

US



Data Activation 3rd Party Integration

Instructions

## Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Purpose of this Document .....	3
1.2	Target Audience .....	3
1.3	Considerations .....	3
<b>2</b>	<b>Design Concepts.....</b>	<b>4</b>
2.1	What is retargeting? .....	4
2.2	Customer Data Platform (CDP) .....	4
2.2.1	Identity Mapping.....	4
2.2.2	What is Pixel tracking.....	5
2.2.3	What is Server-to-Server tracking.....	5
<b>3</b>	<b>Identity Mapping .....</b>	<b>7</b>
3.1	Custom Event for identity mapping .....	7
<b>4</b>	<b>Pixel Integration (Client Side) .....</b>	<b>9</b>
4.1	Requirements.....	10
4.1.1	Demdex Client Value .....	10
4.1.2	CI360 Tag Deployed .....	10
4.1.3	CI360 Pixel Placeholder .....	10
4.2	Create Spot in CI360 .....	11
4.3	Create Custom Event .....	12
4.4	Create Creative with Adobe pixel .....	13
4.5	Define Web Task .....	14
4.6	Trigger Adobe Pixel via SAS Creative .....	17
4.7	Verification.....	17
<b>5</b>	<b>API (Server-to-Server) .....</b>	<b>19</b>
5.1	Requirements.....	19
5.1.1	Regional Server Host Name .....	19
5.1.2	Demdex Client Value .....	20
5.1.3	CI360 Tag Deployed .....	20
5.2	Connector setup.....	21
5.3	Event setup .....	23
5.4	Task Setup.....	24
5.5	Verification.....	25
<b>6</b>	<b>Appendix .....</b>	<b>26</b>
6.1	External References .....	26
6.2	Source Code .....	26
6.2.1	Adobe Pixel Code .....	26
6.2.2	Adobe API Endpoint .....	27
6.2.3	SAS CI360 Tag Example Code .....	27
6.2.4	SAS Generic Pixel Placeholder.....	28
6.2.5	SAS Adobe Placeholder .....	28
6.3	Connector JSON Payloads .....	28

# **1 Introduction**

## **1.1 Purpose of this Document**

The purpose of this document is to help guide in the integration of CI360 with Adobe products. This will allow activation on Adobe's platform through Adobe Audience Manager (AAM) and by extension, the rest Adobe's products and other 3<sup>rd</sup> party tools connected to AAM.

This includes collecting and sending segment information to Adobe's servers through API with the connector framework and connect using Engage: Digital to inject a pixel using a spot.

## **1.2 Target Audience**

Activation Service Offering users and those who will be implementing solutions to send information to users for the CI360 Software Stack

## **1.3 Considerations**

Due to the fast changing nature of APIs and technology be sure to read the Adobe documentation to understand what is needed as some things might have changed. Some of these solutions might not be directly applicable to all users.

## 2 Design Concepts

### 2.1 What is retargeting?

Retargeting is a method for putting the organization's brand in front of consumers after they have left the website or app and persuading them to reconsider an offer or value proposition from the brand. It often refers to reaching out to lost visitors by Display Ads and its typically performed by using the third-party platform's event (tracking pixel and cookie) to achieve.

### 2.2 Customer Data Platform (CDP)

In this article, we are going to use CI360 to combine 1<sup>st</sup> party data (online and offline), marketing data, and third-party data. Hence, CI360 can be act as CDP to provide unified customer data view for enrich the re-targeting for online campaign which run off-site platform (such as Google Ads), and there are two steps and three methods to achieve this:

1. Identity Mapping (between each platform solutions)
2. Sending Data to third-party platform:
  - Pixel Tracking
  - Server-to-Server Tracking

#### 2.2.1 Identity Mapping

The following diagram shows how marketing tool act as “middle agent” to map identifier and later to exchange data between organization website / app, marketing automation tool as well as third-party platform:

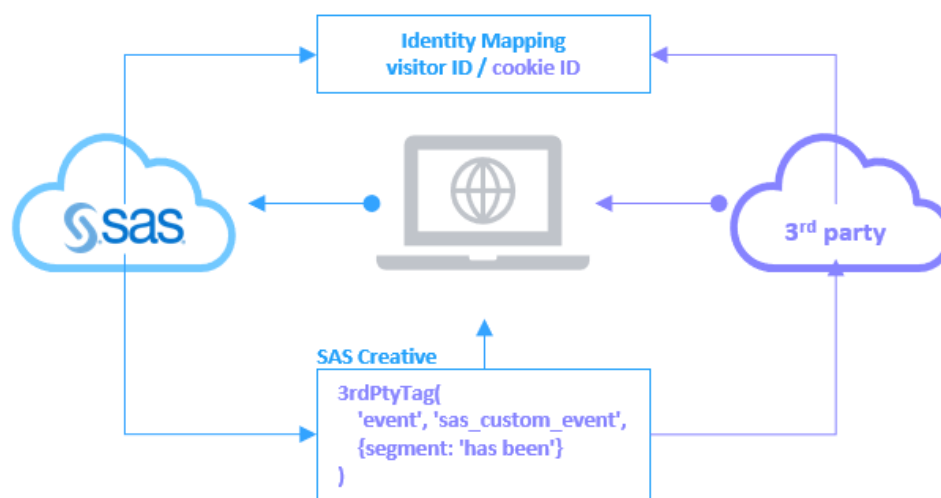


In CI360, the identity mapping can be done by Custom Event with additional attribution, it will be cover in later sections.

### 2.2.2 What is Pixel tracking

Pixel tracking are snippets of code provided from third-party platform that are loaded on the website, in specific pages, to track visitors in-site or in-app behavior. The code is usually executed on client side (visitor's web browser or app) sending the information along with unique cookie ID / visitor ID to third-party platform.

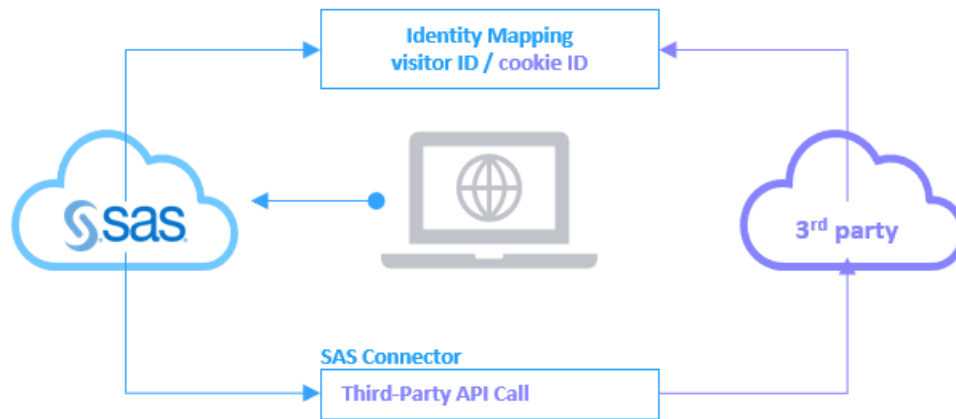
In this offering, we are going to use pixel integration via SAS CI360 Creative to serve third-party tracking code. This approach is different to traditionally third-party tracking which only embedded in the specified page; CI360 will decide the right moment of the customer journey to fire the event or share the segment.



