Launch by Adobe

Reference Architecture Guide for Single-Page Applications (ReactJS with Redux)

by Adobe Customer Solutions

Version 1.0/March 12, 2018

Table of Contents

Reference Architecture Overview

Platform Scope

This Reference Architecture Guide contains best practices for implementing the Adobe Experience Cloud in single-page applications built with the ReactJS framework and using Launch by Adobe as the tag management platform. As a companion to this guide, a reference implementation using these best practices can be found at:

http://launchreactjs.businesscatalyst.com

Other Launch Reference Architectures

Additional Reference Implementation Guides will show you how to:

1. Implement the Experience Cloud in a standard website (pre-requisite for this ReactJS Guide)
2. Implement more robust features of the Experience Cloud in a standard website
3. Implement the Experience Cloud in a Video player
4. Implement the Experience Cloud in Accelerated Mobile Pages (AMP)
5. Implement the Experience Cloud in Single Page Applications (SPAs) using the AngularJS framework

Guide Version History:

v1.0 March12 , 2018 – Initial version

Goal

The goal of this document is to aid you in configuring Launch to deploy the Adobe Experience Cloud ID Service, Adobe Target, Adobe Analytics, and Adobe Audience Manager in your ReactJS application.

Assumptions

This guide assumes:

- You have working knowledge of the Adobe solutions you are implementing (i.e. Adobe Analytics, Adobe Target, Adobe Audience Manager). A technical background is helpful, but not required.
- Your Marketing Cloud Organization has been provisioned for Launch and you have full access.
- You are familiar with Launch by Adobe and already know how to deploy extensions, rules, data elements and publishing tags. It is assumed that you have already reviewed the basic website implementation guide located here, and have configured:
    - Adapters
    - Environments
    - Adobe Target Extension
- You have a ReactJS environment up and running
- You have the Launch Embed codes added on your website.

Single-page Application Instructions & Setup

Load Launch Library Synchronously

At the time of writing this document, we have only tested synchronous loading of Launch on ReactJS sites using Redux. We are currently testing asynchronous loading. Please check back later if you are interested in asynchronous.

you can view the source code of this page to see an example placement of the tags in the HTML document. http://launchreactjs.businesscatalyst.com/.

Data Layer Recommendations

A data layer is a JavaScript object which is used to store page/visitor information and can be shared with 3rd party tools and applications running on your website. The information from the data layer can be read using Adobe Launch and then sent to Adobe Marketing Cloud tools/services and other 3rd party tools that are installed via Launch extensions.

A data layer is a requirement for this implementation and Adobe Customer Solutions strongly encourages you to leverage one in any of its tag management implementations.

Data Layer Implementation

Adobe Launch is compatible with any data layer schema. If you have not already implemented a data layer or are unfamiliar with them, Adobe Customer Solutions recommends the Customer Experience Digital Data Layer Spec. See example at https://www.w3.org/2013/12/ceddl-201312.pdf.

Data Layer Location

If you have any basic data layer structure and data that you wish to load during the initial page load, you must load this before the Launch Library Header. This ensures that it is readily available for Launch when it executes the Library Loaded event or Page Top event rules.

Updating Data Layer

It is recommended to keep your data layer updated whenever a state/view change happens within your SPA application. Any information that you wish to relay to a 3rd party tool during the state/view change should be made available in the data layer. This includes page information, visitor information, event information that was triggered by the page or the user, etc. All this information could be valuable input for your Analytics or Optimization tool.

Data Layer Events

We must add a minimum of three custom events to efficiently communicate state/view changes in a SPA application. (Event names are suggestive)

event-view-start: This event should fire on the view start of the view/state that is loading. All data layer variables needed for testing and optimization tools should be loaded or updated before this event. Every view change should be considered as a start of a new view. For a better user experience (minimal to no flicker), this event should be fired as early as possible once the data layer has been updated with essential new data required for optimization tools for the new view.

event-view-end: This event should fire once the actual view/state has loaded and is ready for user interaction. This event should be fired even when a view/state change has occurred and all SPA components on the page have finished loading. Once of the best places to fire this event will be the componentDidMount lifecycle event of your container components. These components should get mounted once for every view change. This will be an ideal place to update your Data Layer with all the variables that needs to be tracked before firing this event.

- **event-action-trigger:** This event should fire when any event occurs on the page except view/state load. This could be a click event or a state change without a view change.

Example Code for triggering events

a) A library exporting re-usable functions for firing the above-mentioned events,

```
const fireViewEndCustomEvent = () => {
  var event=new CustomEvent('event-view-end');
  var obj = document.querySelector("body"); obj.dispatchEvent(event);
}

const fireViewStartCustomEvent = (data) => {
  var event=new CustomEvent('event-react-view-start', data);
  var obj = document.dispatchEvent(event);
}

const fireActionTriggerCustomEvent = (target, data) => {
  console.log("SDSDSDSDS");console.log(target, data);
  var event=new CustomEvent('event-action-trigger', data);
  var obj = target.dispatchEvent(event);
}

export {
  fireViewEndCustomEvent,
  fireViewStartCustomEvent,
  fireActionTriggerCustomEvent
}
```

b) The root App component firing the event-view-start on componentWillMount and route change

```
componentWillMount() {

 digitalData.user[0].profile[0].attributes.isBirthDay = true;
 fireViewStartCustomEvent({ 'detail': { "mbox" : "target-global-mbox" }});

 const unlisten = this.props.route.history.listen((location, action) => {
  digitalData.user[0].profile[0].attributes.highBuyingPower = true;
  if(location.action == "PUSH") {
   fireViewStartCustomEvent({ 'detail': { "mbox" : "target-global-mbox" }});
  }
 });

}
```

c) A container component's componentDidMount firing the event-view-end

```
componentDidMount() {
      digitalData.page.pageInfo.pageID = "about-demo--page";
      digitalData.page.pageInfo.pageName = "ACS Demo - About Demo Page";
      fireViewEndCustomEvent();
}
```

d) event-trigger-action fired inside a Redux action

The data that needs to be tracked are defined on the elements as data attributes for maximum re-use of code.

```
export const addToCart = (productId, target) => {
  fireActionTriggerCustomEvent(target, { detail : { "linkName" : target.getAttribute("data-
link-name") , "action" : target.getAttribute("data-track-action") }});
    return dispatch => {
      dispatch({
        type: 'REQUEST_ADD_TO_CART'
      });
   }
}
```

General Launch Configuration & Settings

Rule Configuration

A SPA configuration should consist of at least three rules: (Rule names are suggestive)

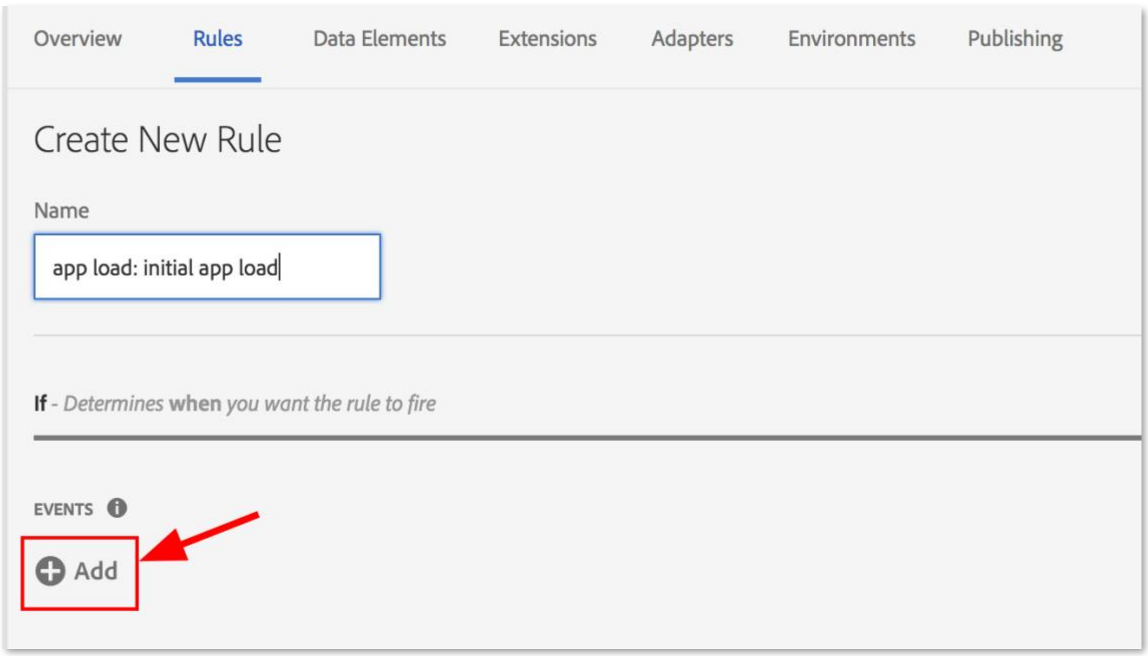
1. **app load: initial app load** – triggered at the top of the page
2. **page load: event-view-start** – triggered by the custom event 'event-view-start' that's dispatched when the view (content) starts to render
3. **page load: event-view-end** – triggered by the custom event 'event-view-end' that's dispatched when the view (content) is finished rendering

1. app load: initial app load

This rule is used for anything that needs to be loaded or fired at the top of the page. For example, if you're using Adobe Target, you would add an action to load Adobe Target.

