Integrate the Platform Web SDK to process customer consent data using the Adobe 2.0 standard

The Adobe Experience Platform Web SDK allows you retrieve customer consent signals generated by Consent Management Platforms (CMPs) and send them to Adobe Experience Platform whenever a consent-change event occurs.

The SDK does not interface with any CMPs out of the box. It is up to you to determine how to integrate the SDK into your website, listen for consent changes in the CMP, and call the appropriate command. This document provides general guidance on how to integrate your CMP with the Platform Web SDK.

Prerequisites

This tutorial assumes that you have already determined how to generate consent data within your CMP, and have created a dataset containing consent fields that has been enabled for Real-time Customer Profile. To learn more about these steps, see the overview on consent processing in Experience Platform before returning to this guide.

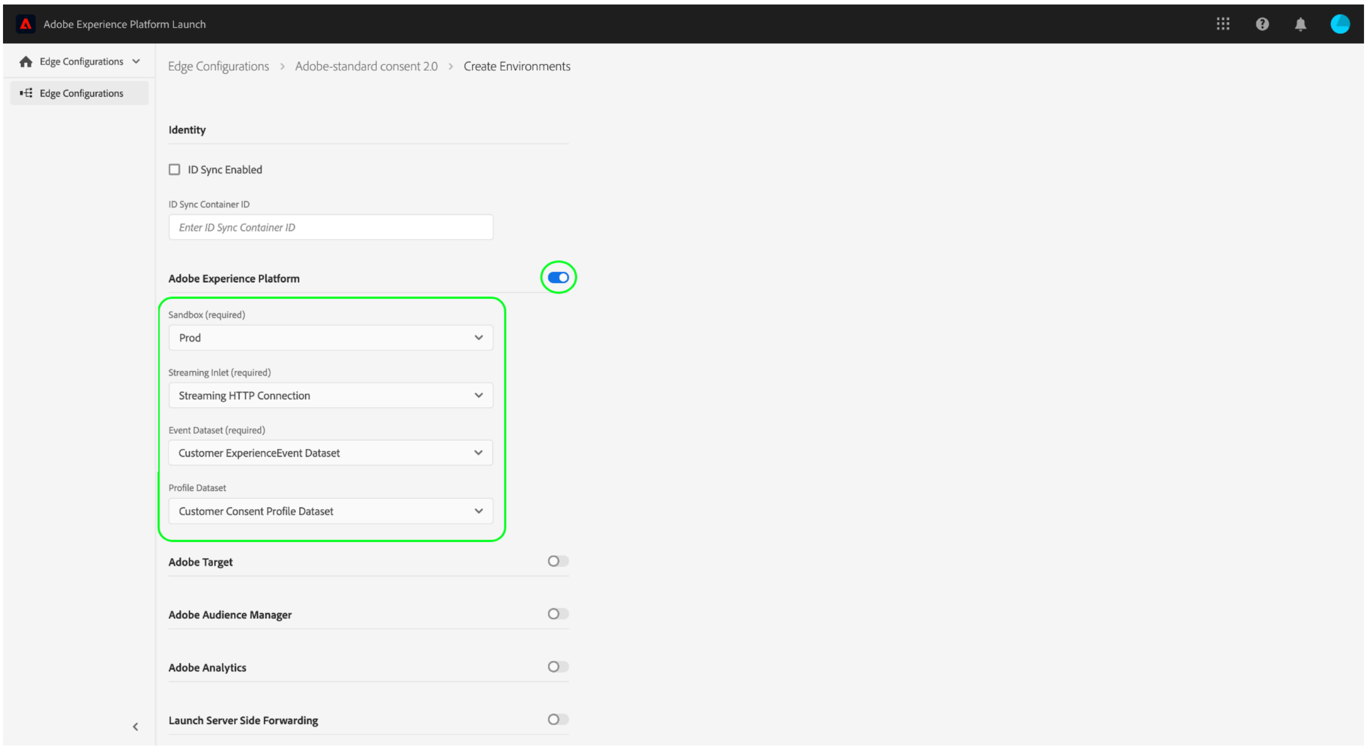
In addition, this guide requires a working understanding of Adobe Experience Platform Launch extensions and how they are installed in web applications. Refer to the following documentation for more information:

- Platform Launch overview
- Quickstart guide
- Publishing overview

Set up an edge configuration

In order for the SDK to send data to Experience Platform, you must have an existing edge configuration for Platform set up in Adobe Experience Platform Launch. In addition, the [!UICONTROL Profile Dataset] you select for the configuration must contain standardized consent fields.

