comScore

UNIFIED DIGITAL MEASUREMENT

comScore Library Reduced Requirements Streaming Tag

JavaScript Implementation Guide

document version: 4.2.3; released on May 4, 2017

Contents

Implementation Quick Start

This section contains abridged instructions for experienced users to implement the Streaming Tag using version 6.x of the comScore Streaming library.

> ⚠ **Please do not use classes, methods or properties which do not appear in this documentation unless you have received explicit instructions to do so from your comScore account team or comScore Tag Support.**

1. Confirm you have taken care of all the preparation items (🔗 *details on page 4*).
2. Include the relevant JavaScript libraries in your web page or application project (🔗 *details on page 6*).
   - For stand-alone implementations on a web page intended to be shown in web browsers on desktop and mobile devices: include the comScore streaming JavaScript library.
   - For implementations in OTT application projects: ensure the *latest* version of the comScore OTT JavaScript library is implemented and include the comScore streaming JavaScript library in your project.
   - For implementations in Cordova mobile application projects: ensure the *latest* version of the comScore Cordova plugin is implemented.
3. Create a `ns_.ReducedRequirementsStreamingAnalytics` instance (🔗 *details on page 6*).
   - For stand-alone implementations your comScore Publisher ID needs to be provided for initialization:

     ```
     11.  var streamingAnalytics = new ns_.ReducedRequirementsStreamingAnalytics( { publisherId: '1234567' } );
     ```

   - For implementations in an OTT application or Cordova mobile application no initialization settings are needed:

     ```
     11.  var streamingAnalytics = new ns_.ReducedRequirementsStreamingAnalytics();
     ```

4. Prepare the metadata which needs to be supplied for each of the content and advertisement assets (🔗 *details on page 7*).
5. Determine the appropriate media type for each of the content and advertisement assets (🔗 *details on page 10*).
6. Tag playback start and end of the content segment or advertisement, including the prepared metadata and media type (🔗 *details on page 11*).

After successfully following these steps the comScore library will be collecting data for your streaming media playback. The library will send measurements whenever playback is started.

1 Introduction

The comScore Streaming Tag provides accurate and comprehensive streaming media analytics functionality. This enablescomScore to receive measurement insights critical to answering questions about audio and video stream usage, including advertising messages.

This documentation is equally applicable to both video and audio streaming measurement. Please clarify with your comScore account team what type of media you should be implementing this tag onto (video and/or audio). Please do not implement this tag onto media types other than those you have been instructed to by your comScore account team.

This documentation applies to the implementation of comScore streaming library version 6.x.

1.1 Intended Use

These instructions and the accompanying comScore library are intended to be used with media players running inside web pages intended to be shown in web browsers on desktop and mobile devices or with media players running inside OTT applications and Cordova mobile application and offer a JavaScript API. If your OTT application is developed in another programming language then please contact your comScore account team to ask for guidance.

1.2 Preparation

Please complete the following checklist before implementing the Streaming Tag in your application project:

1. If you plan on using the Streaming Tag in your OTT application or Cordova mobile application then make sure you have implemented the comScore Application Tag (either the OTT library or the comScore Cordova plugin) in your project and that you have updated the comScore Application Tag to the latest version.
2. If you are updating an existing Streaming Tag implementation, then please refer to 🔗 *Appendix B: Migrating an Existing a Previous Implementation on page 15* to see if there are any relevant steps mentioned for your situation.
3. If you plan on using the Streaming Tag stand-alone, on a web page intended to be shown in web browsers on desktop and mobile devices then confirm your comScore *Publisher ID* - also known as the *Client ID* or *c2 value* - which is a number with at least 7 digits, provided by comScore.
4. Clarify with your comScore account team what type of media you should be implementing the Streaming Tag for (video and/or audio).
5. Familiarize yourself with the tagging instructions and the description of the tagging methodology (refer to 🔗 *Tagging Methodology on page 5*) in this document.
6. Determine the media asset metadata values that need to be collected.
7. Make sure you are using a media player that has an API which allows you to detect the player state and allows you to access details like the current playback position and relevant media asset metadata.