EVENTS

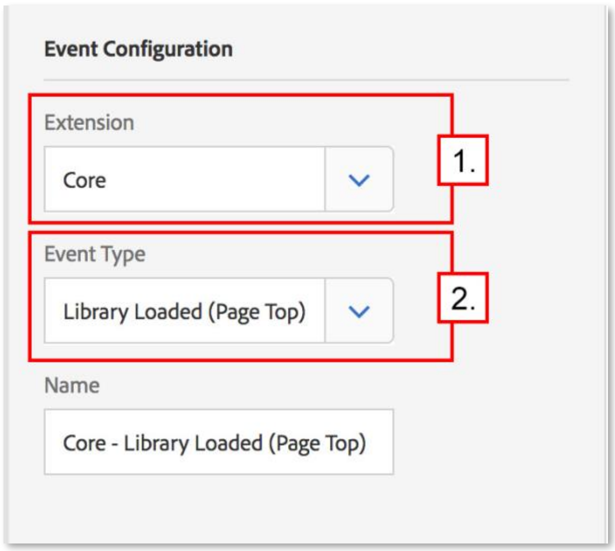
Under the Create New Rule page, provide an appropriate name for the rule, and click on the "Add" button (screenshot below) under the "EVENTS" section.



Name: app load: initial app load

If - Determines when you want the rule to fire

Under the Add Event Configuration page, follow the steps outlined below.

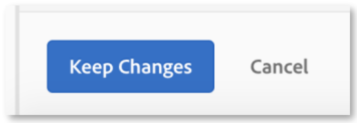

Event Configuration

Extension: Core

Event Type: Library Loaded (Page Top)

Name: Core - Library Loaded (Page Top)

1. Select "Core" from the Extension dropdown
2. Select "Library Loaded (Page Top)" from the Event Type dropdown

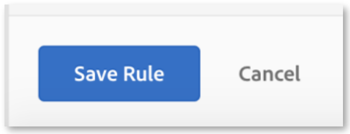


Click the Keep Changes button on the bottom of the page to Save the configuration and return to the Rule creation page.

CONDITIONS
None

EXCEPTIONS
None

ACTIONS
These will be added under the specific Experience Cloud solution sections later in this document.



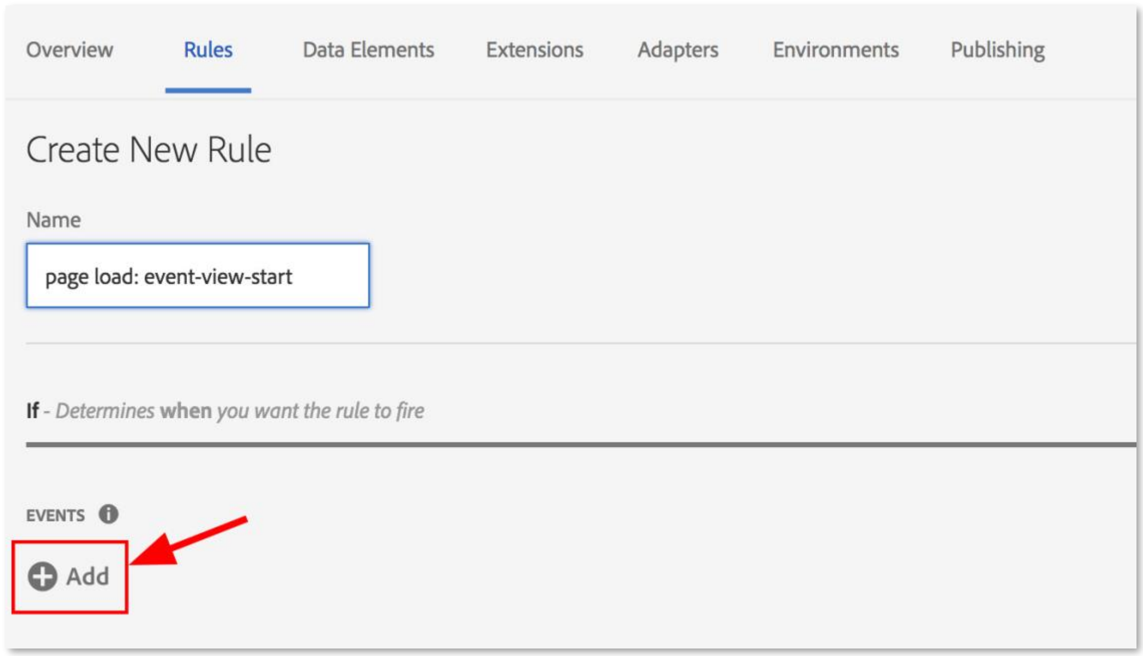
Click on the Save Rule button on the bottom of the Rule creation page to Save the Rule.

2. page load: event-view-start

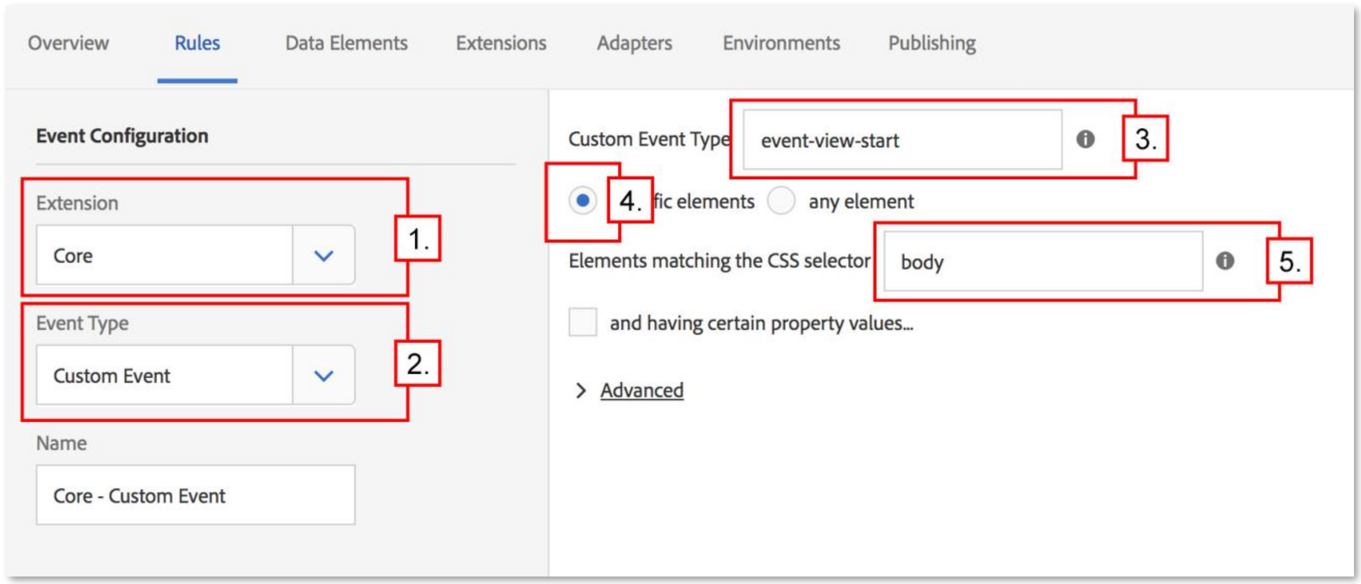
This rule is used for anything that needs to be loaded or fired after any libraries/tools have loaded, but before the view (content) has rendered. Even though , we will not be using this event rule directly in this guide due a pending feature verification, we will be setting this up now so that we can migrate easily when the feature is ready. We will instead be using a Custom Code to listen to this event. More about it on the upcoming sections.

EVENTS

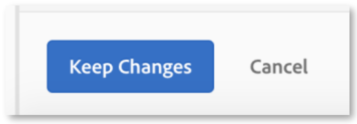
Under the Create New Rule page, provide an appropriate name for the rule, and click on the "Add" button (screenshot below) under the "EVENTS" section.



Under the Add Event Configuration page, follow the steps outlined below.



1. Select "Core" under the Extension dropdown.
2. Select "Custom Event" under the Event Type dropdown.
3. Enter "event-view-start" in the Custom Event Type input box.
4. Select Specific Elements radio button.
5. Enter "body" in the Elements matching the CSS selector input box. If you have a more specific selector such as a root DOM container ID, you could use that instead of "body". Please ensure that you dispatch the custom event to the same selector mentioned here.

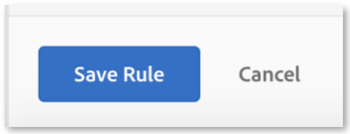


Click the Keep Changes button on the bottom of the page to Save the configuration and return to the Rule creation page.

CONDITIONS
None

EXCEPTIONS
None

ACTIONS
These will be added under the specific Experience Cloud solution sections later in this document.



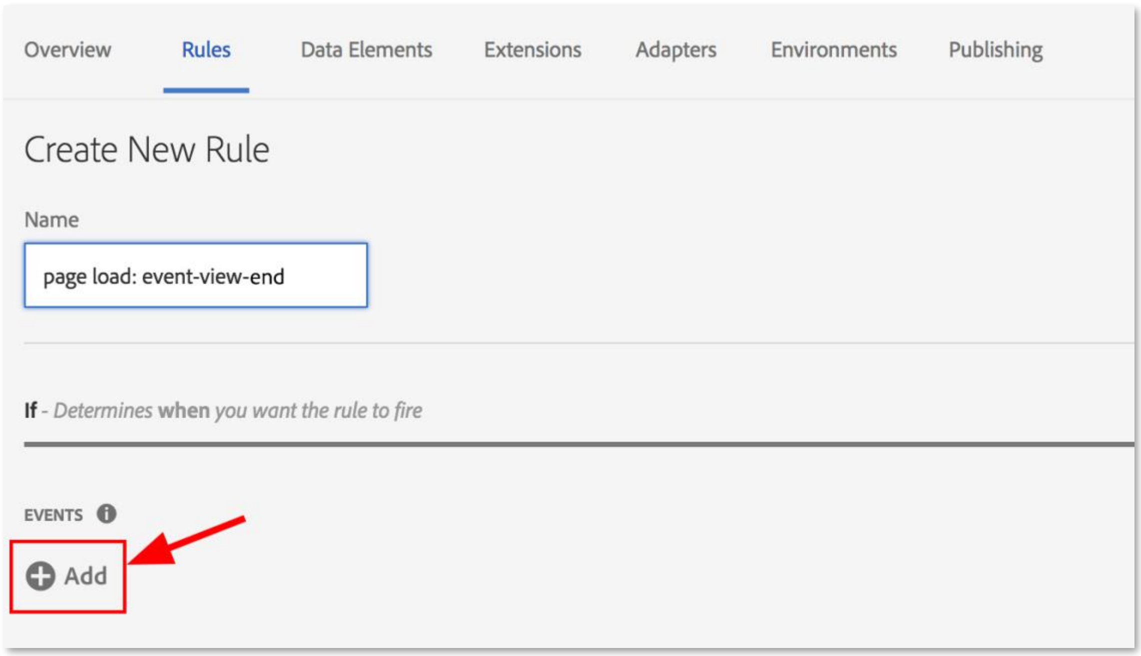
Click on the Save Rule button on the bottom of the Rule creation page to Save the Rule.