### 2.2.3 What is Server-to-Server tracking

Server-to-Server tracking is using APIs call to fire tracking event directly from the solution backend to third party platform which extended the tracking beyond visitor exited the website or app. It is also enhancing the site security as server to server tracking doesn't require any third-party coding to be installed on the site and tracking event can be control and triggered by solution owned by the organizations.

In this offering, we are going to use Server-Side API framework (Connector) to execute third-party tracking server call. This approach allows to share event / segment without embedded third-party code on the customer site to maintain security / data privacy. As well as, for the visitor beyond the site exit where pixel integration unable to reach out.




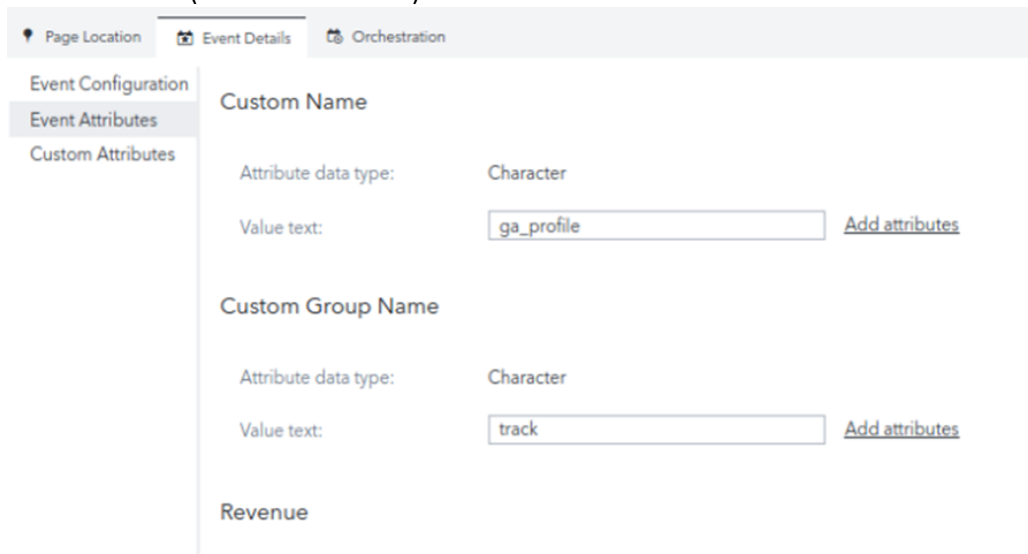
### 3 Identity Mapping


In order to implement these integrations the the first step is mapping third-party platform ID with SAS ID. We will collect the ID from third party platforms for future use.

#### 3.1 Custom Event for identity mapping

This event will be setup to create to collect the Adobe Experience Cloud ID from the client's computer. The ID we want is stored in the gaGlobal.vid variable. Note that this event won't necessarily fire our external event and Server-to-Server communication through the API, we will do that with an **External System Task** using this event as a trigger which we can further customize to add additional conditions and segmentation before sending data to .

- Click **Events** in the left navigation menu
- Create a new event by clicking on 
- You can create any type of event, but for this we will create a **Custom Event** based on a **Page View**
- Choose the page you want this to fire on (this could simply be every page if you restrict using segments in a later step, or it could be specific pages)
- Set the **Custom Name** and the **Custom Group Name**, note that these are the values that will be sent to Adobe. (Write these down)



- Create a new Custom Attribute by clicking 
- Choose **URL Details > Cookie**
- Set the attribute name to **s\_ecid**
- Cookie name to **s\_ecid**
- Return: **After**
- After:\* | *Note: The value is in the form "MID/123123123123" which is why we are capturing everything after the /*
- Attribute data type: **Character**
- Personally identifiable information: **Unchecked** (note that we keep this unchecked so we can connect other information from other systems and connect it with our own identifiers if needed)

later in our data)

s\_ecid

Attribute name: \*

s\_ecid

Source attribute:

Cookie

Cookie name: \*

s\_ecid

Return:


Everything ▼

Attribute data type:

Character ▼

Personally identifiable information: ⓘ

☐

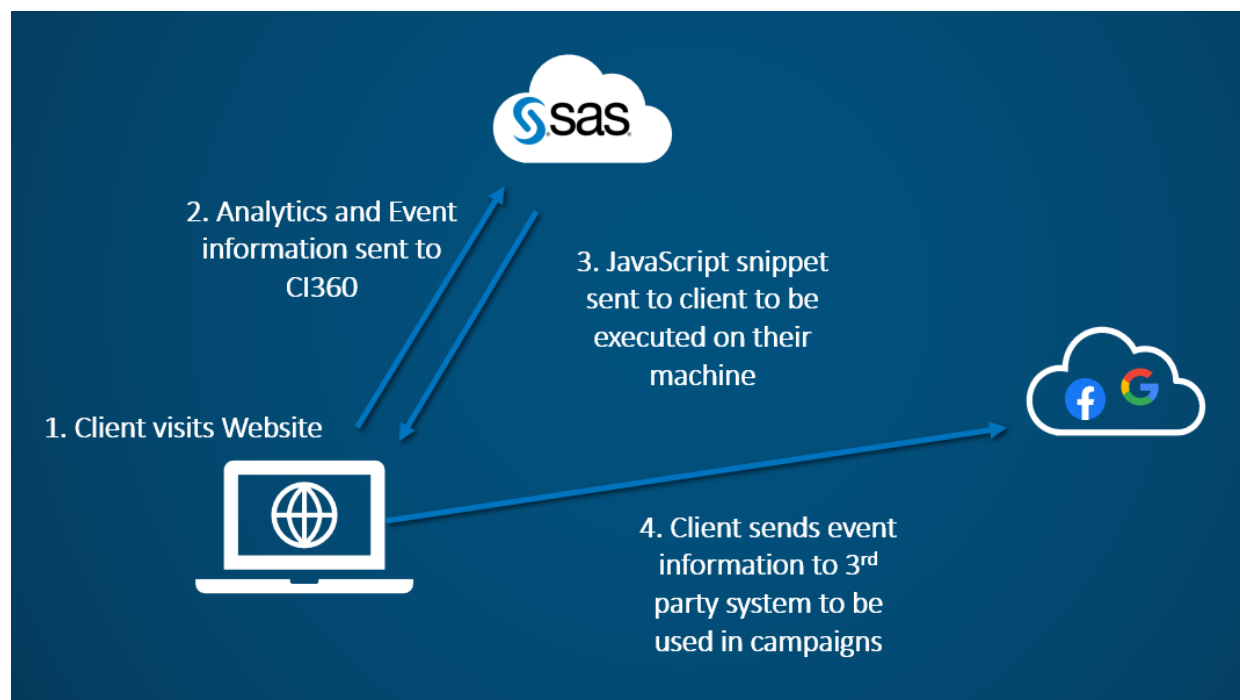
- n. Click  to save the event naming it whatever you want (remember the event name won't be used to send to Adobe, the Event Attributes will be used instead)