> ℹ️ **About retrieving comScore Publisher details and comScore SDKs...**

1. Use a web browser to visit 🔗 *comScore Direct (http://direct.comscore.com/clients/Video.aspx)* .
2. Log in to comScore Direct with your user account and password if you are prompted to.
   If you are a comScore client but do not have a comScore user account then please contact your comScore account team. If you are not yet a comScore client please sign up for a user account through comScore Direct.
3. Confirm you are on the *Mobile App* tab and click on *Get Tag*.
4. The Publisher ID is shown in the popup.
5. Please contact your comScore account team to get (updates to) comScore SDKs.

2 Tagging Methodology

Your media player plays content media and very likely also advertisement media. In the tagging methodology each coherent content media and each individual advertisement media is represented by exactly one *Asset*. Here 'coherent' refers to the content as a whole, regardless of whether or not the data stream for the content media is cut up into separate streams or Files[1] that are seamlessly switched between during playback.

To illustrate this, assume the media player loads content with a pre-roll ad break, two mid roll ad breaks and a post-roll ad break. Except for the second mid-roll ad break all ad breaks have one advertisement assigned.

Media player time line
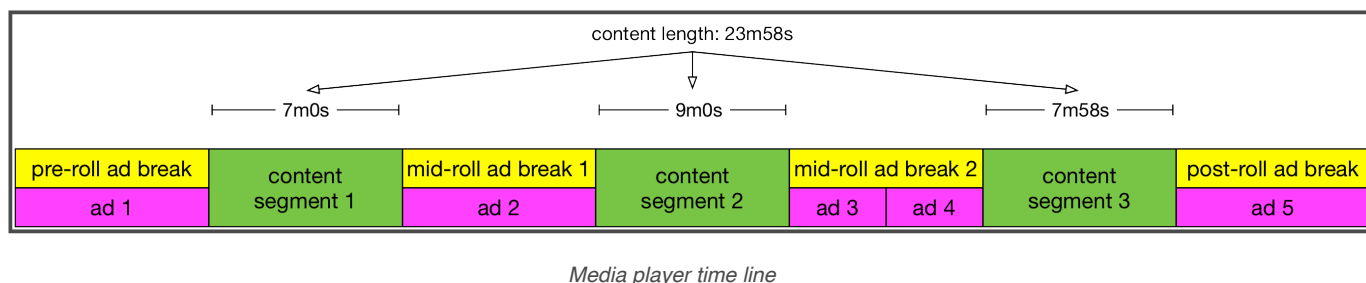
The two mid-roll ad breaks divide the media player time line for the content into 3 *Segments* or *Parts*. The 3 segments have the lengths 7m0s, 9m0s and 7m58s, totalling to 23m58s (not surprisingly, that is equal to the content media length). To be clear, this puts the cue points at positions 0m0s (the pre-roll ad break), 7m0s (the first mid-roll ad break), 16m0s (the second mid-roll ad break) and 23m58s (the post-roll ad break).

Each discrete content media and individual advertisement media is represented by exactly one *Asset*. All data for an asset is collected by specifying *Labels* in a `Object` of which the key/value pairs are set to the label name/value pairs. This example media player time line has six *Assets*, one for the content and five for the advertisements. Note that content media is represented with a single *Asset* regardless of the amount of *Segments* the content media is divided into by mid-roll ad breaks.

Each asset that can be tagged for also has a *Media Type*, like "linear pre-roll advertisement". The *Media Types* available to the implementation are defined as enums.

---

(1) This refers to cases where, for example, content with a length of 40m0s is cut into two pieces: one stream which contains positions 0m0s to 21m0s and another stream that contains positions 21m0s to 40m0s.

In the implementation the `Objects` with asset metadata and the media type are provided as arguments in calls to playback state change notification methods. Those calls occur whenever the media player starts or stops playback.