After creating a new configuration or selecting an existing one to edit, select the toggle button next to **[!UICONTROL Adobe Experience Platform]**. Next, use the values listed below to complete the form.


Adobe Experience Platform Launch
Edge Configurations
Edge Configurations
Edge Configurations > Adobe-standard consent 2.0 > Create Environments

Identity

☐ ID Sync Enabled

ID Sync Container ID: Enter ID Sync Container ID

Adobe Experience Platform ●

Sandbox (required): Prod

Streaming Inlet (required): Streaming HTTP Connection

Event Dataset (required): Customer ExperienceEvent Dataset

Profile Dataset: Customer Consent Profile Dataset

Adobe Target ○

Adobe Audience Manager ○

Adobe Analytics ○

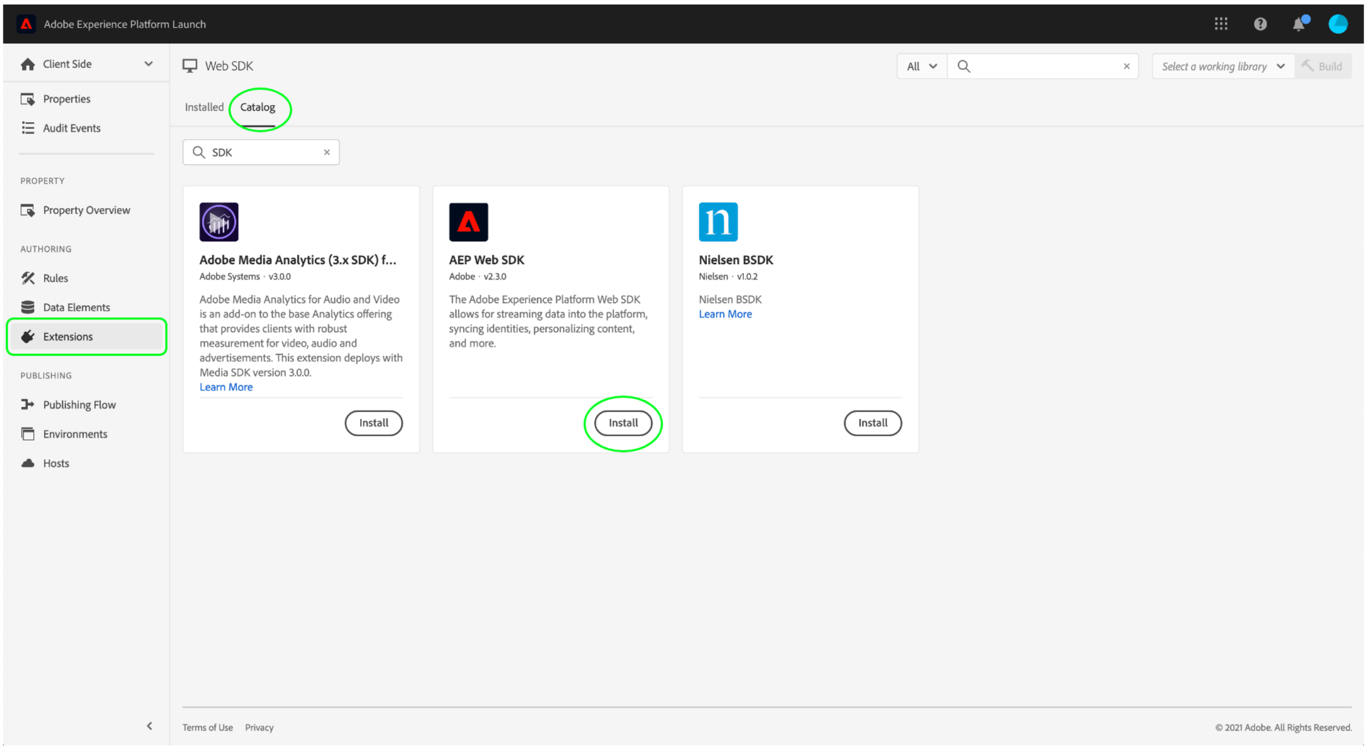
Launch Server Side Forwarding ○

Edge configuration field	Value
[!UICONTROL Sandbox]	The name of the Platform sandbox that contains the required streaming connection and datasets to set up the edge configuration.
[!UICONTROL Streaming Inlet]	A valid streaming connection for Experience Platform. See the tutorial on creating a streaming connection if you do not have an existing streaming inlet.
[!UICONTROL Event Dataset]	An [!DNL XDM ExperienceEvent] dataset that you plan on sending event data to using the SDK. While you are required to provide an event dataset in order to create a Platform edge configuration, please note that sending consent data directly via events is not currently supported.
[!UICONTROL Profile Dataset]	The [!DNL Profile]-enabled dataset with customer consent fields that you created earlier.

