  
<FeedbackNotice xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <CampInfoPkg>  
 <Version minSchemaVersion="2" minSchemaMinorVersion="1"/>  
 <MsoOrder name="<MSO\_NAME>">  
 <!-- this order is for all syscodes for the MSO -->  
 <Orders>  
 <!-- parent Bundle attributes mimic the current revision -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a"  
 decisionOwner="NBC.COM" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00" gname="IBM\_Market\_Awareness"   
 guid="ntDbws5AEd6ukwAf4hSEQQ" epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ"   
 priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Qualifiers are removed as they are not required -->  
 <!-- inner Bundle attributes mimic the current revision -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_1" guid="ntDbws5AEd6ukwAf4hSEQR" epsid="0" peid="sOHzM85AEd6ukwAf4hSEQL"   
 priority="3" productFamily="urn:canoe:safi:pfamily:single:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement attributes mimic the current revision -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00"  
 creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="+cYTZM5AEd6ukwAf4hSEQQ" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <!-- AssetReference attributes mimic the current revision -->  
 <AssetReference epsid="0" peid="/ld7ekaAEd6ukwAf4hSEQQ" apply="valAndPlace" assetType="media">  
 <!-- AssetMetadata is removed as it is not required -->  
 <!-- ContentMgmt element is removed as it is not required -->  
 <!-- Start the Asset Readiness tracking when Content@beginRequired -->  
 <!-- The CIP consumer includes recState (pnd, act, pse, wnd, cls), mediaState (unknown|processing|ready|private:.+)   
 and executionState (notReady|processing|ready|private:.+) in the StateFeedback element, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </AssetReference>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </Placement>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 <!-- CategoryExclusion is removed as it is not required -->  
 </Bundle>  
 <!-- Flight is removed as it is not required -->  
 <!-- The CIP consumer includes recState (pnd, act, pse, wnd, cls), mediaState (unknown|processing|ready|private:.+)   
 and executionState (notReady|processing|ready|private:.+) in the StateFeedback element; revision  
 and reportAsOf are mimiced from the current revision; reportInterval and reportEndTime are  
 ommitted as they are not needed. -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S"  
 reportEndTime="2011-10-07T09:00:00-05:00" mediaState="ready"  
 executionState="ready" recState="act">  
 <Statistics>  
 <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions">218</Statistic>  
 <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="Views">164</Statistic>  
 </Statistics>  
 </StateFeedback>  
 <!-- All Phase elements are included to allow reporting through StateFeedback element -->  
 <!-- When bundle is defined, goal is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetGoal" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="goal">100000</Arg>  
 <Arg variable="goalType">views</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 <Phase epsid="1" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetRetDataPath" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="PriUrl">http://ara.ws-cnv.com:8080/am</Arg>  
 <Arg variable="ApSecUrl">https://cmgjb01.corp.acxiom.net/caas.canoe-ventures/safi/i1.1/iaf </Arg>  
 <Arg variable="SmSecUrl">http://10.13.20.51:8080/safi/i1.1/sms</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 </Bundle>  
 </Orders>  
 </MsoOrder>  
 </CampInfoPkg>  
</FeedbackNotice>

1. FeedbackResponse (OK and FAIL)

<!-- Example FeedbackResponse for "OK" Canoe SaFI Interface Specifications v2.8 -->  
<UpdateResponse xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" msgResult="OK">  
 <Notes></Notes>  
</UpdateResponse>

<!-- Example FeedbackResponse for "FAIL" Canoe SaFI Interface Specifications v2.8 -->  
<UpdateResponse xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" msgResult="FAIL">  
 <Notes>document invalid </Notes>  
</UpdateResponse>