3 Implement the Streaming Tag

As you work with the comScore library you might see classes, methods and properties which do not appear in this documentation. Those elements are typically exposed because that is necessary for the library to work in the way it was intended to or to support customized implementations for which you would receive separate instructions from your comScore account team or comScore Tag Support.

The use of those elements without instructions from comScore could severely impact the behaviour of the comScore library, the quality of the collected data or reporting capabilities.

> ⚠️ **Please do not use classes, methods or properties which do not appear in this documentation unless you have received explicit instructions to do so from your comScore account team or comScore Tag Support.**

If you have any questions or concerns about the instructions in this document or about any classes, methods or properties which you find in the comScore library then please contact your comScore account team or comScore Tag Support.

3.1 Include the Library

> ℹ️ **About the instructions in this section...**
>
> The instructions in this section do not apply if you are implementing the Streaming Tag in a Cordova mobile application. The comScore Cordova plugin already includes the Streaming Tag library. No further steps are required to include the library.

To start using the Streaming Tag you will have to make sure the JavaScript File with the comScore streaming library is available in your HTML document. You can load this JavaScript File from the HTML document's `<head>` section, using:

```
<script src="comscore.streaming.min.js" type="text/javascript"></script>
```

This File contains JavaScript code with definitions of classes used by the Streaming Tag implementation. You can load the File dynamically (or asynchronously) as long as you make sure the File is loaded before any code statements that use the comScore streaming library are executed.

3.2 Create a `ns_.ReducedRequirementsStreamingAnalytics` Instance

Depending on your implementation environment the instance of `ns_.ReducedRequirementsStreamingAnalytics` needs to be created with or without a configuration object.

3.2.1 Stand-alone Implementation in a Web Page

If you are implementing the Streaming Tag stand-alone, on a web page intended to be shown in web browsers on desktop and mobile devices, then you need to create a new instance and initialize it with configuration settings which specify your comScore Publisher ID:

```
11.  var streamingAnalytics = new ns_.ReducedRequirementsStreamingAnalytics( { publisherId: '1234567' } );
```

> Note that the `publisherId` property name is case-sensitive.

3.2.2 Implementation in an Application

If you are implementing the Streaming Tag in an OTT application or Cordova application, then you need to also implement the Application Tag with the comScore OTT library or comScore Cordova plugin. The Streaming Tag will reuse the initialization settings from the Application Tag implementation. Create an instance without providing any configuration settings:

```
11.  var streamingAnalytics = new ns_.ReducedRequirementsStreamingAnalytics();
```

3.3 Prepare Asset Metadata

Ensure you have a `Object` with metadata label values for the *Asset* your media player is going to play. The `Object` will be provided as an argument in the calls to the playback state change notification methods.

The table below lists all standard labels allowed for use in your Streaming Tag implementation, with their intended purpose and expected values and the comScore products they are typically used for. If you have any questions about which values to populate or if any of this data is not available for your implementation to use, then please contact your comScore account team.

> ⚠️ **About providing advertisement asset metadata...**
>
> For any *advertisement* assets **only** include the *Asset Length* label (`ns_st_cl`) in the `Object` with asset metadata unless you are otherwise instructed. Other metadata labels are usually not needed for advertisement assets.

Standard labels available for the Streaming Tag implementation

Products Legend **V** = *Video Metrix* **X** = *Cross Platform Product Suite*

Label	Item	Description	Presence	Products	Example value(s)
ns_st_ci	Unique Content ID	Provide your internal unique identifier for the content media *Asset*. Used in the report calculations logic to identify unique content assets. Provide value 0 if your media player does not use or have access to unique content media identification.	mandatory	V	13784