## 4 Pixel Integration (Client Side)

For this integration we will initiate a pixel call from the customer's browser visiting a website through the use of a Engage: Digital Spot. Using the Pixel Integration we will send data from the customer's computer to the user's

1. First we collect information needed
2. Deploy the tags on the site
3. Create a spot
4. Setup a creative to hold the pixel in CI360
5. Define a task to initiate when a user qualifies to be sent to Adobe
6. Verify that it works



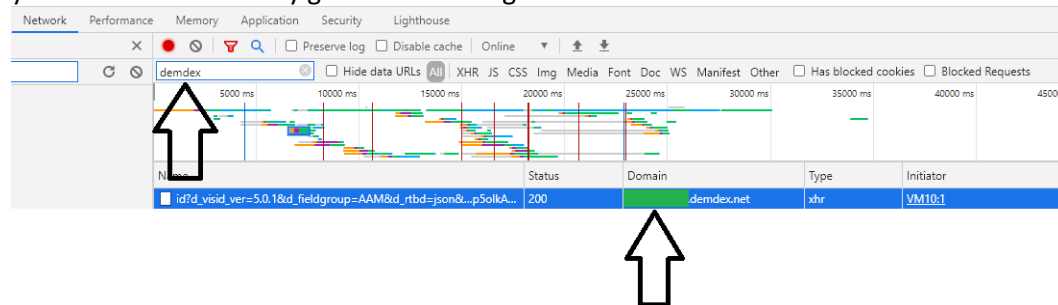
*Note: The creative and spot only need to be created once as they can be reused with different events and tasks in order to send different segments to Adobe Audience Manager*

*Note: Adobe Audience Manager (AAM) does not have any tags that you traditionally would generate in a 3<sup>rd</sup> party service. A pixel code must be create manually using a standard format used by the DCS API. This document attempts to standardize the format for simplicity sake but can be further modified for other situations using the information in the appendix as well as linked documentation to Adobe's documentation. This also assumes that there is one CI360 tenant and one AAM instance, some code might need to be modified for testing environments or otherwise*

## 4.1 Requirements

### 4.1.1 Demdex Client Value

The specific subdomain for the customer's AAM instance, this is a subdomain setup by Adobe for each customer and is unique to them. The format of the domain will be **<aam client subdomain>.demdex.com** so searching for 'demdex' in the network tab of the customer's site will show you this value hidden by green in the image below



### 4.1.2 CI360 Tag Deployed

Add in the CI360 tag into the site. Please copy the code from your CI360 from the **General Settings > SAS Tag Instruction** section.

SAS Ci360 Tag Example:

```
<!-- SAS Ci360 Tag -->
<script>
(function(ci){
  var ef=window[ci]=function(){
    return ef.q.push(arguments);
  };
  ef.q=[];ef.a={};
})('ci360');
</script>
<script async data-efname='ci360' id='ob-script-async'
a='<tenant ID>'
src='https://<server>/js/ot-all.min.js'></script>
```

### 4.1.3 CI360 Pixel Placeholder


This tag is deployed so that we can later create a spot against and not conflict with any other code on the website. Please do note that if you're making several pixel calls for different services, it would be best practice to deploy the below SAS Pixel Placeholder code with several different div's using different IDs, as setting them up the same way will conflict.

```
<!-- SAS Pixel Placeholder -->
<div id="sas_adobe_pixel"></div>
```

**Tip:** both **SAS Ci360 Tag** and **SAS Pixel Placeholder** can be embedding via tag manager products without need to place in the source code of the site which can be constrained by development procedures. Many times SAS CI360 tag is deployed using tag manager products and the pixel placeholder is no different.

## 4.2 Create Spot in CI360

This will create a spot for CI360 where we can then inject content, in this case we will inject a tag in order to send data from the website client to the Adobe servers

- a. From navigation bar, click **Spots**
- b. Click the  button to create a new spot
- c. Select **Web** as the spot type
- d. Navigate to the **URL** of the web page with **SAS CI360 Tag** and **SAS Pixel placeholder**
- e. Click on the **Select selector** button and type in “**div#sas\_adobe\_pixel**” as below (note that this is specific to the code provided, you can use any div name but it must match the tag that you inject using your tag manager from the above requirements):


☐ Navigate ☒ Select Selector:

- f. Then, click **Create Spot**
- g. In **Spot Details Tab**,
  - i. Select **Spot Configuration** from the menu bar
    1. Change the Page matches under Spot Location section to **Any Page**:


### Spot Location

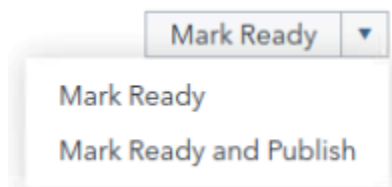
Specify the page or pages on which the spot appears.

Page matches:

- ii. select **Spot Attribution** from the menu bar
  1. click the  button to add new Attribution for the spot
  2. select **URL path (entire path)** as Attribute
  3. click **OK**
  4. amend the Attribute name to “**page\_url**”:

Attribute name: \*

- h. click the  button to save the Spot
- i. type in a name for the Spot and then click **Save**
- j. Go to the **Orchestration Tab**
  - i. Click on **Mark Ready and Publish** from drop-down menu:




- ii. Click **Yes** when promoted by **Publishing the item will change the status to Scheduled or Active**

**Tip:** The example below fires an Adobe event if user had been visited X page Y times during Z period.

### 4.3 Create Custom Event

Here we will create a Custom Event, this is required as the code used in our creative to trigger the pixel will use merge tags in order to replace information to be sent to AAM. The instructions will provide how to create an event and which data will be sent to AAM but you can create your own event with different criteria and naming.

- a. From navigation bar, click **Events**
- b. Click the  button for create a new event
- c. Select **Custom Event**
- d. Select **Page View** as event type
- e. Enter the URL for re-targeting
- f. Click **Create Event**
- g. In the **Event Details** Tab
  - i. Go to **Event Configuration**
    1. To confirm the site is correct:

#### Pages in Event

Specify the page or pages that are included for this event.

Selected page ▼

Page URL:

- ii. Go to **Event Attributes**

*Note: These values below are going to be used as replacement values later for variables sent to Adobe's servers. The values "**has\_been**" and "**tracked**" are both example values that should be changed accordingly with the thought in mind that these values will be used for segmentation on Adobe's side and should be unique for each event. Be Sure to **Write these down** for use later if you changed them or used your own values*


1. In Custom Name's value text: type in "**has\_been**":

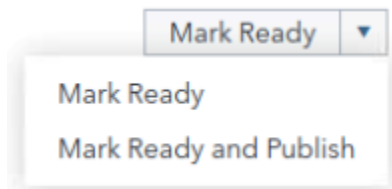
This is a custom value that will be collected in Adobe and stored in "c\_ci360event" signal.