3. page load: event-view-end

This rule is triggered at the bottom of the page and is used for anything that needs to be loaded or fired after the view (content) has rendered.

EVENTS

Under the Create New Rule page, provide an appropriate name for the rule, and click on the "Add" button (screenshot below) under the "EVENTS" section.



Overview | Rules | Data Elements | Extensions | Adapters | Environments | Publishing

Create New Rule

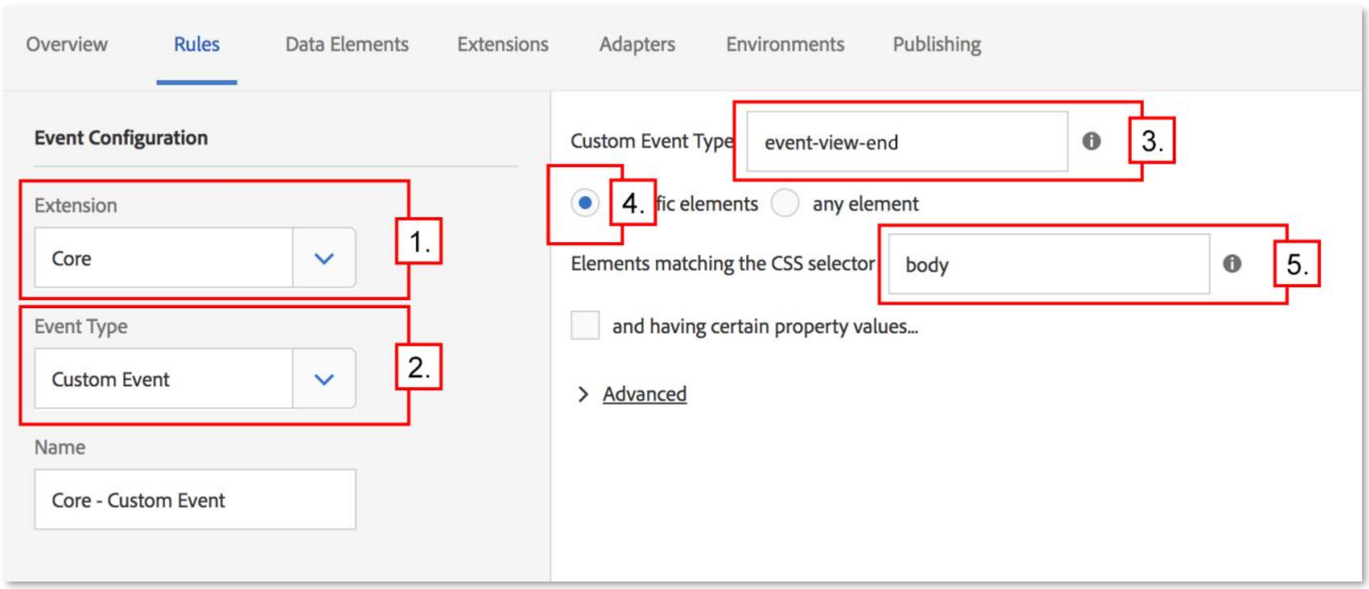
Name: page load: event-view-end

If - *Determines* **when** *you want the rule to fire*

EVENTS

Add

Under the Add Event Configuration page, follow the steps outlined below.



Overview | Rules | Data Elements | Extensions | Adapters | Environments | Publishing

Event Configuration

Extension: Core

Event Type: Custom Event

Name: Core - Custom Event

Custom Event Type: event-view-end

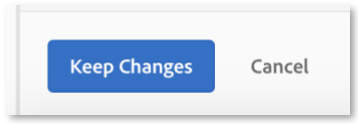
[✓ specific elements] any element

Elements matching the CSS selector: body

and having certain property values...

Advanced

1. Select "Core" under the Extension dropdown.
2. Select "Custom Event" under the Event Type dropdown.
3. Enter "event-view-end" in the Custom Event Type input box.
4. Select Specific Elements radio button.
6. Enter "body" in the Elements matching the CSS selector input box. If you have a more specific selector such as a root DOM container ID, you could use that instead of "body". Please ensure that you dispatch the custom event to the same selector mentioned here.

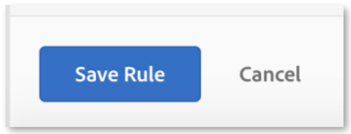


Click the Keep Changes button on the bottom of the page to Save the configuration and return to the Rule creation page.

CONDITIONS
None

EXCEPTIONS
None

ACTIONS
These will be added under the specific Experience Cloud solution sections later in this document.



Click on the Save Rule button on the bottom of the Rule creation page to Save the Rule.

Experience Cloud ID Tagging & Configuration

Configure the Experience Cloud ID Service according to the documentation in the Basic Setup Guide.

Adobe Target Configuration & Tagging

Overview

Adobe Target is the Adobe Marketing Cloud solution that provides everything you need to tailor and personalize your customers' experience, so you can maximize revenue on your web and mobile sites, apps, social media, and other digital channels.

Prerequisites & Product-Specific Setup

In order to use Adobe Target, your contract with Adobe must be provisioned for it and you must have completed the Launch Basic setup and Target extension setup in the Basic Setup Guide.

Once you've confirmed this you can install the Adobe Target extension. To do this, follow the steps below.

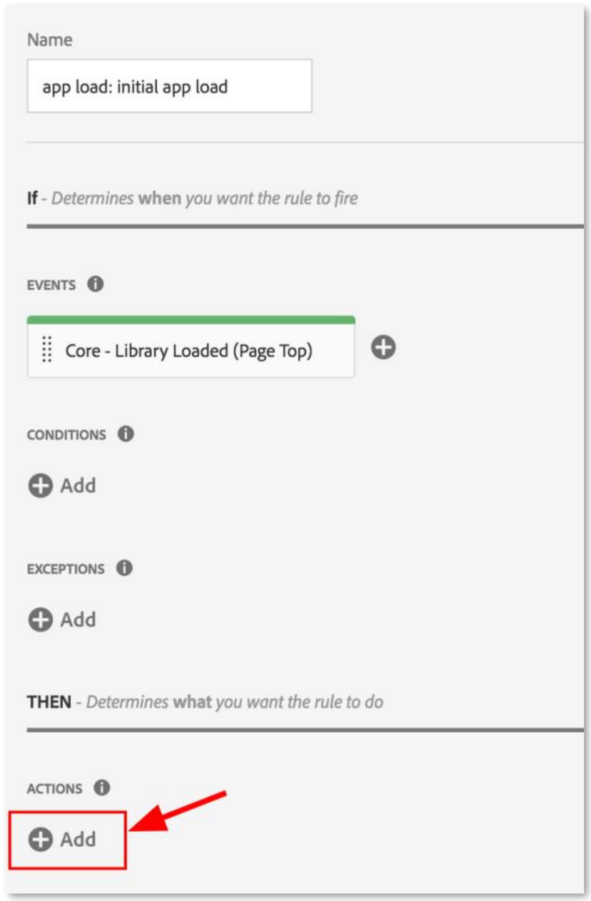
Further documentation can be found at the following URL: https://marketing.adobe.com/resources/help/en_US/experience-cloud/launch/c_extension-target.html

Rule Configuration

In order to properly load the Target library, we need to add an action to the 'app load: initial app load' rule we previously created.

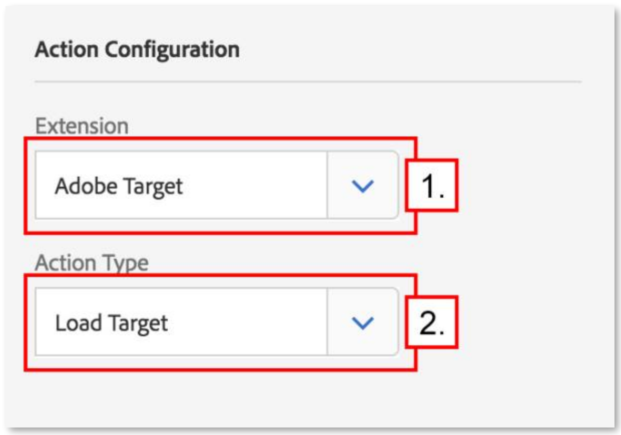
Add Actions

Navigate to the 'app load: initial app load' rule and click the Add button in the Actions section of the Edit Rule page.

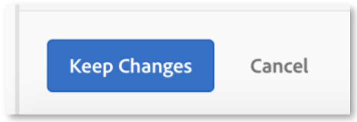

Name
app load: initial app load
If - Determines when you want the rule to fire
EVENTS
Core - Library Loaded (Page Top)
CONDITIONS
Add
EXCEPTIONS
Add
THEN - Determines what you want the rule to do
ACTIONS
Add

Load Target Action

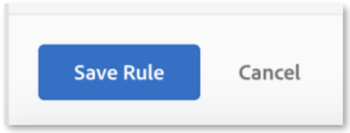
On the Action configuration page, follow the steps outlined below.