Label	Item	Description	Presence	Products	Example value(s)
ns_st_cl	Asset Length	The length of the individual *Asset*, i.e., the available amount of media. Expects a value **in milliseconds**. If your media player or content metadata database reports media length values in seconds then please multiply the reported values by 1000. If the media length is unknown or cannot be determined then please provide value 0.	mandatory	V X	1260000
ns_st_pu	Publisher Brand Name	Collect the consumer-facing brand name of the media publisher that owns the content.	mandatory	V X	• ABC • ESPN • CNN • NFL
ns_st_pr	Program Title	Top level content title (i.e., the name of the overall program, show, or content series). Can be used with label *Episode Title* (ns_st_ep) to tag TV shows on program and episode level.	mandatory	V X	• Modern Family • Harry Potter 7 • Game 16: Eagles vs Patriots
ns_st_tpr	Program ID	Top level content ID, which can be used for matching and grouping purposes (for example when the program title appears with multiple variations for the same program). *(This should not be confused with the Unique Content ID ns_st_ci which identifies an individual asset.)*	optional	V X	• 53617155
ns_st_ep	Episode Title	Sub level content title (i.e., the title of the specific episode). Can be used with label *Program Title* (ns_st_pr) to tag TV shows on program and episode level.	mandatory	X	• Rash Decisions • Season 2 Teaser
ns_st_tep	Episode ID	Sub level content ID, which can be used for matching and grouping purposes (for example when the episode title appears with multiple variations for the same episode of a specific program). *(This should not be confused with the Unique Content ID ns_st_ci which identifies an individual asset.)*	optional	V X	• 53617155
ns_st_sn	Episode Season Number	The season number for episodic content. It is recommended to use a value with 2 digits. Can be omitted or left blank for non-episodic content.	mandatory	X	05
ns_st_en	Episode Number	The episode number for episodic content. It is recommended to use a value with 2 digits[2].	mandatory	X	17
ns_st_ge	Content Genre	Content genre description. Multiple values can be provided as a comma-separated string.	mandatory	V X	• Comedy • Sports • Fantasy,Drama

---

[2] For episodic content with more than 99 episodes in a season it is recommended to use values with 3 digits, e.g., 017.

Label	Item	Description	Presence	Products	Example value(s)
`ns_st_ia`	Advertisement Load Flag	Indicates whether the streamed media carries the same advertisement load that was used during the TV airing. Use value `1` if the advertisement load is the same. Can be omitted otherwise. This flag helps comScore differentiate if the stream is carrying the same ad load as TV. Often digital video inventory is clubbed together with TV inventory and is served with the same ad load. The CPM$^{(3)}$ for digital inventory with TV ad load is different from the CPM for any other ad load. If for any reason your backend or workflow requires all media metadata to have values for the same set of labels, then please make sure you use value `0` for any streamed media which did **not** carry the same advertisement load as during the TV airing. The flag is considered 'set' when the label **is present with any non-empty, non-zero value**.	mandatory	X	`1`
`ns_st_ddt`	Digital Airdate	The date on which the content was made available for streaming consumption. Expects a value in `yyyy-mm-dd` format. This airdate helps comScore establish monetization windows (live, day +1, day +3, ...) forany given episode or show. The monetization windows are used to calculate commercial and program ratings.	mandatory	X	`2011-08-30`
`ns_st_tdt`	TV Airdate	The date on which the content aired on TV. Expects a value in `yyyy-mm-dd` format. This airdate helps comScore establish monetization windows (live, day +1, day +3, ...) for any given episode or show. The monetization windows are used to calculate commercial and program ratings.	mandatory	X	`2011-08-30`
`ns_st_st`	Station Title	The title of the station or channel for which content was recorded or where content is made available.	mandatory	V X	▪ `ESPN3` ▪ `BBC2`
`c3, c4, c6`	Video Metric Dictionary Classification	These labels determine which entity the clip will credit to in the Video Metrix dictionary. These mandatory values are used to determine the entity the clip will credit to in the Video Metrix dictionary. They do not have specific pre-defined meanings. You should work with your comScore account team to establish what these labels' values should be, based on your desired dictionary goals. **All three of these labels must always be passed.** Unused labels must still be passed with the literal string value '`*null`'.	mandatory	V	`*null`

(3) CPM is the advertising cost per impression.