2. In Custom Group Name's value text: type in "**tracked**":

This is a custom value that will be collected in Adobe and stored in "c\_ci360eventcategory" signal

- iii. The configuration should look similar to the below:


- h. click the  button to save the Custom Event
- i. type in “**Event: Adobe Load**” (this is just a placeholder name, you can use any event you want), then click **Save**
- j. Go to the **Orchestration** Tab
  - i. Click on **Mark Ready and Publish** from drop-down menu:



- ii. Click **Yes** when prompted by **Publishing the item will change the status to Scheduled or Active**

## 4.4 Create Creative with Adobe pixel

Here we will define the creative that will run code/call the pixel needed when the user visits the page and qualifies for our task (which we define in a later [step](#)).

- a. From navigation bar, click **Creatives**
- b. Click the  button for create new Creative
- c. Select **HTML** as creative type
- d. Copy the **Adobe Event Tracking Pixel** into the body tag replacing everything as below being sure to replace the “<AAM client subdomain>” with the customer’s value on their website. You can find the code needed [here](#) in the code section (Adobe\_Pixel\_Code.html).

```


<html>

<head>
  <meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
  <title></title>
</head>

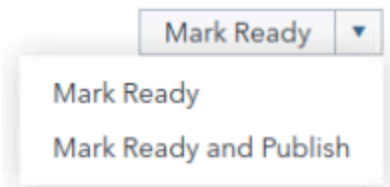
<body>
  <script>
    function getCookie(name)*{
      let matches = document.cookie.match(new RegExp(
        "(?:^|; )" + name.replace(/([\\$?*|{}()\[\]\\\/\+\^])/g, '\\\\$1') + "=([^\s;]*)"
      ));
      return matches ? decodeURIComponent(matches[1]) : undefined;
    }
    var s_ecid = getCookie("s_ecid");
    if (s_ecid.length > 0) {
      s_ecid = s_ecid.substring(8);
      var pixel = document.createElement("IMG");
      pixel.setAttribute("src", "https://<aam_client_subdomain>.demdex.net/event?c_ci360event=
{{eventAction}}&c_ci360eventcategory={{eventCategory}}&mid=" + s_ecid);
      pixel.setAttribute("height", "1");
      pixel.setAttribute("width", "1");
      document.body.appendChild(pixel);
    }
  </script>
</body>

```

e. Click **Done**

- f. click the  button to save the creative
- g. type in **"Creative: Adobe Pixel"** then click **Save**
- h. Go To **Orchestration** Tab



- a. Click on **Mark Ready and Publish** from drop-down menu:



- b. Click **Yes** when prompted by **Publishing the item will change the status to Scheduled or Active**

## 4.5 Define Web Task

Here we will create a task, this connects the Spot and the Creative with Segmentation in order to conditionally fire the Adobe pixel on the client machine. We will use an event and information defined in the event, combined with merge tags to create the send information to AAM servers

- From navigation bar, click **Task**
- Click the  button for create new Task
- Select **Web** as Task type
- Choose the **Spot** created from [Step 2.2](#)
- Go to **Targeting** Tab
  - Click the  button to add criteria
  - Select the **Custom Event** defined in [Step 2.3](#)

**Select Criteria**

Filter

Items (85)

Name	Description
Custom Event	
Adobe Load v2	

- iii. Optional: Define Even Occurrence Detail by click the  button:

**Event Occurrence Details**

All of the following

Adobe Load v2

Event must occur: at least one time in the previous week

Add the attribute conditions that must be present when the event occurs.

**Event Occurrence Details**

Specify the number of times the event must occur in order to be eligible.

Occurrences:

Specify the time period in which all occurrences must happen.

Date range: ☒ Relative ☐ Absolute ☐ None

Event must occur in the following time period:

Previous weeks

Event must occur: at least one time in the previous week

- iv. Or leave it by default as **once per pervious weeks**
- f. Go to **Content** Tab
- i. Click **Select Content**
- ii. Choose **Creatives** from the drop-down menu
- iii. Select the **Creative** defined from [Step 2.4](#)


**Select Content**

Filter

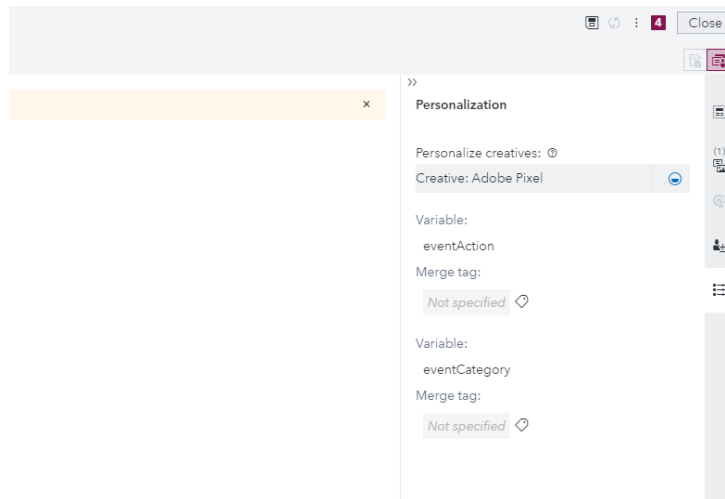
Creatives


New Creative

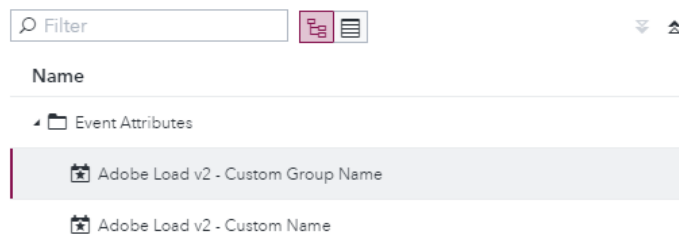
Name	Status	Date Modified	Modified By
Creative: Adobe Pixel	Designing	Oct 1, 2020, 1:37:19 AM	jeffrey.stroud@sas.com