1. UpdateNotice Sample of multiple asset productFamily

<!-- Example parent productFamily="vod", inner productFamily="multiple" - Canoe SaFI Interface  
 Specifications v2.9 -->  
<UpdateNotice xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <CampInfoPkg>  
 <Version minSchemaVersion="2" minSchemaMinorVersion="1"/>  
 <MsoOrder name="<MSO\_NAME>">  
 <!-- this order is for all syscodes for the MSO -->  
 <Orders>  
 <!-- parent Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a"  
 decisionOwner="NBC.COM" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00" gname="IBM\_Market\_Awareness"   
 guid="ntDbws5AEd6ukwAf4hSEQQ" epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ"   
 priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- All Qualifiers appear in the parent Bundle -->  
 <CommonQualifierGroup condition="or">  
 <Qualifier>  
 <!-- Target preroll opportunities on the Office -->  
 <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">The Office</Term>  
 <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">NBC.COM</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target midroll opportunities when they are the first opportunity -->  
 <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  
 <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">1</Term>  
 <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">NBC.COM</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target preroll opportunities on the Office -->  
 <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">preroll</Term>  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">The Office</Term>  
 <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">BRAVO.COM</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target midroll opportunities when they are the first opportunity -->  
 <Term source="ent" operation="eq" xpath="typeOfPlacementOpportunity" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">midroll</Term>  
 <Term source="ent" operation="eq" xpath="order" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/placementopportunity/1">1</Term>  
 <Term source="ent" operation="eq" xpath="providerid" namespace="cablelabs.com/namespaces/metadata/xsd/core/1">BRAVO.COM</Term>  
 </Qualifier>  
 </CommonQualifierGroup>  
 <!-- child1 Bundle (1 of 4 child Bundles) - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->   
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_1" guid="ntDbws5AEd6ukwAf4hSEQR" epsid="0"   
 peid="sOHzM85AEd6ukwAf4hSEQR" priority="3" productFamily="urn:canoe:safi:pfamily:multiple:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"  
 epsid="0" peid="+cYTZM5AEd6ukwAf4hSEQR" priority="3"  
 productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="group">  
 <AssetReference epsid="0" peid="/ld7ekaAEd6ukwAf4hSEQR" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">CSMK1600002900000000</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS, OFF. EQUIP. &amp; STATIONERY</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS &amp; DATA PROCESSING EQUIPMENT</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS,COMPONENTS &amp; ACCESS</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">OFFICE AUTOMATION SYS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-29T00:00:00.000-05:00" endRequired="2011-10-04T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </Placement>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 <!-- When bundle is defined, rotation of asset is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQR" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="assignAssetRotationValue" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="rotationValue">20</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the opportunity -->  
 <CategoryExclusion scope="opportunity"/>  
 </Bundle>  
 <!-- child2 Bundle (2 of 4 child Bundles) - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->   
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_1" guid="ntDbws5AEd6ukwAf4hSEQB" epsid="0"  
 peid="sOHzM85AEd6ukwAf4hSEQB" priority="3" productFamily="urn:canoe:safi:pfamily:multiple:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="+cYTZM5AEd6ukwAf4hSEQB" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="/h0Rdc5AEd6ukwAf4hSEQB" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">CSMK1600002800000000</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS, OFF. EQUIP. &amp; STATIONERY</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS &amp; DATA PROCESSING EQUIPMENT</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS,COMPONENTS &amp; ACCESS</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">OFFICE AUTOMATION SYS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-29T00:00:00.000-05:00" endRequired="2011-10-04T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </Placement>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 <!-- When bundle is defined, rotation of asset is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQB" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="assignAssetRotationValue" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="rotationValue">40</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the session -->  
 <CategoryExclusion scope="session"/>  
 </Bundle>  
 <!-- child3 Bundle (3 of 4 child Bundles) - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->   
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_1" guid="ntDbws5AEd6ukwAf4hSEQD" epsid="0"  
 peid="sOHzM85AEd6ukwAf4hSEQD" priority="3" productFamily="urn:canoe:safi:pfamily:multiple:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="+cYTZM5AEd6ukwAf4hSEQD" priority="3"  
 productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="session">  
 <AssetReference epsid="0" peid="/ld7ekaAEd6ukwAf4hSEQD" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">CSMK1600002800000000</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS, OFF. EQUIP. &amp; STATIONERY</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS &amp; DATA PROCESSING EQUIPMENT</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS,COMPONENTS &amp; ACCESS</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">OFFICE AUTOMATION SYS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-29T00:00:00.000-05:00" endRequired="2011-10-04T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </Placement>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 <!-- When bundle is defined, rotation of asset is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQD" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="assignAssetRotationValue" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="rotationValue">80</Arg>  
 </ProcessRuleSelector>  
 </Phase>  
 <!-- Prevent insertion of any ad from this Placement if an ad of the same category already exists in the group -->  
 <CategoryExclusion scope="group"/>  
 </Bundle>  
 <!-- child4 Bundle (4 of 4 child Bundles) - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->   
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_2" guid="ntDbws5AEd6ukwAf4hSEQF" epsid="0" priority="3"   
 peid="sOHzM85AEd6ukwAf4hSEQF" productFamily="urn:canoe:safi:pfamily:multiple:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="#fuiwdcAEd6ukwAf4hSEQD" priority="3"  
 productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="mapxjz5AEd6ukwAf4hSEQF" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">CSMK1600002800000000</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS, OFF. EQUIP. &amp; STATIONERY</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS &amp; DATA PROCESSING EQUIPMENT</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">COMPUTERS,COMPONENTS &amp; ACCESS</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">OFFICE AUTOMATION SYS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-29T00:00:00.000-05:00" endRequired="2011-10-04T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </Placement>  
 <!-- When bundle is defined, rotation of asset is communicated -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQF" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="assignAssetRotationValue" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="rotationValue">50</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 <!-- Prevent insertion of any ad from this Placement if an ad of the same category already exists in the session -->  
 <CategoryExclusion scope="session"/>  
 </Bundle>  
 <!-- Set window 10/1 - 10/04, M-F, from 9:00 am to 1:00 pm -->  
 <!-- Flight is established in the parent Bundle -->  
 <Flight startDateTime="2011-10-01T09:00:00-05:00" endDateTime="2011-10-04T09:00:00-05:00" startTime="09:00:00" duration="PT04H00M00S"  
 sat="false" sun="false"/>  
 <!-- Statefeedback mediaState and executionState and well as statistics are summarized in the  
 parent Bundle -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S"  
 reportEndTime="2011-10-07T09:00:00-05:00">  
 <Statistics>  
 <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions"/>  
 <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="Views"/>  
 </Statistics>  
 </StateFeedback>  
 <!-- When bundle is defined, goal is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetGoal"  
 ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="goal">100000</Arg>  
 <Arg variable="goalType">views</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>   
 </ProcessRuleSelector>  
 </Phase>  
 <!-- When bundle is defined, return data paths are communicated -->  
 <Phase epsid="1" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetRetDataPath" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="PriUrl">http://ara.ws-cnv.com:8080/am</Arg>  
 <Arg variable="ApSecUrl">https://cmgjb01.corp.acxiom.net/caas.canoe-ventures/safi/i1.1/iaf </Arg>  
 <Arg variable="SmSecUrl">http://10.13.20.51:8080/safi/i1.1/sms</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 </Bundle>  
 </Orders>  
 </MsoOrder>  
 </CampInfoPkg>  
</UpdateNotice>

