comscore

Pageview Candidate

Implementation Guide

document version: 2.2.0; released on September 7, 2020

for further information, please contact:
Comscore
Tag Support
+1 866 276 6972

Tagging

A *tag* is a piece of scripting or markup that is placed on a website or another web based content asset. Tags are sometimes referred to as tracking pixels or beacons. They are used to measure the consumption of digital content by an end-user. *Tagging* is the process of adding a Comscore tag — also called *measurement code* or *SDK*[1] — to your digital content. Each time a tagged piece of content — e.g., a web page, a video stream or an application view in a (mobile) application — is used by an end-user, the tag sends data via an HTTP request to Comscore's data collection servers.

Pageview

pageview is defined as a page that has been fully loaded into a browser, as a result of a user initiated action. Generally a page consists of an HTML file plus all of the images/objects requested by the HTML code.

Business rules are applied to URLs that are deemed non-legitimate:

1. All pages in a redirect, with the exception of the destination page (have a return code of 302)
2. URLs stopped by a user or partially downloaded pages
3. URLs with the following extensions: *.GIF*, *.JPG*, *.VBS*, *.BMP*, *.ZIP*, *.RAM*, *.MOV*, *.MP3*, *.MP2*, *.AVI*, *.MPG*, *.WAV*, *.PDF*, *.PNG*, *.SWF*
4. Ad Banners and Pop-Up or Pop-Under pages
5. Refreshed and auto-refreshed web pages
6. Web crawlers, spiders, bots or other automated engines

This definition supports almost all data requests on the Web and allows for the most comparable and accurate reports in the market.

Dynamic Content

Advanced web techniques such as AJAX and JSON allow for a differentiated delivery mechanism where the standard user-initiated HTML content is not always provided within the request. In many instances an XML Content URL is executed to deliver the user requested content.

Additionally, for dynamically-loaded web content the contents of a page may change substantially without a page load occurring. This behavior is not captured by default with standard Website Tag implementations.

In this case, the user-requested content needs to be '**promoted**' to HTML calls within Comscore systems. This 'promotion' ensures that these '**candidates**' are as actual pageviews. We also need to ensure that a tag is fired for each of these dynamic page loads.

---

(1) SDK is an acronym for Software Development Kit which is typically a set of tools and documentation for the creation of software.

Pageview Candidate Solution Background

Comscore's unified reporting uses a comparison of two different data collection methodologies (panel + census):

- **Census**: Tags specifically send selected data values to Comscore servers
- **Panel**: This traffic collection occurs on the client-side on individual machines used by members of our worldwide panel

The *Pageview Candidate* implementation allows us to more accurately measure a client's panel-based visitation and engagement by allowing Comscore to:

- Identify non-HTML traffic (JSON, JavaScript, XML, etc.) and
- Promote it to an HTML pageview credit as normal internet traffic through our data processing.

Our census tag provides us a lot of information, but not enough to credit these cases when a client may be the beneficiary of extra pageviews on the panel side. In order for our systems to credit such dynamic content, we need to have an additional implementation step in conjunction with the traditional tag.

Qualification Criteria for Pageview Candidate

The Comscore Dictionary team will review dynamic user experiences and advise if the Pageview Candidate solution is an appropriate mechanism for measurement.

Generally, the following serve as qualifying candidates:

1. Picture Galleries/Slideshows
2. Infinite Scroll Pages
3. Single-Page Web Applications

Note:

1. An XML content URL call that is made in tandem with an HTML call **does not count as a Pageview Candidate** as this results in double-counting the same content. An example would be a picture gallery as soon as one loads a web page.
2. Auto-advancing slideshows or picture galleries are **not** eligible for pageview credit. Credit may only be received in cases where the slideshow or gallery is advanced by user action.
3. Infinite scroll pages may receive pageview credit when the user scrolls down a page and new content is loaded, but may **not** receive credit if the user scrolls up through previously-loaded content.
4. A video startup/ pop-up screen is **not** permitted for Pageview Candidate tagging.

The basic qualifying criteria are:

1. More than 50% of the content should change and the content that changes should be page content (not video).
2. The page is given a credit only when the dynamic content is loaded as a result of user-initiated action.