When finished, select **[!UICONTROL Save]** at the bottom of the screen and continue following any additional prompts to complete the configuration.

Install and configure the Platform Web SDK extension

Once you have created an edge configuration as described in the previous section, you must then configure the Platform Web SDK extension that you will ultimately deploy on your site. If you do not have the SDK extension installed on your Platform Launch property, select **[!UICONTROL Extensions]** in the left navigation followed by the **[!UICONTROL Catalog]** tab, and then select **[!UICONTROL Install]** under the Platform SDK extension within the list of available extensions.

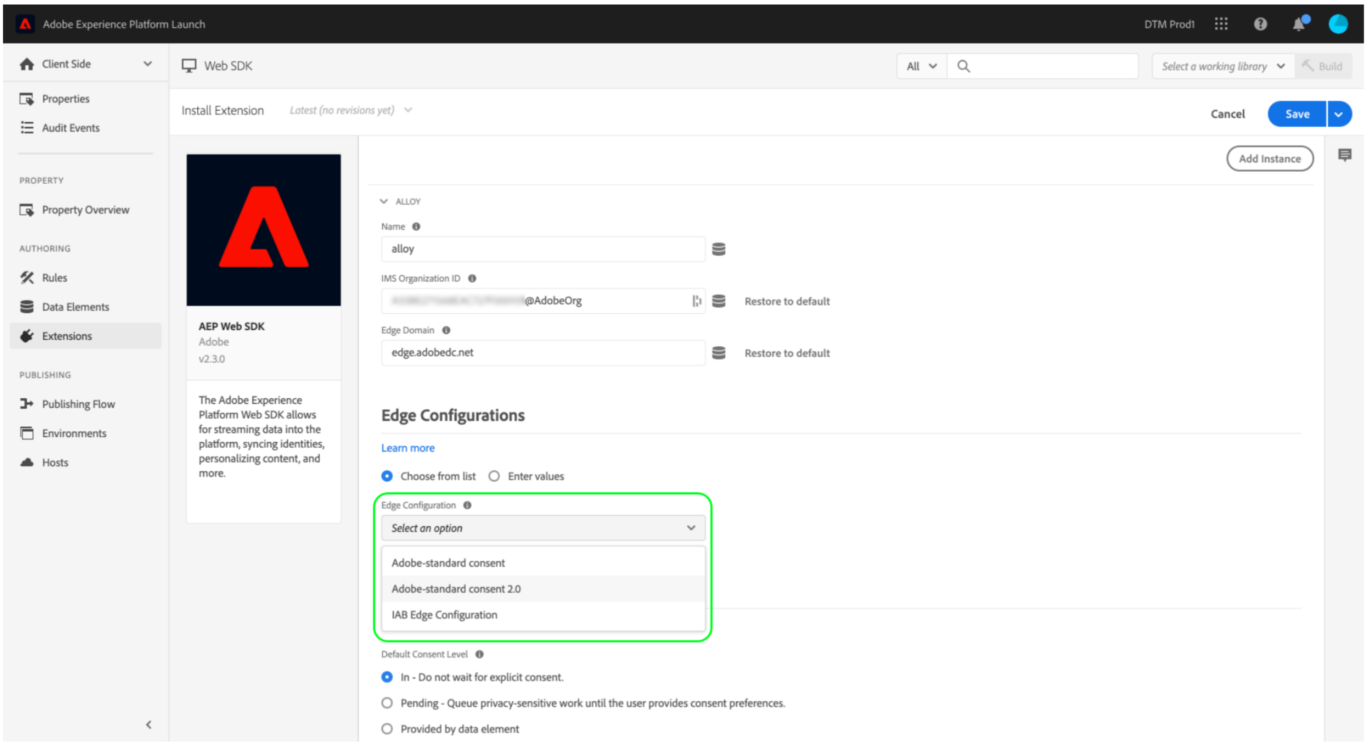


Adobe Media Analytics (3.x SDK) f...
Adobe Systems · v3.0.0
Adobe Media Analytics for Audio and Video is an add-on to the base Analytics offering that provides clients with robust measurement for video, audio and advertisements. This extension deploys with Media SDK version 3.0.0.
Learn More

AEP Web SDK
Adobe · v2.3.0
The Adobe Experience Platform Web SDK allows for streaming data into the platform, syncing identities, personalizing content, and more.

