TrustArc

Google Consent Mode

Implementation Guide

Revision History

Document Version	Date	Description
1.0	07-22-2021	First revision of the document.
1.1	04-09-2022	Changed the order of loading the consent code.
1.2	11-21-2023	Added the additional tag settings from CCM 2023.11.01 release
1.3	01-29-2024	Updated instructions for Integrating the Consent Mode v.2.0 Using Google Template and Using JavaScript
1.4	08-07-2024	Updates instructions for CoMo v2.0 including deployment strategy and tags verification
1.5	04-28-2025	Updated link for sample code for the event listener used in the Google Consent Mode Integration
1.6	07-03-2025	Added more information on CMP ID and additional parameters when Integrating the Consent Mode v.2.0 Using Google Tag Template section
1.7	09-24-2025	Updated Support and Deployment Strategies sections; Added more information on Creating a Banner That Meets Google's Banner Requirements

About This Document

This document provides guidance on how to implement Google Consent Mode, a feature that provides organizations with the capability to adjust the behavior of Google tags (Google Analytics, Google Ads Conversion, Remarketing, and Floodlight) and third-party tags configured in the Tag Manager, to comply with privacy regulations while still collecting a minimum level of engagement data.

Target Audience

This document is intended for web developers, programmers, back-end developers, or trusted parties managing tags on a website.

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 in coordination with their legal counsel.

Table of Contents

Overview

Google is enhancing its enforcement of the EU user consent policy to ensure a privacy-focused digital advertising ecosystem using Google Consent Mode. Consent Mode allows you to customize the behavior of your Google tags in response to the consent status of your users. Using Google Consent Mode, you have the flexibility to specify whether consent has been granted for Google products like Analytics and Ads. In this mode, Google's tags dynamically adjust the consent preferences, and the tags can be adjusted to read those preferences in a standardized way. By leveraging consent signals, Google can recover potential lost conversions resulting from changes in consent preferences.

Support

We are committed to providing fast, high-quality support for all clients implementing Google Consent Mode using the TrustArc Consent Manager.

TrustArc provides a dedicated support path for Google Consent Mode and certification-related inquiries. This includes:

- A **specialized support email** google-consent-mode@trustarc.com that you can use for any issue related to Google Consent Mode v2, or integration verification.
- For Customers with a managed service, we provide direct access to a **Technical Account Manager (TAM)** trained in Google's CMP certification criteria.
- An option to directly contact Google's CMP Support Team by copying cmp-support-helper@google.com. With client approval, TrustArc can engage directly with Google for complex issues or certification blockers.

TrustArc offers the following response time commitments for all issues related to Google Consent Mode:

Customer Tier	First Response Time	Response Type
Premium / Enterprise Clients	Within 1 Business Day (24 hours)	A technical, resolution-oriented response containing clear next steps or troubleshooting actions.
Standard Clients	Within 2 Business Days (48 hours)	

How to Contact Us

Use Case	Email / Channel
General CMP Support	support@trustarc.com
Google Consent Mode Issues	google-consent-mode@trustarc.com
Urgent Certification or Integration Escalations	tam@trustarc.com (Technical Account Management Team)
Escalate to Google	Copy cmp-support-helper@google.com

Disable Google Consent Mode

If you decide not to use Consent Mode, you can disable it by preventing TrustArc from transmitting Consent Mode signals to Google Tag Manager. Make sure that you pause the template in your Tag Manager or remove the gtag function from your source code.

As an alternative option, you can use our standard GTM guide. This method involves using firing and blocking triggers to selectively load tags. You can configure them in a way that ensures they are not loaded before obtaining user consent, especially for tags that track your visitors.

Tag Settings

Google Consent Mode provides tag settings that will manage trackers using conversion modeling for consent mode. For more information, please follow this link.

TrustArc's Cookie Consent Manager supports the following consent types:

- **analytics_storage**
- **ad_storage**
- **ad_personalization**
- **ad_user_data**
- **functionality_storage**
- **personalization_storage**
- **security_storage**

analytics_storage

With the **analytics_storage** tag setting, Google Consent Mode controls the behavior of analytics cookies on your website based on the consent state of your end-users. Google Analytics will then adjust its data collection based on the granular consent choice of each individual user. Note that using built-in consent types without additional permissions required, Google tags will still load in the user's browser. Google will apply the necessary controls to prevent data from flowing to the tags.

If users don't give consent to analytics cookies, your website should still receive aggregate and basic measurement data, such as:

- Timestamps of visits to your website
- User-agent, i.e. whether users landed on your website
- Referrer, i.e. how the user landed on your website
- Whether current or prior page in user's navigation includes ad-click information in the URL
- Random number per each page load

ad_storage

With the **ad_storage** tag setting, Google Consent Mode controls the behavior of advertising cookies on your website based on the consent state of your end-users, i.e. if a user does not consent to advertising cookies, Google Consent Mode will ensure that all advertising-related Google tags will not set cookies on the users' devices.

If users do not give consent to advertising cookies, your website will still be able to show contextual advertisements based on anonymized data.

Google Consent Mode will enable your website to measure conversions related to a specific campaign on an aggregate level rather than on an individual user level, providing insights into the performance of your advertising while helping to comply with privacy requirements.

Additionally, Google Consent Mode allows users to update their consent state and the configuration of tag behaviors based on user location, i.e. automatically ensures that no cookies are set without consent for users inside the EEA, while not requiring consent for the use of cookies for users in the US.

ad_personalization

With the **ad_personalization** tag setting, Google Consent Mode controls whether data can be used for ads personalization (for example, remarketing). This refers to Google Ad Personalization selection of cookie category level that is opted-in on Google Consent Mode.

ad_user_data

With the **ad_user_data** tag setting, Google Consent Mode controls whether personal data is sent to a Google core platform service. This refers to Google Personal Data selection of cookie category level that is opted-in on Google Consent Mode.

functionality_storage

With the **functionality_storage** tag setting, Google Consent Mode enables storage that supports the functionality of the website or app (for example, language settings). This refers to Google Functionality selection of cookie category level that is opted-in on Google Consent Mode.

personalization_storage

With the **personalization_storage** tag setting, Google Consent Mode enables storage related to personalization (for example, video recommendations). This refers to Google Personalization selection of cookie category level that is opted-in on Google Consent Mode.

security_storage

With the **security_storage** tag setting, Google Consent Mode enables storage related to security, such as authentication functionality, fraud prevention, and other user protection. This refers to Google Security selection of cookie category level that is opted in on Google Consent Mode.

Supported Services

Google Consent Mode supports a number of Google's products, including:

- **Google Analytics**
- **Google Ads (Google Ads Conversion Tracking and Remarketing)**
- **Google Tag Manager**
- **Gtag**
- **Floodlight**
- **Conversion Linker**

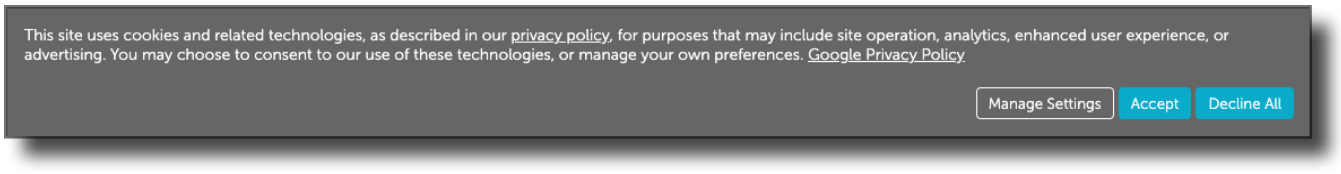
To check the latest support, please visit this site.

Creating a Banner That Meets Google's Banner Requirements

TrustArc offers an out-of-the-box consent banner designed to comply with Google's Consent Mode requirements. Implementing this banner correctly ensures that you meet Google's policies for using their advertising and measurement products, such as Google Ads, Google Analytics, and Floodlight. This banner automatically includes a link to the Google Privacy Policy if you designate all Google Core Platform Services (CPSs) for data reception. If you intend to designate only a subset of Google CPSs, consult your Technical Account Manager to include a document link in your banner that discloses the specified Google CPSs.

NOTE: For specific customizations or to support advertisers designating all Google CPSs, collaborate with your Technical Account Manager or our support team to add this link to your Cookie Banner. If you intend to designate only a subset of Google CPSs, consult your Technical Account Manager to include a document link in your banner that discloses the specified Google CPSs.

