Salesforce Krux Implementation Guide

Contents

[Overview 2](#_Toc520113702)

[Prerequisites 2](#_Toc520113703)

[Control Tag Guide 3](#_Toc520113704)  
Sample Control Tag 3  
Implementing Control Tag 3  
Verify the Control Tag 4  
Control Tag Helper 4

[Policy Regime 5](#_Toc520113706)  
Policy Regime Association method 6  
Consent Defaults 7

[GDPR Salesforce Overview and Consent Management 9](#_Toc520113711)  
Consent Management 9  
JavaScript Tag for Salesforce DMP Consent 10  
Steps to implement consent via Tealium 12  
Test the Consent Tag 13

[Event Pixel Guide 14](#_Toc520113712)  
Creating Events in Krux Console 14  
Event Types 15  
Event Pixel Verification 15

Appendix [16](#_Toc520113713)Code for VMware Global Control Tag 16  
Code for VMware Global Consent Tag 16  
Control Tag Helper Bookmark URL 17

**Overview**

This document is a generic Technical Specifications document, intended as the standard guide for capturing Analytics’ data into Salesforce Krux DMP. It provides guidance to assist the relevant teams in implementation. Most of this document's content is technical in nature as the intended audiences will be implementing the solution and validating the implementation.

**Prerequisites**

Following are the prerequisites required before implementing Krux Control Tag on any web property:

* To abide by GDPR policy, Evidon Consent Banner should be present on the web property to take user’s opt-in or opt-out choices depending on the region, the user belongs to.
* Data Layer is present and actively populates attributes. (Assumption)  
  (Data Layer implementation is not mandatory.)

**Control Tag Guide**

Control Tag is asynchronous and is added to website pages to activate the Salesforce DMP. The JavaScript code should be placed in the <head> section of website pages to take advantage of all platform features.

**Generate Control Tag:**

To create the Control Tag for any site, use the Salesforce DMP Console. Visit Manage-->Sites in the top navigation of Salesforce DMP Console.

**Sample Control Tag**

<!-- BEGIN Salesforce DMP ControlTag for " YourSampleSite " -->

<script class="kxct" data-id="<*YourSampleSiteID*>" data-timing="async" data-version="3.0" type="text/javascript">

window.Krux||((Krux=function(){Krux.q.push(arguments)}).q=[]);

(function(){

   var k=document.createElement('script');k.type='text/javascript';k.async=true;

   k.src=(location.protocol==='https:'?'https:':'http:')+'//[cdn.krxd.net/controltag/](http://cdn.krxd.net/controltag/)<*YourSampleSiteID*>';

   var s=document.getElementsByTagName('script')[0];s.parentNode.insertBefore(k,s);

}());

</script>

<!-- END Salesforce DMP ControlTag -->

**Implementing the Control Tag:**

* Get Control Tag(s) for the website by visiting Manage-->Sites in the top navigation of Salesforce DMP.
* Click the </> (Actions) button on the right.
* Choose JavaScript then click the Copy to Clipboard button.

**Implementing the Control Tag using Tealium:**

* Add Krux Control Tag from the list of Tags provided in Tags section of the desired profile.
* Enter the ConfID of respective Control Tag from Krux Console.

**Verify the Control Tag**

1.     Navigate to a page on your site where the Control Tag is deployed.

2.     Click the Chrome Menu at the top right of your Chrome browser window.

3.     Select  More Tools > Developer Tools.  
 The Developer Tools pane will open at the bottom of your browser.

4.     At the top of Developer Tools pane are several tabs. Click the Network tab.

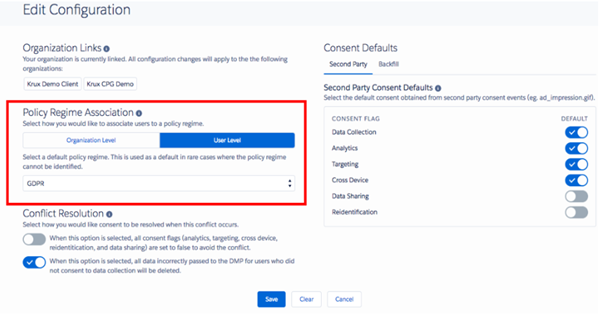
5.     Type “pixel.gif” and Filter the beacons.