- iv. Click **OK**
- v. Click the  button from the right-hand side menu bar

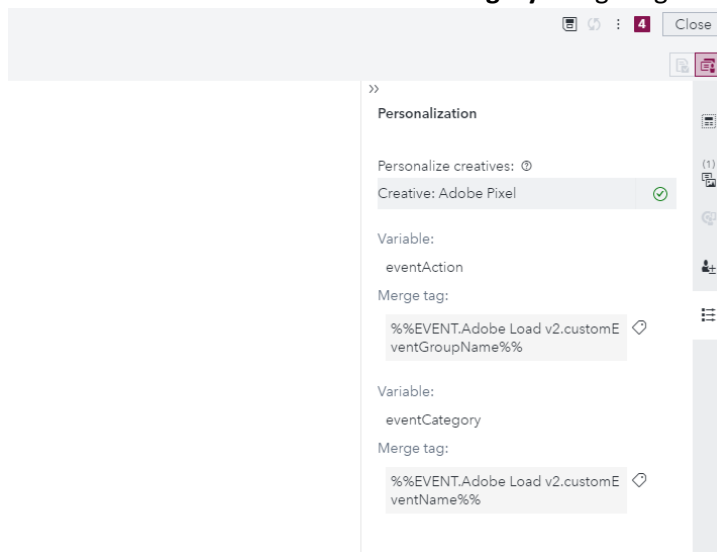
- vi. Screen should be displayed as below:



- vii. Click the  button from the **eventAction** merge tag field
- viii. Select the Event Attribute with **Custom Group Name**:

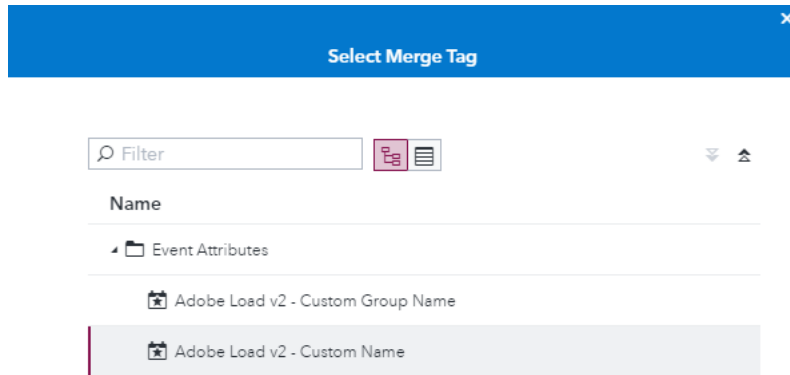


- ix. then, click **OK**
- x. Click the  button from the **eventCategory** merge tag field





- xi. Select the Event Attribute with **Custom Name**:

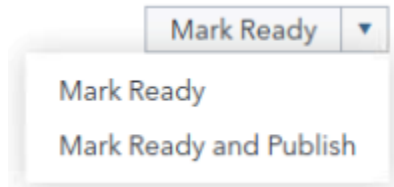


- xii. then, click **OK**.

- xiii. click the  button to save the task

- g. Go To **Orchestration** Tab

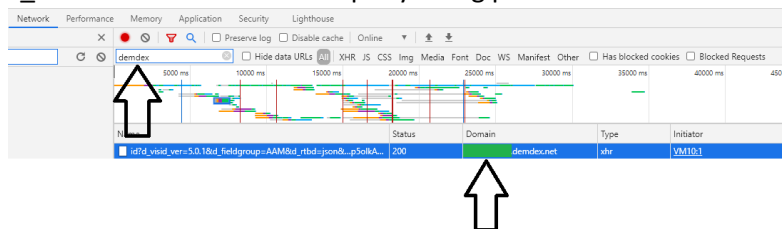
- i. Click on **Mark Ready and Publish** from drop-down menu:



- ii. Click **Yes** when prompted by the following: **Publishing the item will change the status to Scheduled or Active**

## 4.6 Trigger Adobe Pixel via SAS Creative

- Go to the page where your **Custom Event** is fired created in [Step 2.3](#)
- Then refresh the page or visit other page contain the **SAS Pixel Placeholder**
- The “**Event: Adobe Pixel**” Custom Event should be now triggered the Adobe Pixel via SAS Creative
- Go to **Network** tab from the browser developer tools
  - Find the request by typing in “**Demdex**”
  - Click on the demdex request, you should see several requests, choose the one with **c\_ci360match** as one of the query string parameter



## 4.7 Verification

Here we will verify if the Pixel is properly sending data to AAM servers. Using the values we send to Adobe we will see if those values show up in the **Unused Signals Reports** section of the AAM interface.

Once verified we can then use these in traits and then segments for activation on other platforms connected to AAM

- a. Retrieve the values from the events we set before that you wrote down from the Event Setup
- b. Check the video [here](#) for how to use the unused signal report
- c. You will now have two keys you can look for to verify in the unused signal report
  - i. Key 1
    - i. Key: **c\_ci360event**
    - ii. Value: **has\_been**
  - ii. Key 2
    - i. Key: **c\_ci360eventcategory**
    - ii. Value: **tracked**
- d. Once these are verified you can then setup traits and segments in AAM

## 5 API (Server-to-Server)

This section will show how to make a server-to-server connection using a CI360 Connector. This doesn't require any bridge code as it is a one-way communicator from the CI360 tenant to the AAM servers.

1. First we collect information needed
2. Deploy the tags on the site
3. Create a Connector with an Endpoint for the specific client
4. Setup an Event to fire
5. Define a task to initiate when a user qualifies to be sent to Adobe
6. Verify it works

*Note: The connector is setup once and can be reused for events/task moving forward*



### 5.1 Requirements

#### 5.1.1 Regional Server Host Name

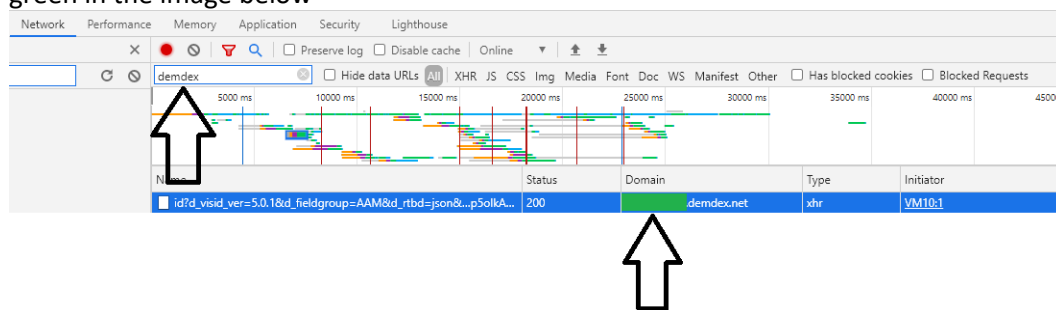
DCS Region ID (dcs_region)	Region (and Location)	Host Name
ID 3	Southeast Asia (Singapore)	apse.demdex.net

ID 4	South America (São Paulo, Brazil)	sae.demdex.net
ID 6	Europe (Dublin, Ireland)	irl1.demdex.net
ID 7	US East (Virginia, USA)	use.demdex.net
ID 8	South Pacific / Oceania (Sydney, Australia)	apse2.demdex.net
ID 9	US West (Oregon, USA)	usw2.demdex.net
ID 11	Asia (Tokyo, Japan)	tyo3.demdex.net
ID 12	India	ind1.demdex.net

