

Consent Management Platform (CMP) – 1.1.0

Full Documentation

(Effective May, 2018)

Introduction

This document contains instructions to install the Rakuten Consent Management Platform (CMP) on your site. The CMP is a JavaScript function that can conditionally collect consent from a user, contingent on that user falling within GDPR regulation. Read each section carefully and follow the instructions in order (top to bottom).

Only one CMP should be installed on any website at a given time. Please ensure that there is not already a CMP installed before you begin the installation instructions below.

Implementing the CMP

This CMP code needs to be implemented at the top of the <body> section of your HTML and should be placed on every page of the site, including any conversion pages. Including this code in a common <body> section of your website can facilitate implementation if that is an option.

The CMP Code

The CMP code is provided below and can be customized to fit your needs. This version contains default values and can be pasted directly on your site without edit.

```
<script>
var commandQueue = [];
var cmp = function(command, parameter, callback) {
  commandQueue.push({
    command: command,
    parameter: parameter,
    callback: callback
  });
};
cmp.commandQueue = commandQueue;
cmp.config = {
  // customPurposeListLocation: '',
  layout: "modal",
  blockBrowsing: true,
  storePublisherData: false,
  storeConsentGlobally: true,
  logging: false,
  localization: {},
  forceLocale: null,
  gdprAppliesGlobally: false,
  repromptOptions: {
    fullConsentGiven: 360,
    someConsentGiven: 30,
    noConsentGiven: 30
  },
  geoIPVendor: 'https://cdn.digitrust.mgr.consensu.org/1/geoip.json',
  testingMode: 'normal',
  layout: "footer",
  showFooterAfterSubmit: true,
  // logoUrl: "https://www.example.com/image.png",
  css: {
    "color-primary": "#0a82be",
    "color-secondary": "#eaeaea",
    "color-border": "#eaeaea",
    "color-background": "#ffffff",
    "color-text-primary": "#333333",
    "color-text-secondary": "#0a82be",
    "color-linkColor": "#0a82be",
    "color-table-background": "#f7f7f7",
    "font-family": "'Noto Sans', 'Helvetica Neue', Helvetica, Arial, sans-serif",
    "custom-font-url": "https://fonts.googleapis.com/css?family=Noto+Sans",
  }
};
window.__cmp = cmp;
</script>
<script src = 'https://cdn.digitrust.mgr.consensu.org/1/cmp.complete.bundle.js' async>
</script>
```

The code above will provide a consent window to all visitors falling within GDPR regulation, in cases when such visitors haven't provided any consent information yet OR their given consent information is outdated.

Pre-defined and defaults can, be changed by editing the configuration values. Please refer to section "Configuring the CMP" in order to understand what needs to be configured.

GDPR requires to allow any visitor to review their choices. Please refer to section "Implementing the manual Trigger" for more details.

Configuring the CMP

Configuration Parameters

The CMP code can be customized to fit your needs. The version above contains default values and can be pasted directly on your site without edit. However, the following parameters can be customized if needed:

- `blockBrowsing` - see [Customizing the look and feel of the tool](#)
- `css` – see [Customizing the look and feel of the tool](#)
- `customPurposeListLocation` – see [Customizing purposes and vendors](#)
- `forceLocale` – see [Localization of CMP](#)
- `gdprAppliesGlobally` – see [Controlling to when to load CMP tool](#)
- `geoIPVendor` – see [Controlling to when to load CMP tool](#)
- `layout` – see [Customizing the look and feel of the tool](#)
- `localization` – see [Localization of CMP](#)
- `logging` – see [Enabling more information \(for developers\)](#)
- `logoUrl` – see [Customizing the look and feel of the tool](#)
- `repromptOptions` – see [Controlling to when to load CMP tool](#)
- `showFooterAfterSubmit` – see [Customizing the look and feel of the tool](#)
- `storeConsentGlobally` – see [Storing of consent data](#)
- `storePublisherData` – see [Customizing purposes and vendors](#)
- `testingMode` – see [Controlling when to load the CMP tool](#)

Customizing purposes and vendors

storePublisherData (Boolean):

Tells the CMP if it should load the custom purposes file and store the custom purpose consent data in a first party cookie. The location of the custom purposes file must be specified in the customPurposeListLocation variable.