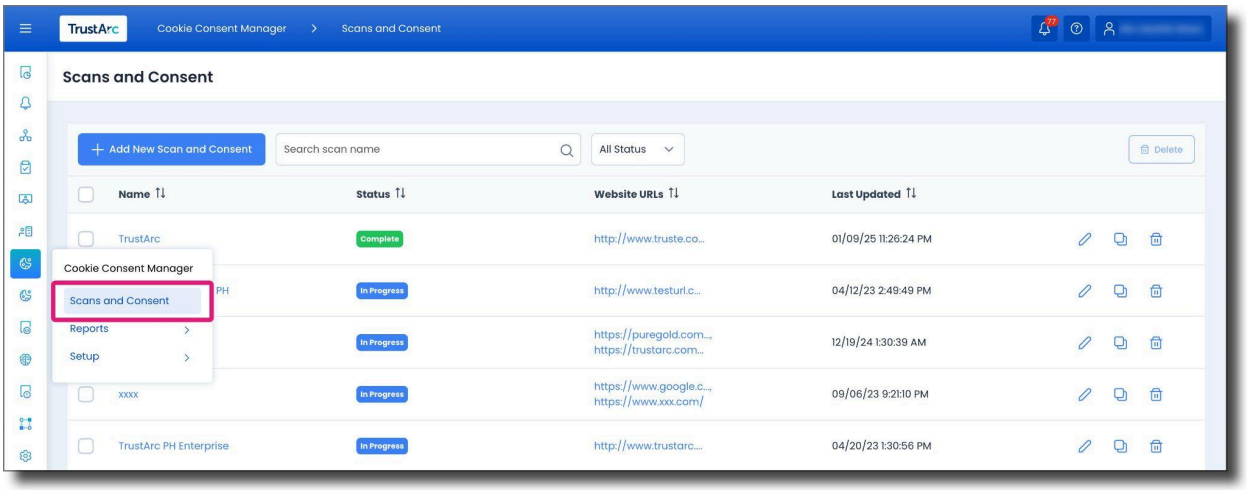
Sample Banner with Google Privacy Policy link:



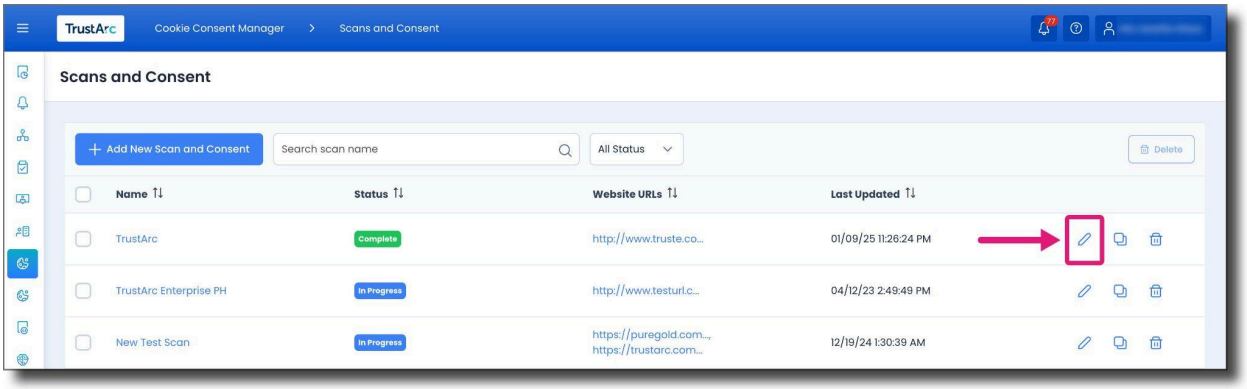
Adding Cookie Banner

To add a banner design, follow these steps:

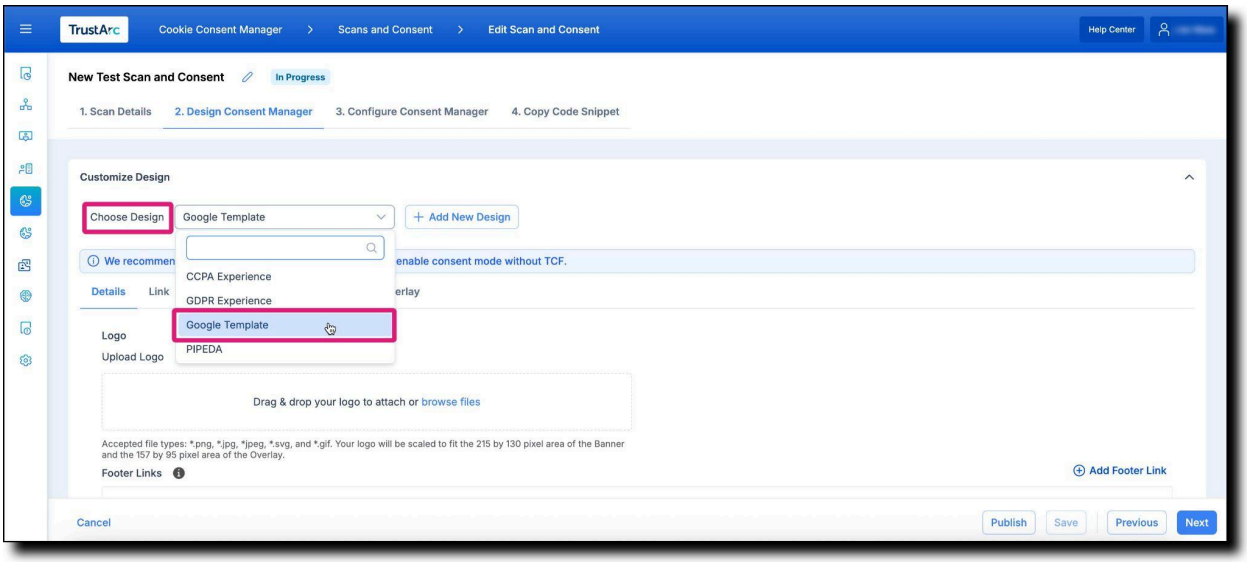
1. From the *Cookie Consent Manager* platform navigation panel, select **Scans and Consent**.



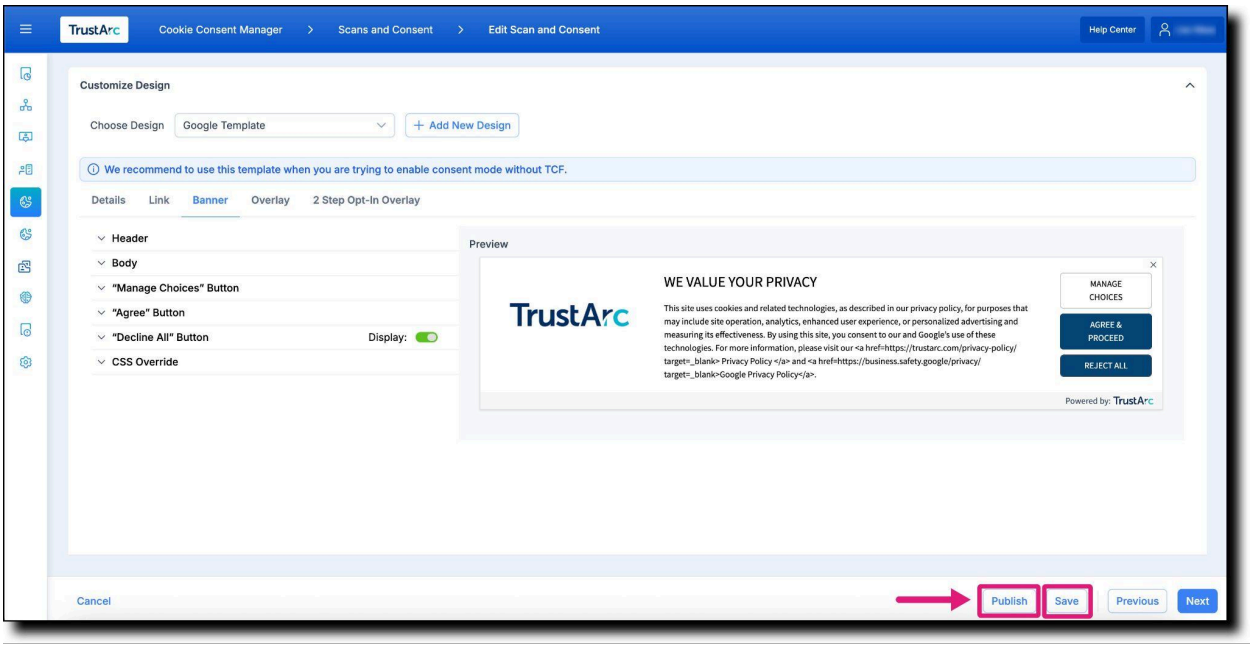
2. Click either the **CCM Instance Name** or the **Edit** button of the Consent Manager you will update.