Regional server identification is required to send data to the DCS API which is the AAM interface that allows for real-time data ingestion. You can send to any of these data regions (based on AWS server regions) and it will be duplicated across to the others, but to ensure delivery is as fast as possible the geo-location/region where the websites' main visitor reside should be used **For Reference:** [Adobe DCS Region ID](#)

### 5.1.2 Demdex Client Value

The specific subdomain for the customer's AAM instance, this is a subdomain setup by Adobe for each customer and is unique to them. The format of the domain will be **<unique value>.demdex.com** so searching for "demdex" in the network tab of the customer's site will show you this value hidden by green in the image below



### 5.1.3 CI360 Tag Deployed




Add in the CI360 tag into the site (note that if you're making several pixel calls for different services, it would be best practice to deploy the below SAS Pixel Placeholder code with several different div's using different IDs, as setting them up the same way will conflict). Please copy the code from your CI360 from the **General Settings > SAS Tag Instruction** section.

```

SAS Ci360 Tag Example:
<!-- SAS Ci360 Tag -->
<script>
(function(ci){
  var ef=window[ci]=function(){
    return ef.q.push(arguments);
  };
  ef.q=[];ef.a={};
})('ci360');
</script>
<script async data-efname='ci360' id='ob-script-async'
  a='<tenant ID>'
  src='https://<server>/js/ot-all.min.js'></script>

```


## 5.2 Connector setup

- a. Navigate to **General Settings > Connectors**
- b. Click on the  button
- c. Enter the details
  - i. Name: **Adobe AAM**
  - ii. Description: **\*Anything you want can go here to describe this\***
  - iii. Support Contact: Set this up as anyone that would receive emails in case of errors
  - iv. Set as **Active** by changing the slider from  to 
  - v. Attach an image. One can be found in the code repository.
  - vi. Click **Apply**

Adobe AAM

**Details** ⓘ

Specify information to configure a connector.

Code:	CONN6
Name: *	<input type="text" value="Adobe AAM"/>
Description:	<input type="text" value="For connecting directly with Adobe Audience Manager for use in external activation of users on ad platforms"/>
Type:	Custom
Support contact:	<input type="text" value="jeffrey.stroud@sas.com"/>
Mark Active:	<input checked="" type="checkbox"/>
Associated image:	 <div style="display: inline-block; vertical-align: middle; text-align: left;"> <small>ⓘ</small>  <a href="#">Change</a>  <a href="#">Remove</a> </div>

- d. Re-Open the Adobe Connector
- e. Click **New Endpoint**
  - i. Name: **Adobe AAM**
  - ii. Description: **\*Anything you want can go here to describe this\***

- iii. URL:  
[https://<Regional Server Host Name>.demdex.net/event?c\\_ci360event={{\\$event}}&c\\_ci360eventcategory={{\\$eventcategory}}&mid={{\\$s\\_ecid}}](https://<Regional Server Host Name>.demdex.net/event?c_ci360event={{$event}}&c_ci360eventcategory={{$eventcategory}}&mid={{$s_ecid}})
- iv. Method: **GET**
- v. Time-out: **15 seconds**
- vi. Authorization: **n/a** note: Adobe does not use authentication for sending data to their service
- vii. Headers: **Key=Host, Value=<aam client subdomain>.demdex.com** (replacing <aam client subdomain> from the [requirements section](#))
- viii. In the picture below, the host is in **Southeast Asian Region** which means the host is **apse.demdex.net**

**New Endpoint**

Details | Orchestration

▼ Details  
Specify information to configure an endpoint. ⓘ

Name: \* Adobe AAM

Description:

URL: \* [https://apse.demdex.net/event?c\\_ci360event={{\\$event}}&c\\_ci360eventcategory={{\\$eventcategory}}&mid={{\\$s\\_ecid}}](https://apse.demdex.net/event?c_ci360event={{$event}}&c_ci360eventcategory={{$eventcategory}}&mid={{$s_ecid}}) Define variable ⓘ

Method: GET

Time-out: 15 minutes

Connector: Adobe AAM

> Authorization

▼ Headers  
Add headers. ⓘ

Key *	Value	Description
Host	sas.demdex.com	Add description


Save and add more Save Cancel

- f. Click on **define variables** (this will replace the {{variables}} in the URL with values from the JSON payload that will be sent each time the connector is invoked and these values will be populated from values we create in the custom event and task we create). Unfortunately at this time, there are no easy way to view the JSON payloads without setting up your own server to capture these values first but once you know their locations you can use JSON path selectors to define them (Check appendix for website to test your JSON payload and selectors) **Check to see if merge tags available from other.**
  - i. Set event to “**\$.customName**”
  - ii. Set eventcategory to “**\$.customGroupName**”
  - iii. Set s\_ecid to “**\$.properties.s\_ecid**”

**Define variable**


Key	Value *	Description
event	\$.eventName	
eventcategory	\$.customName	
s_ecid	\$.properties.s_ecid	


OK Cancel

- g. Click OK CI
- h. Click Save to save the endpoint
- i. In the Endpoints section, click  to enable the endpoint
- j. Click Apply
- k. You will now have a new connector as show below with 1 endpoint setup for Adobe Audience Manager
- l. Optional:
- m. Setup the connector to proxy through an on-premise agent

Connectors


Add and manage configurations for connectors. ⓘ

Filter 

<input type="checkbox"/>	Name	ID	Type	Active	Number of Endp...	Date Modified
<input type="checkbox"/>	Adobe AAM	CONN6	Custom		0	Sep 29, 2020

### 5.3 Event setup

This event will be used to collect the Adobe Experience Cloud ID from the client's computer using custom attributes ability to capture cookie data. This will be used later to send data to Adobe's servers identifying the user that triggers the task.

- o. Click **Events** in the left navigation menu
- p. Create a new event by clicking on 
- q. You can create any type of event, but for this we will create a **Custom Event** based on a **Page View**
- r. Choose the page you want this to fire on
- s. Set the **Custom Name** to "**has\_been**"
- t. and the **Custom Group Name** to "**tracked**"

*Note: These values below are going to be used as replacement values later for variables sent to Adobe's servers. The values "**has\_been**" and "**tracked**" are both example values that should be changed accordingly with the thought in mind that these values will be used for segmentation on Adobe's side and should be unique for each event. Be Sure to **Write these down** for use later if you changed them or used your own values.*

Page Location | Event Details | Orchestration

Event Configuration

Event Attributes

Custom Attributes

**Custom Name**


Attribute data type: Character

Value text:  [Add attributes](#)

**Custom Group Name**


Attribute data type: Character

Value text:  [Add attributes](#)

- u. Create a new Custom Attribute by clicking 
- v. Choose **URL Details > Cookie**
- w. Set the attribute name to **s\_ecid**
- x. Cookie name to **s\_ecid**
- y. Return: **After**
- z. After:\* | *Note: The value is in the form "MID/123123123123" which is why we are capturing everything after the /*
- aa. Attribute data type: **Character**
- bb. Personally identifiable information: **Unchecked** (note that we keep this unchecked so we can connect other information from other systems and connect it with our own identifiers if needed later in our data)


s\_ecid

Attribute name: *	s_ecid
Source attribute:	Cookie
Cookie name: *	s_ecid
Return:	Everything ▼
Attribute data type:	Character ▼
Personally identifiable information: ⓘ	<input type="checkbox"/>

Click  to save the event naming it whatever you want (remember the event name won't be used to send to Adobe, the Event Attributes will be used instead)

## 5.4 Task Setup

Here we will create an external system task to invoke the connector using the [event](#) we created as a trigger. In the case of Adobe, all the information is sent through link parameters so we don't need any bridge code, but if needed we can use the content creation combined with merge tags to send additional information to the external system.

