Media Metrix Reporting

Publisher Web Page Impression Tag

document version: 2.9.2; released on May 16, 2024

Contents

> **ℹ** **Any transmission of data to Comscore is subject to the terms of the Data Sharing Agreement which can be accessed on** *⏏ Comscore Direct (https://direct.comscore.com/Signup/Signup.aspx)* **. Your transmission of data to Comscore constitutes acceptance of the Data Sharing Agreement without further action.**

1 Introduction

In our continuous efforts to provide the market with highest quality audience measurement data, Comscore Media Metrix uses the *Unified Digital Measurement* methodology.

1.1 Unified Methodology

Unified Digital Measurement is a best-of-breed approach that puts the consumer — the human, not the machine — at the center of Comscore measurement and relies on Panel data as well as Census or Server data. Data from the Comscore panel provides a 360° view of the consumer including demographics, cross-visitation, etc. while Census data from tags provides overall, site-specific usage activity. Through the advent of measuring consumption of content and ads, Comscore has been measuring more entities and campaigns as consumer's media consumption grows across channels, devices, and environments. Grounded in rigorous methodology and by tying this data to panel observations, Comscore uses these assets together in what Comscore calls Unified Digital Measurement (UDM).

We're ensuring our new iteration — UDM 2.0 — will stand the test of time. The transition to a cookie-less world in any way that works for publishers or brands, whether that's via website or video streaming tags, SDKs for mobile and OTT applications, or via server-to-server data sharing. Comscore works with you on "where your data resides" to ensure accurate and robust reporting. A separate document about UDM 2.0 is available from your Comscore account team or implementation support team.

UDM 2.0 combines first-party data from digital publishers and TV networks in a privacy-preserving manner to ensure audiences are represented with the same granularity and precision as you have come to expect.

This document covers the specifications to implement Comscore Unified Digital Measurement 2.0 tags on a publisher web site.

1.2 What are tags?

A tag, sometimes referred to as a web bug or tracking pixel, is an element included in a web page or other web based content asset and is used to track the occurrences of consumption "events" such as web page impressions. Each time the content asset — for example a publisher's web page — is served upon the end-consumer, the tag call is made and that consumption event is logged on the data collection server. These events include visiting a web site or page within a website, watching a video, delivery of an advertisement or delivery of distributed content. Information such as the IP address of the computer that consumed the content, a timestamp of when the tag was retrieved, the type of browser, cookie values, etc. can be passed along with the tag call.

Additional values are used to capture more granular information about the tag call including where it was viewed, who delivered it, which advertising package, campaign or content channel, etc. it belonged to. These values are typically included as part of most tag calls through query string parameters. When client-side tags are combined with insights provided by the Comscore panel, they deliver significant additional value to internet audience measurement, bringing breadth, flexibility and enhanced granularity to the rich consumer insights and behavioral data of the Comscore panel.

1.3 UDM 2.0 tag specification

A key element for UDM 2.0 is that *third-party* (*3P*) identifiers of the consumer — for example a third-party cookie — may not be available. Reach and demographic reports in Comscore's audience reporting services like Media Metrix and Video Metrix will use information from panels, enhanced with *first-party* (*1P*) publisher-specific data of the consumer provided by the publisher. 1P Data can consist of:

- 1P identifier data (e.g., a first-party cookie set by the Comscore Publisher Tag, an obfuscated value of a first-party publisher cookie value or a login identifier)
- Optional demographics data — age group and gender — provided these are available

Please refer to 🔗 *Appendix C: How to add 1P data to web page impression tags on page 11* for details on providing 1P data in tags.

1.4 What are the options for data collection?

There are two options for a publisher to provide data:

1. Use the same tags you are using today, include your 1P identifier, and insert (hashed or obfuscated) additional consumer information
2. Server-to-server transfer of web/app/streaming analytics data, including 1P identifier, and (hashed or obfuscated) additional consumer information.

This document covers the use of tags for web page impression tagging. For an overview of all options please refer to the aforementioned UDM 2.0 document, available from your Comscore account team or implementation support team.