Implementation Checklist

Standard Website Tag implementation

Put the Comscore page tag (shown below) in a place which is static for all the pages (your header is a good example), replacing `CLIENT_ID` by your Comscore client ID (found on direct.comscore.com).

```
<!-- Begin Comscore Tag -->
<script>
    var _comscore = _comscore || [];
    _comscore.push({ c1: "2", c2: "CLIENT_ID" });
    (function() {
        var s = document.createElement("script"), el = document.getElementsByTagName("script")[0]; s.async = true;
        s.src = "https://sb.scorecardresearch.com/cs/CLIENT_ID/beacon.js";
        el.parentNode.insertBefore(s, el);
    })();
</script>
<noscript>
    <img src="https://sb.scorecardresearch.com/p?c1=2&c2=CLIENT_ID&cv=3.6&cj=1">
</noscript>
<!-- End Comscore Tag -->
```

Tag for Dynamic Content Loads

To receive census credit for dynamic content, when the page loads dynamic content that is eligible to receive pageview credit call the following API:

```
self.COMSCORE && COMSCORE.beacon({c1: "2", c2: "CLIENT_ID"}));
```

This API call will fire an additional page tag for the new content. This requires that the Website Tag is implemented as described in *Standard Website Tag implementation on this page*.

Implement `pageview_candidate` Keyword

To receive panel credit for dynamic content, when the page loads dynamic content that is eligible to receive pageview credit an HTTP request must be observed that returns a text response containing the keyword `pageview_candidate`. This can be either a part of the dynamic content itself, or it can be a standalone dummy call.

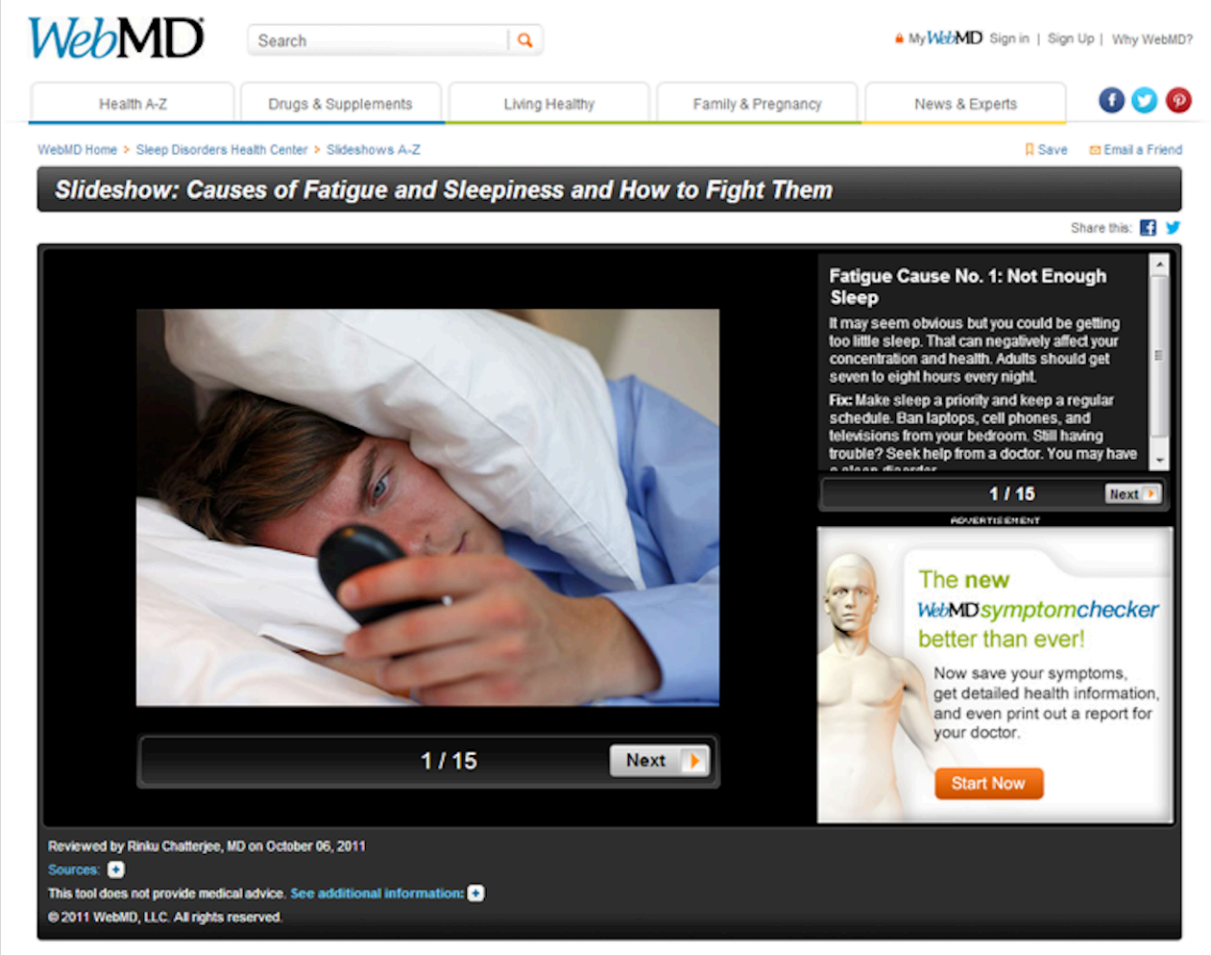
Note:

- The URL that returns the `pageview_candidate` keyword must have the same top-level domain as the page that is displaying the dynamic content.
- The URL that returns the `pageview_candidate` keyword must NOT be cached client-side (i.e., in the browser).

This data is picked up by our panel. This step is part of our methodology to ensure that none of the traffic from your property is lost and we report accurate numbers in your reporting.

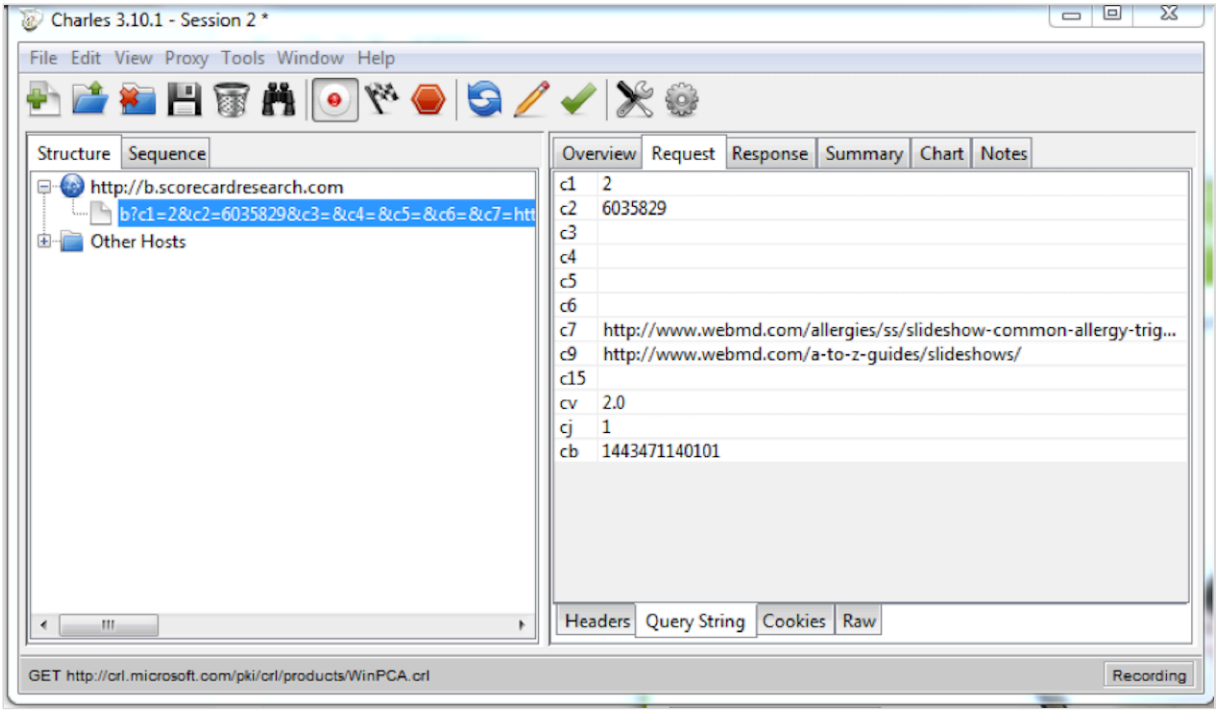
Examples

For a gallery below the image can change using user initiated events and fifty percent or more of the content on the page is new/ refreshed based on the user's request.