1. FeedbackNotice Sample of multiple asset productfamily

<!-- Example parent productFamily="vod", inner productFamily="multiple" - Canoe SaFI Interface  
 Specifications v2.9 -->  
<FeedbackNotice xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <CampInfoPkg>  
 <Version minSchemaVersion="2" minSchemaMinorVersion="1"/>  
 <MsoOrder name="<MSO\_NAME>">  
 <!-- this order is for all syscodes for the MSO -->  
 <Orders>  
 <!-- parent Bundle attributes mimic the current revision -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a"  
 decisionOwner="NBC.COM" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00" gname="IBM\_Market\_Awareness"   
 guid="ntDbws5AEd6ukwAf4hSEQQ" epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ"   
 priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Qualifiers are removed as they are not required -->  
 <!-- inner Bundle attributes mimic the current revision -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_1" guid="ntDbws5AEd6ukwAf4hSEQR" epsid="0"   
 peid="sOHzM85AEd6ukwAf4hSEQR" priority="3" productFamily="urn:canoe:safi:pfamily:multiple:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement attributes mimic the current revision -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"  
 epsid="0" peid="+cYTZM5AEd6ukwAf4hSEQR" priority="3"  
 productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="group">  
 <!-- AssetReference attributes mimic the current revision -->  
 <AssetReference epsid="0" peid="/ld7ekaAEd6ukwAf4hSEQR" apply="valAndPlace" assetType="media">  
 <!-- AssetMetadata is removed as it is not required -->  
 <!-- ContentMgmt element is removed as it is not required -->  
 <!-- Start the Asset Readiness tracking when Content@beginRequired -->  
 <!-- The CIP consumer includes recState (pnd, act, pse, wnd, cls), mediaState (unknown|processing|ready|private:.+)   
 and executionState (notReady|processing|ready|private:.+) in the StateFeedback element, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </AssetReference>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 <!-- TrickModeRestriction is removed as it is not required -->  
 </Placement>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 <!-- AssignAssetRotationValue Phase is included to report syntax erors, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQR" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="assignAssetRotationValue"  
 ruleRepository="urn:canoe:safi:rulesRepository:1.0">  
 <Arg variable="rotationValue">50</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"  
 executionState="notReady" recState="act">  
 <Notes>ruleRepository urn:canoe:safi:rulesRepository:1.0 unknown</Notes>  
 </StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 <!-- CategoryExclusion is removed as it is not required -->  
 </Bundle>  
 <!-- inner Bundle attributes mimic the current revision -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_1" guid="ntDbws5AEd6ukwAf4hSEQB" epsid="0"  
 peid="sOHzM85AEd6ukwAf4hSEQB" priority="3" productFamily="urn:canoe:safi:pfamily:multiple:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement attributes mimic the current revision -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="+cYTZM5AEd6ukwAf4hSEQB" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="/h0Rdc5AEd6ukwAf4hSEQB" apply="valAndPlace" assetType="media">  
 <!-- AssetMetadata is removed as it is not required -->  
 <!-- ContentMgmt element is removed as it is not required -->  
 <!-- Start the Asset Readiness tracking when Content@beginRequired -->  
 <!-- The CIP consumer includes recState (pnd, act, pse, wnd, cls), mediaState (unknown|processing|ready|private:.+)   
 and executionState (notReady|processing|ready|private:.+) in the StateFeedback element, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </AssetReference>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 <!-- AssignAssetRotationValue Phase is included to report syntax erors, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQB" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="assignAssetRotationValue"  
 ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="rotationValue">50</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 <!-- TrickModeRestriction is removed as it is not required -->  
 </Placement>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 <!-- AssignAssetRotationValue Phase is removed as it is not needed -->  
 <!-- CategoryExclusion is removed as it is not required -->  
 </Bundle>  
 <!-- Placement attributes mimic the current revision -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_1" guid="ntDbws5AEd6ukwAf4hSEQD" epsid="0"  
 peid="sOHzM85AEd6ukwAf4hSEQD" priority="3" productFamily="urn:canoe:safi:pfamily:multiple:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="+cYTZM5AEd6ukwAf4hSEQD" priority="3"  
 productMember="urn:canoe:safi:pmember:vodsession:1.0" separationConstraint="session">  
 <AssetReference epsid="0" peid="/ld7ekaAEd6ukwAf4hSEQD" apply="valAndPlace" assetType="media">  
 <!-- AssetMetadata is removed as it is not required -->  
 <!-- ContentMgmt element is removed as it is not required -->  
 <!-- Start the Asset Readiness tracking when Content@beginRequired -->  
 <!-- The CIP consumer includes recState (pnd, act, pse, wnd, cls), mediaState (unknown|processing|ready|private:.+)   
 and executionState (notReady|processing|ready|private:.+) in the StateFeedback element, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </AssetReference>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 <Notes>unknown productMember "urn:canoe:safi:pmember:vodsession:1.0"</Notes>  
 </StateFeedback>  
 <!-- AssignAssetRotationValue Phase is included to report syntax erors, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQD" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="assignAssetRotationValue"  
 ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="rotationValue">50</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 <!-- TrickModeRestriction is removed as it is not required -->  
 </Placement>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 <Notes></Notes>  
 </StateFeedback>  
 <!-- AssignAssetRotationValue Phase is removed as it is not needed -->  
 <!-- CategoryExclusion is removed as it is not required -->  
 </Bundle>  
 <!-- Placement attributes mimic the current revision -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="NBC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 gname="IBM\_Market\_Awareness\_2" guid="ntDbws5AEd6ukwAf4hSEQF" epsid="0" priority="3"   
 peid="sOHzM85AEd6ukwAf4hSEQF" productFamily="urn:canoe:safi:pfamily:multiple:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="#fuiwdcAEd6ukwAf4hSEQD" priority="3"  
 productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="mapxjz5AEd6ukwAf4hSEQF" apply="valAndPlace" assetType="media">  
 <!-- AssetMetadata is removed as it is not required -->  
 <!-- ContentMgmt element is removed as it is not required -->  
 <!-- Start the Asset Readiness tracking when Content@beginRequired -->  
 <!-- The CIP consumer includes recState (pnd, act, pse, wnd, cls), mediaState (unknown|processing|ready|private:.+)   
 and executionState (notReady|processing|ready|private:.+) in the StateFeedback element, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </AssetReference>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 <!-- TrickModeRestriction is removed as it is not required -->  
 </Placement>  
 <!-- StateFeedback used to report syntax validation errors, if present the Notes element  
 specifies the details of any syntax validation errors -->   
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" mediaState="unknown"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 <!-- AssignAssetRotationValue Phase is included to report syntax erors, if present the Notes element  
 specifies the details of any syntax validation errors -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQF" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="assignAssetRotationValue"  
 ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="rotationValue">50</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>   
 <!-- AssignAssetRotationValue Phase is removed as it is not needed -->  
 <!-- CategoryExclusion is removed as it is not required -->  
 </Bundle>  
 <!-- Flight is removed as it is not required -->  
 <!-- The CIP consumer includes recState (pnd, act, pse, wnd, cls), mediaState (unknown|processing|ready|private:.+)   
 and executionState (notReady|processing|ready|private:.+) in the StateFeedback element; revision  
 and reportAsOf are mimiced from the current revision; reportInterval and reportEndTime are  
 ommitted as they are not needed. if present the Notes element specifies the details of any syntax validation errors-->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S"  
 reportEndTime="2011-10-07T09:00:00-05:00" mediaState="ready"  
 executionState="ready" recState="act">  
 <Statistics>  
 <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions">1587</Statistic>  
 <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="Views">645</Statistic>  
 </Statistics>  
 </StateFeedback>  
 <!-- All Phase elements are included to allow reporting through StateFeedback element -->  
 <!-- When bundle is defined, goal is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetGoal" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="goal">100000</Arg>  
 <Arg variable="goalType">views</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"  
 executionState="notReady" recState="act">  
 </StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 </Bundle>  
 </Orders>  
 </MsoOrder>  
 </CampInfoPkg>  
</FeedbackNotice>