3. In the *Design Consent Manager* page, select **Google Template**.



4. Then, click **Save > Publish**.



Important: Clicking **Save** will not reflect the changes in the Consent Manager deployed in your website/s. Changes made to the Consent Manager will only reflect when you click the **Publish** button. If you do not click **Publish**, the server will not pick up the changes and only serve up the CM that was last published.

Deployment Strategies

Independent of the consent behavior, Google Consent Mode can be deployed on a site in one of two methods - a **Basic** or **Advanced** deployment. With a **Basic** deployment (also known as **Zero-Cookie load**), Google Tags are not fired until the user *opts in*. With an **Advanced** deployment (also known as **Standard load**), Google tags send cookieless pings until consent is given; please check this link for more information. The deployment strategy depends on the value of the Consent Model API (Opt-in or Opt-out), which can be configured per country or state in the Consent Manager (CM) settings. It should be mapped in the template under the configuration **Implied Consent Settings**.

NOTE: *Google only requires a validated consent signal as a part of our EU UCP enforcement for these jurisdictions where the EU UCP applies. Implementing in regions beyond may negatively impact campaign performance and is not recommended. To process consent choices in the US, Google has built Restricted Data Processing, a tool to help customers comply with local laws. Not following Google's guidelines could cause significant consequences and drawbacks of implementing consent mode if it is not required. Please review this guidance, to understand geo-based controls for consent mode.*

Implementation

Pre-requisites

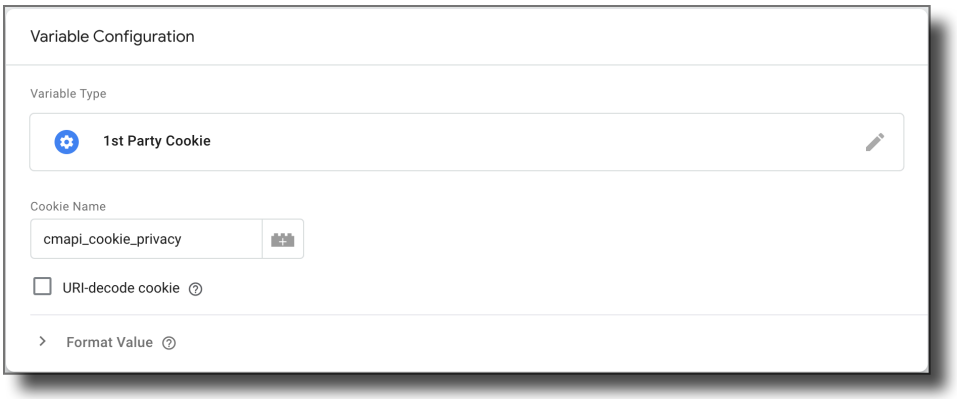
- Google Tag Manager account and integration to the web page
    - Follow the steps in this link on how to set up and install the Tag Manager: https://support.google.com/tagmanager/answer/6103696?hl=en&ref_topic=3441530
    - Follow the steps in this link on how to enable Google Consent Mode on your Tag Manager: https://support.google.com/analytics/answer/9976101?hl=en#:~:text=Enable%20consent%20mode%20for%20websites&text=Website%20developers%20can%20enable%20consent,Manage%20consent%20settings%20(web)
- Configure Google Consent Mode in CCM Advanced when using the JavaScript integration. This is not required when using the Google Tag Template.

Integrating the Consent Mode v.2.0 Using Google Tag Template

Using the community template allows you to easily integrate Google Consent Mode with TrustArc. For best results, ensure that scripts are loaded in the following order:

To implement the Google Consent Mode using the Google Tag Template, follow the steps outlined below:

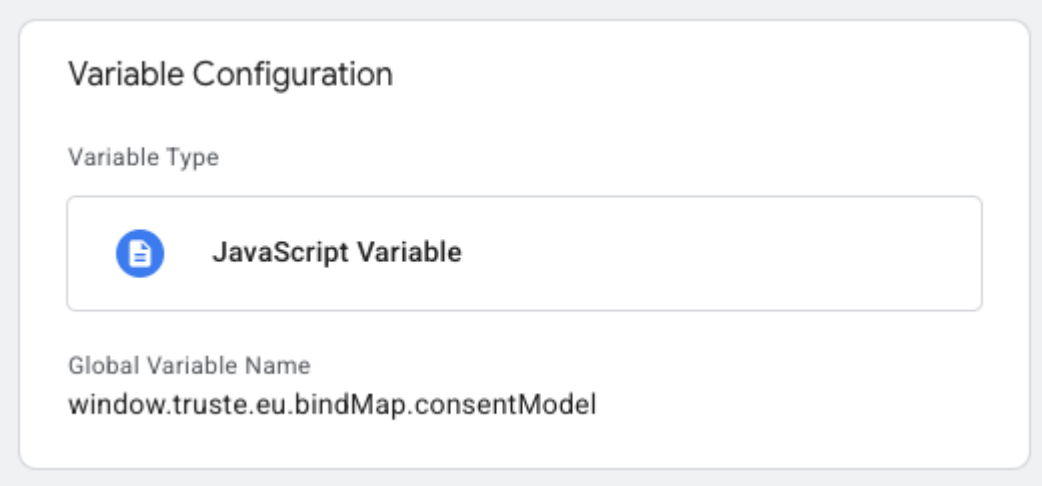
1. On your Google Tag Manager account, go to **Workspace > Variables > User Defined Variables.** Click **New.** Then, click the **Edit** icon to choose a variable type to begin the setup.
2. Under *Page Variables*, select **1st Party Cookie**.
    - Enter *cmapi_cookie_privacy* in the **Cookie Name** field, then save the changes.



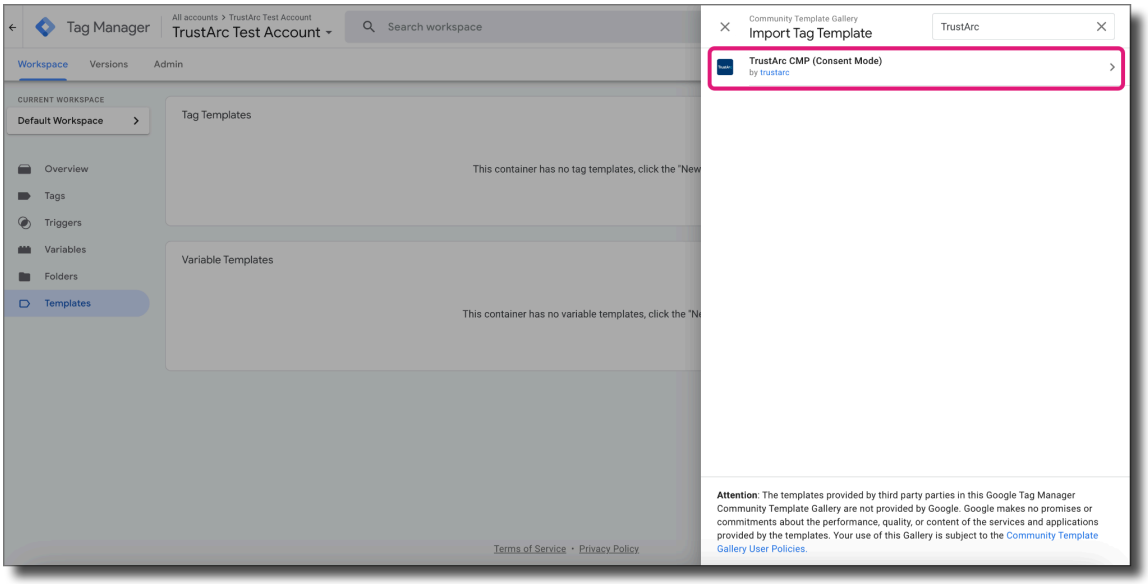
NOTE: If the **Deploy multiple consent managers on the same root domain** feature is enabled, ensure that the CMID is included in the name of the *cmapi_cookie_privacy* cookie.