1. Select 'Adobe Target' from the Extension dropdown
2. Select 'Load Target' from the Action Type dropdown

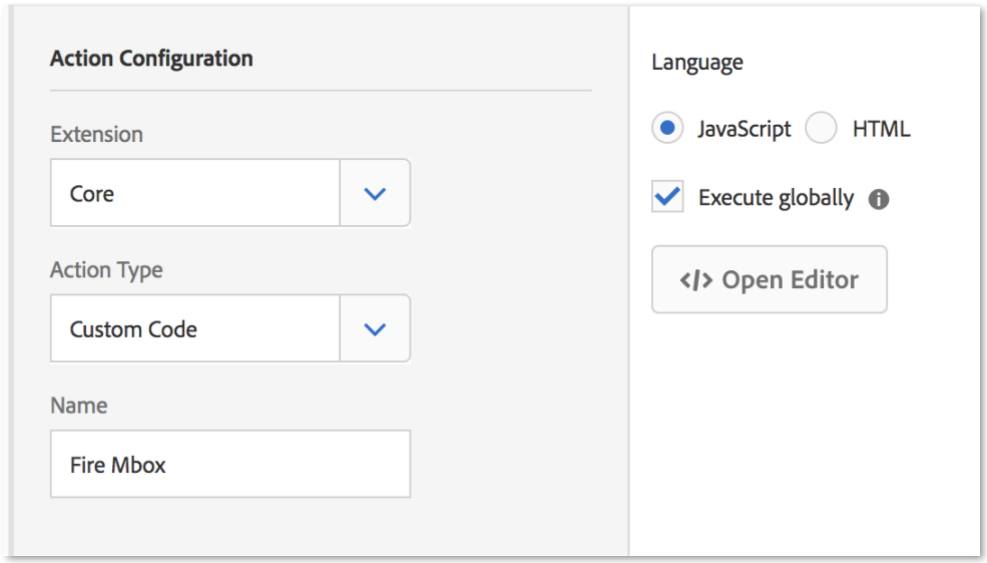


Click the Keep Changes button on the bottom of the page to Save the configuration and return to the Edit Rule page.



Click on the Save Rule button on the bottom of the Rule creation page to Save the Rule. Now that we have ensured that the Adobe Target Library will be loaded, let’s look how we can setup mBox parameters and fire the mBox in the next section.

Add a second action to type Custom Code


Action Configuration

Extension: Core

Action Type: Custom Code

Name: Fire Mbox

Language: [✓ JavaScript] HTML

[✓] Execute globally

Click Open Editor and place the below code.

```
document.addEventListener("event-view-start", function(event) {
  try
  {
    var mbox = event.detail.mbox;

    // For target-global-mbox calls
    window.targetPageParams = function() {
      return {
        "gender": digitalData.user[0].profile[0].attributes.gender,
        "age": digitalData.user[0].profile[0].attributes.age,
        "supportneeded" : digitalData.user[0].profile[0].attributes.highBuyingPower,
        "isBirthDay" : digitalData.user[0].profile[0].attributes.isBirthDay
      }
    }

    //For all mbox calls
    window.targetPageParamsAll = function() {
       return {
        "gender": digitalData.user[0].profile[0].attributes.gender,
        "age": digitalData.user[0].profile[0].attributes.age,
        "supportneeded" : digitalData.user[0].profile[0].attributes.highBuyingPower,
        "isBirthDay" : digitalData.user[0].profile[0].attributes.isBirthDay
      }
    };

    adobe.target.getOffer({
      mbox: mbox,
      success: function(offer) {
        console.info("Adobe Target custom mBox ( " + mbox + " ) request succeeded");
        var e = new CustomEvent("event-target-redux-payload", {detail:
          {
            "status": "success",
            "targetoffers" : offer,
            "mbox" : mbox
          }
        });
        document.dispatchEvent(e);
      },
      error: function(status, error) {
        console.warn("Adobe Target custom mBox ( " + mbox + " ) request did not succeed :: ",
status, error);
         var e = new CustomEvent("event-target-redux-payload", {detail:
          {
            "status": "failed",
            "message" : error
          }
        });
        document.dispatchEvent(e);
      }
    });
  }
  catch(e) {
    console.log("Error in Launch event-view-start handler", e);
  }

});
```

Modify the targetPageParams and/or targetPageParamsAll as required to send the mBox parameters that you wish to send.

Save the rule to Library, Build and Publish.

Adobe Analytics Configuration & Tagging

Overview

Adobe Analytics is an industry-leading solution that empowers you to understand your customers as people and steer your business with customer intelligence.

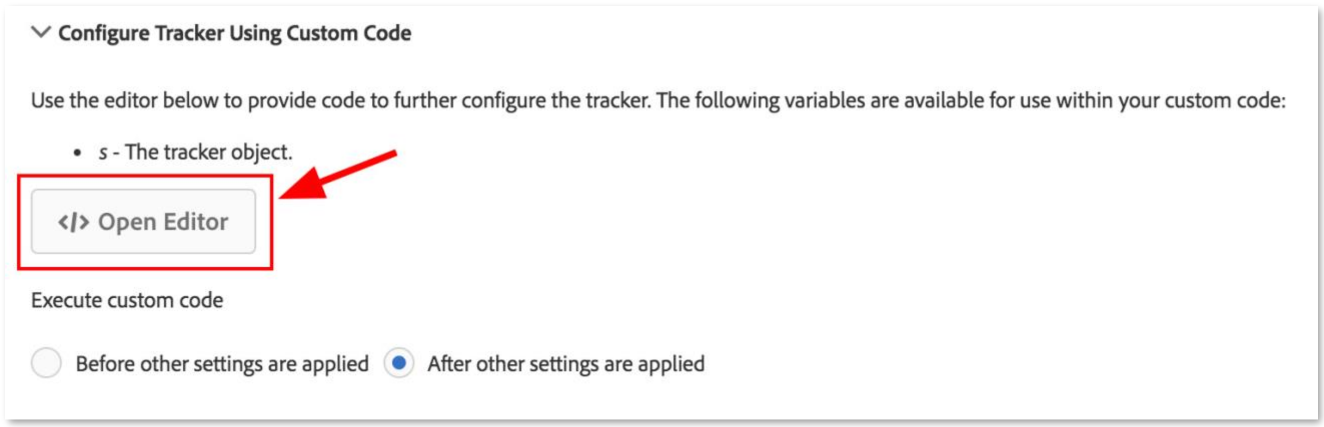
Prerequisites & Product-Specific Setup

This guide assumes you have already installed and configured the Adobe Analytics extension covered in the Basic Setup Guide.

Extension Setup

Additional Configuration

1. **Configure Tracking Using Custom Code**
   Under the Configure Tracking Using Custom Code section, follow the instructions outlined below.



Use the editor below to provide code to further configure the tracker. The following variables are available for use within your custom code:

- *s* - The tracker object.

Execute custom code

[Before other settings are applied] [✓ After other settings are applied]

Click the Open Editor button. This is where you'll add any plugins you plan to use in your implementation. For SPA implementations it's a good idea to use the following plugins:

- apl
- inList

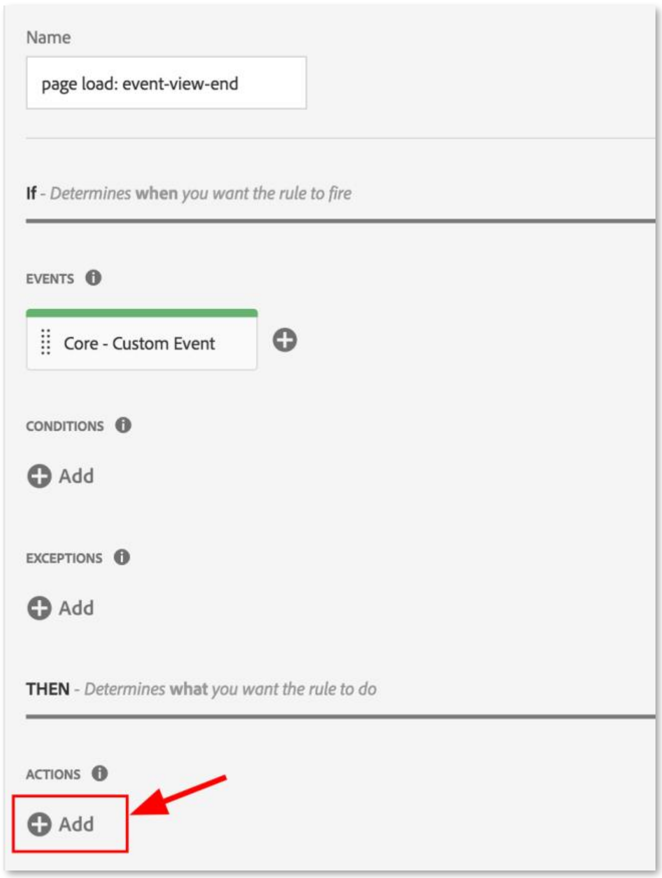
Contact your Adobe consultant to obtain the latest versions of any plugins you may need.

Rule Configuration

In order to capture and send Analytics data to the Adobe servers, we need to add actions to the 'event-view-end' rule we previously created.

Add Actions

Navigate to the 'event-view-end' rule and click the Add button in the Actions section of the Edit Rule page.