1. ProductFamily and ProductMember Definitions



1. Position Qualifier Sample



1. Reporting - Definitions

### Terms and Definitions

|  |  |
| --- | --- |
| Data Point | Definition / Description |
| Scheduled Insertion | The process of placing a specific VOD Ad Asset into a playlist at VOD Session Creation. *Note: In some cases, this can be one and the same as an Ad Decision.* |
| Ad Decision | See Scheduled Insertion |
| View | A View is defined as the first frame of the asset being presented at NPT scale of 1 in the forward direction. If the Viewer leaves the asset entirely (either rewinding before it starts or by routinely beginning the next asset in the playlist) and returns to the first frame presented at NPT scale of 1 in the forward direction, this would count as an additional view. *Note: Can also be referred to as an Impression.* |
| Impression | See View |
| Network | The network of the entertainment asset into which the ad asset was inserted. Network can refer to any type of Content Provider supply VOD content, including Broadcast, Cable and Non-Linear only organizations. *Note: Network refers the branded content network offering (i.e. Bravo, NBC, ExerciseTV, AMC, TBS, etc), and not the parent programming network group (i.e. NBCU, AMC Networks, Turner, etc).* |
| Program Identifier | The CableLabs Provider ID/Asset ID to identify the entertainment asset the ad is being inserted into. *Note: Can also be referred to as an Entertainment Title and/or Main Content Asset.* |
| Play time | Play Time measures the amount of time an asset was viewed in Play Mode (NPT scale of 1 in forward direction) regardless of any trick modes that may have been invoked in between. Play Time is a subset of Run Time; Play Time does not include session set-up/tear-down time or any time relating to FF, Rewind, or Pause (which may all be included in Run Time). Play time may exceed ad duration. Play time is measured for an ad asset each time an ad is entered from program content (the same ad may be entered multiple times during the customer’s viewing of a VOD entertainment asset). |
| Run time | Run Time is measured from the time a view was initiated until the time the stream was exited regardless of any trick modes that may have been invoked in between. Run time may be described as the “wall clock” time that an ad asset was engaged by a customer. Run time is measured for an ad asset each time an ad is entered from program content (the same ad may be entered multiple times during the customer’s viewing of a VOD entertainment asset). |
| Household | A Unique Viewing Subscriber’s Household. A Household can have one or multiple viewing devices (i.e. Set top Box) associated with it. |
| Break | Break is defined as a grouping of placements in one of the following categories:   1. Pre-Roll – group of placements scheduled previous to entertainment content beginning 2. Mid-Roll – group of placements scheduled within the entertainment content. A VOD Session could have multiple Mid-rolls (e.g. Mid-roll 1, Mid-roll 2, etc.) 3. Post-Roll – group of placements scheduled following the end of the entertainment content   Note: For technical audiences, a Break is defined as 1 or multiple Placement Opportunities sharing the same PO Group Index. |
| Placement | A distinct location where non-entertainment content (i.e. advertising, marketing) can be placed.  Note: Placements are not one to one with Placement Opportunities. If a Break is made up of 2 Placement Opportunities, which get satisfied with 4 placements (regardless of inventory ownership), there are 4 placements included in the break. |
| Placement Position | Position is the placement sequence number relative to and within a Break. Note: Positions are not one to one with Placement Opportunities. If a Pre-roll is made up of 2 Placement Opportunities, which get staified with 4 placements (regardless of inventory ownership), the 4 placement positions included in the break are numbered sequentially as 1, 2, 3, 4. |
| Order | A Canoe Order represents the list of Advanced Advertising Products and Services a Programmer enlists the Canoe Stewardship for use. An order may contain one or multiple campaigns. |
| Campaign | A Canoe Campaign is the instruction set container for executing a Canoe Order. A Campaign may contain one or multiple Campaign Items. |
| Campaign Item | The set of Scheduling / Targeting logic surrounding one or multiple Ad Assets for a particular Campaign. A Campaign may have one or multiple Campaign Items.  Note: Can also be referred to as a Contract Line Item and/or Campaign Line Item. |