Example:
`uu5rbz_cmapi_cookie_privacy`

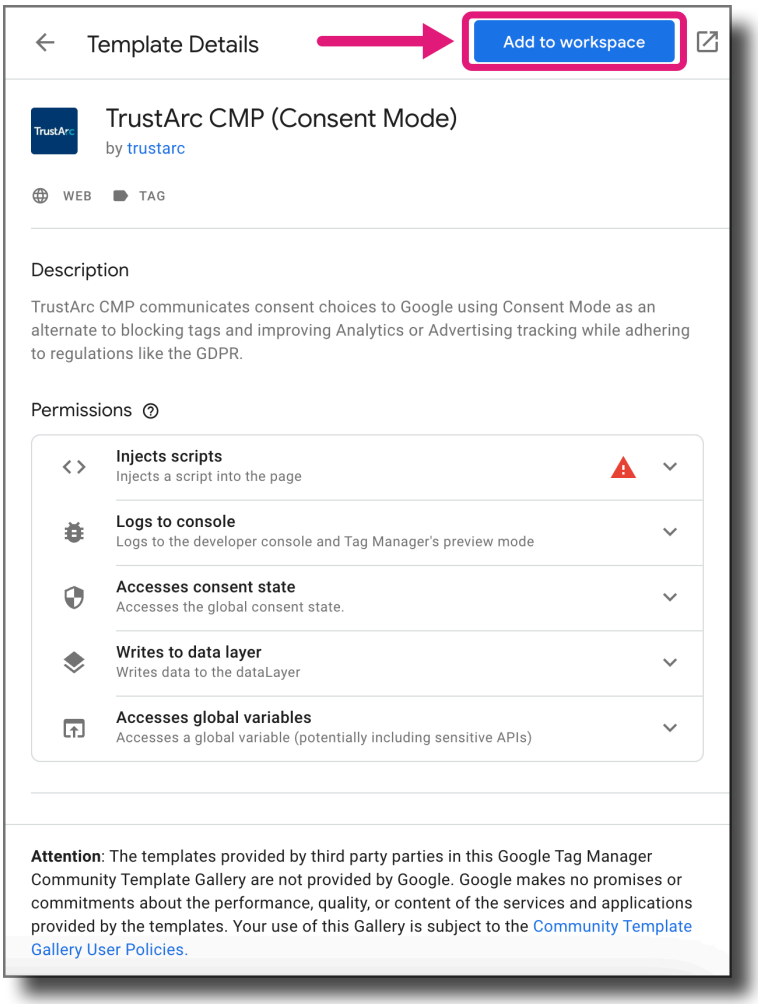
- Next, create a new `JavaScript Variable`, and enter *window.truste.eu.bindMap.consentModel* in the **Global Variable Name** field. Then, save the changes.



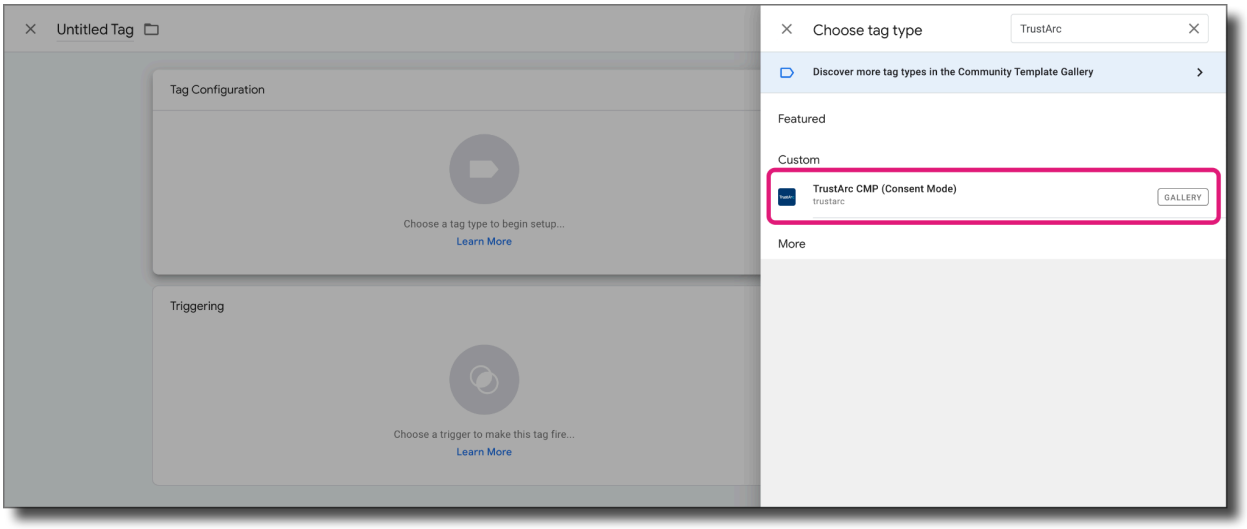
3. Add the **Public TrustArc Template**.
    - Go to **Templates > Tag Templates.** Click **Search Gallery.** Then, click the **Search** icon and enter **TrustArc** in the search bar.



- Click the **TrustArc CMP (Consent Mode)** template from the search result, then click **Add to workspace**.



4. Create a new *Tag Configuration* using the template added in Step 3. Go to the *Tags* section and click **New**. Then, click the **Search** icon, enter **TrustArc**, and click **TrustArc CMP (Consent Mode)**.

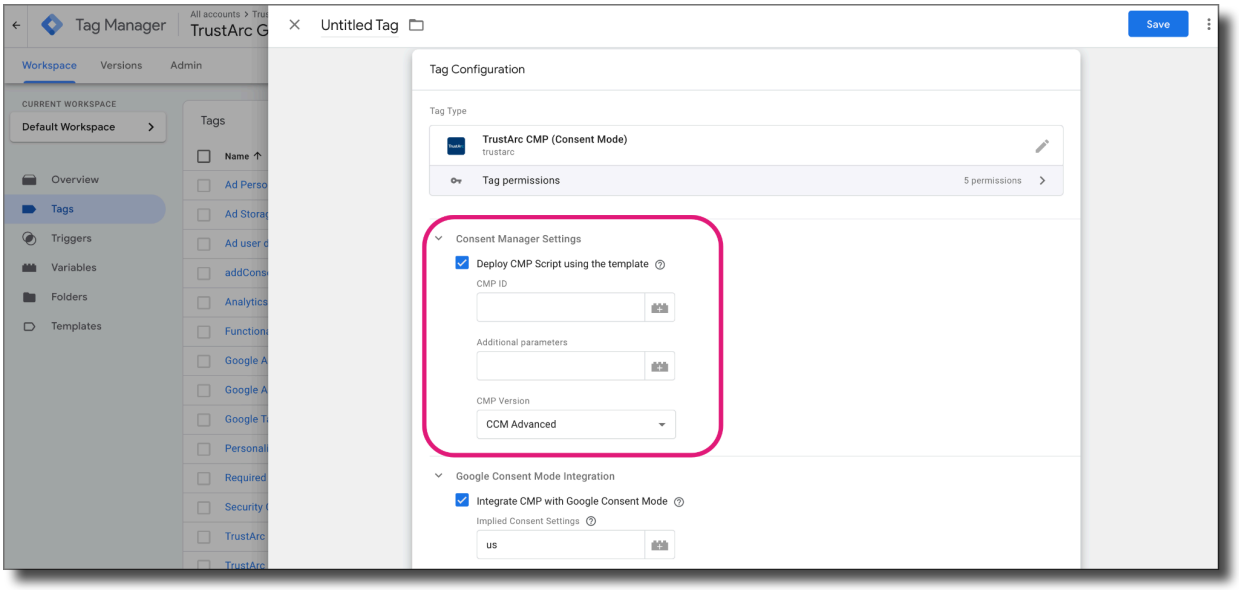


5. *(OPTIONAL)* This step is only necessary if you are deploying the Cookie Consent Manager using the template.