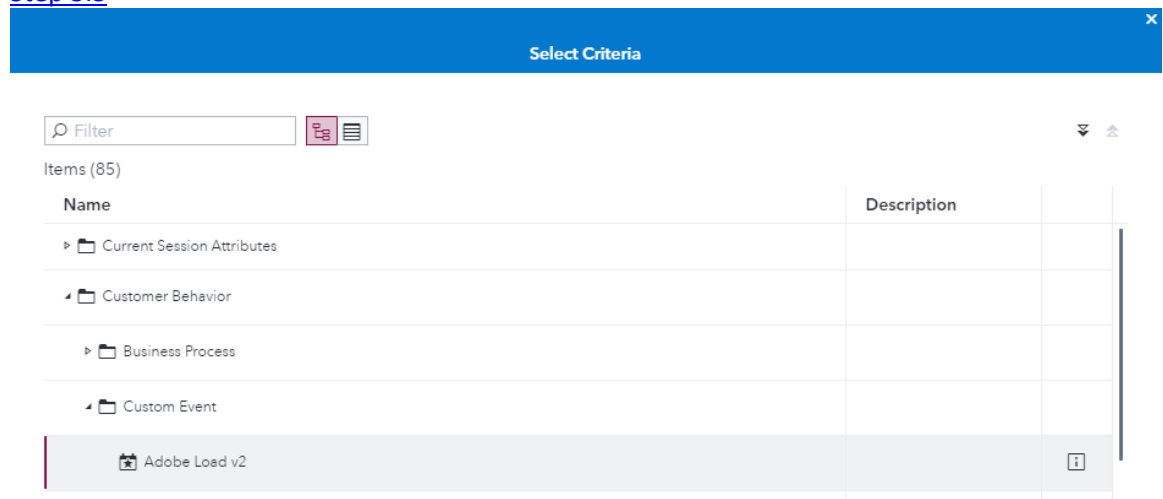
- a. Click on **Tasks** in the side panel
- b. Create a new task by clicking on 
- c. Choose **External System**
- d. Here you have the choice to select an Agent (even if you are using an on-premise agent to send your connector through, selecting an Agent is not required here) so simply click on

 **Select Connector Endpoints** to skip this section

- n. Check the **Endpoint Name/Connector Name** we set before (Adobe AAM/Adobe AAM)
- o. Click **Create Task**



- p. Navigate to the **Orchestration Tab > Trigger** section and specify the Event you created in [Step 3.3](#)



- h. Click **Ok**
- i. Optional
  - i. Add any other targeting criteria you want based on your requirements by navigating to the **Targeting** tab
  - ii. Add a schedule making sure this task only fires during specific times

## 5.5 Verification

Here we will verify if the Pixel is properly sending data to AAM servers. Using the values we send to Adobe we will see if those values show up in the **Unused Signals Reports** section of the AAM interface. Once verified we can then use these in traits and then segments for activation on other platforms connected to AAM

1. Retrieve the values from the events we set before that you wrote down from the Event Setup
2. Check the video [here](#) for how to use the unused signal report
3. You will now have two keys you can look for to verify in the unused signal report
  - i. Key 1
    - i. Key: **c\_ci360event**
    - ii. Value: **has\_been**
  - ii. Key 2
    - i. Key: **c\_ci360eventcategory**
    - ii. Value: **tracked**
4. Once these are verified you can then setup traits and segments in AAM

## 6 Appendix

### 6.1 External References

Item	Link
<b><i>Sending Data to DCS (Pixel/API)</i></b>	<a href="https://docs.adobe.com/content/help/en/audience-manager/user-guide/api-and-sdk-code/dcs/dcs-event-calls/dcs-url-send.html">https://docs.adobe.com/content/help/en/audience-manager/user-guide/api-and-sdk-code/dcs/dcs-event-calls/dcs-url-send.html</a>
<b><i>DCS Regions (API)</i></b>	<a href="https://docs.adobe.com/content/help/en/audience-manager/user-guide/api-and-sdk-code/dcs/dcs-api-reference/dcs-regions.html">https://docs.adobe.com/content/help/en/audience-manager/user-guide/api-and-sdk-code/dcs/dcs-api-reference/dcs-regions.html</a>
<b><i>Making Server-to-Server Calls DCS Calls</i></b>	<a href="https://docs.adobe.com/content/help/en/audience-manager/user-guide/api-and-sdk-code/dcs/dcs-apis/dcs-s2s-calls.html">https://docs.adobe.com/content/help/en/audience-manager/user-guide/api-and-sdk-code/dcs/dcs-apis/dcs-s2s-calls.html</a>
<b><i>Supported Attributes/Link Parameters for DCS Calls (Pixel/API)</i></b>	<a href="https://docs.adobe.com/content/help/en/audience-manager/user-guide/api-and-sdk-code/dcs/dcs-api-reference/dcs-keys.html">https://docs.adobe.com/content/help/en/audience-manager/user-guide/api-and-sdk-code/dcs/dcs-api-reference/dcs-keys.html</a>
<b><i>Adobe Cookie Documentation</i></b>	<a href="https://docs.adobe.com/content/help/en/id-service/using/intro/cookies.html">https://docs.adobe.com/content/help/en/id-service/using/intro/cookies.html</a>
<b><i>JSON Selector (API/Connectors)</i></b>	<a href="http://jsonpath.com/">http://jsonpath.com/</a>
<b><i>Trait Creation</i></b>	<a href="https://docs.adobe.com/content/help/en/audience-manager/user-guide/features/traits/trait-builder/create-onboarded-rule-based-traits.html#basics">https://docs.adobe.com/content/help/en/audience-manager/user-guide/features/traits/trait-builder/create-onboarded-rule-based-traits.html#basics</a>
<b><i>Data Source Creation</i></b>	<a href="https://docs.adobe.com/content/help/en/audience-manager/user-guide/features/data-sources/manage-datasources.html#create-data-source">https://docs.adobe.com/content/help/en/audience-manager/user-guide/features/data-sources/manage-datasources.html#create-data-source</a>