### Data Point Configurations

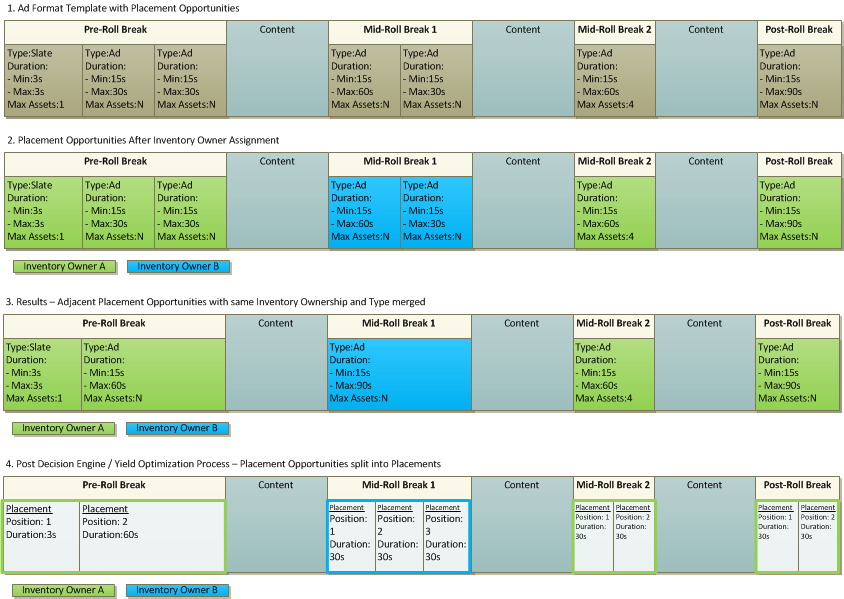
All data summaries in the following table must be configured in the following manner: per Campaign, per Campaign Item, per Ad Asset ID, per Program Identifier, per Network, per Opportunity Type, per Break, per Position.

|  |  |
| --- | --- |
| Data Point | Configuration(s) |
| Total Scheduled Insertions | The total count of Ad Scheduled Insertions being made during the specified reporting time period. |
| Total Views | The total count of Ad Views being made during the specified reporting time period. |
| Total Play time | The Total Ad Play time during the specified reporting time period. |
| Total Run time | The Total Ad Run time during the specified reporting time period. |
| Total Households (Campaign Centric) | The Total number of Households per Campaign that have one or more views recorded during the specified reporting time period. |
| Break / Position Pairings | The total count of Ad Scheduled Insertions per each Break location / Position combination during the specified reporting time period.  The total count of Ad Views per each Break location / Position combination during the specified reporting time period. |

The below table contains an additional data summary, not in the same format as above. This data summary must be configured in the following manner: per Ad Asset ID, per Program Identifier, per Network, per Opportunity Type, per Break, per Position.

|  |  |
| --- | --- |
| Data Point | Configuration(s) |
| Total Households (Ad Asset Centric) | 1. The Total Unique number of viewing Households during the specified reporting time period. (Ad Asset Centric) |