Name

page load: event-view-end

If - *Determines* **when** *you want the rule to fire*

EVENTS

Core - Custom Event

CONDITIONS

Add

EXCEPTIONS

Add

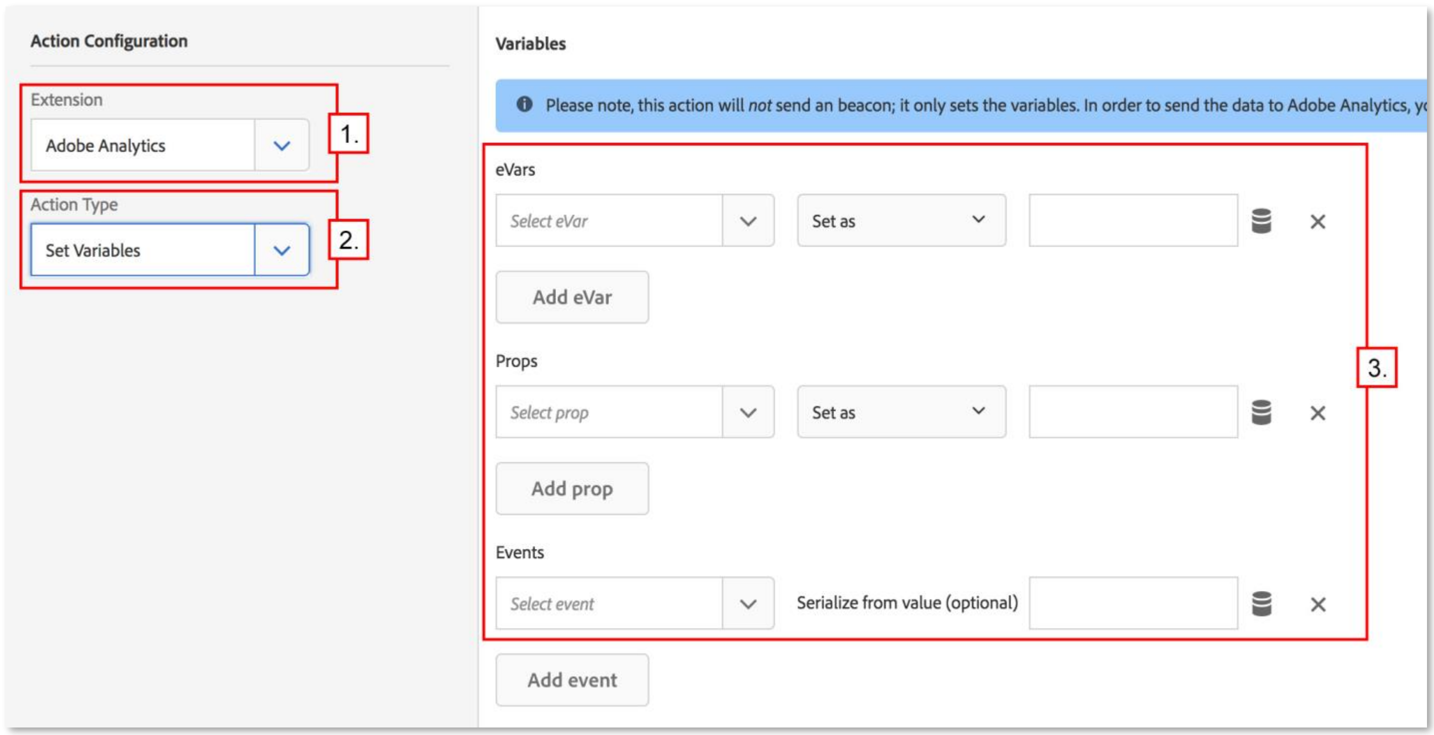
THEN - *Determines* **what** *you want the rule to do*

ACTIONS

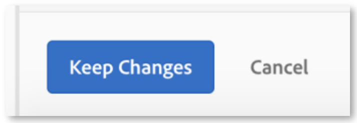
Add

Set Variables Action

On the Action Configuration page, follow the steps outlined below.



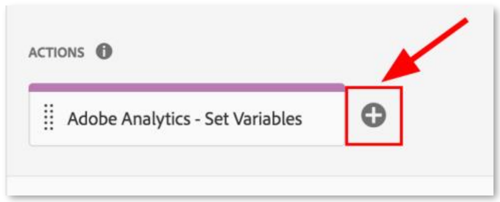
1. Select 'Adobe Analytics' from the Extension dropdown
2. Select 'Set Variables' from the Action Type dropdown
3. Configure your page load variables per your implementations requirements



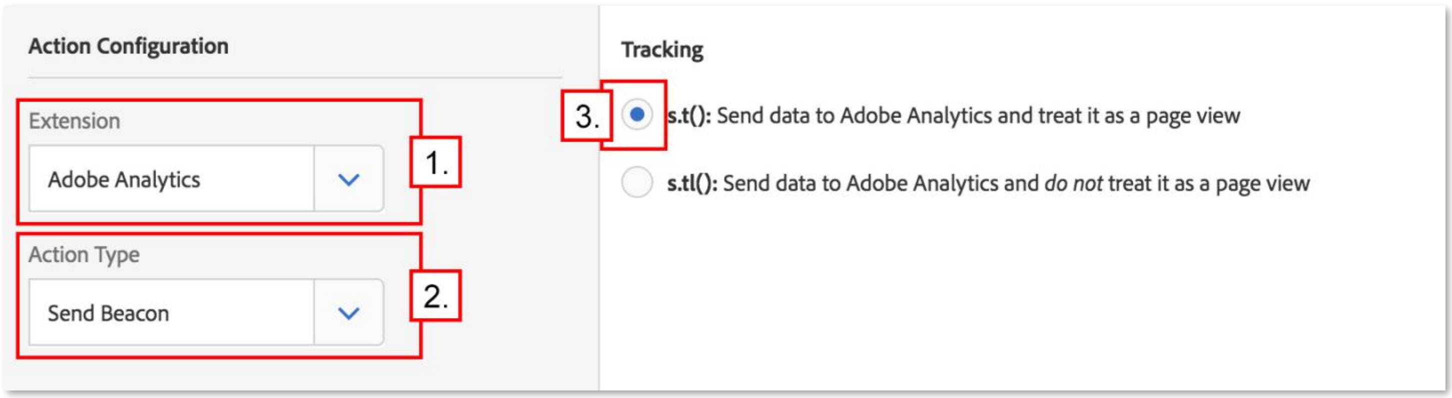
Click the Keep Changes button on the bottom of the page to Save the configuration and return to the Edit Rule page.

Send Beacon Action

Click the 'plus' icon in the Actions section



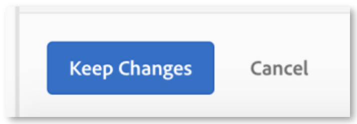
On the Action Configuration page, follow the steps outlined below.



[✓ s.t(): Send data to Adobe Analytics and treat it as a page view]

s.tl(): Send data to Adobe Analytics and *do not* treat it as a page view

1. Select 'Adobe Analytics' from the Extension dropdown
2. Select 'Send Beacon' from the Action Type dropdown
3. Select the 's.t()' radio button in the Tracking section

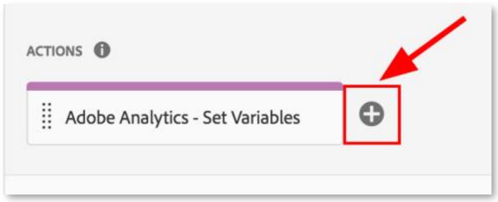


Click the Keep Changes button on the bottom of the page to Save the configuration and return to the Edit Rule page.

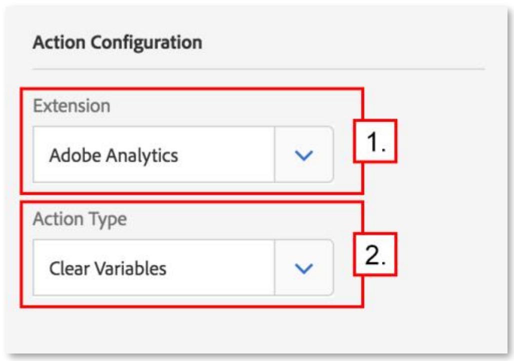
Clear Variables Action

This is a new feature in Adobe Launch and extremely helpful in SPA implementations. In a traditional implementation where the browser reloads between pages, the JavaScript object that stores all of the Analytics variables is automatically cleared out and reset when a new page is visited. This doesn't happen in a SPA because the browser doesn't reload between view/state changes. Therefore, we have to explicitly tell Launch when to clear the object. If we don't, variables will persist between image requests and skew our data.

Click the 'plus' icon in the Actions section



On the Action Configuration page, follow the steps outlined below.

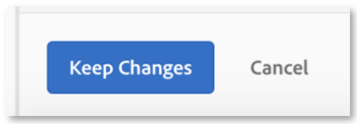


Action Configuration

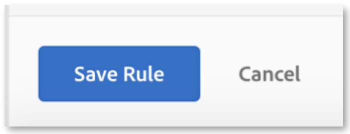
Extension: Adobe Analytics

Action Type: Clear Variables

1. Select 'Adobe Analytics' from the Extension dropdown
2. Select 'Clear Variables' from the Action Type dropdown



Click the Keep Changes button on the bottom of the page to Save the configuration and return to the Edit Rule page.



Click on the Save Rule button on the bottom of the Rule creation page to Save the Rule.

React JS and Redux Framework Instructions & Setup for Adobe Target

ReactJS Environment

We have used the React Boilerplate (https://www.reactboilerplate.com/) for this guide. You should be able to adapt the instructions in this guide with minimal or no changes to your ReactJS implementations.

You can see a demo of the setup mentioned here on this page - http://launchreactjs.businesscatalyst.com/

Please note that the NPM Package is still under review and this section may undergo changes in v1.1 of this document. The anticipated general availability of the package is April. Since the package is not available for general, the following steps below will need support from an Adobe Consultant for customizing the solution to work with your ReactJS/Redux app

Install the NPM package

From https://www.npmjs.com/

On your terminal, run

```
npm install adobe-target-react-redux
```

From Adobe Internal Git

On your terminal, run

```
npm install git+ssh://git@git.corp.adobe.com:bennjose/adobe-target-react-redux.git
```

From a local folder (Only for testing/development)

- Clone the GitHub repo https://git.corp.adobe.com/bennjose/adobe-target-react-redux or download it as zip and unzip it to your folder of choice. (you can download and send the zip folder to customer)
- On your terminal, inside the repo folder, run

```
npm install
```

- To build the package, run

```
npm run build:production
```

- To create an NPM softlink, run

```
npm link
```

- Go inside your ReactJS web application package folder, run

```
npm link adobe-target-react-redux
```

- Build your application to ensure that there are no dependency errors.