1.5 Benefits of tagging

- Census-adjusted metrics (Home + Work) for your site reported in all Media Metrix reports
- Greater consistency with internal and server-side metrics
- Profiling of your channels against wide area of planning criteria
    - Demos, Lifestyle, Online Behavior, Product Ownership, third-party segmentation schemes
- Coverage of smaller sites/channels and niche audiences

2 Implementing the Publisher Tag

The Comscore Publisher Tag uses asynchronous loading of its JavaScript file so that loading and processing of the web page is unaffected by the tag. A `<img>` HTML element is present for cases where JavaScript cannot be executed. The tag uses the parameters listed below, which are required to ensure appropriate data categorization.

Comscore Publisher Tag parameters

Parameter	Required or Optional	Description	Explanation	Example value
c1	**Required**	Tag Type	Pre-populated with fixed value 2	2
c2	**Required**	Comscore Publisher ID	A Comscore-provided number with at least 7 digits that is unique for each publisher[1]. This value is also used in the URL of the *beacon.js* file.	1234567

Comscore combines 1P data from digital publishers and TV networks in a privacy-preserving manner to ensure your audiences are represented with the same granularity and precision as you have come to expect. The tag tries to set a third-party cookie regardless of whether or not JavaScript can be executed. The JavaScript code of the tag has opt-in functionality which will let it additionally try to set a publisher-specific first-party cookie with the name `_scor_uid`.

While the first-party cookie functionality is disabled by default, **it is strongly advised to enable the first-party cookie functionality** by using the `enableFirstPartyCookie: true` setting, as is shown in the code example below. Optionally, you can add additional information about the consumer to enhance your results. Please refer to 🔗 *Appendix C: How to add 1P data to web page impression tags on page 11* for details on providing 1P data.

To implement the tag, copy its HTML code into your web page content and provide the appropriate parameter values. **Here the example Comscore Publisher ID 1234567 is used in multiple locations, which should all be replaced with the value you are assigned.**

```
1.   <!-- Begin Comscore Tag -->
2.   <script>
3.       var _comscore = _comscore || [];
4.       _comscore.push({
5.           c1: "2", c2: "1234567",
6.           options: {
7.               enableFirstPartyCookie: true
8.           }
9.       });
10.
11.      (function() {
12.          var s = document.createElement("script"), el = document.getElementsByTagName("script")[0]; s.async = true;
13.          s.src = "https://sb.scorecardresearch.com/cs/1234567/beacon.js";
14.          el.parentNode.insertBefore(s, el);
15.      })();
16.  </script>
17.  <noscript>
18.      <img src="https://sb.scorecardresearch.com/p?c1=2&c2=1234567&cv=4.4.0&cj=1">
19.  </noscript>
20.  <!-- End Comscore Tag -->
```

---

(1) The Publisher ID is also known as the *Client ID* or *c2 value*.

This code example also includes the `options` property with the `enableFirstPartyCookie` setting set to `true` to enable the first-party cookie functionality of the Publisher Tag.

> **About first-party cookie lists...**
>
> If you enable the first-party cookies and you maintain a list of first-party cookies — for example as part of your privacy statement — then **please make sure to add the `_scor_uid` cookie** to your list. Its purpose is Audience Measurement. Please be aware the Comscore Publisher Tag uses this cookie in accordance with policies surrounding user consent signals as described in *Appendix B: User consent tagging on page 8* .

In addition to the mentioned parameters, the following parameters are automatically collected by the JavaScript code of the tag:

Automatically collected parameters

Parameter	Description
`c7`	Full Page URL
`c8`	Page Title
`c9`	Referring URL
`cs_fpcu`	First-party `_scor_uid` cookie value. Only collected if the first-party cookie functionality is enabled and tag can read a previously set first-party cookie.

The Comscore Publisher Tag will continue to collect data for the same purposes as approved with your current implementation.