### 6.2 Source Code

#### 6.2.1 Adobe Pixel Code

Note: You do need to replace “<AAM client subdomain>” part of the code!

```
<html>
```

```
<head>
```

```
<meta content="text/html; charset=UTF-8" http-equiv="Content-Type" /
```

```
<title></title>
</head>

<body>
<script>
function getCookie(name) {
  let matches = document.cookie.match(new RegExp(
    "(?:^|; )" + name.replace(/[\/\?*\|\}\(\)\[\]\\\\/\+\^]/g, "\\$1") + "=([^\;]*)"
  ));
  return matches ? decodeURIComponent(matches[1]) : undefined;
}
var s_ecid = getCookie("s_ecid");
if (s_ecid.length > 0) {
  s_ecid = s_ecid.substring(8)
  var pixel = document.createElement("IMG");
  pixel.setAttribute("src", "https://<AAM client
subdomain>.demdex.net/event?d_caller=ci360saspixel&c_ci360event={{eventAction}}&c_ci360eventcategory={{e
ventCategory}}&d_mid=" + s_ecid);
  pixel.setAttribute("height", "1");
  pixel.setAttribute("width", "1");
  document.body.appendChild(pixel);
}
</script>
</body>
```

### 6.2.2 Adobe API Endpoint

`https://<Regional_Server_Host_Name>.demdex.net/event?c_caller=ci360sts&c_ci360event={{$event}}&c_ci360eventcategory={{$eventcategory}}&mid={{$s_ecid}}`

Set event to “**\$.customName**”

Set eventcategory to “**\$.customGroupName**”

Set s\_ecid to “**\$.properties.s\_ecid**”

### 6.2.3 SAS CI360 Tag Example Code

Note: This is for examples use only, please copy the code from your CI360 from the General Settings > SAS Tag Instruction section

SAS Ci360 Tag Example:

```
<!-- SAS Ci360 Tag -->
<script>
```

```
(function(ci){
  var ef=window[ci]=function(){
    return ef.q.push(arguments);
  };
  ef.q=[];ef.a={};
})('ci360');
</script>
<script async data-efname='ci360' id='ob-script-async'
  a='<tenant ID>'
  src='https://<server>/js/ot-all.min.js'></script>
```

## 6.2.4 SAS Generic Pixel Placeholder

For generic spot creation

```
<!-- SAS Pixel Placeholder -->
<div id="sas_pixel"></div>
```

## 6.2.5 SAS Adobe Placeholder

For Adobe spot creation

```
<!-- SAS Pixel Placeholder -->
<div id="sas_adobe_pixel"></div>
```

## 6.3 Connector JSON Payloads

Below is a cut down version of the json payload for reference, you can use <http://www.jsonpath.com/> to find the value you want using JSON selector notation

### 6.3.1.1 Custom Event

Example JSON Selectors

<i><b>Selector</b></i>	<b>Value (using the JSON example below)</b>
<i><b>\$.guid</b></i>	d4a1b51b-cc9b-4b25-8d76-1b5021625a5a
<i><b>\$.eventName</b></i>	adobeV4
<i><b>\$.customName</b></i>	adobeAAM
<i><b>\$.sessionID</b></i>	e105513fe064112b551dccc3
<i><b>\$.channelType</b></i>	external
<i><b>\$.date.generatedTimestamp</b></i>	1597647116198
<i><b>\$.externalTenantId</b></i>	Abcdefgh1234567890
<i><b>\$.internalTenantId</b></i>	0000
<i><b>\$.identityId</b></i>	2d25b302-6ee9-35fc-ba41-8c3033f90266
<i><b>\$.visitId</b></i>	930981fa7e3a4c58623bc80d
<i><b>\$.properties.s_ecid</b></i>	5ED56DC9A88AE1AAA57DB12722A90906
<i><b>\$.properties.externalCode</b></i>	TSK_104

<code>\$.identity.identityId</code>	2d25b302-6ee9-35fc-ba41-8c3033f90266
<code>\$.identity.sessionId</code>	e105513fe064112b551dccc3

Ex. Using the date value below in a connector variable you would use `$.date.generatedTimestamp` and the value returned would be 1597647116198

### 6.3.1.2 Example JSON Payload

It is highly recommended that you setup a server to receive the JSON payloads in order to see any differences and confirm the code is correct. This should be used for reference only!

```
{
  "guid": "d4a1b51b-cc9b-4b25-8d76-1b5021625a5a",
  "apiEventKey": null,
  "eventDesignedId": "d780a02f-2e96-4e46-a80a-9885e23bb4ec",
  "eventDesignedName": "adobeV4", //Defined in the external task under Orchestration > Outgoing Information > Event Name
  "eventName": "adobeV4", //Defined in the external task under Orchestration > Outgoing Information > Event Name
  "customName": "adobeAAM", //Custom Name defined in the trigger event (custom event) under Event Details > Event Attributes > Custom Name > Value text
  "eventType": "outboundSystem", //Defines which kind of task, external tasks are outboundSystem tasks, events would be customEvent etc.
  "sessionId": "e105513fe064112b551dccc3", //Defines the sessionId of the current user
  "channelId": "6ffeb2f866cb3044264262ab",
  "channelType": "external", //Defines the channelType
  "ipAddress": null,
  "date": {
    "generatedTimestamp": 1597647116198, //Timestamp in Unix Epoch format
  },
  "externalTenantId": "652335ef4d00013c6c2ca636", //The external Tenant ID (used in the tag on the site)
  "internalTenantId": 8721, //The internal Tenant ID
  "identityId": "2d25b302-6ee9-35fc-ba41-8c3033f90266", //The datahub ID used to define a user
  "page": {
    "loadId": "9df574667a3a4c587dca6ec7",
    "viewSequenceNum": 0,
  },
  "visitId": "930981fa7e3a4c58623bc80d", //The visitID for the user associated with this event
  "visitorGroup": null,
  "visitorState": "returning",
  "visitSequenceNum": null,
},
"properties": { //This section contains most of the custom attributes and custom properties
  "s_ecid": "5ED56DC9A88AE1AAA57DB12722A90906",
  "parent_eventname": "load",
  "externalCode": "TSK_104", //The external code defined in the external task under Properties
  "event_datetime_utc": "1597647116198", //Event time in Unix Epoch
  "parent_event": "customEvent",
  "contributing_guid_1": "a217d4c4-34f9-460a-afb7-0b7ade117c7e"
},
"identity": {
```

```
"identityId": "2d25b302-6ee9-35fc-ba41-8c3033f90266",
"identityType": null,
"identitySource": null,
"identityEventName": null,
"identityAttribute": null,
"identityAssociation": null,
"userId": null,
"visitId": null,
"ipAddress": null,
"sessionId": "e105513fe064112b551dccc3",
"visitorId": null,
"loginEventType": null,
"loginType": null,
"loginValue": null
},
"customGroupName": "adobeaaam",
"extendedCustomEventWithRevenueFlag": false,
"parentEventUid": "9155c61d-dd69-4fbb-9b49-c6f73490b90"
```