Using the adobe-target-react-redux component

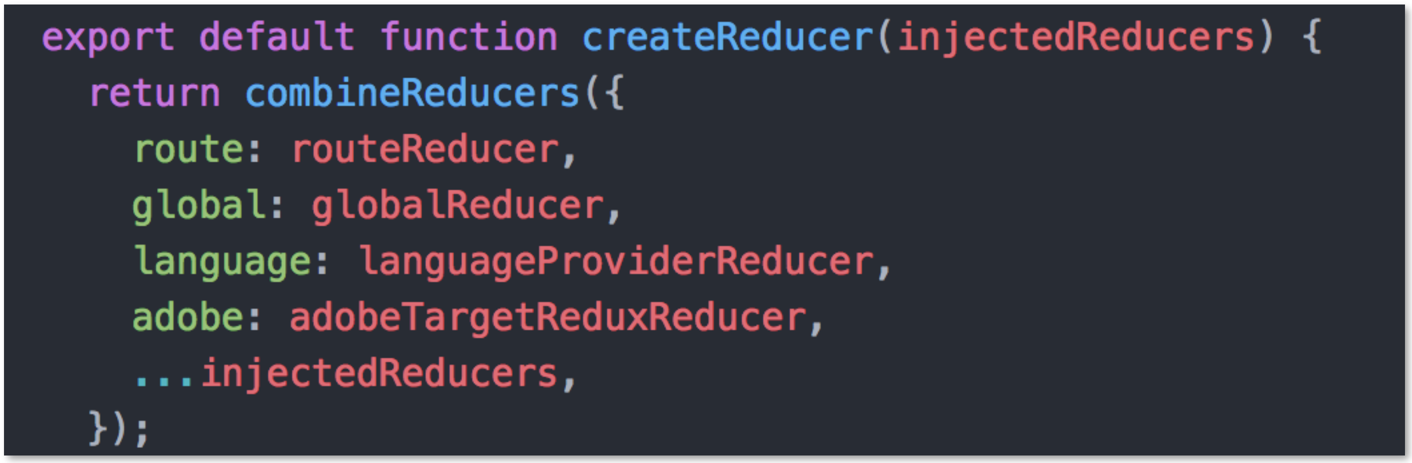
Add the Redux Reducer

Import the Reducer function from the component and combine it with your reducer.

```
import { adobeTargetReduxReducer } from 'adobe-target-react-redux';
```

The reducer key must be named as “adobe”

If you are using combineReducers from ‘redux-immutable’, your code might look something like this.



```
export default function createReducer(injectedReducers) {
  return combineReducers({
    route: routeReducer,
    global: globalReducer,
    language: languageProviderReducer,
    adobe: adobeTargetReduxReducer,
    ...injectedReducers,
  });
```

Connect the Redux Store with Target

This step will the ensure that the component can dispatch actions to the redux store.

On the module where you are creating and initializing the store, import “bindAdobeTargetToStore” function

```
import { bindAdobeTargetToStore } from 'adobe-target-react-redux'
```

After you have initialized the store, pass the “store.dispatch” function to “bindAdobeTargetToStore” function

```
bindAdobeTargetToStore(store.dispatch);
```

Fire the Custom Event

The “event-view-start” listener that we configured in Launch Rule will have to be triggered to send the mBox call.

This event must be triggered in the component which handles the view transitions or routing so that you can fire the action on route change or view change.

Before firing this event, you must ensure that you have updated the data layer with data which needs to be used for targeting and segmenting in Adobe Target campaigns

```
  //Ensure Data layer is ready for new view
    var event=new CustomEvent('event-view-start', { 'detail':
      {
        "mbox" : "target-global-mbox"
      }
    });
    document.dispatchEvent(event);
```

Wrap the components

The final step is to wrap the component(s) that you wish to manipulate using Adobe Target.

Wrapping Strategy

You must decide whether to wrap the root wrapper component or to wrap individual components.

If the page is a good candidate for A/B testing by re-arranging or moving components around, then you must wrap the outermost wrapper component which contains all the components you wish to re-arrange or move.

If the page contains components which can be tested individually by changing content inside each component, then it is recommended that you wrap those individual components.

How to Wrap Components.

Import the Target Component to your Component Class which has the render method.

```
import { AdobeTargetForReact } from 'adobe-target-react-redux';
```

Wrap your JSX with AdobeTargetForReact component and send your JSX as a render prop.

In the below example you can see how a header <div> is wrapped.

```
render() {
    return (
      <AdobeTargetForReact location = "StaticHeaderComponent" render =
         {() => (
          <div>
            <A href="https://twitter.com/mxstbr">
              <Img src={Banner} alt="react-boilerplate - Logo" />
            </A>
            <NavBar>
              <HeaderLink to="/">
                <FormattedMessage {...messages.home} />
              </HeaderLink>
              <HeaderLink to="/features">
                <FormattedMessage {...messages.features} />
              </HeaderLink>
            </NavBar>
          </div>
        )}
       />
    );
  }
```

The location prop should have a unique component name per page, if you have a component which shared across website, then it must unique across website. you can adapt the same regional mBox naming strategy for naming the location. Please note that there won't be any extra mBox calls for each wrapped component. This location is merely for uniquely identifying each component.

Running Target Campaigns

You can use all VEC WYSIWIG actions to modify your components as long as the start and end of DOM modification is contained inside a single Location boundary. So, it is good to plan how you will wrap your components for maximum ease of use.

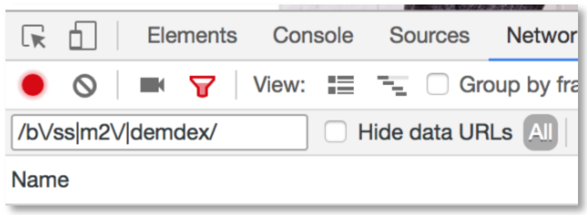
For example: if you are moving a button from place A to B, place A and B should be inside the same Location name/wrapper.

Custom Code support will be available soon for React Redux apps. Please check back later if you wish to use Custom Code.

Validating the Implementation

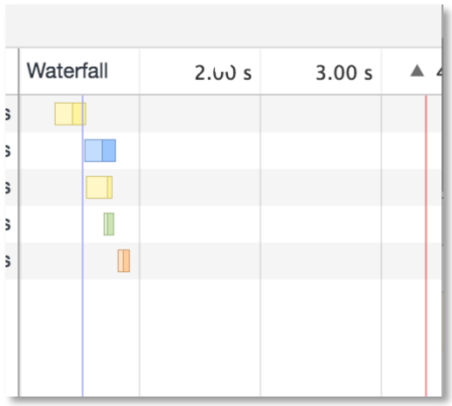
Validate Calls to Adobe Servers

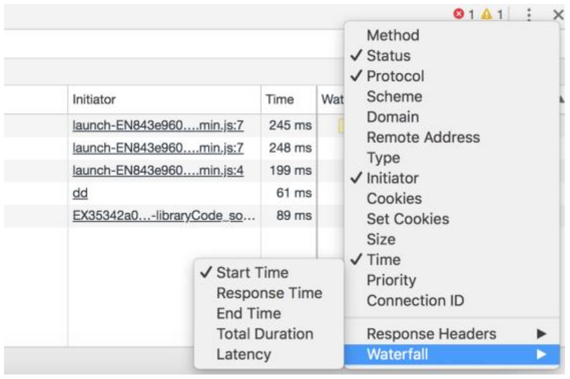
- Do a full re-load/refresh of the page on clean (No cookies) Google Chrome
- Open the developer tool bar's Network Tab
- Enter this regex in the filter - **/b\/ss|m2\//**



/bVss|m2V|demdex/

- This will show all the calls happening related to Adobe Experience Cloud products
- Ensure you are sorting in ascending order of the request start time. If not, you can do this by clicking on the Waterfall diagram. Ensure that the waterfall diagram is showing the Start time.