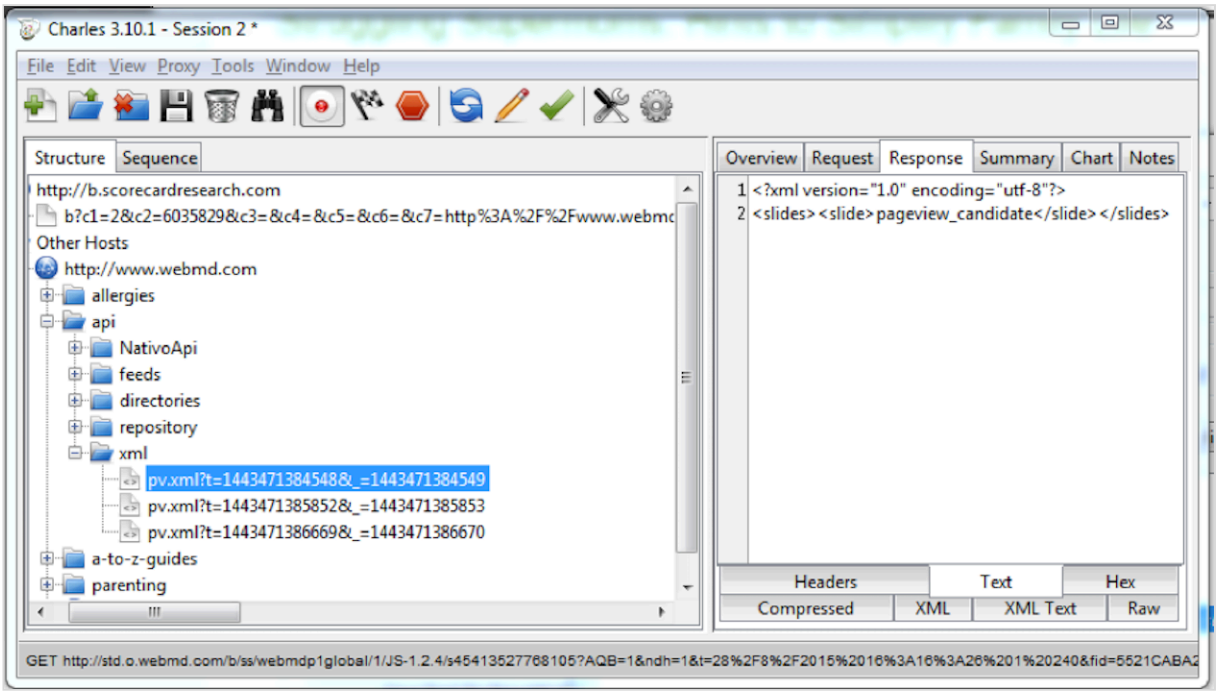


> Please note that *more than 50%* of the content should change.

There should be an HTTP call made to *b.scorecardresearch.com* with query string parameter `c1=2`:



In additional to the above call, there should be a call made to your server where the response has `pageview_candidate` in it.



Response formats

The `pageview_candidate` string can be present in the response in different formats.

JSON

```
{
    key: "value",
    key1: "value",
    arr: [1,2,3,4,5,6,7,8,9],
    comscore: "pageview_candidate"
}
```

XML

```
<!-- pageview_candidate -->
<data>
    <key>value</key>
    <key>value</key>
        <array>
            <value>1</value>
            <value>2</value>
            <value>3</value>
            <value>4</value>
            <value>5</value>
            <value>6</value>
            <value>7</value>
            <value>8</value>
            <value>9</value>
        </array>
</data>
```

Plain Text

```
pageview_candidate
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
```