6.     Verify that you see a “pixel.gif” from the domain “[beacon.krxd.net](http://beacon.krxd.net)” in the item list.

**Control Tag Helper**

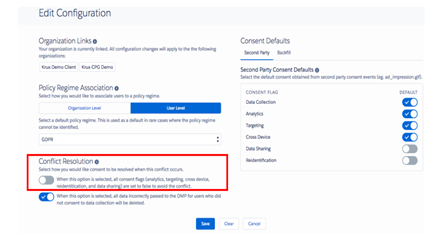
* Control Tag Helper can be used only if Salesforce DMP UI is accessible.
* It can be used to validate the Control Tag on the page, as well as see other information captured via the Control Tag.
* To bookmark the Control Tag Helper, log into Salesforce DMP and navigate to Manage > Control Tag Helper.
* Drag the hyperlink into your browser toolbar or create a new one in your toolbar and manually enter the code provided.
* If the Control Tag did not load, or if it does not exist on the page you are viewing, you will likely see "No Control Tag Detected".

**Policy Regime**

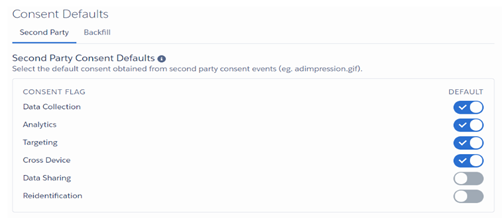
* Consumer Rights can be established and enforced by many different entities.
* There are nationally enforced consumer rights which differ region by region, and there is industry enforced consumer rights from regulators like the DAA and NAI.
* Since each individual may be governed under a different set of policies depending on the geo-location, the DMP introduces the construct of Policy Regime.
* Policy regime is a specific set of privacy policy regulations that apply for a specific user.
* For e.g. Users from EU region are governed by GDPR, while users from USA are governed by Global Standards.
* So, the DMP associates a policy regime with every single device that is observed in each account.
* There are two Association Methods for Policy Regime covered under Salesforce DMP:
  + **Organizational Level.**
  + **User Level**.
  + **Organizational Level**: If in DMP, the policy regime is set at organizational Level, it uses a single set of policy fir all the devices associated with the organization irrespective of the regions.
    - For example, if a customer selects the organization level and sets the policy regime for their organization to GDPR, all devices would be treated under the GDPR, even if the users are from the United States.
    - **User Level**: If in DMP, the policy regime is set at the user level, the DMP evaluates the geographic region associated with the device and applies the appropriate policy regime.
      * For E.g. users in the EU are managed by the GDPR rules, but USA users aren’t.
      * In absence of a user-provided policy regime signal from UI, the DMP will rely on geo location data to associate the user to the appropriate policy regime.
* **Policy Regime Source:**
  + **default** – It is used when the system is unable to get policy regime for a user/organization through all possible methods. This is the last resort setting.
  + **client-config**– These are client specific configuration from the UI. It is used for organizations that associate policy regime at the organization level and when policy regime cannot be derived through either geo-ip or UI request.
  + **geo-ip** - Based on ip lookup for the data subject in question. This is used when the policy regime association is done at the user level and nothing was provided by the user as consent.
  + **UI request**- This is used when the policy regime association is done at the user level and policy regime was specified by the user in the request through API or Consent Tags.
* **Setting Policy Regime Association Method & Defaults**.
  + Select Manage Consumer Rights.
  + You will land on the Consent Report and Policy Regime Setting Page.
  + Select Edit Configuration. The below window will appear.  
                                               
* Here We can select either Organization Level or User Level Policy Regime based on our organization’s traffic diversity.
* For VMware, User Level settings will serve better as the default settings as User Level settings will allow DMP to identify Users based on geo-ip and then apply the respective consent Rules, either GDPR or Global Standards.
* After selecting a default policy regime, we can select either GDPR or Global Standards from the dropdown. This option will be used to assign policy only in case where the policy regime for a user can’t be identified.
  + For example, in any rare case, if geo location of user is not available and if User Level Policy Regime is in action for DMP, then It will be unable to apply Consent Policy based on the geo region. In those cases, setting a default policy will help to assign a policy for such users.
* **Consent Resolution defaults** - cd, sh and re flags require segment processing i.e., ‘al’.
  + To resolve the consent conflict, this button is used.

**True** - This option sets all consent flags to true to avoid the conflict.

**False** - This option sets all consent flags false to avoid the conflict.



* Defining Second Party Consent Defaults:



           These consent flags contain default values based on the global standard policy regime.

**Recommendations:**

* For VMware, since we have user spread across the globe, consisting of regions abiding GDPR and Global Standards as well, we should apply User Level Policy Regime in DMP Default Settings.
* For users, whose consent has not been passed from UI or their consent are not available for DMP, then DMP will apply consent Policy Based on the geo region of the users.
* Apart from Policy Regime Settings, for the users, whose consent aren’t available to DMP nor DMP can locate its geo-location, in such cases we should stick to GDPR policy.
* For 2nd party data, such as ad\_impression.gif, which doesn’t rely on the DMP Control tag to fire, may defy the consent policy and report data to the DMP. For such cases, we can set 2nd Party consent as well for all the 6 parameters in the Consent Management console. Since, we are serving EU as well, default settings should follow GDPR specifications.

