TrustArc

Consent Manager

GTM Integration Guide

(Advanced Service)

Revision History

Document Version	Date (MM-DD-YYYY)	Description
3.3	02-12-2024	Added important note for page refresh
3.4	06-10-2025	Added a JavaScript snippet for page reload functionality in the Verifying a GTM Zero-Tracker Integration section
3.5	01-16-2026	Updated Deployment Strategies, Consent Manager Assets, and GTM Variables and Triggers sections
3.6	04-29-2026	Updated the Tracker Consent section under GTM Variables and Triggers to remove reference to &gtm=1 parameter

About This Document

This guide is designed to provide guidance on configuring your Google Tag Manager (GTM) instance to fire and block tags based on the user-provided consent when performing a Zero-Tracker deployment of TrustArc's Consent Manager (CM) solution.

Target Audience

This document is intended for Implementation and/or Web Development teams.

Disclaimer

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form by any means, electronic or mechanical, for any purpose, without the express written permission of TrustArc Inc. Moreover, this guide is strictly informational in nature and **is not intended to provide legal advice**, as all legal and compliance obligations and interpretations remain the responsibility of users of TrustArc's Consent Manager solution in coordination with their legal counsel.

This guide is **NOT** a substitute for Google's GTM documentation. If you have questions regarding the use, purpose, or other configurations of GTM, please contact your Google Account Manager and/or visit their online documentation for websites.

If you have questions regarding non-GTM related Consent Manager deployment options, please refer to TrustArc's Consent Manager Deployment Guide.

Table of Contents

Overview

The sections outlined in this document provide the instructions to utilize GTM triggers and rules in a **Zero-Tracker** Consent Manager deployment for both **Standard** and **Zero-Tracker Loads**.

Before using this integration guide, please make sure the following items are completed:

- You have identified all tracking tags on your site and their corresponding Bucketing Category.
- All tracking tags corresponding to a non-Required tracker have been moved into GTM.
- You have added your GTM script to all of your site's pages.

Consent Behaviors

The Consent Manager can utilize two different consent behaviors, **Implied Consent** (uninterrupted user navigation on the site) and **Expressed Consent** (interrupted user navigation on the site until the user provides consent), and each country's consent behavior can be set independently.

For **Implied Consent** configured countries, the Consent Manager is available for the user to launch, from either the *Cookie Preferences* button or the *Implied Consent Banner*, but they are not forced to provide consent before interacting with the site. When a user launches the Consent Manager in an **Implied Consent** country they can close the Consent Manager at any time via an *X* or *Cancel* button.

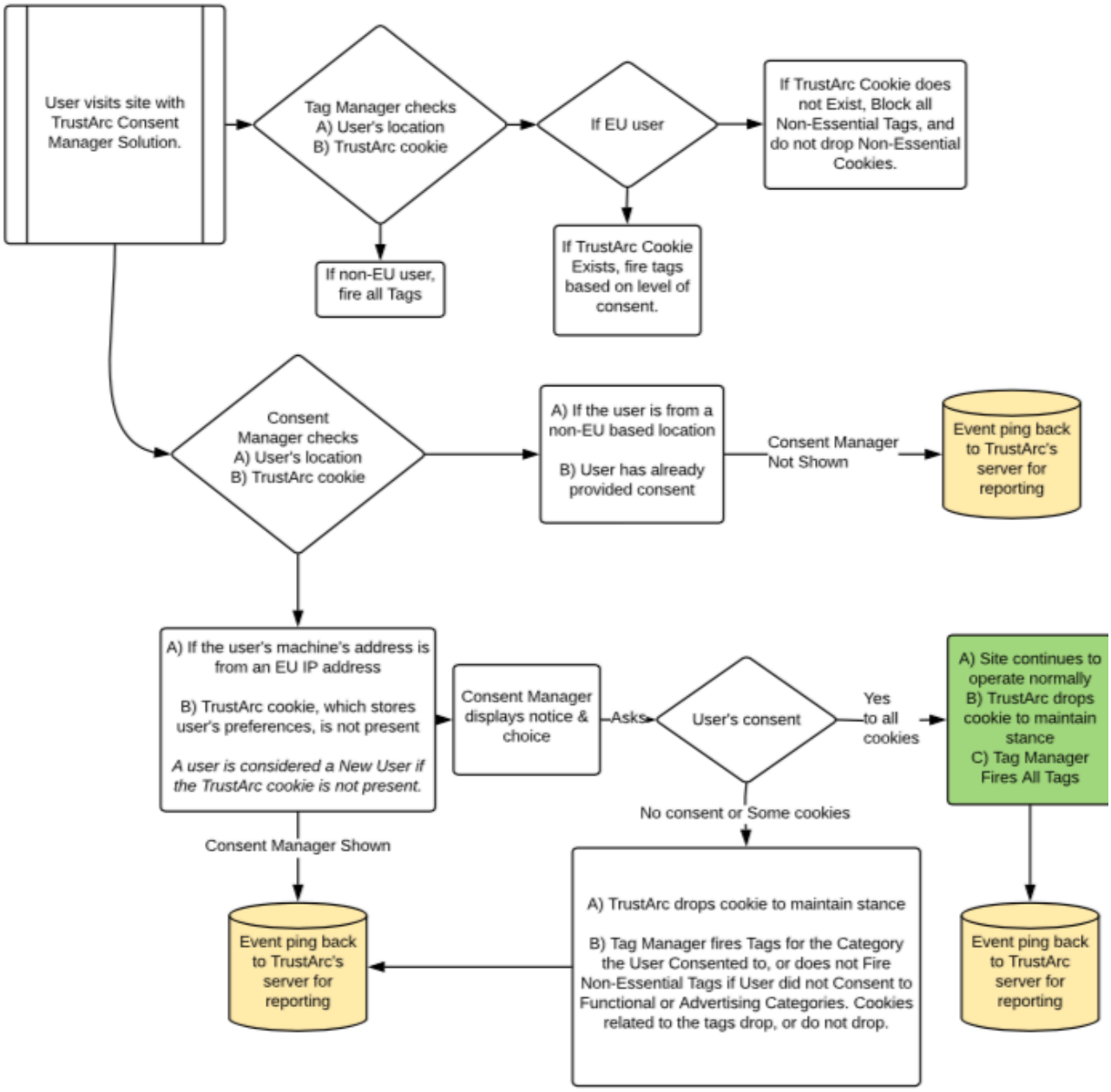
For **Expressed Consent** configured countries, the Consent Manager is launched automatically when a user with no valid consent preference is detected. The user must provide their consent before interacting with the site and the Consent Manager cannot be closed until consent is provided. Once consent has been provided the user can launch the Consent Manager from the *Cookie Preferences* button to review or change their selection but they are not required to provide a new submission.

Deployment Strategies

Independent of the consent behavior, Consent Manager can be deployed on a site in one of two methods - a **Standard** deployment or a **Zero-Tracker** deployment. When the Consent Manager is deployed, there are two different tracker loads which can be used – a **Standard Load (opt-out)** or a **Zero-Tracker Load (opt-in)**. With a **Standard Load**, all trackers are fired until the user *opts out* of tracking. With a **Zero-Tracker** Load, no non-Required trackers are fired until the user *opts in* to tracking. Your consent manager configuration will contain a consent model configuration (opt-in / opt-out) per location to represent each behavior.

User Flow

Below is a decision flow chart based on a new user first encountering a CM which has been integrated with GTM, where GTM is blocking or controlling the firing of tags and the dropping of trackers based on the end user's level of consent in the CM. In the flow chart, non-EU users experience a **Standard Load** while EU users experience a **Zero-Tracker Load**.



Consent Manager Assets

This section outlines the TrustArc specific code which is required to deploy Consent Manager in a GTM integrated **Zero-Tracker** deployment. This code can be deployed either directly to the page code or through GTM.

Important: Please review the important note at the end of the Using GTM to Deploy CM Assets section regarding deploying Consent Manager code via GTM.

Consent Manager Code

You will need to add your instance-specific Consent Manager code, provided by your Technical Account Manager, to each page to be covered by the Consent Manager. An example of this code would be:

```
JavaScript
<div id="consent_blackbar"></div>
<div id="teconsent"></div>
<script async="async"
src="//consent.trustarc.com/notice?domain=client.com&c=teconsent&js=nj&noticeType=bb"
crossorigin=""></script>
```

You will need to place the `consent_blackbar` div on the page location where you would like the consent banner to display - please note that this should be visible to the user when they first view the page so it should be either static at the top of the page or floating at either the top of bottom of the screen. The `teconsent` div should be placed where you would like the *Cookie Preferences* link to display.