True	Custom purposes will be loaded from customPurposeListLocation and the purpose consent information will be written into a cookie on the domain that the CMP is loaded from (This cookie is unique from the vendor consent cookie).
False	The customPurposeListLocation will not be requested and no cookie will be written with custom purpose consent data.

customPurposeListLocation (String):

URL location of the JSON file that holds a list of custom purposes a user can select. These will be pulled into the purposes list in the CMP.

```
E.G. customPurposeListLocation: 'https://www.publisher.com/purposes.json'
```

The JSON should be formatted according to the following pattern:

```
{
  "version": 1,
  "purposes": [
    {
      "id": 1,
      "name": "Custom Purpose 1",
      "description": "Here's a description of the first purpose"
    },
    {
      "id": 2,
      "name": "Custom Purpose 2",
      "description": "Here's a description of the second purpose"
    }
  ]
}
```

Storing of consent data

storeConsentGlobally (Boolean):

Tells the CMP if it should sign the consent data collected from users as a third-party cookie (on the consensu.org domain) or as a first-party cookie. If a user has disabled third-party cookies (or the browser doesn't support third-party cookies), this will always fallback to writing as a first-party cookie.

True	Consent data is written to a cookie on the consensu.org domain (accessible by every CMP registered with the IAB)
False	Consent data will be written to a cookie on the domain that the CMP is loaded from.

Controlling when to load the CMP tool

gdprAppliesGlobally (Boolean):

Controls if the CMP is shown to every site visitor regardless of location or if it is only shown to EU users.

True	The CMP will bypass the user localization logic and show to every user without a current consent cookie
False	The CMP will only show to users without current consent cookies that are deemed to be in the EU

repromptOptions (Dictionary)

Configures when to reprompt the user for consent after they've last received a cookie, in days.

Note: All three sub-configuration values must be specified, even if some are defaults.

fullConsentGiven	Honored if a user has given full consent to every purpose and partner (Int) Range: 0 – 390, Default: 360
someConsentGiven	Honored if a user has given consent to only some purposes and partners Range: 0 – 390, Default: 30
noConsentGiven	Honored if a user has given no consent at all Range: 0 – 390, Default: 30

geoIPVendor (String)

Specifies the location of the vendor responsible for IP to Geo translation.

testingMode (String)

Allows the site owner to bypass normal CMP rendering logic for testing purposes. When fully implemented, this variable should be set to 'normal'. "always show" and "never show" should only be used during installation to confirm that the code has been implemented properly.

"always show"	CMP will always load on pages where it has been implemented
"never show"	CMP will never load unless manually triggered by a button or console command
"normal"	CMP will load normally according to the rest of the configuration settings

Customizing the look and feel of the tool

layout (String):

Controls how the CMP looks when it loads, either in "modal" mode, "footer" mode, or "thin footer" mode.

"modal"	The CMP will load in its "banner" mode in the center of the page.
"footer"	The CMP will load in its "footer" mode at the bottom of the page.
"thin"	The CMP will load in its "thin footer" mode at the bottom of the page.

blockBrowsing (Boolean):

Controls if the CMP blocks users from interacting with the rest of the site before a consent decision has been made.

True	The CMP will block users from interacting with the site before a consent decision has been made. The website will be grayed out behind the CMP, regardless of the layout that has been configured.
False	Users will be allowed to interact with the website while the CMP is still showing. The CMP will persist on the site until the user makes a consent decision.

showFooterAfterSubmit (Boolean):

Controls if the CMP loads the recall footer after a user submits their consent preferences. Set this to False if you are going to setup and style your own manual trigger button or link somewhere on your site.

True	The CMP will load the recall footer after a user interacts with the CMP
False	The CMP will not load the recall footer after a user interacts with the CMP

logoUrl (String):

Sets the URL location of the logo image that will load in the CMP. If this variable is not included or if the value is Null, then no image will be loaded in the tool. If the tool is configured to load in “footer” mode, the image is forced to be a square shape. If the tool is configured to load in the “thin footer” mode, the image will not show, even if one is specified.

```
E.G. logoUrl: "https://www.example.com/image.png"
```

css (Dictionary):

