

VideoCensus

Tagging Implementation

Quick Start Guide

Last modified: October 2011

Introduction

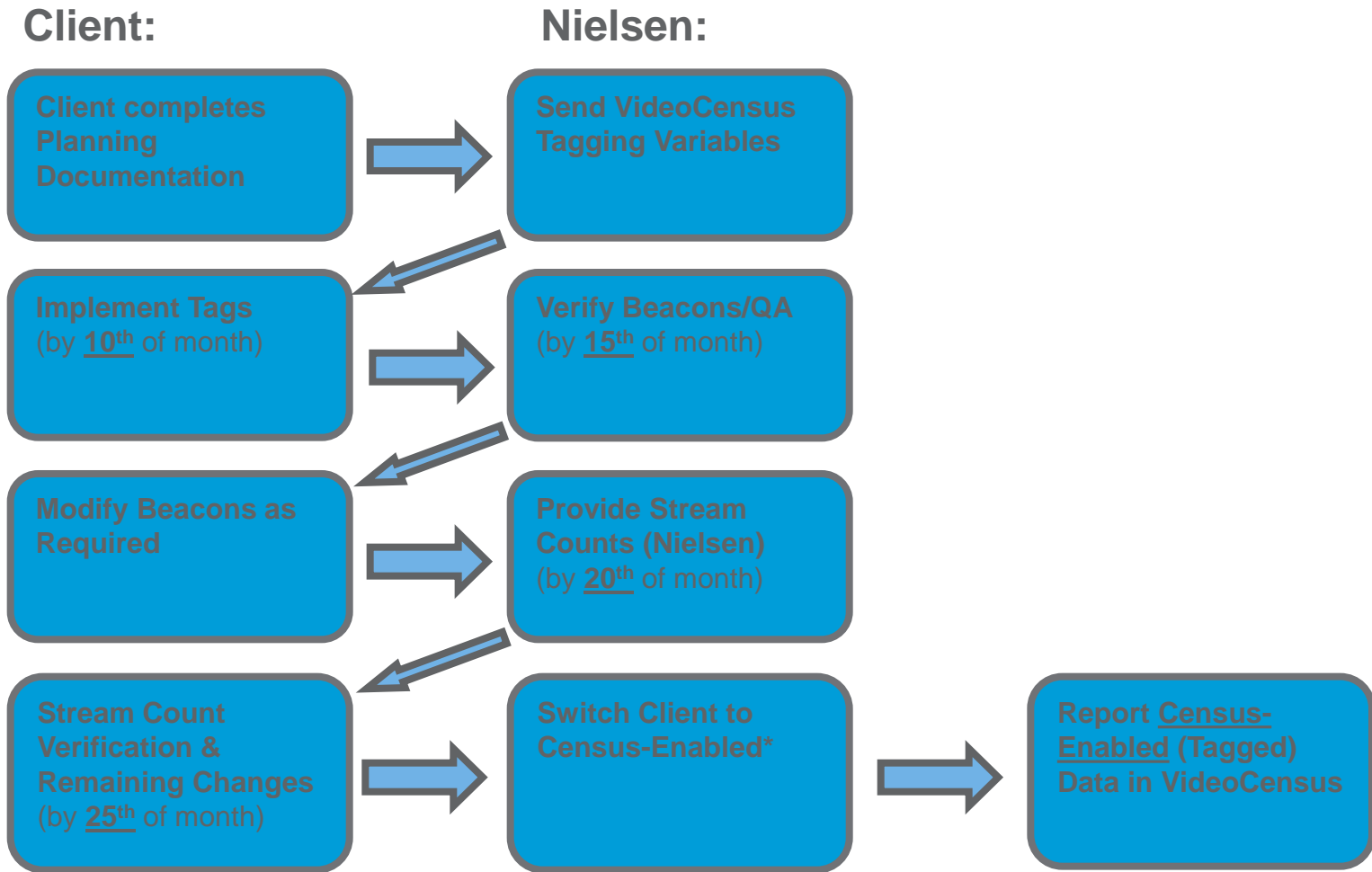
This quick start guide will take you through an implementation of the basic Nielsen video tag for Video Census measurement.

Sections include:-

- Typical implementation time line
- Basic structure / format of the video tag
- JavaScript, Java, Flash and iOS code snippets

Note: For Video Analytics, IAG and Market Intelligence tagging, please contact onlinevideo@nielsen.com for the appropriate documentation.