Important:

- It takes approximately **500ms** for Consent Manager to identify and provide the necessary information for GTM to correctly block trackers for users who have not previously set their consent. If you are using a **Zero-Tracker Load**, you will need to place the Consent Manager `script` to run as early as possible on the page, preferably as the first script to run in the header, to ensure GTM has the required information to properly block the necessary tags.

Please refer to TrustArc's Consent Manager Deployment Guide for more information.

GTM Event Listener

In order to generate the required GTM events, you must add the following code to every page on which the CM is deployed:

JavaScript

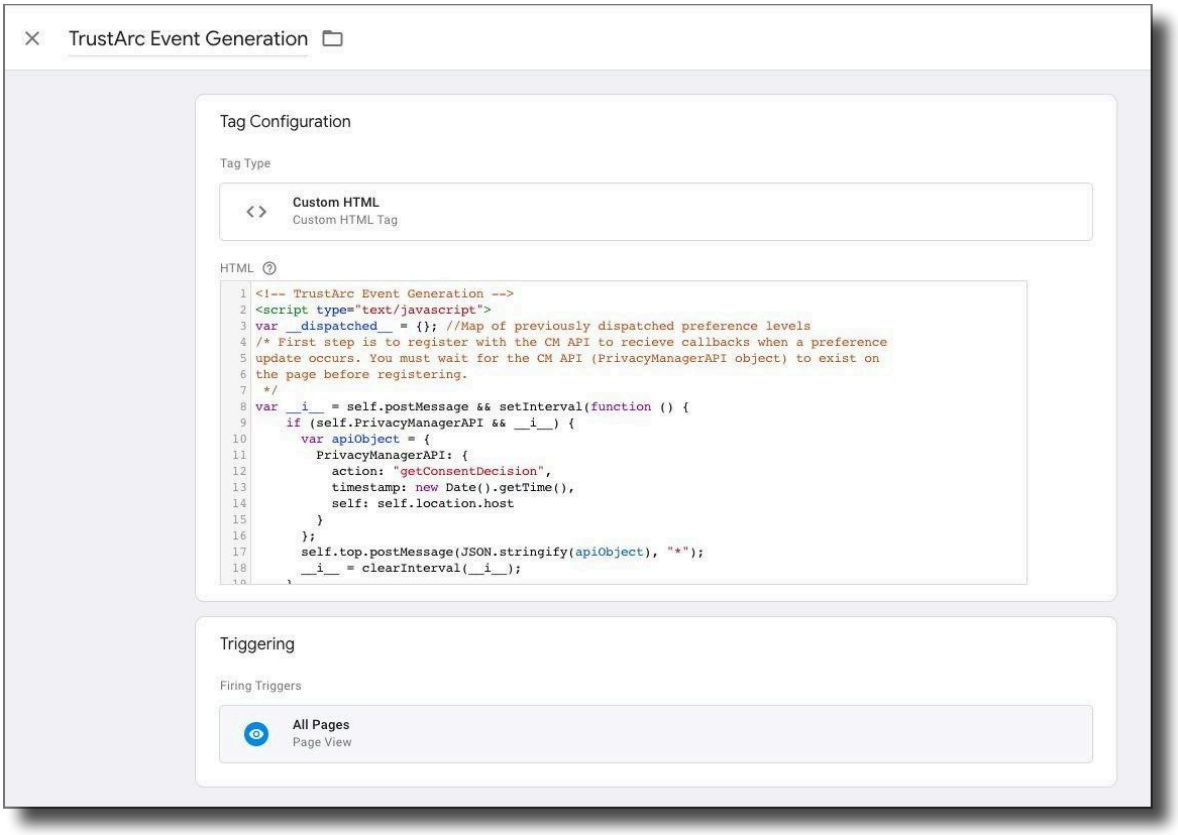
```
var __dispatched__ = {}; //Map of previously dispatched preference levels

/*
First step is to register with the CM API to receive callbacks when a preference update
occurs. You must wait for the CM API (PrivacyManagerAPI object) to exist on the page before
registering.
*/

var __i__ = self.postMessage && setInterval(function(){
       if(self.PrivacyManagerAPI && __i__){
             var apiObject = {PrivacyManagerAPI:
             {action:"getConsentDecision",
                    timestamp: new Date().getTime(),
                    self: self.location.host}};
             self.top.postMessage(JSON.stringify(apiObject),"*");
       __i__  = clearInterval(__i__);
}},50);

/*
Callbacks will occur in the form of a PostMessage event. This code listens for the
appropriately formatted PostMessage event, gets the new consent decision, and then pushes
the events into the GTM framework. Once the event is submitted, that consent decision is
marked in the 'dispatched' map so it does not occur more than once.
*/

self.addEventListener("message", function(e, d){
try{
      if(e.data && (d= JSON.parse(e.data)) &&
           (d = d.PrivacyManagerAPI) && d.capabilities && d.action=="getConsentDecision") {
           var newDecision =
                  self.PrivacyManagerAPI.callApi("getGDPRConsentDecision",
                  self.location.host).consentDecision;
           newDecision && newDecision.forEach(function(label){
                  if(!__dispatched__[label]){
              self.dataLayer && self.dataLayer.push({"event":"GDPR Pref Allows "+label});
                        __dispatched__[label] = 1;
                        }
                  });
           }
      }catch(xx){/** not a cm api message **/}
});
```

NOTE: You can also copy the code here.

Using GTM to Deploy CM Assets

You can add the CM assets to your pages via GTM using Custom HTML tags. When doing so, please ensure that the CM asset tags are fired on **All Pages** on the *Page View* GTM event and that no blocking rules are applied to these tags. You will also need to fire any tag controlled by the user's consent on the *Window Loaded* GTM event.

Important: When using a **Zero-Tracker** Load, it is NOT recommended to use GTM to add the CM script to the page. The recommended placement of the CM script for a **Zero-Tracker** Load is to run as early as possible on the page, preferably as the first script to run in the header, to ensure GTM has the required information to properly block the necessary tags. If you do choose to use GTM to add the CM script, you will be limited to blocking / firing tags on the *Window Loaded* GTM event and note that potential timing issues within GTM may occur. Please refer to the Troubleshooting section of this guide for more information.



Disclaimer: Plugins and browser extensions can block your tag managers and then, block CCM. Please check if your tag manager works with the popular Adblockers. If your tag manager is blocked, you may consider firing the TrustArc scripts outside of the tag manager in the HEAD tag.

GTM Variables and Triggers

This section outlines the configuration of the custom GTM Variables and Triggers used by Consent Manager in GTM.

Variables

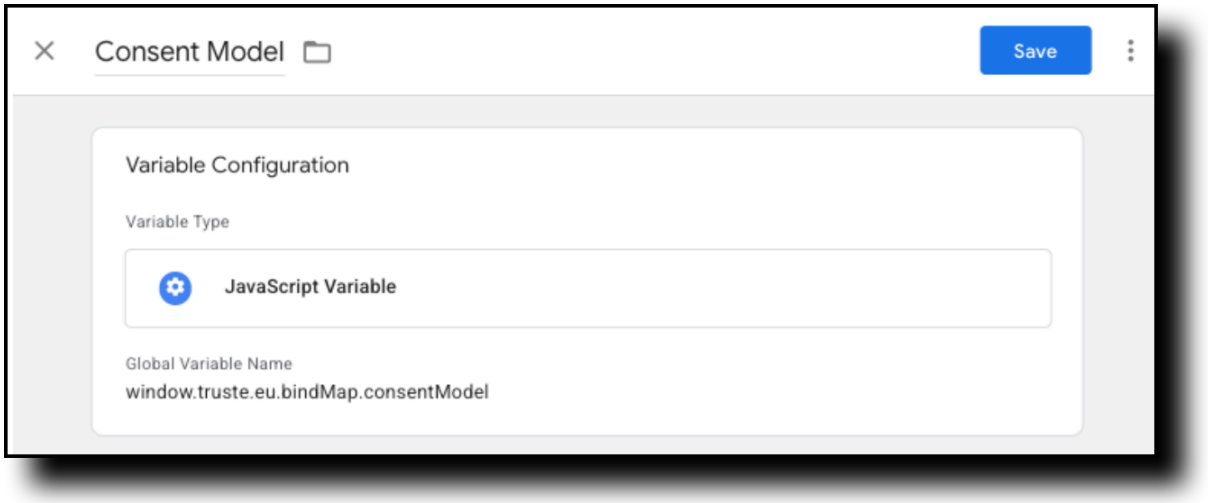
The following GTM variables determine whether a tag is injected into the page. Create the **Tracker Consent** variable. Then, create either the **Consent Model** variable or the **Tracker Behavior** variable, depending on your Consent Manager configuration. If you are unsure which configuration applies, contact your Technical Account Manager before proceeding.