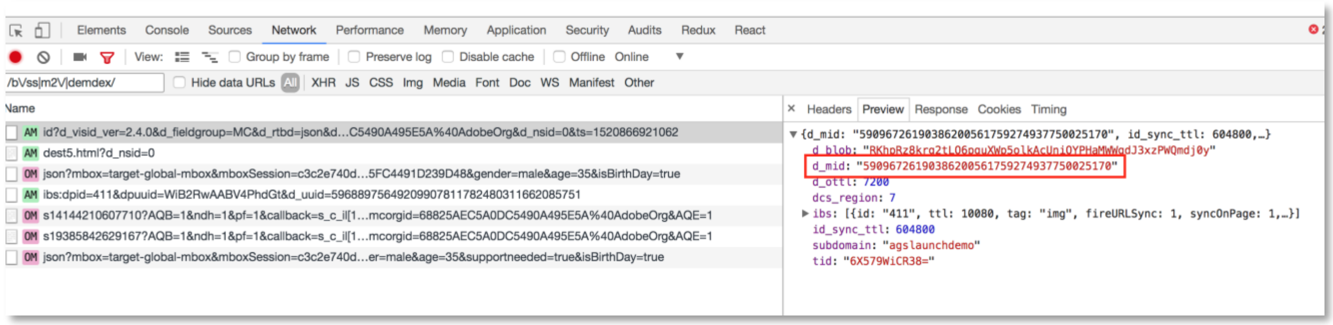




Initiator	Time
launch-EN843e960....min.js:7	245 ms
launch-EN843e960....min.js:7	248 ms
launch-EN843e960....min.js:4	199 ms
dd	61 ms
EX35342a0...-libraryCode_so...	89 ms

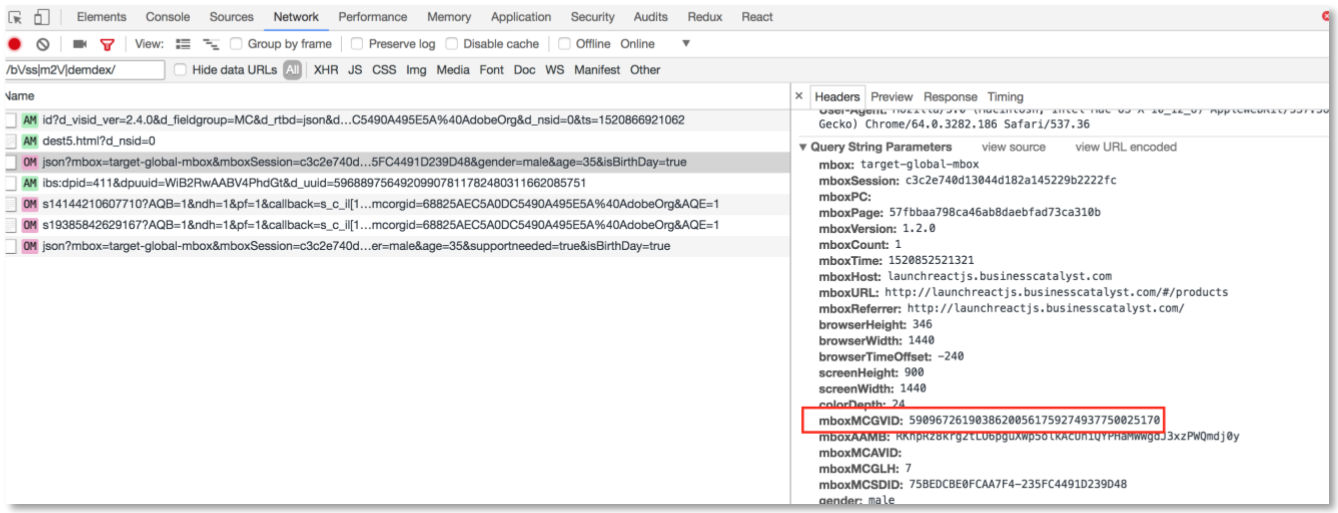
- Ensure you have the order of the calls as below
  - Visitor API call (Ex:- URL starts with http://dpm.demdex.net/id?d_visid_ver=........)
  - Target call (Ex:- <clientcode>.tt.omtrdc.net)
  - Analytics call (Ex:- <logincompany.sc.omtrdc.net)
- Navigate to another view, and ensure for every new view, Target call happens before Analytics call.

- Ensure the mid is matching on all the calls.



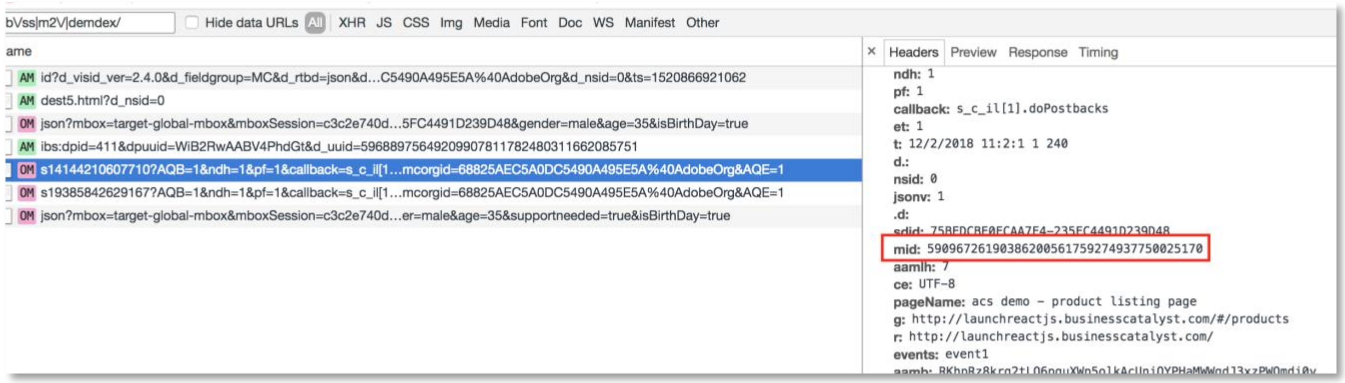
```
AM id?d_visid_ver=2.4.0&d_fieldgroup=MC&d_rtbd=json&d…C5490A495E5A%40AdobeOrg&d_nsid=0&ts=1520866921062
AM dest5.html?d_nsid=0
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…5FC4491D239D48&gender=male&age=35&isBirthDay=true
AM ibs:dpid=411&dpuuid=WiB2RwAABV4PhdGt&d_uuid=59688975649209907811782480311662085751
OM s14144210607710?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM s19385842629167?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…er=male&age=35&supportneeded=true&isBirthDay=true
```

```
▼ {d_mid: "59096726190386200561759274937750025170", id_sync_ttl: 604800,…}
    d_blob: "RKhpRz8krg2tLO6pguXWp5olkAcUniQYPHaMWWgdJ3xzPWQmdj0y"
    d_mid: "59096726190386200561759274937750025170"
    d_ottl: 7200
    dcs_region: 7
  ▶ ibs: [{id: "411", ttl: 10080, tag: "img", fireURLSync: 1, syncOnPage: 1,…}]
    id_sync_ttl: 604800
    subdomain: "agslaunchdemo"
    tid: "6X579WiCR38="
```



```
AM id?d_visid_ver=2.4.0&d_fieldgroup=MC&d_rtbd=json&d…C5490A495E5A%40AdobeOrg&d_nsid=0&ts=1520866921062
AM dest5.html?d_nsid=0
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…5FC4491D239D48&gender=male&age=35&isBirthDay=true
AM ibs:dpid=411&dpuuid=WiB2RwAABV4PhdGt&d_uuid=59688975649209907811782480311662085751
OM s14144210607710?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM s19385842629167?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…er=male&age=35&supportneeded=true&isBirthDay=true
```

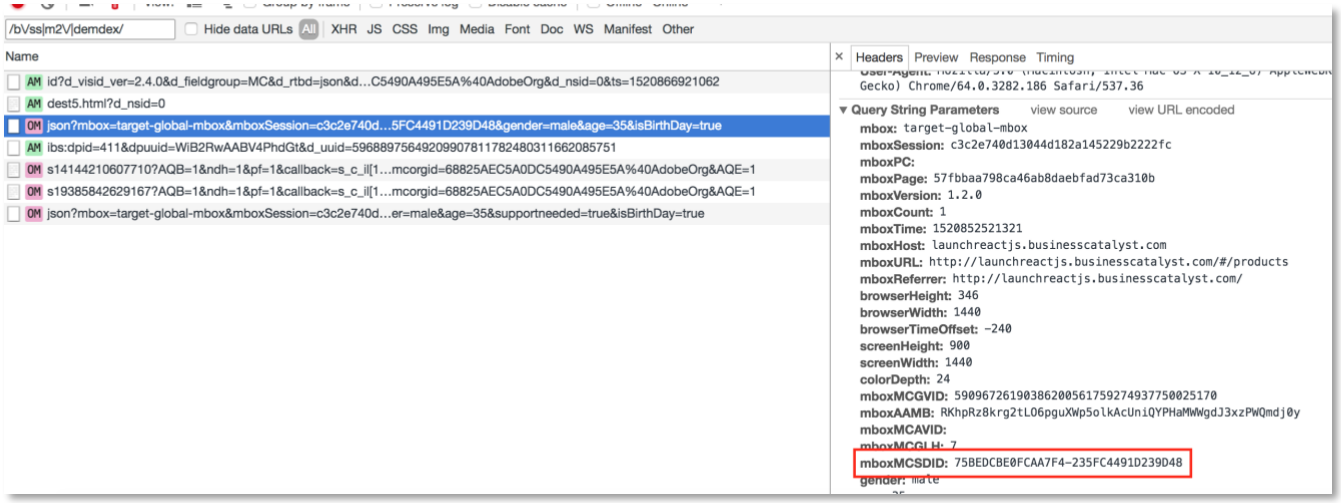
```
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36
  Gecko) Chrome/64.0.3282.186 Safari/537.36
▼ Query String Parameters      view source      view URL encoded
    mbox: target-global-mbox
    mboxSession: c3c2e740d13044d182a145229b2222fc
    mboxPC:
    mboxPage: 57fbbaa798ca46ab8daebfad73ca310b
    mboxVersion: 1.2.0
    mboxCount: 1
    mboxTime: 1520852521321
    mboxHost: launchreactjs.businesscatalyst.com
    mboxURL: http://launchreactjs.businesscatalyst.com/#/products
    mboxReferrer: http://launchreactjs.businesscatalyst.com/
    browserHeight: 346
    browserWidth: 1440
    browserTimeOffset: -240
    screenHeight: 900
    screenWidth: 1440
    colorDepth: 24
    mboxMCGVID: 59096726190386200561759274937750025170
    mboxAAMB: RKhpRz8krg2tLO6pguXWp5olkAcUniQYPHaMWWgdJ3xzPWQmdj0y
    mboxMCAVID:
    mboxMCGLH: 7
    mboxMCSDID: 75BEDCBE0FCAA7F4-235FC4491D239D48
    gender: male
```



```
AM id?d_visid_ver=2.4.0&d_fieldgroup=MC&d_rtbd=json&d…C5490A495E5A%40AdobeOrg&d_nsid=0&ts=1520866921062
AM dest5.html?d_nsid=0
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…5FC4491D239D48&gender=male&age=35&isBirthDay=true
AM ibs:dpid=411&dpuuid=WiB2RwAABV4PhdGt&d_uuid=59688975649209907811782480311662085751
OM s14144210607710?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM s19385842629167?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…er=male&age=35&supportneeded=true&isBirthDay=true
```