> **About the locations where tags can be implemented…**
>
> Publishers should refrain from implementing the tag inside ad calls and/or iframes as this may result in the `c7` parameter — required for correct attribution of traffic — to not populate correctly.
>
> **Please also do not add the tags to pop-up or pop-under locations that may contain page content.**

Appendix A: Additional instructions for special use cases

Content can be delivered in many ways. Some use cases require additional implementation steps. If these additional instructions apply to your content then please consult with your Comscore account team.

Dynamically-generated web pages

In the case of dynamic content such as when using AJAX, infinite scroll, HTML5 History `pushState` and `popState` API or similar technologies, publishers should follow the following steps:

1. Consult with your Comscore account team to ensure that the dynamic content meets Comscore's rules for measurement and will be measured on the Comscore panel.
2. Implement the tag as instructed in this document for the initial page load.
3. When the page content changes for more than 50% due to user interaction, call the JavaScript `COMSCORE.beacon()` function, where the appropriate parameters are passed as argument:

```
81.  <script>
82.      (self.COMSCORE && COMSCORE.beacon({
83.          c1: "2", c2: "1234567",
84.          options: {
85.              enableFirstPartyCookie: true
86.          }
87.      }));
88.  </script>
```

This is the desired method for implementing the tag with dynamic content. The above code example checks for the existence of the `COMSCORE` reference before executing the function call, which will be available after the tag was executed for the initial page load.

Often times for panel measurement it is required to perform additional implementation steps for *Pageview Candidate* measurement. Please consult with your Comscore account team to ensure these additional steps are executed where applicable.

Custom implementation without JavaScript

Sometimes the automatically collected URL in the `c7` parameter does not represent the URL associated with the content the user is viewing. In other cases the content might not be allowed to contain or execute JavaScript code. For these use cases publishers should use **only** the `<img>` HTML code of the tag and populate the parameters with appropriate values, **including the** `c4` **parameter**. The `c4` parameter **must be included** and populated with the full URL associated with the content being consumed by the user (which is not necessarily the URL in the address bar of the web browser):

```
<img src="https://sb.scorecardresearch.com/p?c1=2&c2=1234567&c4=http%3A%2F%2Fwww.website.com%2Fmypage.aspx&cj=1">
```

> ℹ️ **About URL encoding of query string parameter values in the URL of the** `src` **attribute of** `<img>` **HTML element…**
>
> Values of query string parameters in the URL of the `src` must be properly URL encoded as illustrated in the example shown above. This is particularly important when the `c4` parameter is provided with a URL as its value.

Appendix B: User consent tagging

Applicable privacy and data protection laws and regulations may require companies to capture and/or document a user's consent for measurement. For example, the European Union's General Data Protection Regulation ("GDPR") and the Privacy and Electronic Communication Directive 2002/58/EC and the California Consumer Privacy Act ("CCPA") have requirements regarding capturing user consent or providing consumers the ability to opt-out of the sale of personal information, where appropriate. Please note that the implications of applicable privacy and data protection laws and regulations may vary and are best evaluated by each individual business.

Using a Consent Management Platform

If you are using a Consent Management Platform (CMP) which implements iAB Transparency and Consent Framework (TCF) version 2.0 then Comscore Publisher Tag integrates with the CMP to automatically collect user consent. No additional steps are necessary to enable this integration, other than to make sure the Comscore Publisher Tag is in the web page where it can access the CMP as per the TCF 2.0 technical specification.

Sometimes JavaScript code execution might not be possible. Publishers are expected to provide two extra parameters in the `<img>` HTML code of the tag, but only if they can retrieve and incorporate Transparency and Consent data. The two parameters are `gdpr` and `gdpr_consent`, both used according to the Consent Management Platform (CMP) specification.

> **For populating `gdpr_consent` it is important to note Comscore's TCF vendor id is 77.**

If JavaScript code execution is not possible and Transparency and Consent data cannot be retrieved, then please implement manual consent communication in the `<img>` HTML code of the tag as explained in the next section.