Tracker Consent

- **GTM Variable Type**: 1st Party Cookie
- **Cookie Name**: `cmapi_cookie_privacy`
- **Cookie Purpose**: This cookie indicates the user's preference level.
- **Expiration**: 12 months
- **Possible Values**:
    - `permit 1 required` = Opted out of non-Required trackers
    - `permit 1,2 functional` = Opted out of Advertising only
    - `permit 1,2,3` = Accepted all categories
    - `permit 1,2,3,4` = Accepted all categories (plus custom category)

Consent Model

- **GTM Variable Type**: JavaScript Variable
- **Variable Name** : `window.truste.eu.bindMap.consentModel`
- **Variable Purpose**: This cookie indicates the user's consent model, based on their location
- **Possible Values**:
    - `opt-in` = Consent Model is opt-in (zero-tracker load)
    - `opt-out` = Consent model is opt-out (standard load)

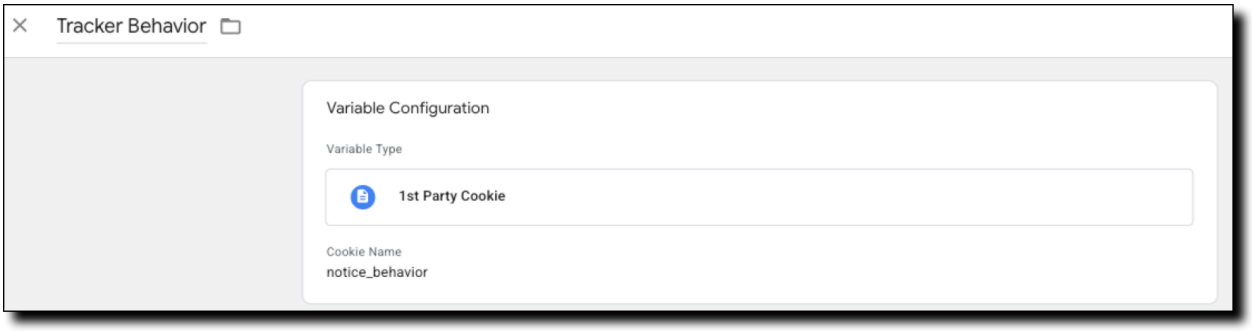


Note: In the CCM Portal, the Technical Account Manager must first confirm whether the Consent Model is set to "Opt-out" or "Opt-in". If either option is selected, this variable is required and should be used.

Tracker Behavior (Optional)

Use this variable only if your implementation uses the legacy cookie notice_behavior.

- **GTM Variable Type**: 1st Party Cookie
- **Cookie Name**: `notice_behavior`
- **Cookie Purpose**: This cookie identifies the user's tracker behavior preference for legacy implementations.



NOTE: In the CCM Portal, verify that the Consent Model is set to **None** before creating this variable. If the Consent Model is set to **None**, this variable is required.

Blocking Triggers

The following GTM triggers are used to block tags from being injected into the page based on the user's provided consent preferences. You will need a trigger for each non-Required consent category included in your CM instance – the examples included below are the triggers for TrustArc's default Functional and Advertising Trackers consent categories. Your GTM configuration should include blocking triggers corresponding to your CM instance's non-Required categories.

Level 2 Preference Blocking Trigger

This trigger blocks the tag from being injected when the user has not consented to Functional Trackers.

- **Trigger Type**: Custom Event
- **Event name**: .* (regex match)
- **This trigger fires on**: *Some Custom Events*
- **Conditions**:
    - **Tracker Consent** *does not contain 2*
    - **Tracker Consent** *contains permit*

Level 3 Preference Blocking Trigger

This trigger blocks the tag from being injected when the user has not consented to *Advertising* Trackers.

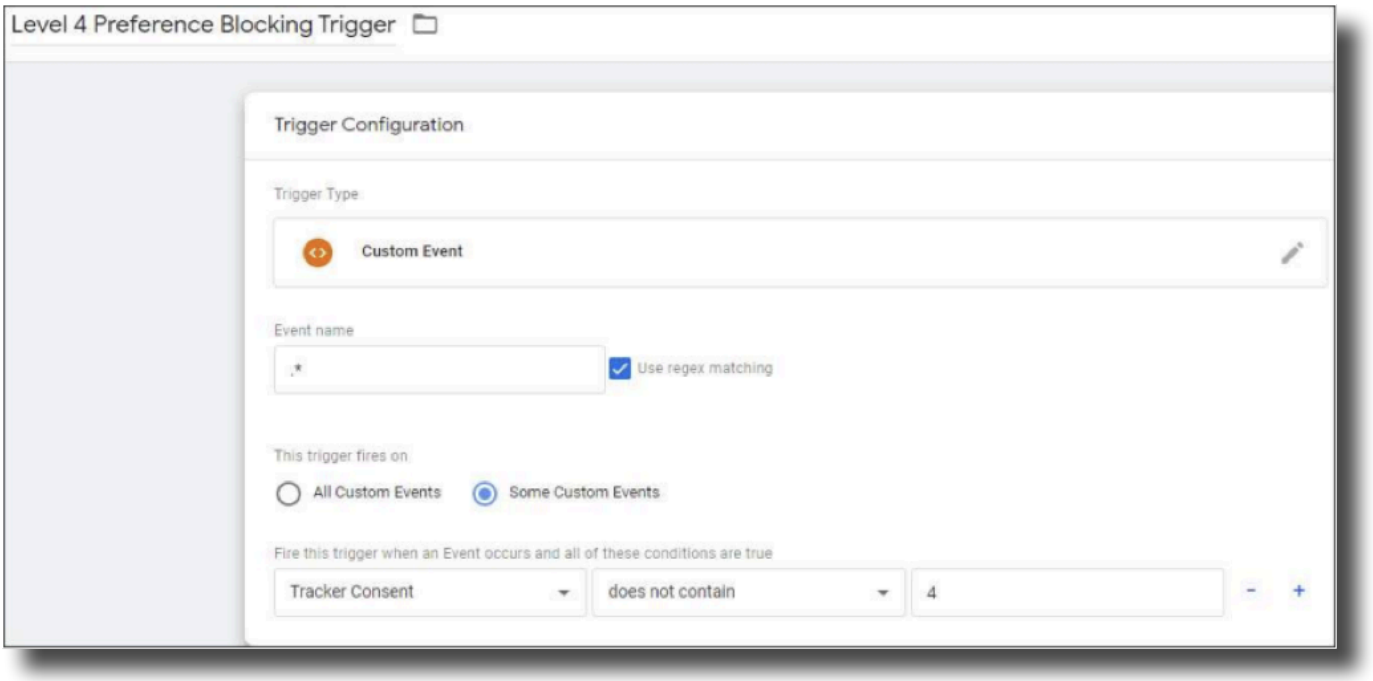
- **Trigger Type**: Custom Event
- **Event name**: .* (regex match)
- **This trigger fires on**: *Some Custom Events*
- **Conditions**:
    - **Tracker Consent** *does not contain 3*
    - **Tracker Consent** *contains permit*

Level 4 Preference Blocking Trigger (Optional)

TrustArc's system supports up to 3 additional custom categories in addition to our 3 out-of-the-box categories, for a total of 6 categories. This is an example of how you would add support for a 4th category. This process can also be repeated for the 5th and 6th category, changing the assigned number accordingly.

This trigger blocks the tag from being injected when the user has not consented to the 4th category.

- **Trigger Type**: Custom Event
- **Event name**: .* (regex match)
- **This trigger fires on**: Some Custom Events
- **Conditions**:
    - **Tracker Consent** *does not contain 4*
    - **Tracker Consent** *contains permit*