Timeline:



*Requires 1 month of active measurement

The basics

- **Send a start call for each video**
- If an ad interrupts: send a new start call when the content resumes
- **For full episodes-**
 - **Sent a start call for each segment the first time it plays.**
 - If the user scrubs backward to a previous segment and an ad plays before the content resumes- send an additional call for that segment.
 - Do not send an additional call for that segment if an ad does not play in between segments.
 - *Ex: a full episode with 5 segments might have 5 start calls, potentially more if the user revisits a prior segment and is interrupted by an ad).*
 - Include the LP parameter for full episode players – see video tag parameters later in this document

The basics (continued)

- **Tagging ads is optional but recommended:** include the ad identifying parameter (c3). Otherwise, do not send a call for ads as this will be flagged during our QA process.
- If another video is clicked while one is in progress you should send a start call for the new video that starts playing.
- Do not send any additional calls for pausing, adjusting video quality, or changing to full screen
- **Important:** please provide stream URL definitions so we have a backup (ask your VideoCensus rep for a template to do this). This will ensure a comprehensive back-up in the event of a problem with your tags.

Video Tag parameters

There are 5 basic parameters for content and ads:

- Two Nielsen-specific (required):

1. **CI (Client ID)**
 - Ex: ci=us-123456
2. **C6 (VideoCensus ID)**
 - Ex: c6=vc,c01

(Nielsen will provide both of these values to you)

- Three fixed (required):

1. **CC (cookie check)**
 - Hard code to cc=1
2. **RND (random number)**
 - Ex: rnd=123456789
 - Must be dynamic per event, do not use scientific notation.
3. **TL (video title)**
 - Should be prefixed by “dav0-”
 - Percent (%) encode the TL value after the “dav0-” prefix

tl=dav0-Encode%20This%20Title

Encoding prevents restricted characters from impacting any processing scripts.

Video Tag parameters (content specific)

- One content-specific (required):
 1. **CG (show name or category name)**
 - TV networks required to use program name here
 - The entire cg value should be percent (%) encoded
cg=Encode%20Program%20Name%20Here
Note: Encoding prevents restricted characters from impacting any processing scripts.
- Optional parameters:
 1. **LP (Long play indicator)**
 - Four sub-parameters:
 - Short form / Long form override. The publisher can explicitly state that this is a short form or long form clip. If set to SF then parameters 2 and 4 will be ignored.
 - Current segment/chapter number. Set to 0 if not known.
 - Length in seconds of this segment/chapter. Set to 0 if not known.
 - Anticipated total number of segments/chapters for this episode. Set to 0 if not known.
 - lp=LF,3,582,6**
 2. **LS (Live stream indicator)**
 - One parameter:
 - Set to Y if this is a live stream
 - Set to N if this is a standard video on demand stream
 - lf=N**
Note: if this parameter is omitted then it is assumed to be video on demand

Video Tag parameters (Ad specific)

- One parameter in addition to those outlined previously is required for Ads
 1. **C3 (Ad indicator)**
 - If this parameter is omitted then the stream will be assumed to be content

c3=st,a

Note: if this parameter is omitted entirely then it is assumed that the video is content, not an Ad.

Formatting the video tag call

- Example content beacon:

- <http://secure-us.imrworldwide.com/cgi-bin/m?ci=us-123456&cg=News%20Summary&tl=dav0-Video%20Content%20Name&c6=vc,b01&lp=3,582,6&lf=N&cc=1&rnd=91423477>

Note that cg and tl are properly encoded

- Example ad beacon:

- <http://secure-us.imrworldwide.com/cgi-bin/m?ci=us-123456&cg=Video%20Ad&tl=dav0-Video%20Ad%20Name&c6=vc,b01&cc=1&rnd=91423477&c3=st,a>

Note the inclusion of the c3 variable.

Note that cg and tl are properly encoded

- See slide 8 for an example of how to generate the above beacon using JavaScript
- See slide 9 for an example of how to generate the above beacon using Flash AS2
- See slide 10 and onwards for an example of how to generate the above beacon using Flash AS3

Tips and tricks

- Whenever possible, declare variables globally.
- **Be sure to tag all video players on your site.** This will ensure you receive credit for all of your video traffic.
- Work with Nielsen to test your calls in a development or staging environment first- this will help you format the calls properly before pushing live to production.
- If you have an upcoming video player change or upgrade- please notify Nielsen in advance to avoid any gaps in your numbers.