Defines the colors and font of the tool. Inputs for the color values must follow standard CSS formatting. The "font-family" CSS must be specified for the any custom fonts to take effect.

```
E.G. css: {
  "color-primary": "#0a82be",
  "color-secondary": "#eaeaea",
  "color-border": "#eaeaea",
  "color-background": "#ffffff",
  "color-text-primary": "#333333",
  "color-text-secondary": "#0a82be",
  "color-linkColor": "#0a82be",
  "color-table-background": "#f7f7f7",
  "font-family": "'Noto Sans', 'Helvetica Neue', Helvetica, Arial, sans-serif",
  "custom-font-url": "https://fonts.googleapis.com/css?family=Noto+Sans",
},
```

Localization of the CMP

forceLocale (String):

Allows the site owner to manually select a language for the CMP text. The input must be an ISO language code available in either the default languages or the Localization dictionary. The input string will be truncated after the first “-“ (e.g., “fr-be” becomes “fr”)

If empty, or not specified, the CMP will attempt to render in the language specified in the user’s browser.

Localization (Dictionary):

Localization Dictionary allows the site owner to specify the wording of the CMP in each language. Language sets specified in the dictionary will override the default dictionary for that language. Please follow the below structures to prepare texts for your language.

Please note that any language customizations must adhere to policy guidelines and translations into the EEA languages will need to be provided.

```

en: {
  intro: {
    title: 'Thanks for visiting ',
    domain: '',
    description: 'Ads help us run this site. When you use our site selected companies may access and use information on
your device for various purposes including to serve relevant ads or personalised content.',
    deviceInformationPopover: `
      <strong>Information that may be used:</strong>
      <ul>
        <li>Type of browser and its settings</li>
        <li>Information about the device's operating system</li>
        <li>Cookie information</li>
        <li>Information about other identifiers assigned to the device</li>
        <li>The IP address from which the device accesses a client's website or mobile application</li>
        <li>Information about the user's activity on that device, including web pages and mobile apps
visited or used</li>
        <li>Information about the geographic location of the device when it accesses a website or mobile
application</li>
      </ul>
    `,
    purposesPopover: `
      <strong>Purposes for storing information:</strong>
      <ul>
        <li>Storage and access of information</li>
        <li>Ad selection and delivery</li>
        <li>Content selection and delivery</li>
        <li>Personalisation</li>
        <li>Measurement</li>
      </ul>
    `,
    acceptAll: 'OK, Continue to site',
    showPurposes: 'Learn more'
  },
  details: {
    title: 'Privacy preferences',
    back: 'Back',
    save: 'OK, Continue to site',
    showVendors: 'Show all companies',
    enableAll: 'Enable all',
    disableAll: 'Disable all'
  },
  purposes: {
    active: 'Active',
    inactive: 'Inactive',
    disclaimer: 'We and selected companies may access and use information for the purposes outlined. You may customise
your choice or continue using our site if you are OK with the purposes. You can see the ',
    disclaimerVendorLink: 'complete list of companies here.',
    showVendors: 'Show companies',
    hideVendors: 'Hide companies',
    featureHeader: 'This will include the following features:',
    company: 'Company'
  },
  vendors: {
    company: 'Company',
    offOn: 'Allow',
    description: 'Companies carefully selected by us will use your information. Depending on the type of data they
collect, use, process and other factors, certain companies rely on your consent while others require you to opt-out. For information
on each partner and to exercise your choices, see below. Or to opt-out, visit the ',
    or: 'or ',
    sites: ' sites.',
    description2: 'Customise how these companies use data on the ',
    description2Link: 'previous page.',
    description3: 'You can control the preferences for all companies by ',
    description3Link: 'clicking here.'
  },
  footer: {
    message: 'Read more about access and use of information on your device for various purposes',
    deviceInformationHeader: 'Information that may be used:',
    deviceInformation: `
      <ul>
        <li>Type of browser and its settings</li>
        <li>Information about the device's operating system</li>
        <li>Cookie information</li>
        <li>Information about other identifiers assigned to the device</li>
        <li>The IP address from which the device accesses a client's website or mobile application</li>
        <li>Information about the user's activity on that device, including web pages and mobile apps
visited or used</li>
        <li>Information about the geographic location of the device when it accesses a website or mobile
application</li>
      </ul>
    `,
    purposesHeader: 'Purposes for storing information:',
    purposes: `
      <ul>
        <li>Storage and access of information</li>
        <li>Ad selection and delivery</li>
        <li>Content selection and delivery</li>
        <li>Personalisation</li>
        <li>Measurement</li>
      </ul>
    `
  }
}, fr: {
  . . .
}

```

Enabling more information (for developers)

logging (String|Boolean):

Configures the logging for the CMP. Makes it possible to enable the CMP to provide more information about the events and actions it performs.