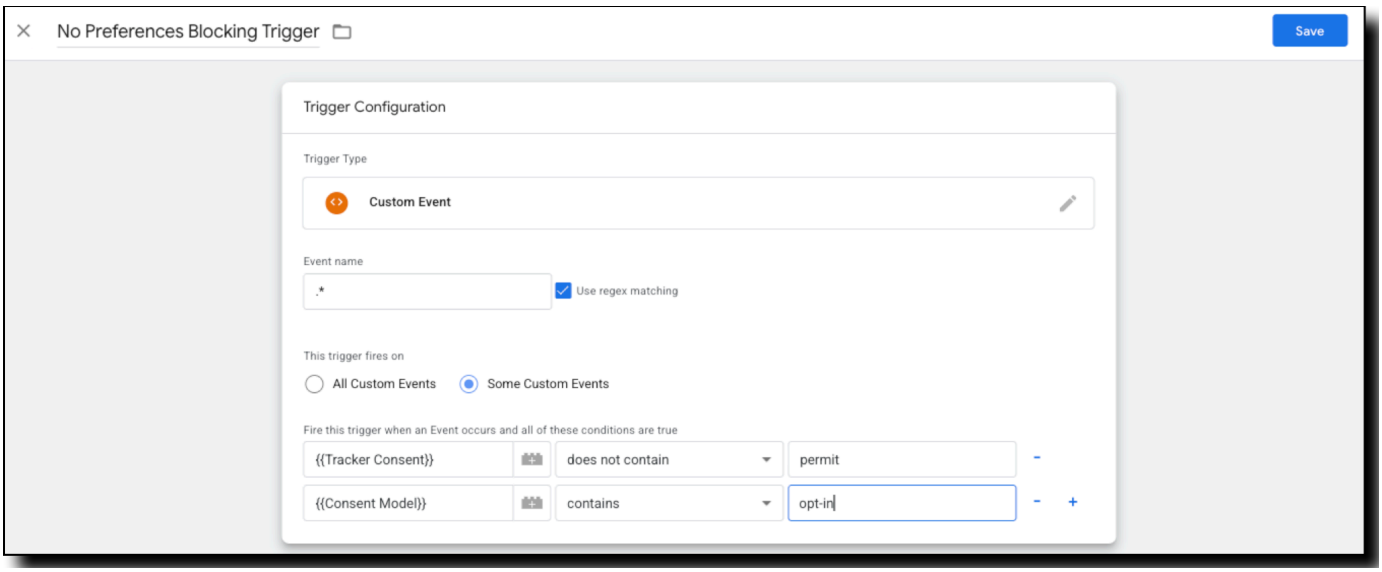
No Preferences Blocking Trigger

This trigger prevents the tag from being injected when the user has not provided consent. The trigger checks whether the Consent Model contains *opt-in* to achieve a Zero-Tracker Load. Work with your Technical Account Manager to confirm the correct configuration of the No Preferences Blocking Trigger for your implementation.

Consent Model Variable

When the Consent Model variable is being used, this should serve as the condition for configuring the No Preferences Blocking Trigger.

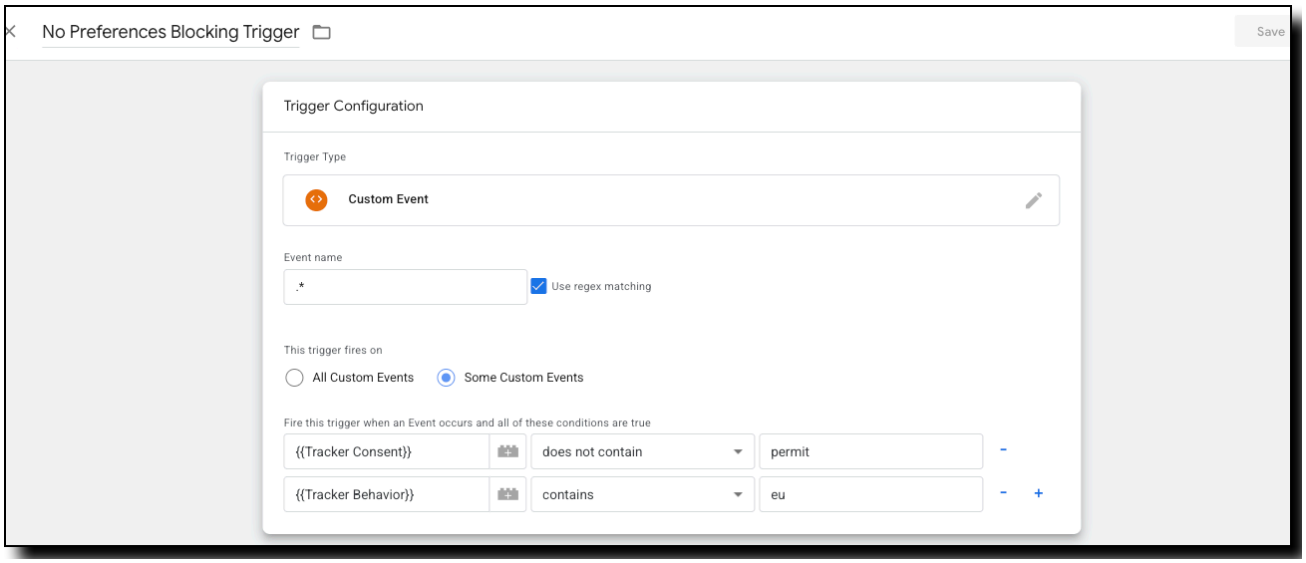
- **Trigger Type**: Custom Event
- **Event name**: .* (regex match)
- **This trigger fires on**: Some Custom Event
- **Conditions**:
    - **Tracker Consent** *does not contain permit*
    - **Consent Model** *contains opt-in*



Tracker Behavior Variable

When the **Tracker Behavior** variable is being used, this should serve as the condition for configuring the **No Preferences Blocking Trigger**.

- **Trigger Type**: Custom Event
- **Event name**: .* (regex match)
- **This trigger fires on**: Some Custom Event
- **Conditions**:
    - **Tracker Consent** *does not contain permit*
    - **Tracker Behavior** *contains eu*

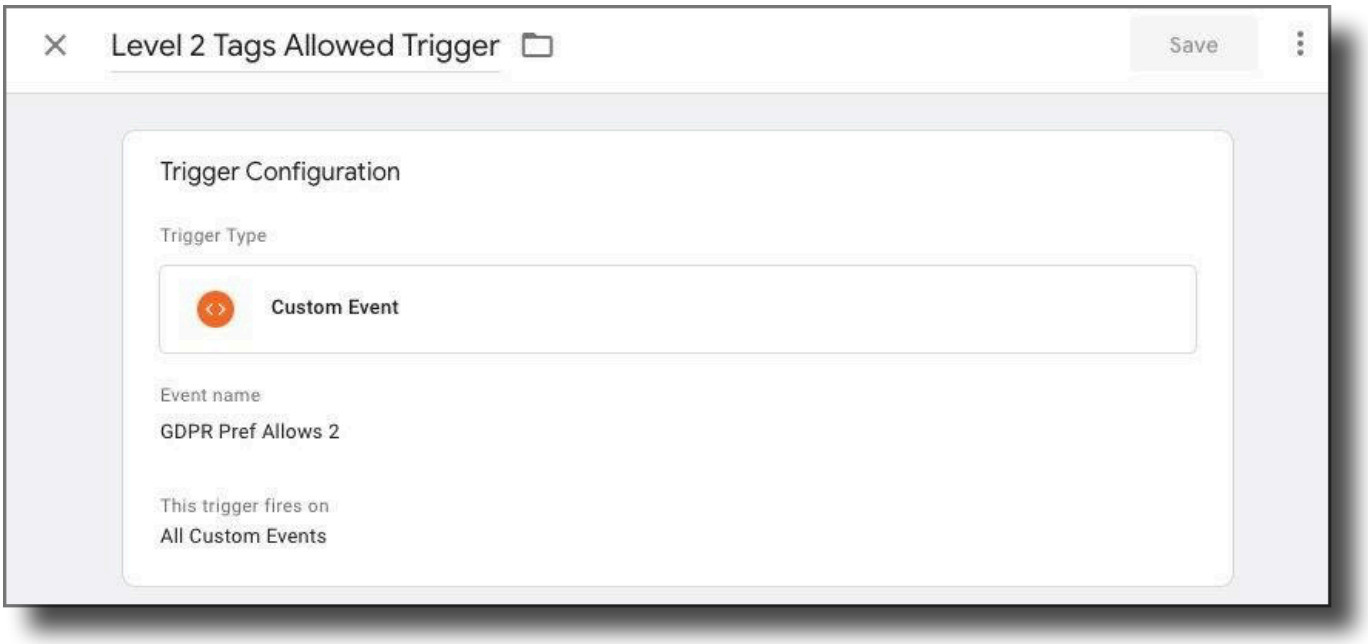


Event Triggers

The following GTM triggers are used to load tags once a user has provided consent without requiring a full refresh of the page. When using these GTM events please ensure you are including the Event Listener code described in the GTM Event Listener section of this guide.