JavaScript Example

The following sample code may help with implementing the VideoCensus tag in JavaScript:

```
var davImg = new Image();
function _scdav(_sctitle,_scgroup)
{
//      _sctitle = Title of the Stream
//      _scgroup = Client Defined Variable
var ScImgSrc;
var ScRandom = Math.ceil(Math.random()*1000000000);
    ScImgSrc = 'http://secure-us.imrworldwide.com/cgi-bin/m?ci=us-123456';
    //ScImgSrc += '&cg=' + escape(_scgroup);      //Program/Section Name
    ScImgSrc += '&tl=dav0-' + escape(_sctitle);    //Title of the Stream
    //ScImgSrc += '&c3=st,a' + escape('StreamType'); //Use only if the video stream
is an advertisement
    ScImgSrc += '&c6=vc,b01' + escape("");      //VideoCensus ID - specified by
Nielsen - varies per entity
    ScImgSrc += '&cc=1';//Cookie Check (Always on)
    ScImgSrc += '&rnd=' + ScRandom;

    davImg = "";
    davImg = new Image();
    davImg.src = ScImgSrc;
}
```

Flash AS2 Example

This integration is performed by including a call to our servers within the ActionScript of a Flash movie each time content changes.

```
// START Nielsen Video Measurement Instrumentation
// COPYRIGHT 2007 The Nielsen Company
// Start - Global Variables
scCI="us-123456";
// End - Global Variables
// Page impression Code
scCG="Sports%20Video";
scTL= "dav0-" + escape("NFL Draft");
scC6="vc,b02";
loadMovieNum("http://secure-us.imrworldwide.com/cgi-bin/m?ci=" + scCI + "&cg=" + scCG + "&tl=" + scTL
+ "&c6=" + scC6 + "&cc=1" + "&rnd=" + Math.ceil(Math.random()*100000000), 100);
// END Nielsen Video Measurement Instrumentation
```

The loadMovieNum section of the code executes a call to download a 1x1 pixel from secure-us.imrworldwide.com with all the appended values necessary for tracking.

You will need to execute a call to our servers for each new stream request.

If you only stream content on User Demand, you can also implement the call within the on(release) {} section of a movie. Here is an example:

```
on(release) {

    scCI="us-123456";
    scCG="Sports%20Video";
    scTL= "dav0-" + escape("NFL Draft");
    scC6="vc,b02";
    loadMovieNum("http://secure-us.imrworldwide.com/cgi-bin/m?ci=" + scCI + "&cg=" + scCG + "&tl=" + scTL
+ "&c6=" + scC6 + "&cc=1" + "&rnd=" + Math.ceil(Math.random()*100000000), 100);
}
```

Flash AS3 Example (pg 1 of 4)

This integration is performed by including a call to our servers within the ActionScript of a Flash movie each time content changes. This example extends over the next five slides.

```
import flash.external.ExternalInterface;

function _scOnStreamStart(_scCl,_scCG,_scTL,_scC6,_scSD,_scOU)
{
    var screquest:URLRequest=new URLRequest("http://secure-us.imrworldwide.com/cgi-bin/m?ci=" +
    _scCl+ "&cg=" + _scCG + "&tl=dav0-" + _scTL + "&sd=" + _scSD + "&c6=vc," + _scC6 + "&ou=" +
    _scOU + "&rnd=" + Math.ceil(Math.random()*100000000));
    var myloader:Loader = new Loader();
    myloader.load(screquest);
}

function _scOnStreamPing(_scCl,_scCG,_scTL,_scC6,_scDU)
{
    var screquest:URLRequest=new URLRequest("http://secure-us.imrworldwide.com/cgi-bin/m?ci=" +
    _scCl+ "&cg=" + _scCG + "&tl=dav1-" + _scTL + "&du=" + _scDU + "&c6=vc," + _scC6 + "&rnd=" +
    Math.ceil(Math.random()*100000000));
    var myloader:Loader = new Loader();
    myloader.load(screquest);
}

function _scOnStreamEnd(_scCl,_scCG,_scTL,_scC6,_scDU)
{
    var screquest:URLRequest=new URLRequest("http://secure-us.imrworldwide.com/cgi-bin/m?ci=" +
    _scCl+ "&cg=" + _scCG + "&tl=dav2-" + _scTL + "&du=" + _scDU + "&c6=vc," + _scC6 + "&rnd=" +
    Math.ceil(Math.random()*100000000));
    var myloader:Loader = new Loader();
    myloader.load(screquest);
}
```

Flash AS3 Example (pg 2 of 4)