"debug"	Used only for development/testing. Logs nearly every event occurring in the CMP to stdout
"info"	Logs to stdout when the CMP is processing a command or sending out a notification to event listeners.
"warn"	Logs to stdout when there is something that prevents the CMP from working entirely as intended, e.g. cookies are disabled or config values are invalid.
"error"	Logs to stdout whenever an error occurs that causes the CMP to not work at all, e.g. the vendors list not loading or not being able to write cookies to a domain.
True	Equivalent to "debug"
False	Disables logging

Implementing the Manual Trigger

GDPR requires that consent tools make it as easy to withdraw consent as it was to give it. In this spirit, this tool can be recalled at any point by placing a manual trigger on the page.

Manual Trigger Code

To include a trigger to show the CMP window at any time, include an element anywhere on the page with an "onclick" event with the value of `window.__cmp('showConsentTool')`

```
<button onclick = "window.__cmp('showConsentTool')"> Revisit Consent Settings </button>
```

Other CMP Commands

The CMP is also able to perform a handful of functions intended to speed up the development process and QA

CMP Commands

showConsentTool – core command that when executed calls and displays CMP window for the visitor.

```
<button onclick = "window.__cmp('showConsentTool')"> Show Consent Tool </button>
```

renderCmpIfNeeded – this command is a helper to control how often each visitor is provided with CMP options. It takes configuration defined within `repromptOptions` dictionary. Website owners can, however, implement their own rendering of CMP by utilizing core `showConsentTool` command (see below).

```
<button onclick = "window.__cmp('renderCmpIfNeeded')"> Load CMP If Needed </button>
```

getConsentData – command allowing user to retrieve consent data as a JSON object via handler function. Response contains whether GDPR is applicable to visitor, if consent has a global scope, and the consent string itself

```
window.__cmp('getConsentData', null, function(result) {console.log(result)});
```

Output of above example would be written in console of browser developer tools:

```
{gdprApplies: true, hasGlobalScope: true, consentData: "BOOHN7tOOHN7tABABAENAY-  
AAAAQB7_____b9_3__7v9uT7Kr_K7VfxiHGQr2hGVA8KBOAo"}
```

getPublisherConsents – Similar to `getConsentData`, but only returns the list of publishers that a user has consented to.

```
window.__cmp('getPublisherConsents', null, function(result) {console.log(result)});
```

getVendorConsents – Similar to getConsentData, but only returns the list of vendors that a user has consented to.

```
window.__cmp('getVendorConsents', null, function(result) {console.log(result)});
```

ping – Used to determine if the CMP is active on a page

Returns:

```
{  
  gdprAppliesGlobally: true/false  
  cmpLoaded: true  
};
```

addEventListener – allows publishers to add a callback that will be executed when certain events occur within the CMP.

```
window.__cmp('addEventListener', 'onSubmit', function(response) { console.log(response) })
```

Possible events to bind to:

onSubmit	Fired when the user finalizes their decision in the UI and closes the modal
consentNotRequired	Fired when the CMP determines the user does not need to see the CMP.
cmpReady	Fired when the CMP is loaded and ready.
consentStringUpdated	Fired when the user makes any sort of written change to a consent string.

gdprInScope – returns an object with additional information about the state of the CMP, and why it is in that state.

```
window.__cmp('gdprInScope', null, function(response) { console.log(response) })
```

Information contained in the returned object:

cmpShown	Has the cmp been shown yet? (If the CMP doesn't need to show (i.e. the user already has a consent cookie) this will be false)
----------	---

gdprAppliesGlobally	Is the CMP configured to always show regardless of location/language?
gdprAppliesLanguage	Is the CMP showing because the user's browser language is set to a location in the EEA?
gdprAppliesLocation	Is the CMP showing because the user's IP address resolved to a location in the EEA?
submitted	Has consent been submitted?

decodeMetadata – decodes the metadata string from the getVendorConsents command.

```
window.__cmp('decodeMetadata', null, function(response) { console.log(response) })
```