Label	Item	Description	Presence	Products	Example value(s)
ns_st_ce	Complete Episode Flag	Specifies the content media to be a full episode rather than an excerpt. Use value 1 when the content media is a full episode. Can be omitted otherwise. This flag helps us identify if the streaming content is episodic, long-form, or premium in nature.It also indicates whether the show or episode will be explicitly broken out in the dictionary. If for any reason your backend or workflow requires all media metadata to have values for the same set of labels, then please make sure you use value 0 for any media (content and/or advertisements) which is **not** a full content episode. The flag is considered 'set' when the label **is present with any non-empty, non-zero value**.	mandatory	V X	1
		Asset Labels Specifically for Live Streams			
		These labels are expected to only be supplied for *Assets* which represent live streamed media. Live streamed media are provided to viewers as multicast, simulcast, or unicast streams where the viewer typically cannot control the playback position.			
ns_st_ft	Feed Type	Specified the type of feed provided on the live stream. Intended to be used on live streams using the same feed as was used for the live TV broadcast. Currently only used for implementations in the US where it can have the following values: ▪ EASTHD ▪ WESTHD ▪ EASTSD ▪ WESTHD	optional	X	

> ⚠ **About providing label values as `String` values...**
>
> Please make sure to always provide label values as strings even if the value format is numerical or if the value is a flag (i.e., 1 or 0). This is particularly important when adding elements to with labels.

Oftentimes, the label values you want to provide contain characters that cannot be passed in a URL (spaces, exclamation marks, commas, etc.). You do not have to URL encode or escape these characters. The comScore library will take care of URL encoding these characters.

To illustrate the use of the content media metadata labels some examples with a selection of labels are provided in 🔗 *Appendix A: Content Asset Metadata Examples on page 14*. The examples show what kind of label values are expected for commonly used types of content media.

3.4 Determine the Media Type

As part of the implementation you need to indicate the type of media. The chosen type will be provided as an argument in the calls to the playback state change notification methods.

Media type enums for content media assets

(The table shows a shortened reference for brevity. The full reference is

ns_.ReducedRequirementsStreamingAnalytics.ContentType.)

Content media type enum		Description
`ContentType.LongFormOnDemand`	on demand, long form	Content with strong brand equity or brand recognition, usually created or produced by media and entertainment companies using professional-grade equipment, talent, and production crews that hold or maintain the rights for distribution and syndication.
`ContentType.ShortFormOnDemand`	on demand, short form	
`ContentType.Live`	live / simulcast	
`ContentType.UserGeneratedLongFormOnDemand`	on demand, long form	Content with little-to-no brand equity or brand recognition. User- generated content (UGC) has minimal production value, and is uploaded to the Internet by non-media professionals.
`ContentType.UserGeneratedShortFormOnDemand`	on demand, short form	
`ContentType.UserGeneratedLive`	live / simulcast	
`ContentType.Bumper`		Bumpers - also known as "billboards" or "slates" - are static promotional images usually run before video content and usually lasting fewer than 5 seconds with or without a voice over. They are frequently not true video streams in the technical sense. Ideally, these would not be tagged due to their nature. This value can be used in cases where the bumpers have to be tagged.
`ContentType.Other`		Used if none of the above categories apply.

Media type enums for advertisement media assets

(The table shows a shortened reference for brevity. The full reference is `ns_.ReducedRequirementsStreamingAnalytics.AdType.`)

Advertisement media type enum		Description
`AdType.LinearOnDemandPreRoll`	linear, pre-roll	Linear ads delivered into a media player and presented before, in the middle of, or after content is consumed by the user. For video the ad completely takes over the full view of the media player.
`AdType.LinearOnDemandMidRoll`	linear, mid-roll	
`AdType.LinearOnDemandPostRoll`	linear, post-roll	
`AdType.LinearLive`	linear, live / simulcast	Linear ads delivered before, in the middle of, or after a live / simulcast stream of content. For video the ad completely takes over the full view of the media player.
`AdType.BrandedOnDemandPreRoll`	during linear pre-roll	Branded media that a user may intentionally view (like content), or it may be served to a user during an ad break (like an advertisement).
`AdType.BrandedOnDemandMidRoll`	during linear mid-roll	
`AdType.BrandedOnDemandPostRoll`	during linear post-roll	
`AdType.BrandedOnDemandContent`	as content	
`AdType.BrandedOnDemandLive`	during live / simulcast	
`AdType.Other`	Used if none of the above categories apply.	