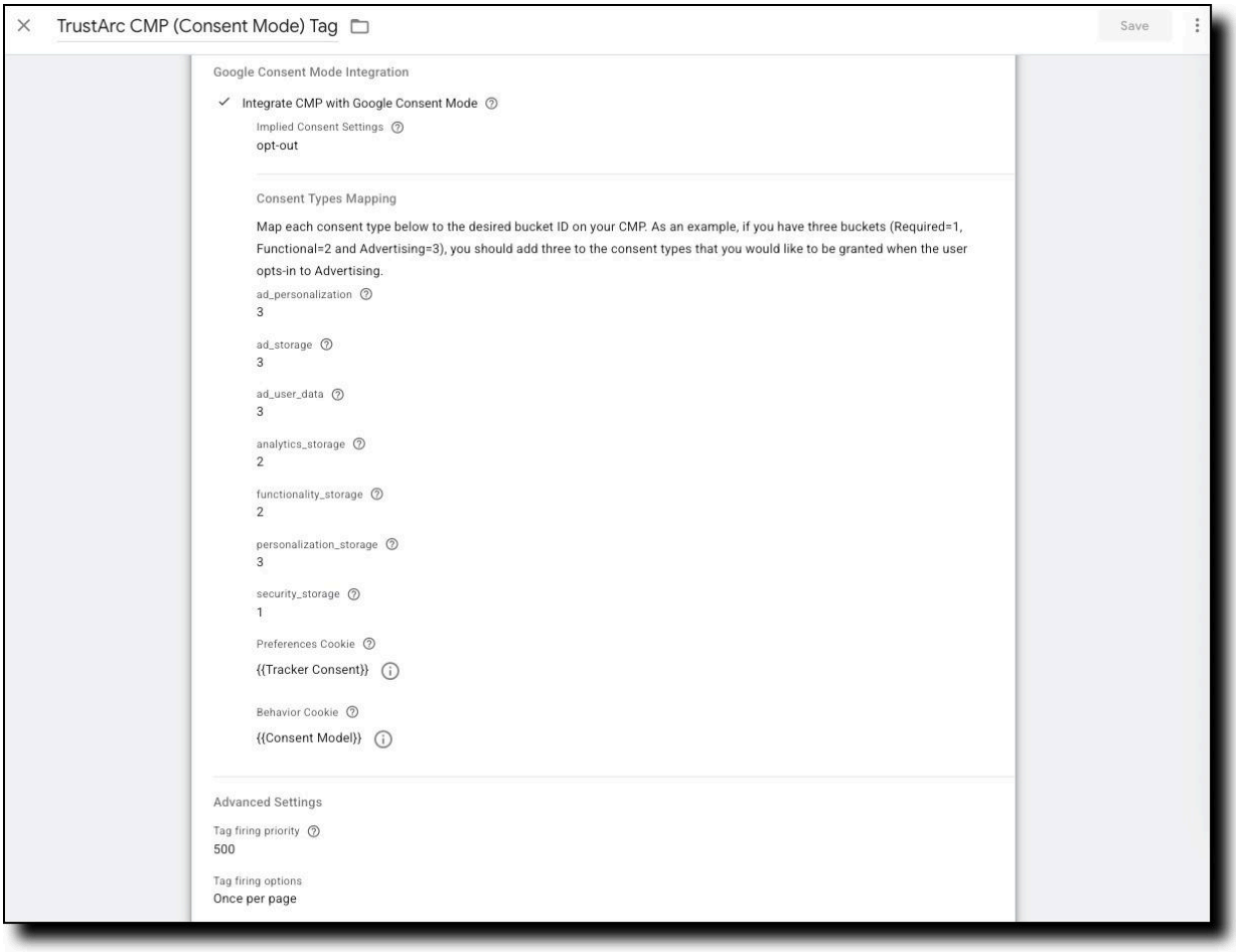
   Under the *Consent Manager Settings*, select the **Deploy CMP Script using the template** checkbox to use the TrustArc template to deploy the CMP script. Then, enter the following:

   - **CMP ID** – The client identifier domain used in the CMP script. Contact your Technical Account Manager (TAM) to verify the correct CMP ID.
     - For CCM Advanced, the CMP ID will be in a domain format: *mydomain.com*.
     - For CCM Pro, the CMP ID will be a randomly generated value in a format similar to the following: *abcxyz*
   - **Additional parameters** – `&c=teconsent&js=nj&noticeType=bb&gtm=1`

   Then, select the **CMP Version** you are using. (For CCM Pro, enter the letters only from the script which can be obtained from Step 4).



6. Under *Google Consent Mode Integration,* select the **Integrate CMP with Google Consent Mode** checkbox to use the TrustArc template to integrate with Google Consent Mode. Then, set the configurations:



- Set the **Implied Consent Settings** according to the configuration used for *Basic* (opted in by default) or *Advanced* (opted out by default). When using the Consent Model variable, just add "opt-out," and it will set the default consent as GRANTED for all locations that are currently mapped as "opt-out" in your consent manager settings. Possible values for the Consent Model are as follows, which represents the consent experience:

    - **Opt-in** - All non-required trackers are blocked and will only fire once the user provides consent.
    - **Opt-out** - All trackers are fired and will only be blocked once the user opts out.

  NOTE: Another option is to use the *notice_behavior* cookie instead of the Consent Model variable. It will store the information used to check the country configurations.

  For example, the default consent for countries configured with *US* manager can be set to **granted**, and the default consent for countries configured with *EU* can be set to **denied**.

  In **CCM Advanced**, the acceptable values (*in lowercase*) are

  - *(us)* - for the United States
  - *(eu)* - for the European Union
  - *(us, none)* - where *'none'* would cover unprovisioned countries.

  In **CCM Pro**, the acceptable values are more diverse, such as:

  - *(na)* — North America
  - *(eu)* — European Union
  - and other country-specific values

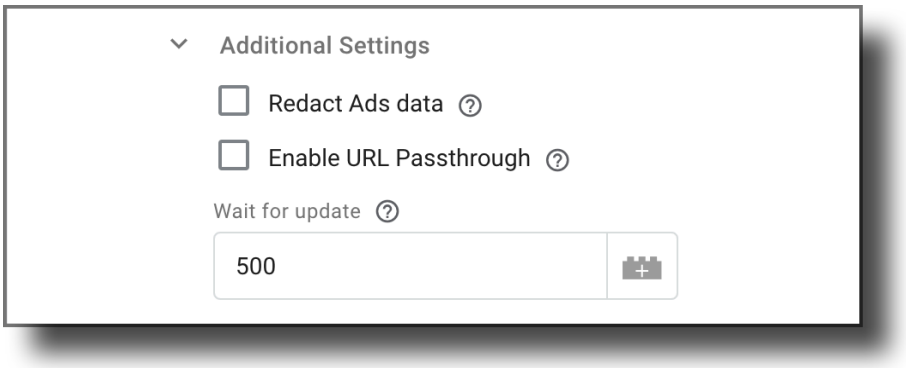
  For a full list of values for CCM Pro, please refer to available values for the *notice_behavior* cookie in the *Implementation Guide*.

  Important: Please review Google's recommendation for setting different settings for regions like EEA and the US.

- **Consent Types Mapping -** This is used for setting the consent state. Map each consent type to the desired bucket ID on the CMP. For example, if you have three buckets *(Required = 1, Functional = 2, and Advertising = 3)*, add 3 to the consent type field that you would like to be granted when users opt-in to *Advertising*.

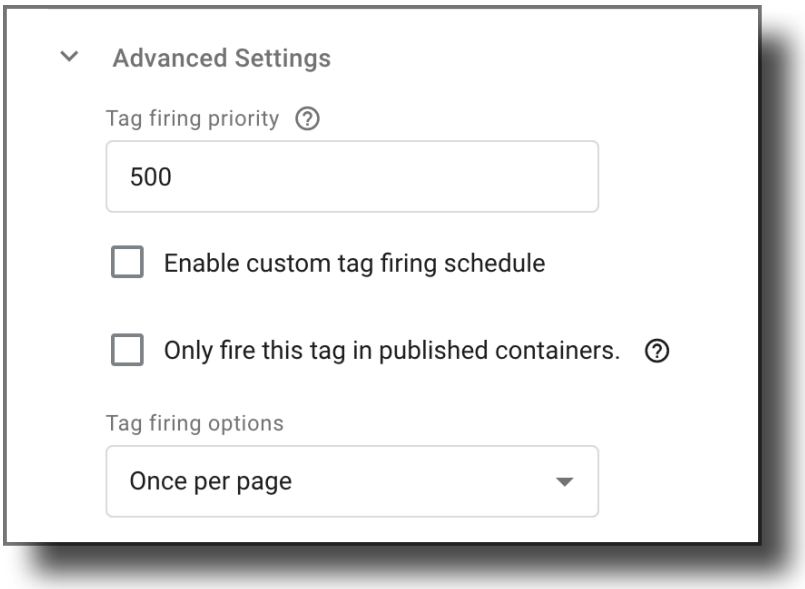
- Set the **Preferences Cookie** to the variable added in Step 2 for *cmapi_cookie_privacy.* This is used to retrieve and set the consent according to the user's preferences.

- Set the **Behavior Cookie** to the variable added in Step 2 for JavaScript Variable, *window.truste.eu.bindMap.consentModel.* This is used to initialize the consent settings according to the user's location and inform which locations are set with opt-in or opt-out experience.
- **Additional Settings:**



  - **Redact Ads Data** - When `ads_data_redaction` is **true** and `ad_storage` is **denied**, ad click identifiers sent in network requests by Google Ads and Floodlight tags will be redacted. Network requests will also be sent through a cookieless domain.
  - **Enable URL Passthrough** - To improve ad click measurement quality when `ad_storage` is **denied**, you can optionally elect to pass information about ad clicks through URL parameters across pages using URL passthrough.
  - **Wait for update** - Specify a millisecond value to control how long to wait before data is sent. This wait is necessary to allow the Cookie Consent Manager to load. The recommended setting is 500 (ms). If you do not set a wait time, Google tags might fire before the consent setting defaults, and you may see a warning message in the console.
- *(OPTIONAL)* Enable the **Fire Custom Event when consent changes** feature if you want TrustArc to fire a custom event for each consent type every time consent is loaded or changed.