1. Break, Placemnt, & Position illustrations



1. Reporting – Sample UpdateNotice Messages

3 Update Notices that resolve to the Data Set Samples

<!-- Sample for use in reporting definition - Canoe SaFI Interface  
 Specifications v3.0 -->  
<UpdateNotice xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <CampInfoPkg>  
 <Version minSchemaVersion="2" minSchemaMinorVersion="1"/>  
 <MsoOrder name="<MSO\_NAME>">  
 <!-- this order is for all syscodes for the MSO -->  
 <Orders>  
 <!-- parent Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a"  
 decisionOwner="ABC.COM" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQ1" epsid="0" peid="sOHzM85AEd6ukwBUNDLE01"   
 priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- All Qualifiers appear in the parent Bundle -->  
 <CommonQualifierGroup condition="or">  
 <Qualifier>  
 <!-- Target "Body of Proof, Ep 1" on abc -->  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">Body of Proof, Ep 1</Term>  
 <Term source="ent" operation="eq" xpath="uriID"  
 namespace="cablelabs.com/namespaces/metadata/xsd/core/1">abc.com/Title/CNVT0100016H</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target "Body of Proof, Ep 1" on abc -->  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">Body of Proof, Ep 1</Term>  
 <Term source="ent" operation="eq" xpath="uriID"  
 namespace="cablelabs.com/namespaces/metadata/xsd/core/1">dxd.com/Title/CNVT0100016H</Term>  
 </Qualifier>  
 </CommonQualifierGroup>  
 <!-- child Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="ABC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQR" epsid="0"  
 peid="sOHzM85AEd6ukwBUNDLE02"   
 priority="3" productFamily="urn:canoe:safi:pfamily:single:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement - creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00"  
 creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="sOHzM85AEd6ukwBUNDLE03" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="sOHzM85AEd6ukwASSETR04" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">ads.abc.com/Title/CNVT0100008</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1"> AUTO.,AUTO.ACCESS &amp; EQUIP</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTOS, TRUCKS &amp; MISC VEHICLES- FACTORY</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">LIGHT TRUCKS &amp; VANS-FACTORY: NEW &amp; CPO</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">TRUCKS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-28T11:00:00-05:00" endRequired="2011-11-30T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Phase 3 - Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 </Placement>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the opportunity -->  
 <CategoryExclusion scope="opportunity"/>  
 </Bundle>  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="a" decisionOwner="ABC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQR" epsid="0"  
 peid="sOHzM85AEd6ukwBUNDLE05"   
 priority="3" productFamily="urn:canoe:safi:pfamily:single:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement - creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00"  
 creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="sOHzM85AEd6ukwBUNDLE06" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="sOHzM85AEd6ukwASSETR07" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">ads.abc.com/Title/CNVT0100010</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1"> AUTO.,AUTO.ACCESS &amp; EQUIP</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTOS, TRUCKS &amp; MISC VEHICLES- FACTORY</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">LIGHT TRUCKS &amp; VANS-FACTORY: NEW &amp; CPO</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">TRUCKS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-28T11:00:00-05:00" endRequired="2011-11-30T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Phase 3 - Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 </Placement>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the opportunity -->  
 <CategoryExclusion scope="opportunity"/>  
 </Bundle>  
 <!-- Set window 10/1 - 10/31, M-F, from 9:00 am to 1:00 pm -->  
 <!-- Flight is established in the parent Bundle -->  
 <Flight startDateTime="2011-11-01T09:00:00-05:00" endDateTime="2011-11-30T09:00:00-05:00" startTime="09:00:00" duration="PT04H00M00S"  
 sat="false" sun="false"/>  
 <!-- Statefeedback mediaState and executionState and well as statistics are summarized in the  
 parent Bundle -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S"  
 reportEndTime="2011-12-03T09:00:00-05:00">  
 <Statistics>  
 <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions"/>  
 <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="Views"/>  
 </Statistics>  
 </StateFeedback>  
 <!-- When bundle is defined, goal is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetGoal" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="goal">100000</Arg>  
 <Arg variable="goalType">views</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 </Bundle>  
 </Orders>  
 </MsoOrder>  
 </CampInfoPkg>  
</UpdateNotice>

<!-- Sample for use in reporting definition, Campaign #1, CI #2, 2 asset - Canoe SaFI Interface  
 Specifications v3.0 -->  
<UpdateNotice xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <CampInfoPkg>  
 <Version minSchemaVersion="2" minSchemaMinorVersion="1"/>  
 <MsoOrder name="<MSO\_NAME>">  
 <!-- this order is for all syscodes for the MSO -->  
 <Orders>  
 <!-- parent Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="100000002"  
 decisionOwner="ABC.COM" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQ4" epsid="0" peid="sOHzM85AEd6ukwBUNDLE08"   
 priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- All Qualifiers appear in the parent Bundle -->  
 <CommonQualifierGroup condition="or">  
 <Qualifier>  
 <!-- Target "Desperate Housewives, Ep 501" on abc -->  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">Desperate Housewives, Ep 501</Term>  
 <Term source="ent" operation="eq" xpath="providerID"  
 namespace="cablelabs.com/namespaces/metadata/xsd/core/1">abc.com</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target "Desperate Housewives, Ep 501" on soap -->  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">Desperate Housewives, Ep 501</Term>  
 <Term source="ent" operation="eq" xpath="providerID"  
 namespace="cablelabs.com/namespaces/metadata/xsd/core/1">soap.com</Term>  
 </Qualifier>  
 </CommonQualifierGroup>  
 <!-- child Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="100000002" decisionOwner="ABC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQ5" epsid="0"  
 peid="sOHzM85AEd6ukwBUNDLE09"   
 priority="3" productFamily="urn:canoe:safi:pfamily:single:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement - creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00"  
 creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="sOHzM85AEd6ukwBUNDLE10" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="sOHzM85AEd6ukwASSETR11" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">ads.abc.com/Title/CNVT0100008</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1"> AUTO.,AUTO.ACCESS &amp; EQUIP</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTOS, TRUCKS &amp; MISC VEHICLES- FACTORY</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">LIGHT TRUCKS &amp; VANS-FACTORY: NEW &amp; CPO</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">TRUCKS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-28T11:00:00-05:00" endRequired="2011-11-30T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Phase 3 - Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 </Placement>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the opportunity -->  
 <CategoryExclusion scope="opportunity"/>  
 </Bundle>  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="100000002" decisionOwner="ABC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQ6" epsid="0"  
 peid="sOHzM85AEd6ukwBUNDLE12"   
 priority="3" productFamily="urn:canoe:safi:pfamily:single:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement - creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00"  
 creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="sOHzM85AEd6ukwBUNDLE13" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="sOHzM85AEd6ukwASSETR14" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">ads.abc.com/Title/CNVT0100012</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTO.,AUTO.ACCESS &amp; EQUIP</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTOS, TRUCKS &amp; MISC VEHICLES- FACTORY</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">LIGHT TRUCKS &amp; VANS-FACTORY: NEW &amp; CPO</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">TRUCKS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-28T11:00:00-05:00" endRequired="2011-11-30T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Phase 3 - Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 </Placement>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the opportunity -->  
 <CategoryExclusion scope="opportunity"/>  
 </Bundle>  
 <!-- Set window 10/1 - 10/31, M-F, from 9:00 am to 1:00 pm -->  
 <!-- Flight is established in the parent Bundle -->  
 <Flight startDateTime="2011-11-01T09:00:00-05:00" endDateTime="2011-11-30T09:00:00-05:00" startTime="09:00:00" duration="PT04H00M00S"  
 sat="false" sun="false"/>  
 <!-- Statefeedback mediaState and executionState and well as statistics are summarized in the  
 parent Bundle -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S"  
 reportEndTime="2011-12-03T09:00:00-05:00">  
 <Statistics>  
 <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions"/>  
 <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="Views"/>  
 </Statistics>  
 </StateFeedback>  
 <!-- When bundle is defined, goal is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetGoal" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="goal">100000</Arg>  
 <Arg variable="goalType">views</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 </Bundle>  
 </Orders>  
 </MsoOrder>  
 </CampInfoPkg>  
</UpdateNotice>