**GDPR Salesforce Overview and Consent Management**

* **Consent Management:**

- To obtain consent from User before collecting data about them and to honor user’s request how to use their data, we use Consent Management.

- Salesforce DMP provide 5 ways to record and manage consent of users. These 5 methods are:

                                                               i.      API.

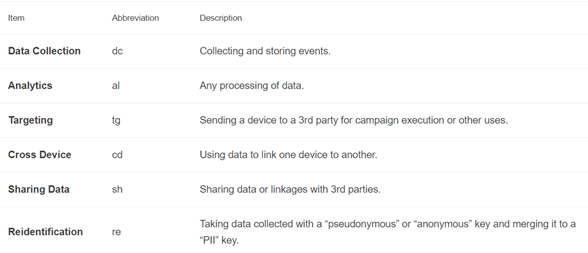
                                                               ii.      JavaScript Consent Tag.

                                                              iii.      SDK.

                                                              iv.      Inside DMP console.

                                                              v.      S3 File Upload.

* Below are the Consent Flags used to indicate specific consent given by a user:



These flags are either set to 0 or 1 while passed from a user to DMP.

- When user agrees to the consent either by clicking I Agee button (For EU region on VMware) or by navigating through the site, a consent Tag must be sent from the UX to the salesforce DMP to record the user level consent for that user

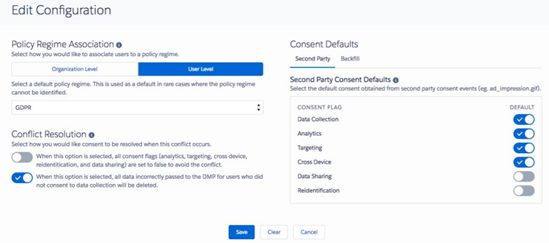
  An example of JavaScript Consent Tag:



       Refer. **JavaScript Consent Tag Format** Section for more Details.

       Once the above Krux consent tag fires, a beacon is sent to DMP which sets the consent of the user.  
      C:\40caba5fcae84b162e17bfac1c55a0b8

* 1. In these cases, the DMP sets default consent based on the applicable policy regime as follows:  
     



* **JavaScript Tag for Salesforce DMP Consent :**  
    
  **JavaScript Consent Tag Format:**

**Krux ('ns:NAMESPACE', 'consent:ROUTE', parameters, callback);**

**NAMESPACE** — the namespace used on the DPM Customer's site. (vmwarecm2 for VMware profile)

**ROUTE** — the type of API request. The four available routes are get, set, remove, and portability

            Required route for Boolean values is 'set'.

**Parameters** - For the JavaScript Consent Tag, you should use the following format to set the consent flag parameters:

**dc**: true, (Data Collection)

**tg**: true, (Targeting)

**al**: true, (Analytics)

**cd**: true, (Cross-device linking)

**sh**: false, (Data Sharing)

**re**: true (Re-identification)

**Callback** - A function that will be called after the asynchronous request completes. For 'set' route callback function is optional.

**Function format:**

**i.      Errors object** — An object containing any errors encountered in validating the request.

**ii.      body object** — An object that contains the settings object and the following fields:

* **idt string** — The type of identifier, must be bk (bridge key) or device.
* **dt** string — The type of the idv (e.g., idfa).
* **idv** string — An ID value, which may be a Krux cookie, first party ID, advertiser ID (from Android), IDFA (from iOS), bridge key value (e.g., hashed), or some other identifier.
* **bk** string — Bridge key name; the attribute name of the bridge key passed in. This can be any bridge key attribute defined in your account.

1. Example:

                             Krux('ns:mynamespace', 'consent:set', {

                             idt: 'device',

                             dt: 'idfa',

                             idv: '6D92078A-8246-4BA4-AE5B-76104861E7DC',

                             dc: true,

                             al: true,

                             tg: true,

                             cd: true,

                             sh: false,

                             re: false

                             });

* **Consumer Rights Management:**
  1. **Consent Summary:** Shows how many users you collected a consent signal from, and how many users you do not have a consent setting for.
  2. **Consent Distribution:**  Shows the count of users that gave consent for each consent category.
* **Configuration:**
  + 1. **Organization Links** - Information about all the Organization links for your account.
    2. **Policy Regime Association** - User level/Organization level & GDPR/Global Standards.
    3. **2P Consent defaults** - Defining desired consent defaults for each consent flag.
    4. **Consent Resolution defaults** - cd, sh and re flags require segment processing i.e. **al**.

To resolve the consent conflict, this button is used.