3.5 Tag Playback State Changes

As your media player loads and plays content and advertisement media call the appropriate API method and provide your prepared metadata and media type as arguments to the call.

3.5.1 Start of Playback of Content Parts

Call the `playVideoContentPart` API method on the `ns_.ReducedRequirementsStreamingAnalytics` object instance and provide the metadata `Object` as well as the appropriate media type.

```
31.    streamingAnalytics.playVideoContentPart(metadata,
       ns_.ReducedRequirementsStreamingAnalytics.ContentType.LongFormOnDemand);
```

3.5.2 Start of Playback of Advertisements

Call the `playVideoAdvertisement` API method on the `ns_.ReducedRequirementsStreamingAnalytics` object instance and provide the metadata `Object`[4] as well as the appropriate media type.

```
33.    streamingAnalytics.playVideoAdvertisement( { ns_st_cl: "31000" },
       ns_.ReducedRequirementsStreamingAnalytics.AdType.LinearOnDemandMidRoll);
```

3.5.3 End of Playback

The following cases are considered *end of playback*:

- When playback of a content part ends or is paused.
- When playback of an advertisements ends.

In these cases please call the `stop` API method on the `ns_.ReducedRequirementsStreamingAnalytics` object instance.

```
39.    streamingAnalytics.stop();
```

Please note that if your player is playing media and playback is interrupted for *any reason* you must call the `stop` API method. This includes the following scenarios:

- The media player stops playback when the (mobile) device locks the screen or puts the application in the background.
- The media player pauses playback because the user scrolls/swipes the content and the media player ends up being out of view.

When playback resumes please call the appropriate start of playback notification method. Please note that these are example scenarios. Other scenarios might apply to your player too.

> **About calling the API methods in the correct order...**
>
> If your media player environment internally triggers player state change events asynchronously or triggers the end of playback of one media asset (content or advertisement) after it triggers the start of playback of another media then please pay close attention to the order in which you call the API methods.
>
> In these cases, if the API methods were to be called in the same order as the media player triggers state changes, then the comScore library will not accurately measure playing time.

3.5.4 Example Playback Scenario Transcript

If the media player plays the assets from example media player time line 🔗 *provided on page 5* , then the following sequence of activity in the player and corresponding Streaming Tag API calls would apply:

---

(4) Unless otherwise instructed only the *Asset Length* label (`ns_st_cl`) should be provided for advertisements.

Example transcript

(using simplified method names without arguments)

Step	Media	Streaming Tag API Call	
		Playback Starts	**Playback Stops**
1	pre-roll ad break (ad 1)	playVideoAdvertisement	stop
2	content segment 1	playVideoContentPart	stop
3	mid-roll ad break 1 (ad 2)	playVideoAdvertisement	stop
4	content segment 2	playVideoContentPart	stop
5	mid-roll ad break 2: 1st advertisement (ad 3)	playVideoAdvertisement	stop
6	mid-roll ad break 2: 2nd advertisement (ad 4)	playVideoAdvertisement	stop
7	content segment 3	playVideoContentPart	stop
8	post-roll ad break (ad 5)	playVideoAdvertisement	stop

If you have followed the implementation instructions up to this point then the comScore library will be collecting data for your streaming media playback. The library will send measurements whenever playback is started.

Appendix A: Content Asset Metadata Examples

There are different types of video content out there on the internet and each type has certain nuances about how it should be tagged in order to be reported correctly in comScore's Audience measurement products. This section will guide you how to populate the video metadata parameters for the most common types of content available on the internet.