<!-- Sample for use in reporting definition, Campaign #2, CI #1, 2 assets - Canoe SaFI Interface  
 Specifications v3.0 -->  
<UpdateNotice xmlns="urn:cablelabs:safi:xsd:cip:3.0"  
 xsi:schemaLocation="urn:cablelabs:safi:xsd:cip:3.0 OC-SaFI-CIP-3.0.-2.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
 <CampInfoPkg>  
 <Version minSchemaVersion="2" minSchemaMinorVersion="1"/>  
 <MsoOrder name="<MSO\_NAME>">  
 <!-- this order is for all syscodes for the MSO -->  
 <Orders>  
 <!-- parent Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="100000001"  
 decisionOwner="ABC.COM" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQ7" epsid="0" peid="sOHzM85AEd6ukwBUNDLE15"   
 priority="3" productFamily="urn:canoe:safi:pfamily:vod:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- All Qualifiers appear in the parent Bundle -->  
 <CommonQualifierGroup condition="or">  
 <Qualifier>  
 <!-- Target "Desperate Housewives, Ep 501" on abc -->  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">Desperate Housewives, Ep 501</Term>  
 <Term source="ent" operation="eq" xpath="providerID"  
 namespace="cablelabs.com/namespaces/metadata/xsd/core/1">abc.com</Term>  
 </Qualifier>  
 <Qualifier>  
 <!-- Target "Castle, Ep 1" on dxd -->  
 <Term source="ent" operation="eq" xpath="Title" namespace="cablelabs.com/namespaces/metadata/xsd/title/1">Castle, Ep 1</Term>  
 <Term source="ent" operation="eq" xpath="providerID"  
 namespace="cablelabs.com/namespaces/metadata/xsd/core/1">dxd.com</Term>  
 </Qualifier>  
 </CommonQualifierGroup>  
 <!-- child Bundle - campVer will always = "1", creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="100000001" decisionOwner="ABC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQ8" epsid="0"  
 peid="sOHzM85AEd6ukwBUNDLE16"   
 priority="3" productFamily="urn:canoe:safi:pfamily:single:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement - creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00"  
 creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="sOHzM85AEd6ukwBUNDLE17" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="sOHzM85AEd6ukwASSETR18" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">ads.abc.com/Title/CNVT0100008</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTO.,AUTO.ACCESS &amp; EQUIP</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTOS, TRUCKS &amp; MISC VEHICLES- FACTORY</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">LIGHT TRUCKS &amp; VANS-FACTORY: NEW &amp; CPO</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">TRUCKS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-28T11:00:00-05:00" endRequired="2011-11-30T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Phase 3 - Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 </Placement>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the opportunity -->  
 <CategoryExclusion scope="opportunity"/>  
 </Bundle>  
 <Bundle campVer="1" campOrderOwner="canoe-ventures.com" campOrder="100000001" decisionOwner="ABC.COM"  
 revDat="2011-10-01T15:00:00-05:00" revision="3" recRevoked="false" recState="act" creDat="2011-09-28T11:00:00-05:00"   
 guid="ntDbws5AEd6ukwAf4hSEQ6" epsid="0"  
 peid="sOHzM85AEd6ukwBUNDLE19"   
 priority="3" productFamily="urn:canoe:safi:pfamily:single:1.0">  
 <StwShpData>  
 <Contact role="Sales" name="TJ Moretto" email="tj.moretto@canoe-ventures.com"/>  
 <Contact role="Operations" name="Jamie Donnenfeld" email="jamie.donnenfeld@canoe-ventures.com"/>  
 </StwShpData>  
 <!-- Placement - creDat will remain the same as it represents the creation of the element. revDate is the date/time the current  
 UpdateNotice was created, revision will be incremented with each UpdateNotice, all instances of  
 priority within a parent bundle and its descendent elements will be equal. -->  
 <Placement revision="3" recRevoked="false" recState="act" revDat="2011-10-01T15:00:00-05:00"  
 creDat="2011-09-28T11:00:00-05:00"   
 epsid="0" peid="sOHzM85AEd6ukwBUNDLE20" priority="3" productMember="urn:canoe:safi:pmember:vodplacement:1.0" separationConstraint="opportunity">  
 <AssetReference epsid="0" peid="sOHzM85AEd6ukwASSETR21" apply="valAndPlace" assetType="media">  
 <AssetMetadata xpath="uriID"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/core/1">ads.abc.com/Title/CNVT0100010</AssetMetadata>  
 <AssetMetadata xpath="IndustryGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTO.,AUTO.ACCESS &amp; EQUIP</AssetMetadata>  
 <AssetMetadata xpath="MajorGroup"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">AUTOS, TRUCKS &amp; MISC VEHICLES- FACTORY</AssetMetadata>  
 <AssetMetadata xpath="PCCDescription"  
 namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">LIGHT TRUCKS &amp; VANS-FACTORY: NEW &amp; CPO</AssetMetadata>  
 <AssetMetadata xpath="ProductCategory" namespace="http://www.cablelabs.com/namespaces/metadata/xsd/ad-id/1">TRUCKS</AssetMetadata>  
 <ContentMgmt beginRequired="2011-09-28T11:00:00-05:00" endRequired="2011-11-30T09:00:00-05:00"/>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </AssetReference>  
 <!-- Phase 3 - Preventing fast forward for the placement -->  
 <TrickModeRestriction>  
 <TrickModeExclusion type="trick" scale="0.0,100"/>  
 </TrickModeRestriction>  
 </Placement>  
 <!-- Prevent insertion of any ad from this Placement if an ad of he same category already exists in the opportunity -->  
 <CategoryExclusion scope="opportunity"/>  
 </Bundle>  
 <!-- Set window 10/1 - 10/31, M-F, from 9:00 am to 1:00 pm -->  
 <!-- Flight is established in the parent Bundle -->  
 <Flight startDateTime="2011-11-01T09:00:00-05:00" endDateTime="2011-11-30T09:00:00-05:00" startTime="09:00:00" duration="PT04H00M00S"  
 sat="false" sun="false"/>  
 <!-- Statefeedback mediaState and executionState and well as statistics are summarized in the  
 parent Bundle -->  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00" reportInterval="PT00H15M00S"  
 reportEndTime="2011-12-03T09:00:00-05:00">  
 <Statistics>  
 <Statistic datamap="urn:cablelabs:vod:placements:campaign.stats" variable="Insertions"/>  
 <Statistic datamap="urn:cablelabs:vod:viewership:campaign.stats" variable="Views"/>  
 </Statistics>  
 </StateFeedback>  
 <!-- When bundle is defined, goal is communicated -->  
 <Phase epsid="0" peid="sOHzM85AEd6ukwAf4hSEQQ" event="defined" revision="3" recRevoked="false" recState="act"   
 revDat="2011-10-01T15:00:00-05:00" creDat="2011-09-28T11:00:00-05:00">  
 <ProcessRuleSelector ruleId="SetGoal" ruleRepository="urn:canoe:safi:rulesRepo:1.0">  
 <Arg variable="goal">100000</Arg>  
 <Arg variable="goalType">views</Arg>  
 <StateFeedback revision="3" reportAsOf="2011-10-01T15:00:00-05:00"></StateFeedback>  
 </ProcessRuleSelector>  
 </Phase>  
 </Bundle>  
 </Orders>  
 </MsoOrder>  
 </CampInfoPkg>  
</UpdateNotice>