7. Update the **Advanced Settings** section for the tag as follows:



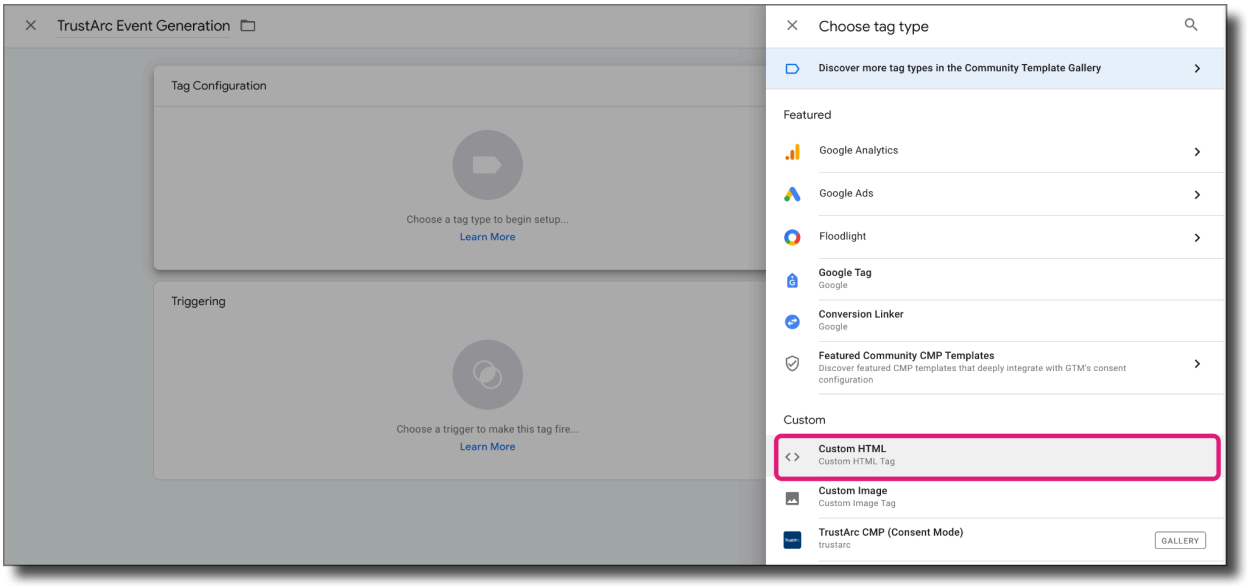
- Define the **Tag firing priority** as *500.*
- Select *Once per page* from the **Tag firing options** dropdown menu.

8. Set the **Firing Triggers** to **Consent Initialization - All Pages.** Enter a name for the tag and click **Save.**

CCM Deployment Script

Deploy the Integration Studio to define a callback function that can be used to detect the changes in the preferences of the users on the website. This script will communicate with the TrustArc template.

1. Go to the *Tags* section, click **New > Tag Configuration > Custom HTML**.



Copy and paste the following JavaScript in the **HTML** field:

https://gist.github.com/trustarctam/31e662f074aff8831283b2f95665977f

JavaScript

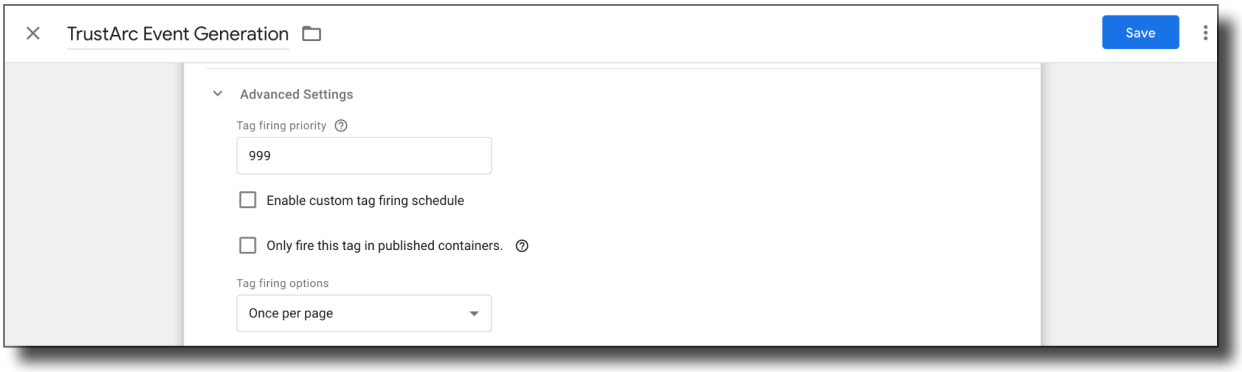
```
<script>
var consentListeners = [];

window.addConsentListenerTA = function (callback) {
  consentListeners.push(callback);
};

var onConsentChange = function (consent, behaviour) {
  consentListeners.forEach(function (callback) {
    callback(consent, behaviour);
  });
};

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") {

          console.log("On consent changed!");

            // Callback function to trigger GCM Template
            //var notice_behavior =
window.truste.eu.bindMap.consentModel;//window.truste.util.readCookie("notice
_behavior") ||
```

```
(truste.eu.bindMap.behavior+','+truste.eu.bindMap.behaviorManager);
          var notice_behavior =
window.truste.util.readCookie("notice_behavior") ||
(truste.eu.bindMap.behavior+','+truste.eu.bindMap.behaviorManager);
            var cmapi_cookie_privacy =
window.truste.util.readCookie("cmapi_cookie_privacy") || '';
          onConsentChange(cmapi_cookie_privacy, notice_behavior);
            self.dataLayer && self.dataLayer.push({"event":"Consent
Changed"});

/*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;
}
});*/
            console.log("After On consent changed!");

        }
    } catch (xx) {

    }
});

var _taInterval;
var _taAttempts = 0;
var _taGoogleTagWasSetLate = function () {

  if(_taAttempts > 50)
    clearInterval(_taInterval);

  if(window.google_tag_data && window.google_tag_data.ics &&
window.google_tag_data.ics.wasSetLate) {
     console.warn("WARNING: Tags are firing before consent is initialized.
Please ensure that the consent mode default is initialized before firing
tags.");
     clearInterval(_taInterval);
   }
```

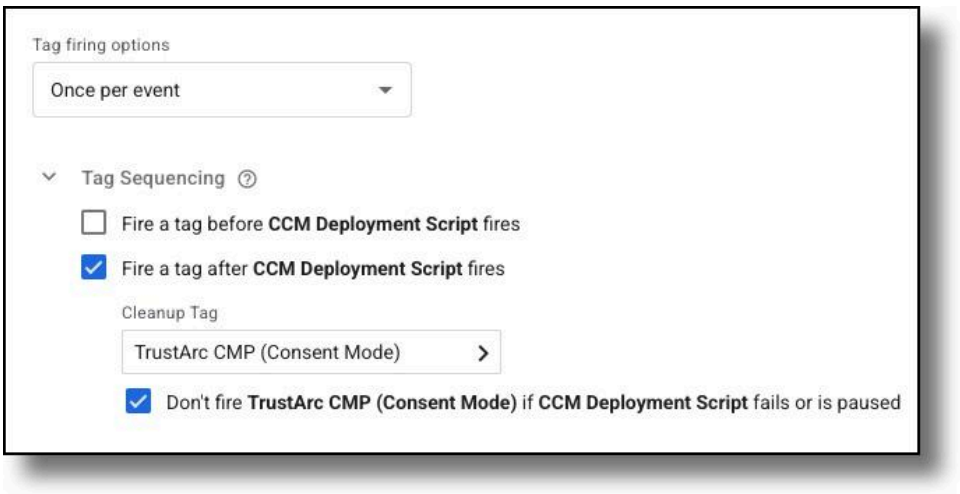
```
  _taAttempts++;
}
_taInterval = setInterval(_taGoogleTagWasSetLate, 200);

</script>
```

2. Update the **Advanced Settings** section for the tag as follows:



- Define the **Tag firing priority** as 999.
- Select *Once per page* from the **Tag firing options** dropdown menu.

Important: ⚠️ The firing priority of the Event GenerationTag must be with a higher value than the Tag Template. This is because event generation creates functions that are required in the template once it is loaded. For example, the tag firing priority for the Event Generation can be 999, and 500 for the Consent Manager template.