**True** - This option sets all consent flags to true to avoid the conflict.

**False** - This option sets all consent flags false to avoid the conflict.

* + 1. **Deleting Erroneous Data Collection -** When this option is selected, all data incorrectly passed to the DMP for users who did not consent to  data collection will be deleted.

**Steps to implement Consent Via Tealium.**

1. In the Tealium Profile for microsites, Create an extension with Type advanced JavaScript.
2. Create the Scope as All Tags.
3. Paste the below code in the editor, save and publish in dev/QA to test the same.

Code: { Krux('ns:AccountNameSpace', 'consent:set', {

dc: true,

al: true,

tg: true,

cd: true,

sh: false,

re: false

}, function(errors, body) {

if (errors) {

console.error(errors);

} else {

}

});

**Test the Consent Flags:**

1. On page load of domain pages, open Network Tab.
2. Filter requests with consent/set.
3. A network request like below will be visible:

<https://consumer.krxd.net/consent/set/e36c883f-24ff-4249-96ed-c50da4ed1a5b?idt=device&dt=kxcookie&dc=1&al=1&tg=1&cd=1&sh=0&re=0&callback=Krux.ns.vmwarecm2.kxjsonp_consent_set_1>

1. The status of the request will be 200.
2. In query String parameter, all the 6 flags can be verified.

**Event Pixel Guide**

**1.** **Events Guide in Krux** :

* Event-based framework allows users to create dynamic segmentation strategies by capturing user gesture, click, and behavior as an ‘Event’.
* There are two use cases for Events :  
  1.Capturing clicks, link clicks, form submit etc.  
  2. Pages where JavaScript snippet can't be used, ex- emails, newsletters, partner website where krux JavaScript Tags aren’t allowed.
* Used to understand user interactions on the page, building and targeting against a particular audience that interacts with the page in specific ways.
* Places where Events are used:  
                        1. Site Interactions.  
                        2. Shopping Carts.  
                        3. Selecting Preferences.  
                        4. Subscription Process and Newsletters.  
                        5. Emails.

**2. Creating Events in Krux**:

          1. Manage-> Capture -> Events.  
          2. Click on Create Event.  
          3. Fill in Event Name.  
          4. Select Event Type from Dropdown. (krux click tracker can forward to a URL).  
          5. Define Event Category for classification.  
          6. Select Conversion event, if that event is participating in attribution Modeling.  
          7. To send additional info with the event, create event attributes.  
          8. click save and Generate code.

**3. Event Types:**

In the create Event Tab, we have Event Type Drop down which has predefined type of events. Brief on the events present are mentioned below:

1. Ad – Event type selected when an Ad click needs to be tracked.  
2. Click – Any custom click event can be tracked by creating Click event type.  
3. Form – Any form Submit event can be tracked.  
4. Page View – Any page view can be selected with this condition.  
5. Purchase – successful Purchase of a product can be covered in event type.  
6. Resource Download – Any file download like pdf, exe etc. can be covered.  
7. Social – To track social button clicks as an event.  
8. Subscriptions – To track a subscription of any newsletter etc.  
9. Video – Video events.  
10. Krux Click Tracker - Used in the case where after clicking on an email or subscription letter, user is redirected to any landing page.

**3. Generating JavaScript Code Snippet for Events**:

* The JavaScript Code snippet will work only on pages with Salesforce DMP control Tags present on it.
* There are many event types present in the drop down of the Event Type section while creating Events.
  1. Some the common ones are ad, click, form, registration, subscription, social etc.
  2. One of them is Krux click Tracker, which is used in the case where after clicking on an email or subscription letter, user is redirected to any landing page.
* After clicking save and Get Code, we’ll get the following screen.
* The Image shows an event code for Krux Click Tracker event type.
* Below is the code snippet generated for Events.
* Multiple attributes can also be passed via custom events.
* To do so, define event attributes and generate the event code. That attribute will be dynamically assigned when placed on the page where it will be called.

 Using Krux Event Pixel Tag in Tealium:

1. Tealium has Krux event pixel Tag which can be added.
2. In the Krux event pixel Tag, event ID, event Type and Client Id has to be configured, which will be provided by the Salesforce DMP.
3. Then event attributes can be mapped in data Mapping Tab, once they are created in Salesforce DMP.

-For Custom Events, an extension can be created in Tealium and custom JavaScript code can be written.

**4.** **Event Pixel Verification**:

           1. Navigate to the page with Event Tag implemented on it.  
           2. Open Developer’s Console and then Network Tab.  
           3. Type “event.gif” in the search Box.  
           4. Look for the beacon with domain name “[beacon.krxd.net](http://beacon.krxd.net)”.  
           5. Verify the corresponding attributes in the beacon.  
           6. Alternatively, same can be verified using krux Console Debugger.

**APPENDIX**