Nielsen BSDK
Nielsen · v1.0.2
Nielsen BSDK
Learn More

When configuring the SDK, under **[!UICONTROL Edge Configurations]**, select the configuration you created in the previous step.



AEP Web SDK
Adobe
v2.3.0
The Adobe Experience Platform Web SDK allows for streaming data into the platform, syncing identities, personalizing content, and more.

ALLOY
Name: alloy
IMS Organization ID: [redacted]@AdobeOrg
Edge Domain: edge.adobedc.net

Edge Configurations
● Choose from list ○ Enter values
Edge Configuration: Select an option
- Adobe-standard consent
- Adobe-standard consent 2.0
- IAB Edge Configuration

Default Consent Level
● In - Do not wait for explicit consent.
○ Pending - Queue privacy-sensitive work until the user provides consent preferences.
○ Provided by data element

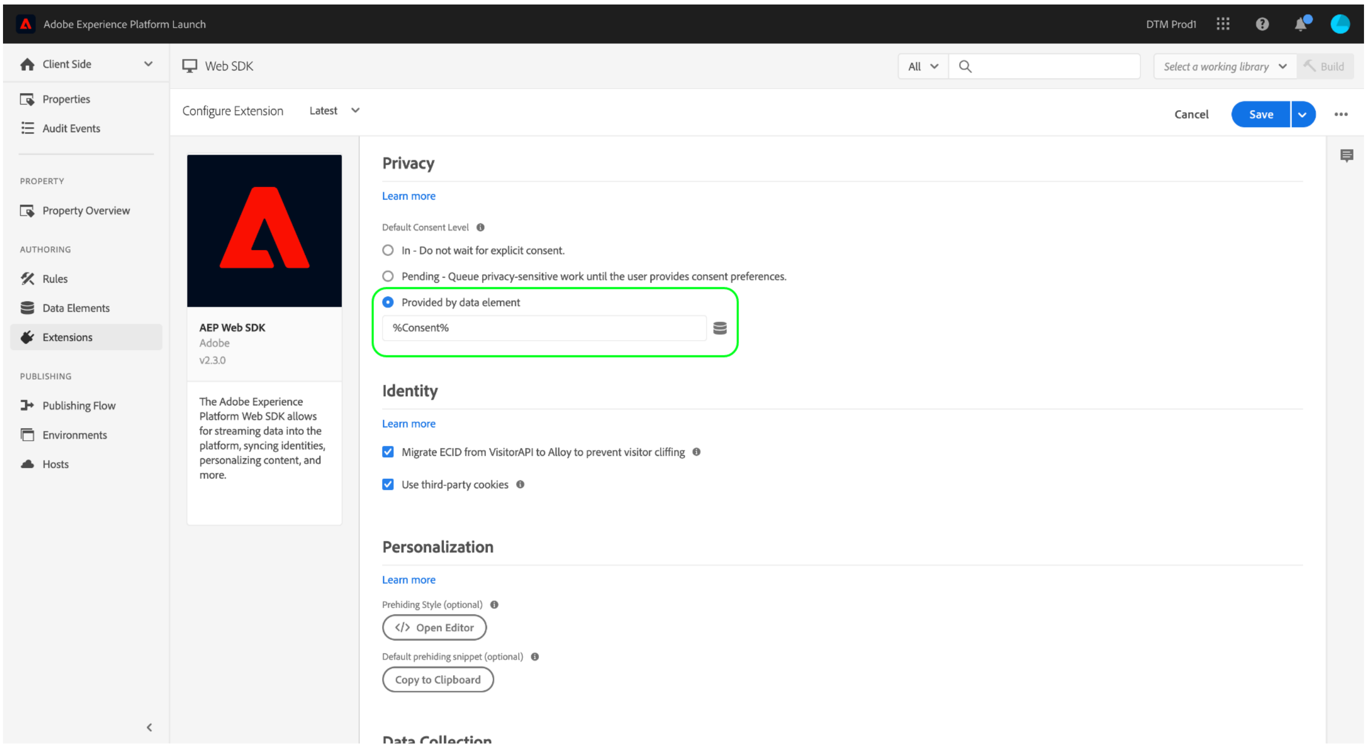
Select **[!UICONTROL Save]** to install the extension.