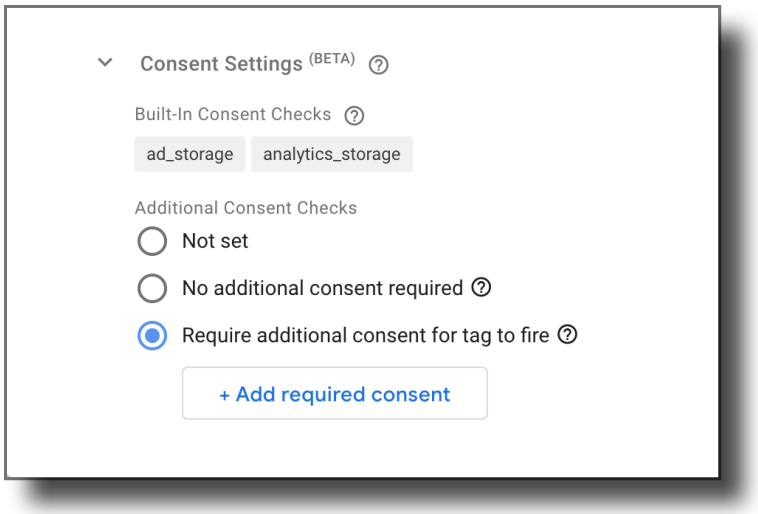


3. Set the **Firing Triggers** to **Consent Initialization - All Pages.** Then, click **Save.**

Review Blocking Triggers

You must review and remove any applied blocking triggers from tags that support Consent Mode (such as Google Ads, Analytics, Floodlight, and Conversion Linker) and review the firing trigger to only load after consent is given like **"All pages."** Please check this site for more information about controlling tags using Consent Mode.

Google Tags are typically created with built-in checks. You can adjust the consent setting by opening a *Tag*, click on **Advanced Settings**, and expand the **Consent Settings** section.



The three options available for **Additional Consent Checks** are:

- **Not set** - The tag remains in the default state with no extra consent checks for the tag.
- **No additional consent required** - Select this option if your tag does not need extra consent checks beyond its built-in ones. In the example, `ad_storage` and `analytics_storage` are built-in the tag.
- **Require additional consent for tag to fire** - Select this option to ensure that the tag only

activates if all specified consent types are **granted**. Use the **+ Add required consent** button to add more consent types supported by your Consent Management Platform.

Integrating the Consent Mode Using JavaScript (gtag Implementation)

When you have installed the TrustArc tag on your site, you can enable the Google Consent integration by adding the following code snippet to your website template, below the `gtag.js` or GTM code itself.

Using Google Consent requires a pre-defined order of events. For more information, please check this link.

1. Load Gtag to set the default consent
2. Load the Cookie Consent Manager script
3. Load the Tag Manager script
4. Update Gtag using the update consent

NOTE 1: Please review the *sample notice script* in the table below. Your Technical Account Manager can help with a finalized script customized for your website and will help you map the Consent Types according to your cookie bucketing.

NOTE 2: If you are not using Google Tag Manager to manage your tags, you must manually set consent with gtag. To do this, you can inspect the dataLayer and the URL parameters in the *Network* tab of the *Developer Tools*. For consent types not being used, please omit them from the code below. These methods are only applicable to Google products and are described in the Google Consent Mode settings available on this link.

NOTE 3: Replace `AW-YYYYYY` with the ID of the Google Tag ID property to which you want to send data.

NOTE 4: If you are a CCM Pro customer, you may refer to this link for a sample code for the event listener used in the Google Consent Mode Integration. The event listener is responsible for sending the gtag consent 'default' and 'update', and also for mapping the consent types with the cookie buckets. You can also reach out to the support at support@trustarc.com if you have any questions.

SAMPLE NOTICE SCRIPT FOR CCM ADVANCED

Copy the code snippet from the Gist into your site inside a script tag.

https://gist.github.com/trustarctam/6cf12f2869e956aab5eb23c5e94ecf02

```
const WAIT_FOR_UPDATE = 500; // Whether to wait for an update after the default initialization
const IMPLIED_LOCATION = ""; // Default implied location. Can be set to eu, us, opt-out, etc.
const CONSENT_CONFIG = "notice_behavior"; // "consent_model" (JavaScript) or "notice_behavior" (Cookie)

// This is where you can configure the Google Consent Mode and the default values for the consent initialization
// Bucket Mapping
const REQUIRED = 1;
const FUNCTIONAL = 2;
const ADVERTISING = 3;

const consentTypesMapped = {
    "security_storage"          : REQUIRED,
    "functionality_storage"     : FUNCTIONAL,
    "ad_personalization"        : ADVERTISING,
    "ad_storage"                : ADVERTISING,
    "ad_user_data"              : ADVERTISING,
    "analytics_storage"         : ADVERTISING,
    "personalization_storage"   : ADVERTISING
};
```

JavaScript

```
<!-- sample notice script -->
<script type="text/javascript" async="async"
src="https://consent.trustarc.com/notice?domain=<yourdomain.com>&c=teconsent&
text=true&country=gb&gtm=1&js=nj&noticeType=bb"></script>

<!-- Global site tag (gtag.js) - Google Ads: CONVERSION_ID -->
<script async src="https://www.googletagmanager.com/gtag/js?id=AW-YYYYYY">
</script>

<!-- TrustArc CMP will send updates using Gtag -->
```

Make sure to add the above code below your gtag.js or GTM code snippet, and if you have chosen to rename the dataLayer, ensure that you change `window.dataLayer` to the actual name of the dataLayer.

Relevant values:

- **ad_storage** – Controls cookie behavior for advertising purposes, including conversion measurement. If a user does not provide consent for ads cookies, Google tags will not use cookies for advertising purposes
- **analytics_storage** – Controls analytics cookie usage
- **ad_personalization** – Controls whether data can be used for ads personalization
- **ad_user_data** – Controls whether personal data is sent to a Google core platform service
- **functionality_storage** – Enables storage that supports the functionality of the website or app such as language settings
- **personalization_storage** – Enables storage related to personalization such as video recommendations
- **security_storage** – Enables storage related to security such as authentication functionality, fraud prevention, and other user protection
- **ads_data_redaction (optional)** – When `ads_data_redaction` is **true** and `ad_storage` is **denied**, ad click identifiers sent in network requests by Google Ads and Floodlight tags will be redacted. Network requests will also be sent through a cookieless domain. Not included in the update on consent.
- **wait_for_update** – Since `notice` script loads asynchronously, it might not always run before the Google Tags (when and where `notice` tag is placed can affect the timing). To account for this, specify 'wait_for_update' along with a millisecond value to control how long to wait before sending data.

You may change the values of the default consent types like `ad_storage` and `analytics_storage` to **granted** and set `ads_data_redaction` to **false** if you want to default to an opt-in before the end-user has submitted consent, for example allowing for default opt-in under CCPA.

Validating the Consent Mode Operation

To verify if the Consent Mode is operating as expected, you need to use the tag assistant. For more information, please check this site.

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.

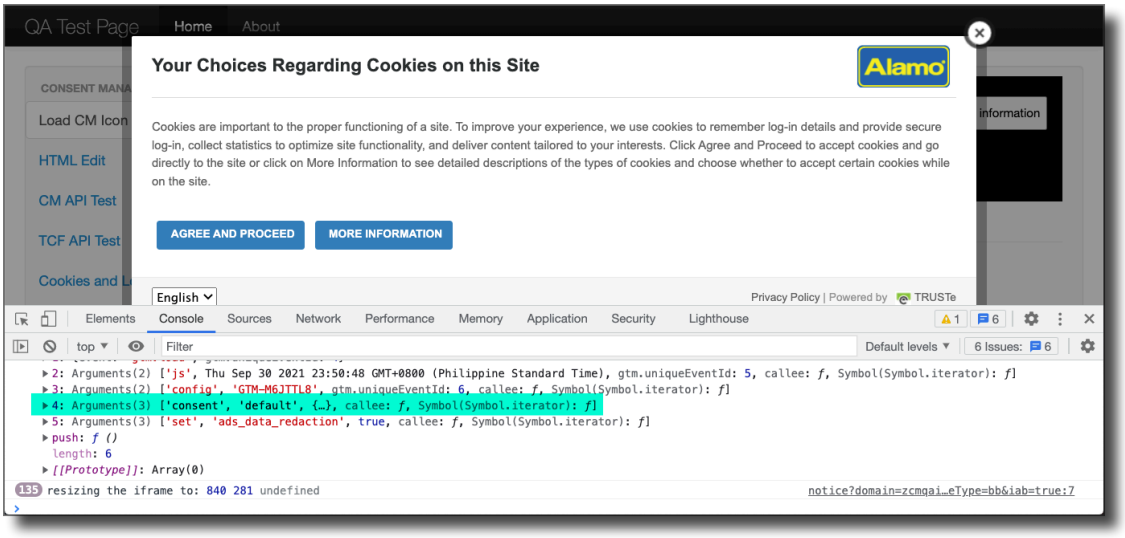
Validating Integration Using the Javascript

To verify if Consent Mode is operating as expected when you are not using the public template, please follow the steps below.