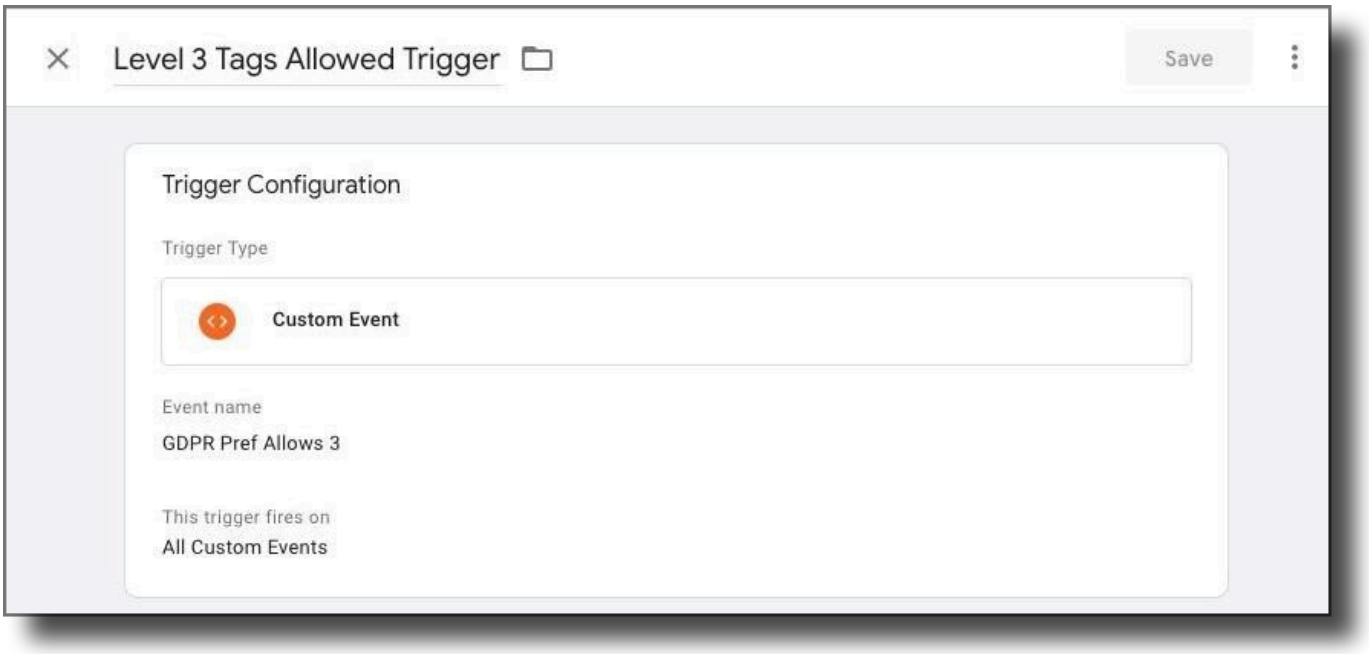
Level 2 Tags Allowed Trigger

- **Trigger Type**: Custom Event
- **Event name**: GDPR Pref Allows 2
- **This trigger fires on**: *All Custom Events*



Level 3 Tags Allowed Trigger

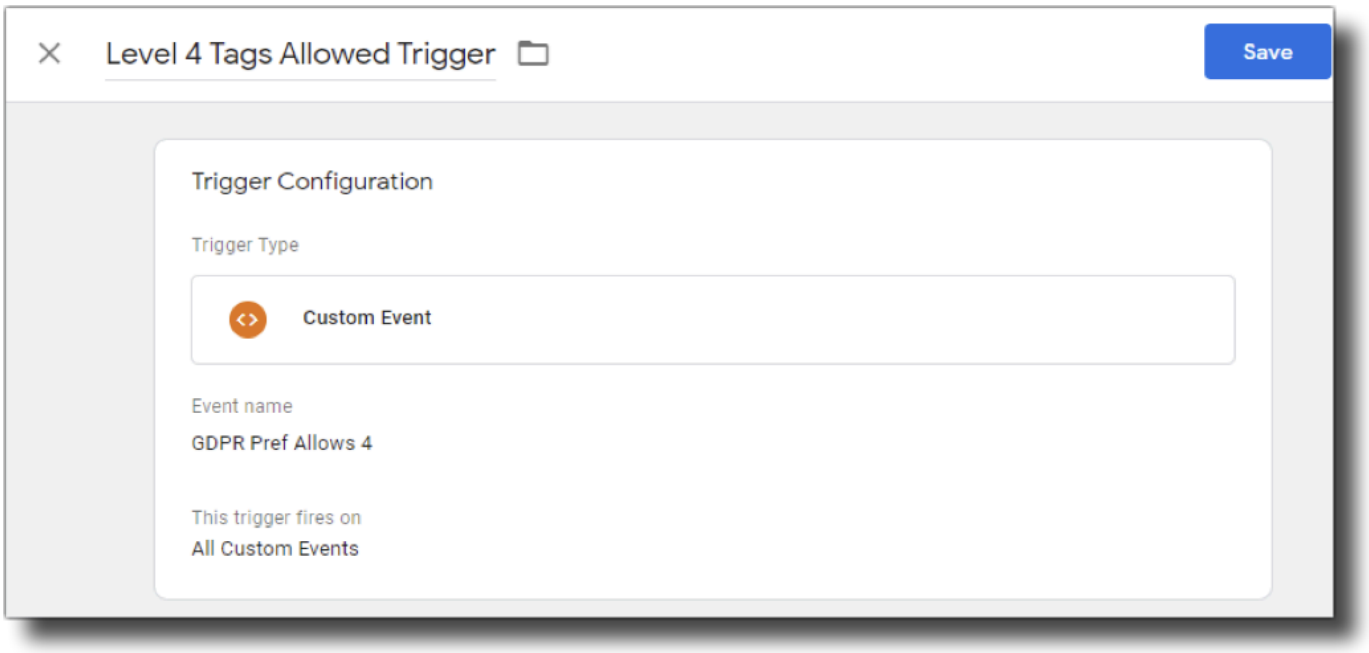
- **Trigger Type**: Custom Event
- **Event name**: GDPR Pref Allows 3
- **This trigger fires on**: *All Custom Events*



Level 4 Tags Allowed (Optional)

This optional rule would only be created and added as a firing trigger if you have more than the 3 out-of-the-box categories, as referenced earlier in this guide.

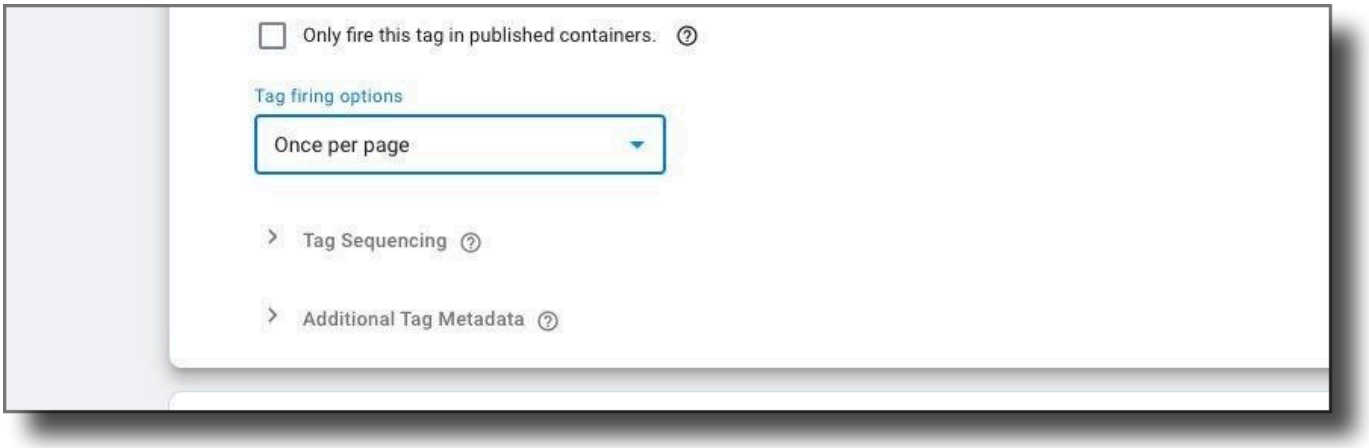
- **Trigger Type**: Custom Event
- **Event name**: GDPR Pref Allows 4
- **This trigger fires on**: *All Custom Events*



GTM Tags

This section outlines the application of the appropriate GTM Triggers to your GTM Tags to control their inclusion on the page based on the user's consent. The examples below are for applying the triggers for TrustArc's default *Functional* and *Advertising* Trackers consent categories. Your GTM configuration should include blocking triggers corresponding to your CM instance's non-Required categories.

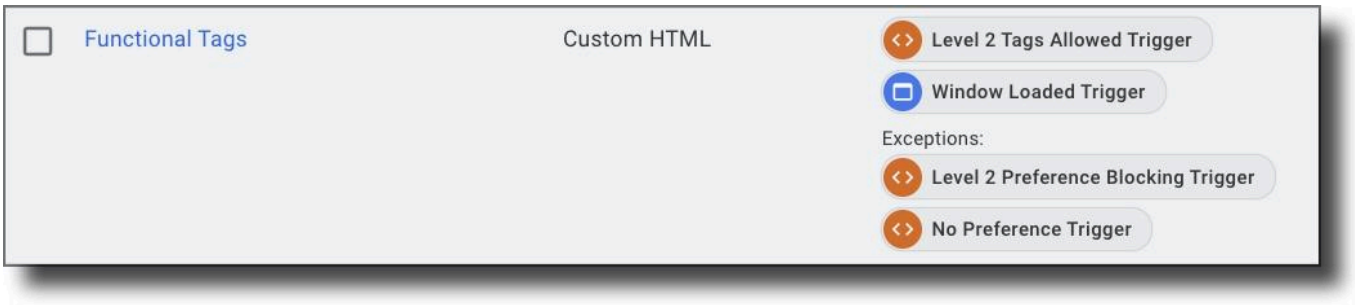
Important: In addition to adding the appropriate triggers to your tags, you will need to update each tags' *Tag firing options* to **Once per page**. This is to ensure that a change to the user's consent does not unexpectedly trigger a tag to fire multiple times on a single page load.