Manually communicating consent

If you are not using a Consent Management Platform then you need to manually communicate user consent by adding parameter `cs_ucfr` to the collected data. The required values for this user consent parameter are:

Label `cs_ucfr` values for communicating user consent

Value	Interpretation	Usage
0	User has not given consent or has opted out	Use this value to indicate the user 1. has been asked for consent where the user did not give consent, or 2. enabled the option to opt out (e.g., opt out of the sale of personal information)
1	User has given consent	Use this value to indicate the user has been asked for consent where the user has given consent to collect data for measurement
	User has not taken an action	Use an empty string value (i.e., blank) to indicate the user has not taken an action

> **About including parameter `cs_ucfr` when not communicating user consent…**
>
> If you do not communicate consent for a user, then **do not populate** label `cs_ucfr`.

A publisher must add parameter `cs_ucfr` with an appropriate value to the web page impression tagging by adding the parameter and its value to the HTML and JavaScript code. With this change of the HTML and JavaScript code the publisher **should not change any other collected data values**.

For example, assuming the user has given consent, the aforementioned tag code would be changed into:

```
1.   <!-- Begin Comscore Tag -->
2.   <script>
3.       var _comscore = _comscore || [];
4.       _comscore.push({
5.           c1: "2", c2: "1234567", cs_ucfr: "1",
6.           options: {
7.               enableFirstPartyCookie: true
8.           }
9.       });
10.
11.      (function() {
12.          var s = document.createElement("script"), el = document.getElementsByTagName("script")[0]; s.async = true;
13.          s.src = "https://sb.scorecardresearch.com/cs/1234567/beacon.js";
14.          el.parentNode.insertBefore(s, el);
15.      })();
16.  </script>
17.  <noscript>
18.      <img src="https://sb.scorecardresearch.com/p?c1=2&c2=1234567&cs_ucfr=1&cv=4.4.0&cj=1">
19.  </noscript>
20.  <!-- End Comscore Tag -->
```

Likewise, assuming the user has given consent, the aforementioned code statement for tagging dynamically generated web pages would be changed into:

```
81.  <script>
82.      (self.COMSCORE && COMSCORE.beacon({
83.          c1: "2", c2: "1234567", cs_ucfr: "1",
84.          options: {
85.              enableFirstPartyCookie: true
86.          }
87.      }));
88.  </script>
```

Bypass user consent requirement for first-party cookie

When the first-party cookie functionality is enabled the Publisher Tag inspects available user consent signals to determine setting the cookie is allowed.

Based upon the self-determination of a website/publisher that no applicable jurisdiction requires it to obtain consent for cookies, Comscore offers the option for the implementer to indicate that the website/publisher is exempt from any user consent requirements. If this applies, the implementer can use the `bypassUserConsentRequirementFor1PCookie: true` setting:

```
1.   <!-- Begin Comscore Tag -->
2.   <script>
3.       var _comscore = _comscore || [];
4.       _comscore.push({
5.           c1: "2", c2: "1234567",
6.           options: {
7.               enableFirstPartyCookie: true,
8.               bypassUserConsentRequirementFor1PCookie: true
9.           }
10.      });
11.
12.      (function() {
13.          var s = document.createElement("script"), el = document.getElementsByTagName("script")[0]; s.async = true;
14.          s.src = "https://sb.scorecardresearch.com/cs/1234567/beacon.js";
15.          el.parentNode.insertBefore(s, el);
16.      })();
17.  </script>
18.  <noscript>
19.      <img src="https://sb.scorecardresearch.com/p?c1=2&c2=1234567&cv=4.4.0&cj=1">
20.  </noscript>
21.  <!-- End Comscore Tag -->
```

Appendix C: How to add 1P data to web page impression tags

The following optional parameters can be added to provide 1P identifier data and additional demographics data about the consumer:

Tag parameters for 1P data

Parameter	Required or Optional	Description	Explanation	Example value
cs_fpid	Mandatory[(2)]	1P identifier	Contains the pseudonymized 1P identifier value, which could be either a user_id, login (preferred), OpenID or a first-party cookie.	1570113661206_6059410249
cs_fpit	Optional	Type of identifier in cs_fpid	Specifies the type of identifier ▪ li: logged-in ID ▪ lo: logged-out ID ▪ o: OpenID ▪ c: first-party cookie If cs_fpit is not included alongside cs_fpid then Comscore will assume the identifier is a first-party cookie.	c
cs_fpdm	Optional	1P demographics data	Contains an obfuscated value of the demographics data which belong to the 1P identifier. This obfuscation calculation is explained further below this table.	39642313001
cs_fpdt	Optional	Type of 1P demographics data in cs_fpdm	Indicates the origin of the demographics values. Accepted values: ▪ 01: collected by publisher ▪ 02: collected through / purchased from third party ▪ 03: mixed sources or modelled ▪ 99: unknown origin	01

> ⚠️ **Please ensure that all four parameters are provided together.** If you cannot populate a suitable value for any of the parameters — e.g., when demographics data is not available — then please use value *null instead.
>
> **Demographics data should not be collected for children.** If the consumer is a child (ages 0 - 17) then please use value *null for cs_fpdm.

The 1P identifier is expected to stick to the same consumer in the same browser where possible, or even across browsers if a user- or login identifier is used. 1P demographics data is calculated as 19991999999 added to to the concatenation of birth date, age group and gender, where birth date has the format yyyyymmdd, age group has the format xx and gender has the format z. In other words: yyyyymmddxxz + 19991999999.

Age groups and gender are shown in the following tables. **The age groups in the range 0 - 17 are not listed because demographics data should not be collected for children.**

Age groups identifiers

Identifier	Age group
00	00
06	18 - 20
07	21 - 24
08	25 - 34
09	35 - 44
10	45 - 54
11	55 - 64
12	65+

Gender identifiers

Identifier	Age group
0	Unknown
1	Male
2	Female
3	Unspecified / Other

Age group and birth date are complementary, but the preference is to collect birth date. When birth date is provided, age group can be omitted by using 0 values and vice versa. To illustrate:

Demographics values examples

Demographics of consumer	Demographics value calculation	Description
Female born on March 13, 1965	19650313002 + 19991999999 = 39642313001	The age group uses zeroes because the birth date is known
Male in the age between 35-45	00000000091 + 19991999999 = 19992000090	The birth date uses zeroes because only the age group was available

Further below is a tag code example, assuming the following 1P data values:

- Identifier: 1605266069802_50777152

- Identifier type is first-party cookie: c

- Demographics values: 39642313001

- Demographics collected by publisher: 01

```
1.   <!-- Begin Comscore Tag -->
2.   <script>
3.       var _comscore = _comscore || [];
4.       _comscore.push({
5.           c1: "2", c2: "1234567",
6.           cs_fpid: "1605266069802_50777152", cs_fpit: "c", cs_fpdm: "39642313001", cs_fpdt: "01",
7.           options: {
8.               enableFirstPartyCookie: true
9.           }
10.      });
11.
12.      (function() {
13.          var s = document.createElement("script"), el = document.getElementsByTagName("script")[0]; s.async = true;
14.          s.src = "https://sb.scorecardresearch.com/cs/1234567/beacon.js";
15.          el.parentNode.insertBefore(s, el);
16.      })();
17.  </script>
18.  <noscript>
19.      <img src="https://sb.scorecardresearch.com/p?c1=2&c2=1234567&cs_fpid=1605266069802_50777152&cs_fpit=c
     &cs_fpdm=39642313001&cs_fpdt=01&cv=4.4.0&cj=1">
20.  </noscript>
21.  <!-- End Comscore Tag -->
```