Content Asset Metadata Examples

Label	TV Show Episode	TV Show Trailer	Live Sports Content	Sports Highlight Clip	Movie	Movie Trailer	Online News Content	Music Video
Station Title (ns_st_st)	Hulu	Youtube	ESPN3	Youtube	Hulu	Youtube	Huffington Post	VEVO
Publisher Brand Name (ns_st_pu)	ABC	ABC	ESPN	NFL	Warner Bros.	Warner Bros.	Huffington Post	VEVO
Program Title (ns_st_pr)	Modern Family	Modern Family	Game 16: Eagles vs Patriots	Game 16: Eagles vs Patriots	Harry Potter 7	Harry Potter 7	Huff Post Live	Taylor Swift
Episode Title (ns_st_ep)	Rash Decisions	Season 2 Teaser	*null	*null	*null	Harry Potter 7 Trailer #3	All is not Well in Hillaryland	Wildest Dreams
Episode Season Number (ns_st_sn)	1	*null	*null	*null	*null	*null	*null	*null
Episode Number (ns_st_en)	2	*null	*null	*null	*null	*null	*null	*null
Genre (ns_st_ge)	Comedy	Comedy	Sports	Sports	Fantasy, Drama	Fantasy, Drama	News	Music
Complete Episode Flag (ns_st_ce)	1	0	1	0	1	0	0	0

A list of suggested Genre (ns_st_ge) values is provided below:

- Action / Adventure
- Adult
- Animation
- Awards
- Comedy
- Food
- Documentary
- Drama
- Educational
- Fantasy
- Foreign Language
- Game Show
- Holiday
- Home & Garden / Home Improvement
- Home Shopping
- Kids
- Lifestyle
- Movies
- Music
- News
- Paid Programming
- Politics / Public Affairs
- Reality
- Religious
- Science Fiction
- Soap Opera
- Sports
- Talk
- Thriller / Horror
- Travel
- Variety

Appendix B: Migrating an Existing a Previous Implementation

These steps apply to Streaming Tag implementations which are migrating from comScore streaming library version 4.x or 5.x.

1. If you are updating the Streaming Tag in your OTT application or Cordova mobile application then make sure you have updated the comScore Application Tag (either the OTT library or the comScore Cordova plugin) in your project to the latest version.

2. If you are updating the Streaming Tag in your OTT application (not a Cordova mobile application) or in a stand-alone implementation, on a web page intended to be shown in web browsers on desktop and mobile devices, then take note of the new File name of the comScore streaming tag library JavaScript File and update your environment accordingly. The old File name contained the word 'streamsense', which is no longer the case.

3. Replace occurrences of `ns_.StreamingTag` with `ns_.ReducedRequirementsStreamingAnalytics`.

The previous step will have caused your object reference for your Streaming Tag implementation to now point to an object instance `ns_.ReducedRequirementsStreamingAnalytics`.

That step should also have caused any occurrences of `ns_.StreamingTag.ContentType` and `ns_.StreamingTag.AdType` to be replaced with `ns_.ReducedRequirementsStreamingAnalytics.ContentType` and `ns_.ReducedRequirementsStreamingAnalytics.AdType`, respectively.

4. Locate the code statement which creates the `ns_.ReducedRequirementsStreamingAnalytics` object instance. Depending on your implementation it could be using a configuration object which uses a property `customerC2` to specify your comScore Publisher ID. The code statement will look like this:

```
11. var streamingAnalytics = new ns_.ReducedRequirementsStreamingAnalytics( { customerC2: '1234567' } );
```

5. If your implementation uses that configuration object, then rename the `customerC2` property on the configuration object into `publisherId`:

```
11. var streamingAnalytics = new ns_.ReducedRequirementsStreamingAnalytics( { publisherId: '1234567' } );
```

About the code statements in an existing comScore library implementation...

It could be that some of the mentioned comScore library classes, API methods or method arguments do not appear in your implementation. If your implementation contains elements which are not mentioned in these migration instructions then please contact your comScore account team or comScore Tag Support for additional instructions.