Functional Tags

For each tag included in the *Functional* Tracker category you will need to add the following triggers in addition to any existing triggers:

- **Firing Triggers**: *Level 2 Tags Allowed Trigger*
- **Exceptions**: *Level 2 Preference Blocking Trigger* and *No Preference Trigger*



Advertising Tags

For each tag included in the *Advertising* Tracker category you will need to add the following triggers in addition to any existing triggers:

- **Firing Triggers**: *Level 3 Tags Allowed Trigger*
- **Exceptions**: *Level 3 Preference Blocking Trigger* and *No Preference Trigger*

Special Trigger Condition Tags

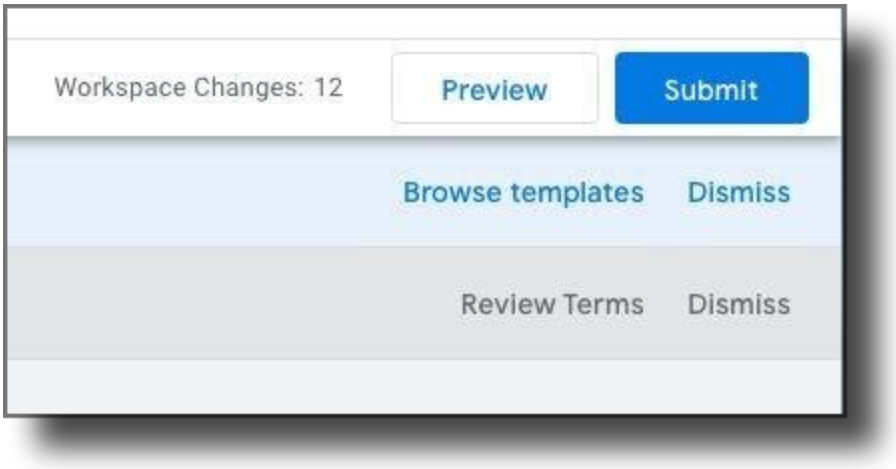
The above examples are for use on tags that are fired on every page. However, it is common for tags to only be fired under specific conditions or on specific pages. In these situations, the *Exception Triggers* would be included as in the above examples but the *GTM Event Firing Trigger* (**Level 2 Tags Allowed** / **Level 3 Tags Allowed**) would not. This prevents these 'special trigger condition' tags from being accidentally fired when a user changes their consent.

Verification and Troubleshooting

This section outlines how to verify a **Zero-Tracker** GTM integration and some common troubleshooting steps.

GTM Preview Mode

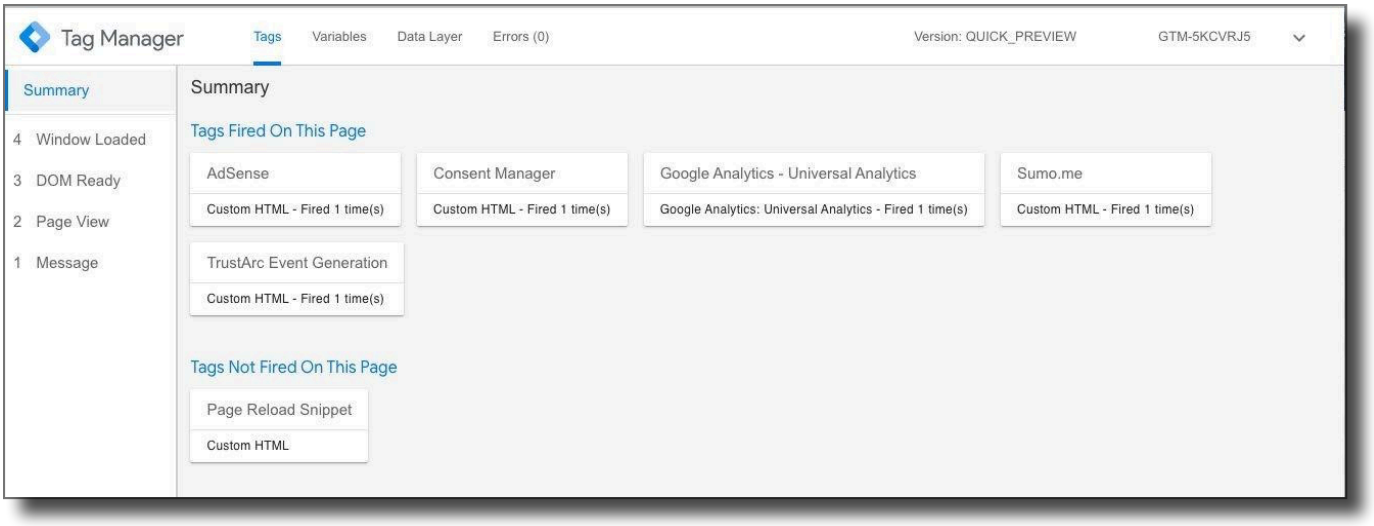
The Preview Mode in GTM can be a vital tool when verifying and troubleshooting a Zero-Tracker integration as it allows you to examine the state of each variable and trigger condition for every tag at each GTM Event. If you have the appropriate permissions for the GTM container, you can launch the Preview Mode by clicking the Preview button next to the Submit button within GTM.



When viewing your site with your browser in Preview Mode you can examine information for each tag fired on each of the GTM events.

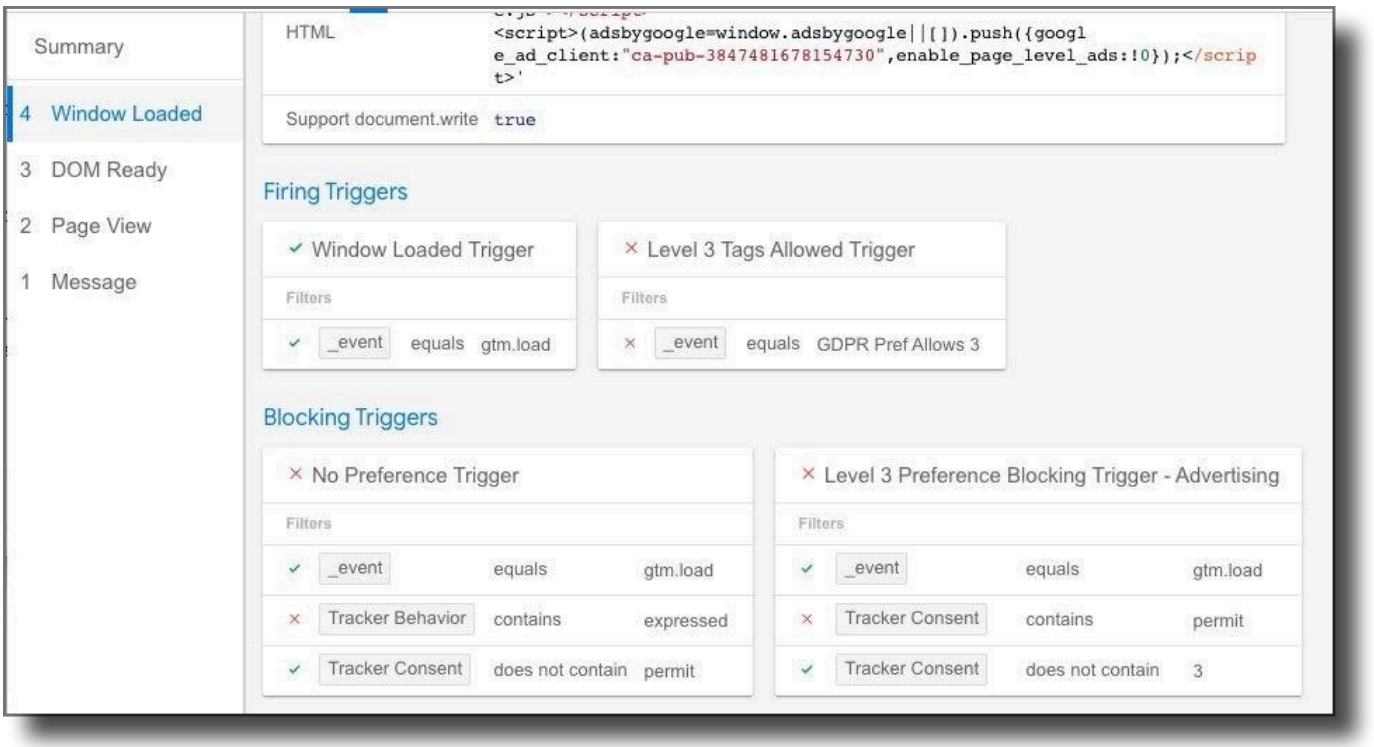
Summary

The Summary information displays the individual tags that were fired and blocked on each GTM Event which occurred.