Flash example continued...

```
function _scPlayVideo(scVideoName,_scCl,_scCG,_scTL,_scC6) {
    var myVideo:NetConnection=new NetConnection();
    myVideo.connect(null);
    var newStream:NetStream = new NetStream(myVideo);
    videoholder.attachNetStream(newStream);
    _scSD = Math.ceil(newStream.duration);
    newStream.play(scVideoName);
    _scOnStreamStart(_scCl,_scCG,_scTL,_scC6,_scSD,_scOU);
    scTextBox.text="Stream Start Record Sent. Playing" + scVideoName;

    // The following code tells Flash to ignore any asynchronous errors
    newStream.addEventListener(AsyncErrorEvent.ASYNC_ERROR, asyncErrorHandler);
    function asyncErrorHandler(myevent:AsyncErrorEvent):void{
        //ignore error
    }
}

newStream.addEventListener(NetStatusEvent.NET_STATUS,statusHandler);
function statusHandler(myevent:NetStatusEvent):void {
    if (myevent.info.code == "NetStream.Play.Stop") {
        _scDU = Math.ceil(newStream.time);
        _scOnStreamEnd(_scCl,_scCG,_scTL,_scC6,_scDU);
        scTextBox.text="Video Stopped";
        clearInterval(intervallID);
    }
    if (myevent.info.code == "NetStream.Play.Start") {
        intervallID = setInterval( doPing,5000 );
    }
}

function doPing():void {
    _scDU = Math.ceil(newStream.time);
    _scOnStreamPing(_scCl,_scCG,_scTL,_scC6,_scDU);
}
```

Flash AS3 Example (pg 3 of 4)

Flash example continued...

```
function button1Click (myeven:MouseEvent):void {
    clearInterval(intervalID);
    var _scCI = "us-netratings";
    var _scCG = "0";
    var _scC6 = "c03";
    var _scTL=escape("NielsenOnline Video 1");
    _scPlayVideo("NOL2.flv",_scCI,_scCG,_scTL,_scC6);
    scTextBox.text="Playing Video NOL2.flv";
};

function button2Click (myeven:MouseEvent):void {
    clearInterval(intervalID);
    var _scCI = "us-netratings";
    var _scCG = "0";
    var _scC6 = "c03";
    var _scTL=escape("NielsenOnline Video 2");
    _scPlayVideo("NOL3.flv",_scCI,_scCG,_scTL,_scC6);
    scTextBox.text="Playing Video NOL3.flv";
};

button1_btn.addEventListener (MouseEvent.CLICK, button1Click);
button2_btn.addEventListener (MouseEvent.CLICK, button2Click);
```

Flash AS3 Example (pg 4 of 4)

Flash example continued...

```
//Set Variables and Start Video
var nsRef:NetStream;
var intervalID:Number;
var _scDU:Number;
var _scSD: Number;
var _scOU:String = ExternalInterface.call('window.location.href.toString');
var myloader:Loader = new Loader();
var _scCI = "us-netratings";
var _scCG = "0";
var _scC6 = "c03";
var _scTL=escape("First Stream");
_scPlayVideo("NOL1.flv",_scCI,_scCG,_scTL,_scC6);
scTextBox.text="Playing Introduction Video";
```


iOS Support

The following sample Objective C code may help with implementing the VideoCensus tag in xcode:

```
double myf=((double)rand()/RAND_MAX)*1000000000;
NSNumber *fl=[NSNumber numberWithInt:myf];
NSNumberFormatter *f=[[NSNumberFormatter alloc] init] autorelease];
[f setGeneratesDecimalNumbers:NO];
NSString *stRandom = [f stringWithNumber:fl];
NSString *title=@"this is a test video";
NSLog(@"My encoded video title: %@", [title
stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding]);

NSString * encodedTitle =
(NSString*)CFURLCreateStringByAddingPercentEscapes(NULL,(CFStringRef)title,NULL,(CFStringRef)
@"!*()*;:@&=+$/?%#[]",kCFStringEncodingUTF8 );

NSString *stURL =[NSString stringWithFormat:@"http://secure-us.imrworldwide.com/cgi-
bin/m?ci=%@&cg=%@&tl=dav0-%@&c6=%@&cc=1&rnd=%@",@"us-502202",
@"testmovie",encodedTitle, @"vc,c06",stRandom];

NSLog(@"%@",stURL);
NSURL *url = [NSURL URLWithString:stURL];
NSString *content=[NSString stringWithContentsOfURL:url];
NSLog(@"content of the url %@",content);
```

Questions?

Please contact the VideoCensus tech team via
onlinevideo@nielsen.com



<http://www.nielsen-online.com/>