Appendix D: Update an existing Comscore Publisher Tag implementation

To update an existing Comscore Publisher Tag Implementation typically only the URL of the *beacon.js* file needs to be updated. This URL is typically found in one of the JavaScript statements. For example:

```
1.  <!-- Begin comScore Tag -->
2.  <script>
3.      var _comscore = _comscore || [];
4.      _comscore.push({ c1: "2", c2: "1234567" });
5.
6.      (function() {
7.          var s = document.createElement("script"), el = document.getElementsByTagName("script")[0]; s.async = true;
8.          s.src = (document.location.protocol == "https:" ? "https://sb" : "http://b") + ".scorecardresearch.com/
    beacon.js";
9.          el.parentNode.insertBefore(s, el);
10.     })();
11. </script>
12. <noscript>
13.     <img src="https://sb.scorecardresearch.com/p?c1=2&c2=1234567&cv=2.0&cj=1">
14. </noscript>
15. <!-- End comScore Tag -->
```

Please replace the old URL with one using the Comscore Publisher ID and let the URL use secure transmission. Please also enable the first-party cookie functionality feature as advised. With these changes, the example code would be changed like this:

```
1.  <!-- Begin Comscore Tag -->
2.  <script>
3.      var _comscore = _comscore || [];
4.      _comscore.push({
5.          c1: "2", c2: "1234567",
6.          options: {
7.              enableFirstPartyCookie: true
8.          }
9.      });
10.
11.     (function() {
12.         var s = document.createElement("script"), el = document.getElementsByTagName("script")[0]; s.async = true;
13.         s.src = "https://sb.scorecardresearch.com/cs/1234567/beacon.js";
14.         el.parentNode.insertBefore(s, el);
15.     })();
16. </script>
17. <noscript>
18.     <img src="https://sb.scorecardresearch.com/p?c1=2&c2=1234567&cv=4.4.0&cj=1">
19. </noscript>
20. <!-- End Comscore Tag -->
```

In the above examples, the Comscore Publisher ID is 1234567 and is used in the URL of the *beacon.js* file.

If your existing implementation has additional parameters being collected aside from the c1 and c2 shown in the examples above then **please do not change those parameters as part of your update**.

Appendix E: Implementation FAQs

1. **Where should I place the tag?**

   We recommend that the clients place the Comscore tag code as close to the top of the page body as possible.

2. **How can I validate my tag is working as intended?**

   Log into Comscore Direct and click on the "Verify URL" button. In the pop-up, enter in the URL that you desire to test. The Verify URL process will make a request to the page and verify that the tag is physically present on the page. For those sites that embed the Comscore tag in a separate file, the Verify URL feature will not detect the presence of the tag.

   We suggest that you load your web page and verify the tag HTTP request calls using a tool like Fiddler or the developer console of your web browser. The instructions for looking for the appropriate tag calls are as follows:

   1. On initial deployment of the Comscore tag, the user's machine will request the Comscore beacon.js file[3]. The resulting HTTP request URL will have the following structure: `https://sb.scorecardresearch.com/cs/1234567/beacon.js` where instead of `1234567` your Comscore Publisher ID will appear.
   2. The beacon.js code is executed and then provides the instruction the web browser to fire the actual tag call, which will have a HTTP request URL with this structure:
      `https://sb.scorecardresearch.com/b?c1=2&c2=1234567&ns__t=1524476354599&ns_c=UTF-8&cv=4.4.0&c8=My%Page%20Title&c7=http%3A%2F%2Fwww.mywebsite.com%2Fmypage.aspx&c9=`
   3. The HTTP request will have a response code of `204: No Content`.
   4. The important parameters to confirm in the tag call are:
      - `c1=2`
      - `c2=` with your specific Publisher ID as assigned by Comscore
      - `c7=` with the full page URL including query string variables

3. **Will placing tags slow the site increasing page load time?**

   No. Comscore uses a CDN and delivers industry-leading response times. It is expected that the Comscore tag will have a response time of 60-90ms.

4. **Can I place the Comscore tag on secure pages?**

   Yes, the JavaScript code of the Comscore tag uses HTTPS URLs. The URL in the `src` attribute of `<img>` HTML element of the tag also uses an HTTPS URL.

---

(3) Unless the beacon.js file has been requested before and has been cached.