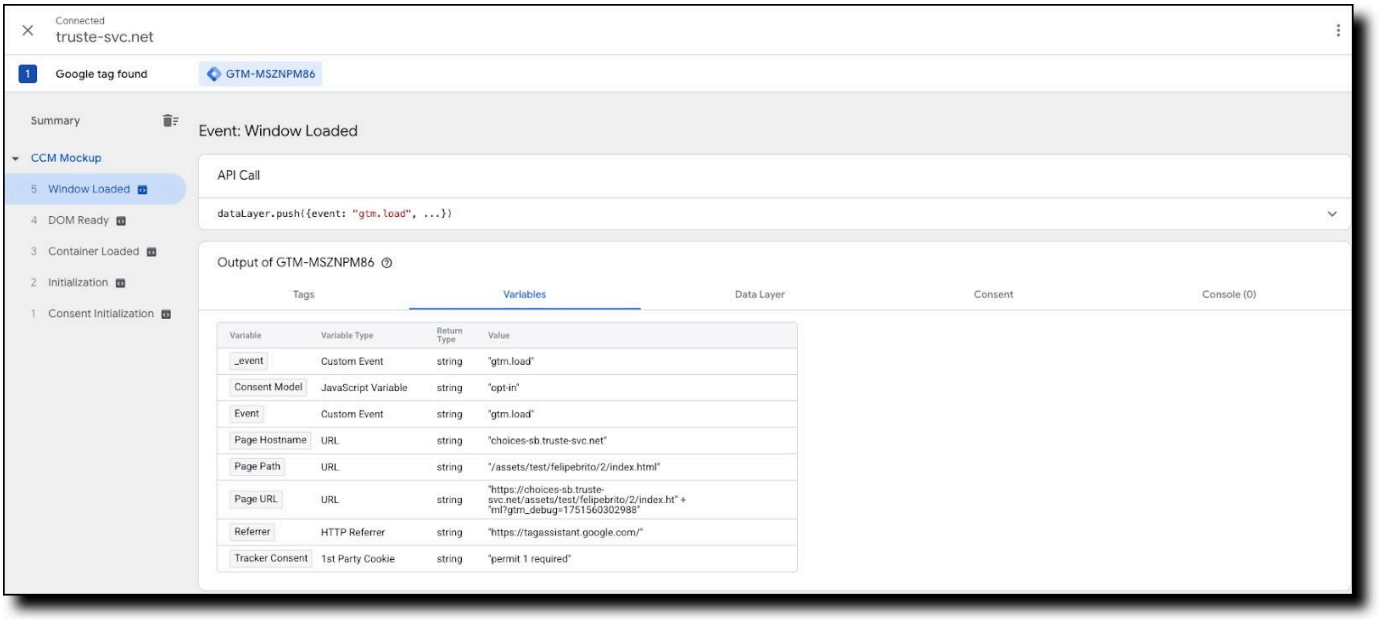
Tags

Clicking on any of the tags when a GTM Event is selected will display a snapshot of the pertinent information for that tag at that event. Included in this information are the states of the individual trigger conditions of any applied trigger.



Variables

Selecting the Variables tab when a GTM Event is selected will display a snapshot of the information contained within each variable.



Verifying a GTM Zero-Tracker Integration

The process of verifying your GTM Zero-Tracker Integration will vary depending on the specifics of your deployment, CM configuration, and overall tools at your disposal. However, in general you will be testing that the correct blocking occurs in three situations:

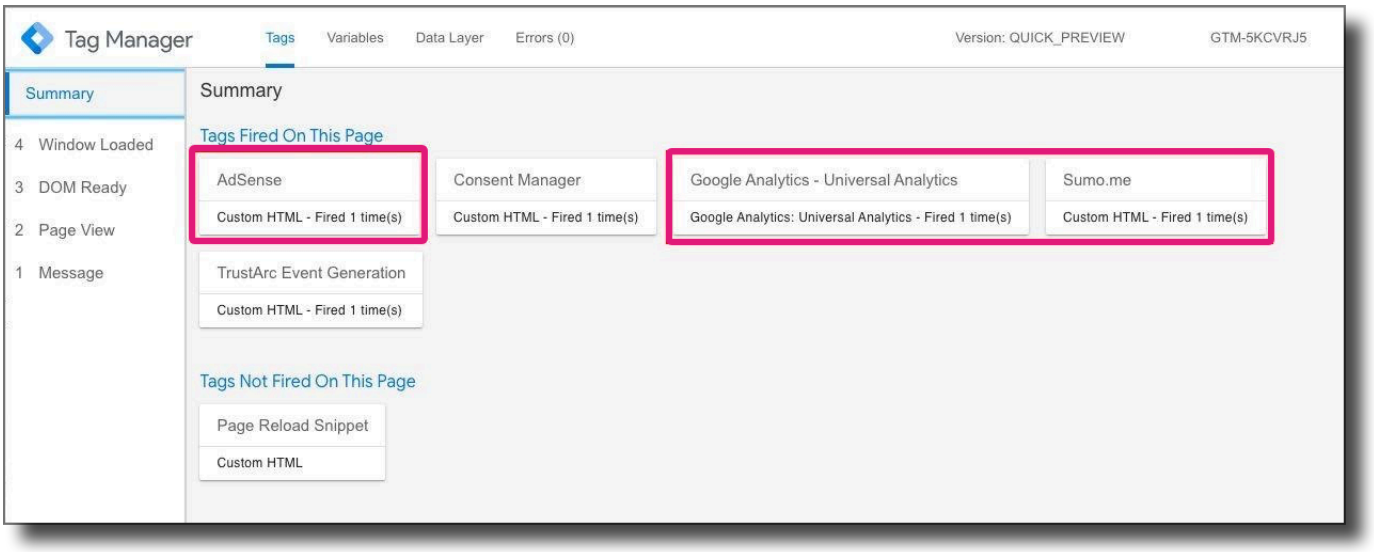
- A new user with Standard Load
- A new user with Zero-Tracker Load
- Changing the existing consent for each category

It is recommended to verify the mechanics and functionality of the GTM Integration with a single tag of each consent category prior to applying the integration to your GTM implementation as a whole.

In the following examples the *AdSense* tag is a **Level 3 tag** and the Google Analytics and Sumo.me tags are **Level 2 tags**.

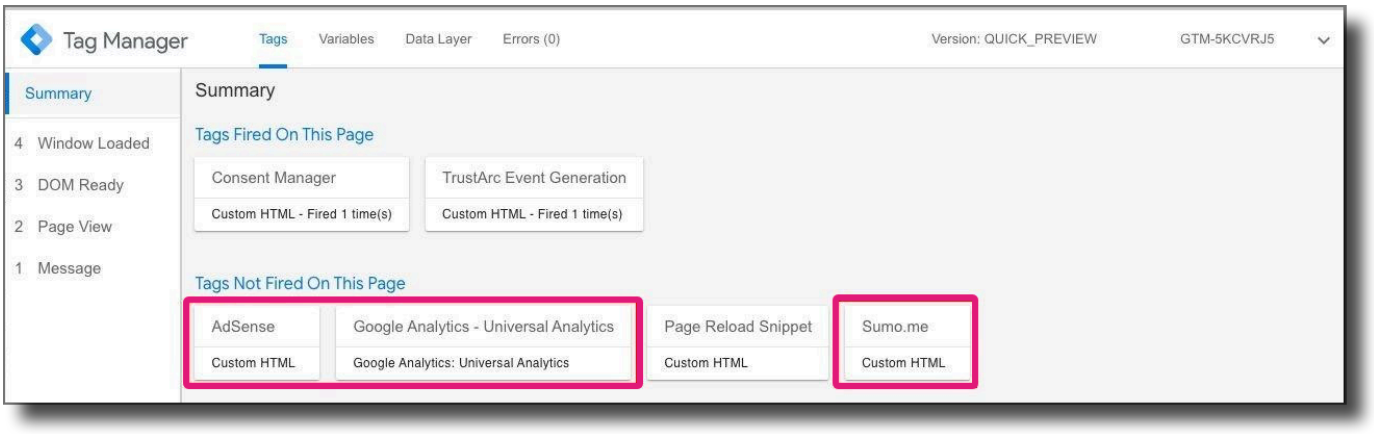
A new user with Standard Load

When verifying a new user who should see a Standard Load, we are simply ensuring that all tags are fired as expected on the initial load. You should see the same tags fire after the integration as before at each GTM Event. If you are using GTM to deploy any of the CM assets you should also see those tags fired on each page.