1. Reporting – Sample Campaign Centric Data Set



1. Reporting – Sample Ad Centric Data Set



1. Issues

This table enumerates open resolved technical issues.

* 1. Open Issues

| **Issue** | **Description** |
| --- | --- |
|  | Resolve use of manipulated content metadata for execution of the campaign item. |
|  |  |

* 1. Resolved Issues

| **Issue** | **Description** |
| --- | --- |
|  | This document does not include requirements for tracking/monitoring campaign and content (ad and title) status. Additional discussions are needed to determine the interface(s) that will support these requirements. |
|  | Implementation for Product Category Separation must be defined. |
|  | Implementation for Ad Separation must be defined. |
|  | CIP doc needs to provide for private qualifier domains |
| 5. | Measurement data discussion deferred pending findings of a task force that will map the requirements to the Service Measurement Data Model. |
|  |  |

1. Contributors and Acknowledgments

The principle authors of this document are:

| Author | Role |
| --- | --- |
| TJ Moretto | Canoe, Sr. Product Manager |
| Bruce Dilger | Canoe, VP Advanced Advertising Architecture |
| Tim Whitton | Canoe, Business Analyst |

Additional contributors and reviewers include:

| Contributor / Reviewer | Role |
| --- | --- |
| Jon Farb | Canoe, Director of Product Management, VOD |
| Jeff Dillon | Canoe, Director of Engineering, VOD |