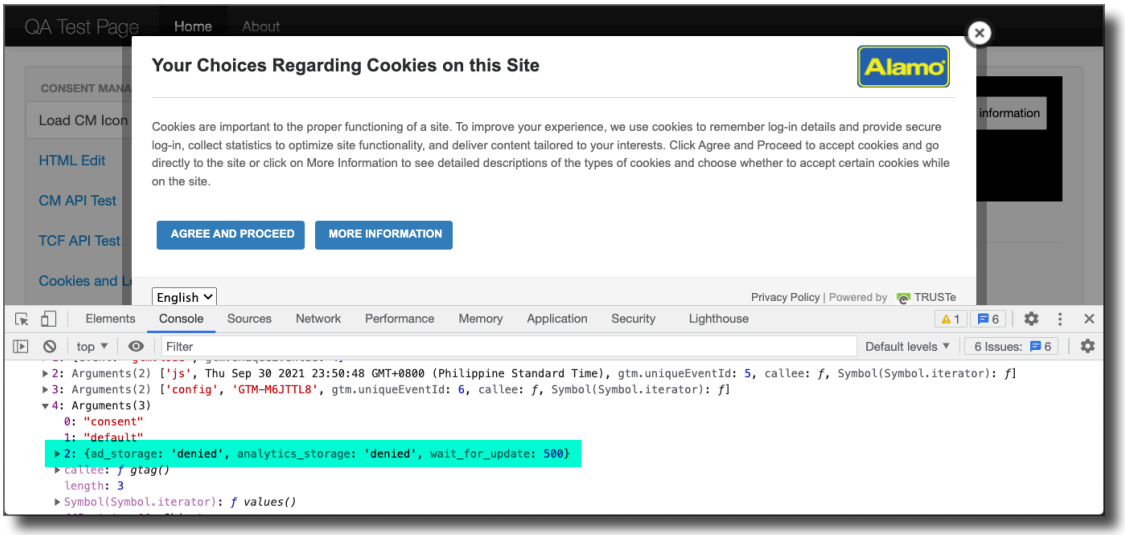
When a page loads without an existing consent cookie and prior to the consent event, the default settings should be registered in the **dataLayer**. This item should have the strings **consent** and **default** and the values like `ad_storage` and `analytics_storage` defined as **granted** or **denied** according to the user's preferences.

To check the initial settings of the dataLayer, follow these steps:

1. In the *Inspect Element* tool, click the **Console** tab.
2. On the command line, type **dataLayer** and press the **Enter** key.
3. Click the chevron icon next to the line that contains **'consent'** and **'default'**.



The default values for *the consent types* are specified in the code snippet. Refer to Integrating the Consent Mode Using JavaScript section.



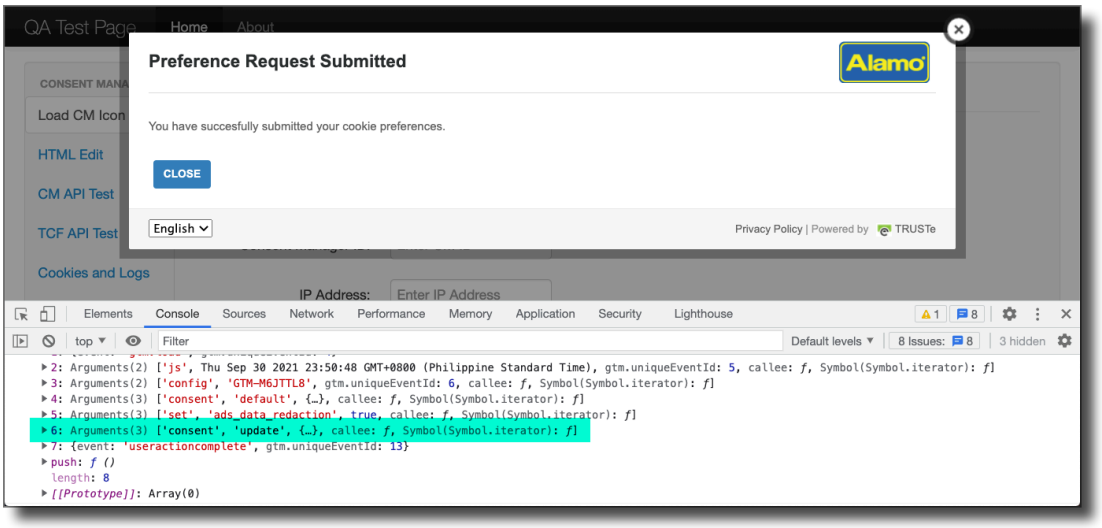
Important: To ensure consent preferences are appropriately honored, once a tag script or code has been executed on a page (i.e., a tag has fired), you will need to configure an automatic page refresh in order for the new consent preference to be reflected. Failure to execute a refresh will result in 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.

Checking the dataLayer Object After Consent is Given

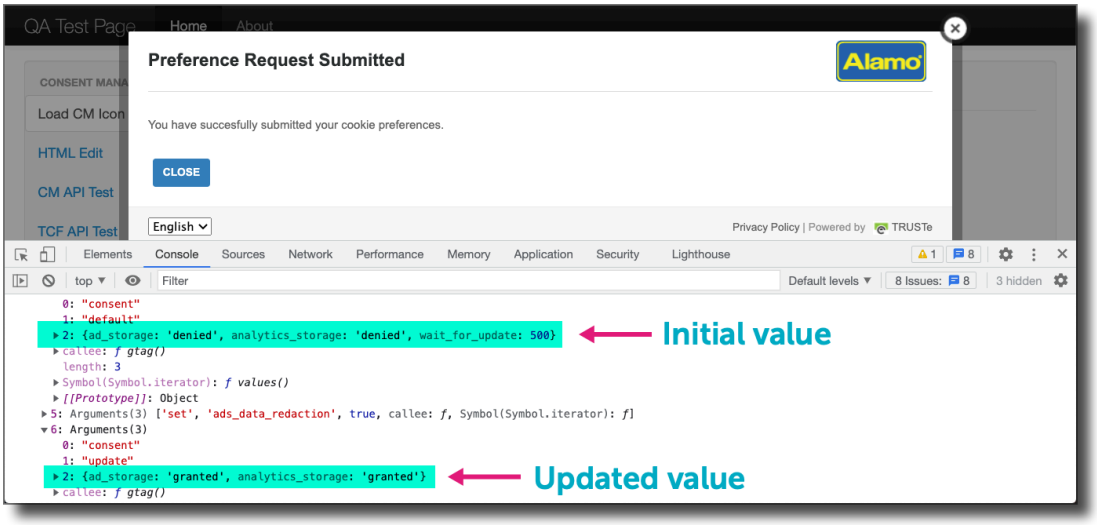
When a page loads with an existing cookie or when a consent event occurs, the dataLayer with strings **'consent'** and **'update'** will indicate values for the consent types like *ad_storage* and *analytics_storage* according to what settings and consent are submitted by the end-user. Please refer to Google's documentation for the latest instructions.

To verify the consent in the dataLayer, follow these steps:

1. In the *Inspect Element* tool, click the **Console** tab.
2. On the command line, type **dataLayer** and press the **Enter** key.
3. Click the chevron icon next to the line that contains **'consent'** and **'update'**.



After consent has been given for all types of cookies, the dataLayer will contain both the initial and updated Consent Mode settings.



Integrating IAB TCF 2.2

Internet Advertising Bureau (IAB) Transparency and Consent Framework (TCF) works as a system for communicating the state of user consent between first parties (i.e. publishers), third parties (i.e. advertisers), and the TrustArc Consent Management Platform (CMP) in use on your website. TCF is a set of technical specifications and policies that help publishers and advertisers comply with the EU's General Data Protection Regulation (GDPR). This is an optional integration available to clients that are publishers.

Clients have the choice of whether to use or not use this integration. When set to true, Google will infer Consent Mode for ad_storage, ad_user_data, and ad_personalization consents from the TC string. For more information about the IAB TCF 2.2 integration, please reach out to your Technical Account Manager.

To integrate IAB TCF, add the following script at the top of every page before the Tag Manager script:

None

```
<script>
   window['gtag_enable_tcf_support'] = true;
</script>
```

Cookie Consent Manager Advanced can also be configured to set the `enableAdvertiserConsentMode` field to true in the TCData object.

Important: When integrating with IAB TCF, the Cookie Consent Manager Script and the IAB stub should be loaded before Google Consent mode is initialized using the template or using the default is initialized through the gtag default consent. To check the latest updates from this integration, please check this site.