A new user with Zero-Tracker Load

When verifying a new user who should see a Zero-Tracker Load, we are verifying that all of the tags that have the Blocking Triggers applied have NOT been fired. This should be done before any user interaction with the Consent Manager.



Once consent has been submitted, and if consent has been provided, the appropriate tags should fire immediately.

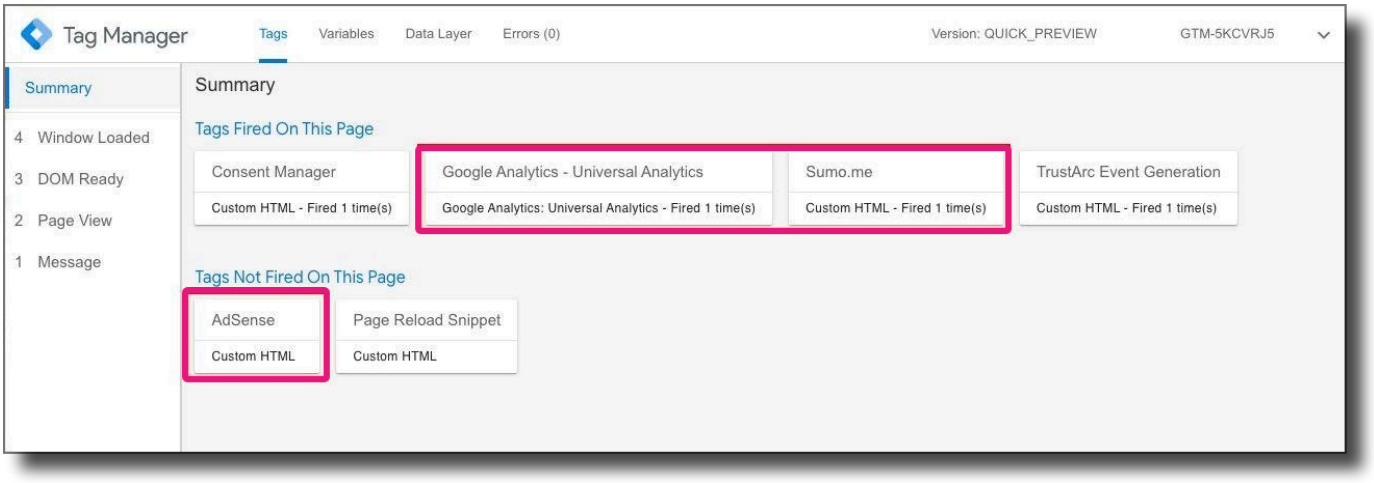
Changing the existing consent for each category

When verifying for a user who has changed their consent we are verifying that the correct tags are blocked or fired based on the change. For example, when verifying the change in consent from *Advertising* to *Functional* we would expect to see any tags with the **Level 3 Preference Blocking Trigger** applied **NOT** to be fired on the next page load. Conversely, a change from *Functional* to *Advertising* should cause any tags with the **Level 3 Preference Blocking Trigger** applied to be fired. A change both to and from each consent category is recommended.

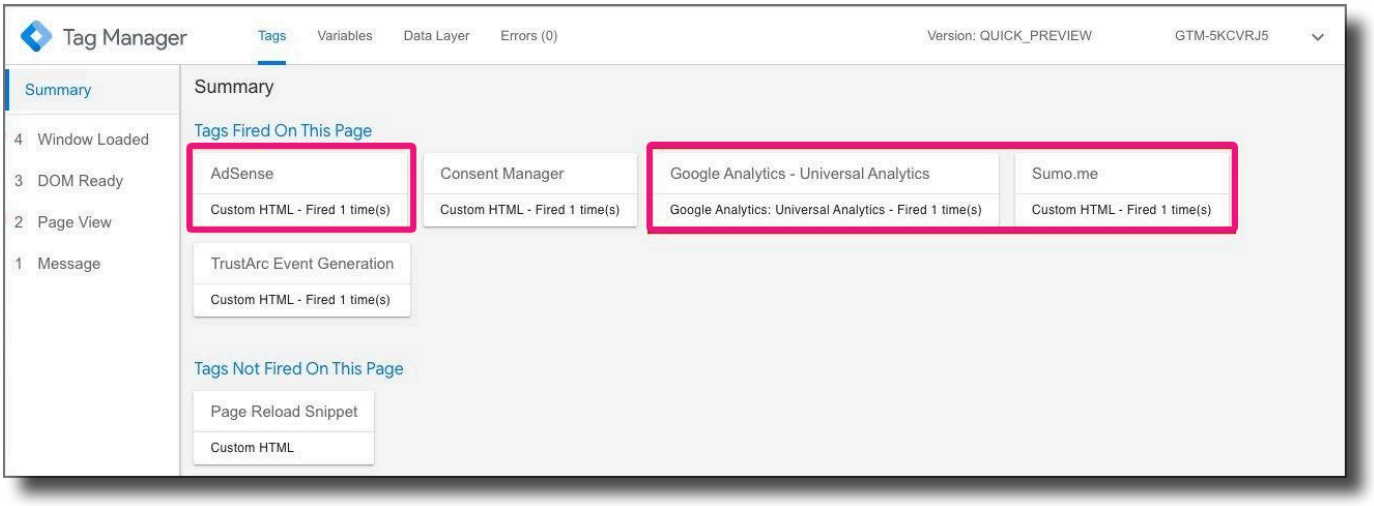
Important: To ensure consent preferences are appropriately honored, once a tag script or code has been executed on a page (i.e., a tag has fired), you will need to configure an automatic page refresh in order for the new consent preference to be reflected. Failure to execute a refresh will result in the prior tracking behavior (e.g., tracking will continue as if an opt-out had not occurred) to persist until a manual refresh is done by a web visitor themselves.

NOTE: TrustArc has developed a JavaScript snippet for page reload functionality, which you may use directly or reference for your own implementation. You can view the code at this link.

With Level 2 enabled / Level 3 disabled



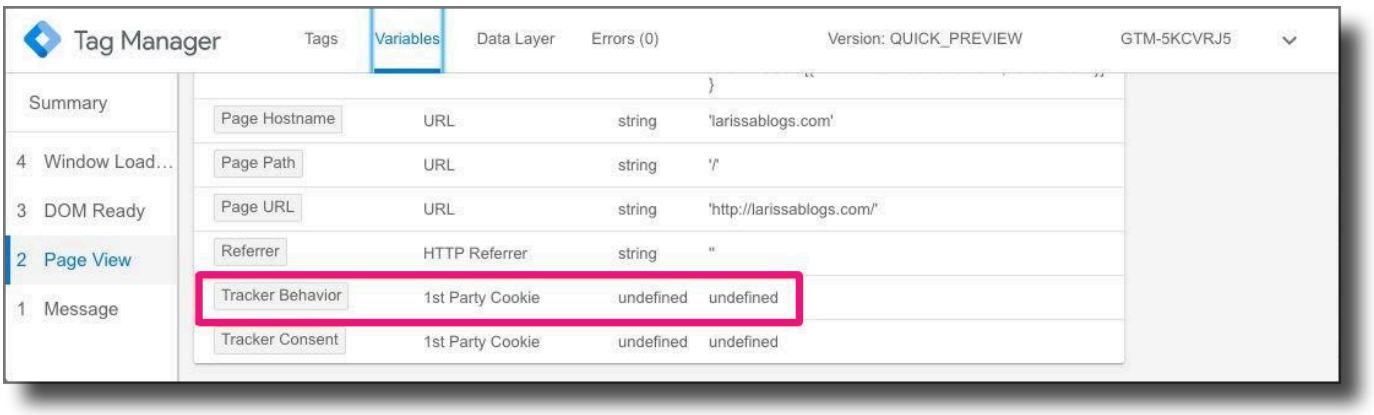
With Level 2 and Level 3 enabled



Troubleshooting

I’m not seeing the expected Zero-Tracker Load occur

The most common cause of a **Zero-Tracker Load** not occurring when expected is that the *Tracker Behavior* GTM Variable does not have data when the tags firing trigger conditions are met, preventing the corresponding blocking trigger conditions from being met as well. This frequently occurs when GTM is used to deploy the CM script to the page and in these cases the CM script should be moved to fire as early in the header of the page as possible. If this is not possible, or if this does not resolve the issue, you will need to change all of the tags trigger conditions to fire on a later event when the *Tracker Behavior* GTM Variable does have data.



In the above snapshot the *Tracker Behavior* variable is `undefined` and does not contain data at the *Page View* GTM Event.