Create a data element to set default consent

With the SDK extension installed, the next step is to create a data element to represent the default consent values for your customers. Select **[!UICONTROL Data Elements]** in the left navigation to navigate to the data element creation dialog.

From here, you must create an [!UICONTROL XDM Object] data element based on the `consents` object. See the Web SDK extension documentation for specific steps on creating an XDM Object data element, and how to set its default values.

Once the data element is created, navigate back to the Web SDK extension config page. Under the [!UICONTROL Privacy] section, select **[!UICONTROL Provided by data element]**, and use the provided dialog to select the default consent data element you created earlier.



The Adobe Experience Platform Web SDK allows for streaming data into the platform, syncing identities, personalizing content, and more.

Privacy

Learn more

Default Consent Level

○ In - Do not wait for explicit consent.

○ Pending - Queue privacy-sensitive work until the user provides consent preferences.

● Provided by data element

%Consent%

Identity

Learn more

☑ Migrate ECID from VisitorAPI to Alloy to prevent visitor cliffing

☑ Use third-party cookies

Personalization

Learn more

Prehiding Style (optional)

</> Open Editor

Default prehiding snippet (optional)

Copy to Clipboard

Data Collection

Deploy the extension on your website

Once you have finished configuring the extension, it can be integrated into your website. Refer to the publishing guide in the Platform Launch documentation for detailed information on how to deploy your updated library build.

Making consent-change commands

Once you have integrated the SDK extension into your website, you can start using the Platform Web SDK `setConsent` command to send consent data to Platform.

There are two scenarios where `setConsent` should be called on your site:

1. When consent is loaded on the page (in other words, on every page load)
2. As part of a CMP hook or event listener that detects changes in consent settings

> [!NOTE]
>
> For an introduction to the common syntax for Platform SDK commands, see the document on executing commands.

The `setConsent` command expects two arguments:

1. A string that indicates the command type (in this case, `"setConsent"`)

2. A payload object that contains a single array-type property: `consent`. The `consent` array must contain at least one object that provides the required consent fields for the Adobe standard.

The required consent fields for the Adobe standard are shown in the following example `setConsent` call:

```
alloy("setConsent", {
  consent: [{
    standard: "Adobe",
    version: "2.0",
    value: {
      collect: {
        val: "y"
      },
      share: {
        val: "y"
      },
      personalize: {
        content: {
          val: "y"
        }
      },
      metadata: {
        time: "2020-10-12T15:52:25+00:00"
      }
    }
  }]
});
```

Payload property	Description
`standard`	The consent standard being used. For the Adobe standard, this value must be set to `Adobe`.
`version`	The version number of the consent standard indicated under `standard`. This value must be set to `2.0` for Adobe-standard consent processing.
`value`	The customer's updated consent information, provided as an XDM object that conforms to the structure of the Profile-enabled dataset's consent fields.

> [!NOTE]
>
> If you are using additional consent standards in conjunction with `Adobe` (such as `IAB TCF`), you can add additional objects to the `consent` array for each standard. Each object must contain appropriate values for `standard`, `version`, and `value` for the consent standard they represent.

The following JavaScript provides an example of how you may define a function that handles consent preference changes on your site, to be used as a callback in an event listener or a CMP hook:

```
var handleConsentChange = function () {
  // Retrieve the consent data and generate a timestamp
```

```
    var categories = getConsentData();
    var d = new Date();
    var collectedAt = d.toISOString();

    //  Map the consent values and timestamp to XDM
    var consentXDM = {
      collect: {
        val: categories.collect !== -1 ? "y" : "n"
      },
      personalize: {
        content: {
          val: categories.personalizeContent !== -1 ? "y" : "n"
        }
      },
      share: {
        val: categories.share !== -1 ? "y" : "n"
      },
      metadata: {
        time: collectedAt
      }
    };

    // Pass the XDM object to the Platform Web SDK
    alloy("setConsent", {
      consent: [{
        standard: "Adobe",
        version: "2.0",
        value: consentXDM
      }]
    });
  });
```

Handling SDK responses

All [!DNL Platform SDK] commands return promises that indicate whether the call succeeded or failed. You can then use these responses for additional logic such as displaying confirmation messages to the customer. See the section on handling success or failure in the guide on executing SDK commands for specific examples.

Next steps

By following this guide, you have configured the Platform Web SDK extension to send consent data to Experience Platform. You can now return to the consent processing overview for steps on how to test your implementation.