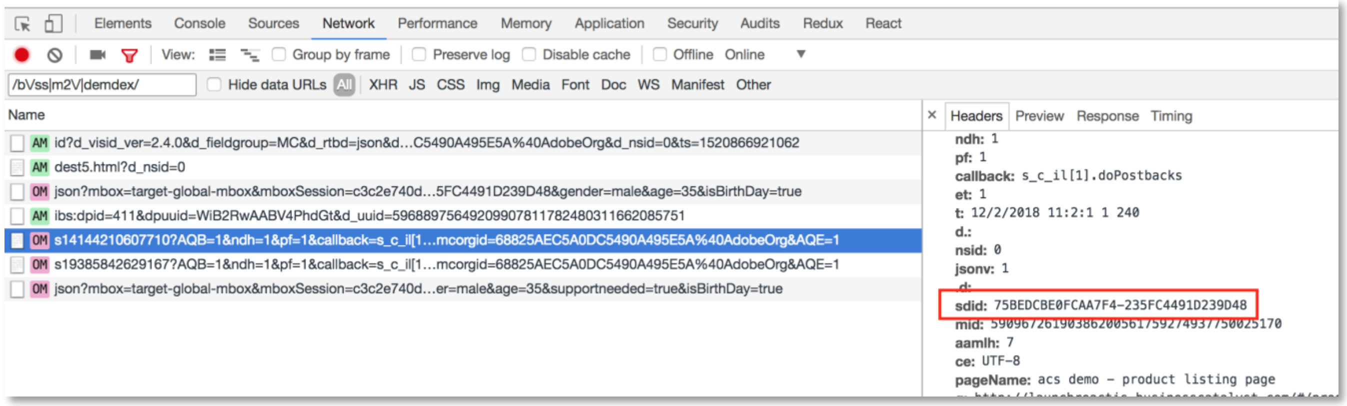
```
ndh: 1
pf: 1
callback: s_c_il[1].doPostbacks
et: 1
t: 12/2/2018 11:2:1 1 240
d.:
nsid: 0
jsonv: 1
.d:
sdid: 75BEDCBE0FCAA7F4-235FC4491D239D48
mid: 59096726190386200561759274937750025170
aamlh: 7
ce: UTF-8
pageName: acs demo - product listing page
g: http://launchreactjs.businesscatalyst.com/#/products
r: http://launchreactjs.businesscatalyst.com/
events: event1
aamb: RKhpRz8krg2tLO6pguXWp5olkAcUniQYPHaMWWgdJ3xzPWQmdj0y
```

- Ensure SDID is matching with between Target and Analytics call per view



```
/bVss|m2V|demdex/   Hide data URLs  All  XHR  JS  CSS  Img  Media  Font  Doc  WS  Manifest  Other

Name
AM id?d_visid_ver=2.4.0&d_fieldgroup=MC&d_rtbd=json&d…C5490A495E5A%40AdobeOrg&d_nsid=0&ts=1520866921062
AM dest5.html?d_nsid=0
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…5FC4491D239D48&gender=male&age=35&isBirthDay=true
AM ibs:dpid=411&dpuuid=WiB2RwAABV4PhdGt&d_uuid=59688975649209907811782480311662085751
OM s14144210607710?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM s19385842629167?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…er=male&age=35&supportneeded=true&isBirthDay=true
```

```
× Headers  Preview  Response  Timing
  Gecko) Chrome/64.0.3282.186 Safari/537.36
▼ Query String Parameters     view source     view URL encoded
  mbox: target-global-mbox
  mboxSession: c3c2e740d13044d182a145229b2222fc
  mboxPC:
  mboxPage: 57fbbaa798ca46ab8daebfad73ca310b
  mboxVersion: 1.2.0
  mboxCount: 1
  mboxTime: 1520852521321
  mboxHost: launchreactjs.businesscatalyst.com
  mboxURL: http://launchreactjs.businesscatalyst.com/#/products
  mboxReferrer: http://launchreactjs.businesscatalyst.com/
  browserHeight: 346
  browserWidth: 1440
  browserTimeOffset: -240
  screenHeight: 900
  screenWidth: 1440
  colorDepth: 24
  mboxMCGVID: 59096726190386200561759274937750025170
  mboxAAMB: RKhpRz8krg2tLO6pguXWp5olkAcUniQYPHaMWWgdJ3xzPWQmdj0y
  mboxMCAVID:
  mboxMCGLH: 7
  mboxMCSDID: 75BEDCBE0FCAA7F4-235FC4491D239D48
  gender: male
```

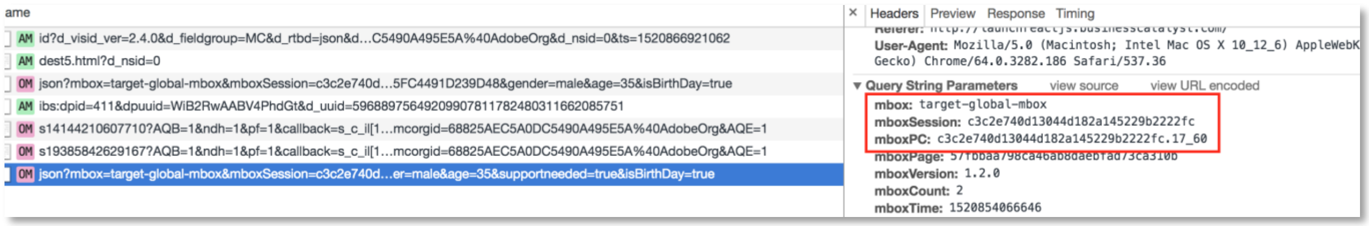


```
/bVss|m2V|demdex/   Hide data URLs  All  XHR  JS  CSS  Img  Media  Font  Doc  WS  Manifest  Other

Name
AM id?d_visid_ver=2.4.0&d_fieldgroup=MC&d_rtbd=json&d…C5490A495E5A%40AdobeOrg&d_nsid=0&ts=1520866921062
AM dest5.html?d_nsid=0
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…5FC4491D239D48&gender=male&age=35&isBirthDay=true
AM ibs:dpid=411&dpuuid=WiB2RwAABV4PhdGt&d_uuid=59688975649209907811782480311662085751
OM s14144210607710?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM s19385842629167?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…er=male&age=35&supportneeded=true&isBirthDay=true
```

```
× Headers  Preview  Response  Timing
  ndh: 1
  pf: 1
  callback: s_c_il[1].doPostbacks
  et: 1
  t: 12/2/2018 11:2:1 1 240
  d.:
  nsid: 0
  jsonv: 1
  .d:
  sdid: 75BEDCBE0FCAA7F4-235FC4491D239D48
  mid: 59096726190386200561759274937750025170
  aamlh: 7
  ce: UTF-8
  pageName: acs demo - product listing page
```

Validate Target Call

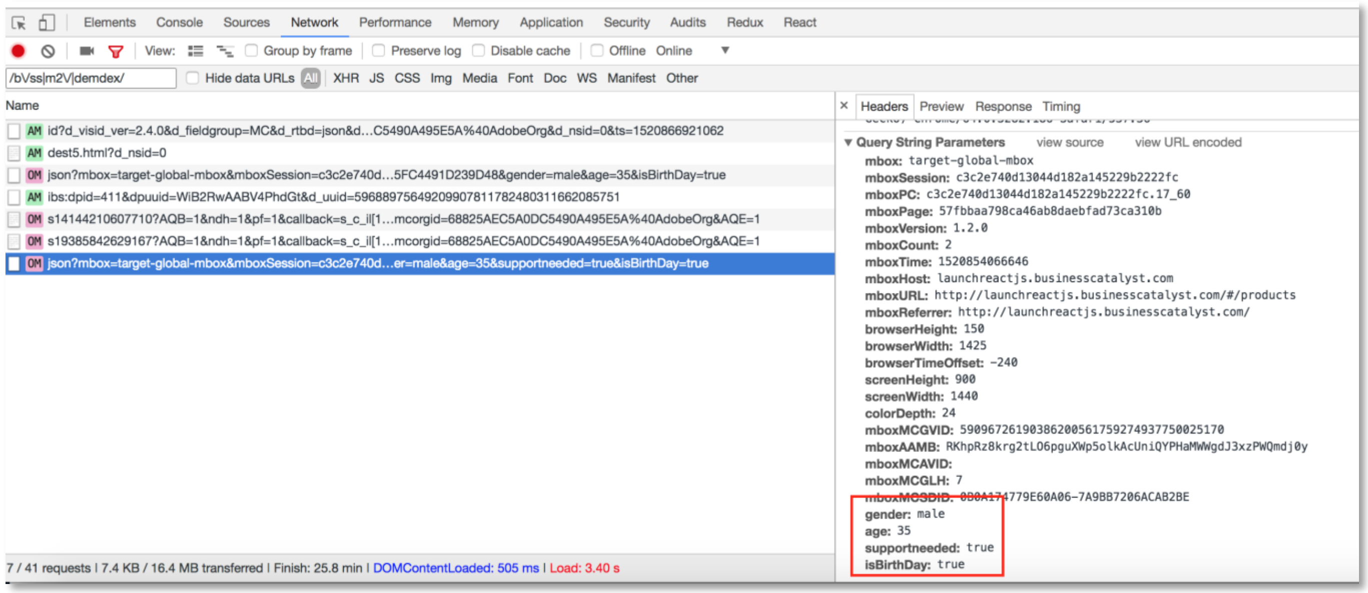
- Ensure that the mbox name matches what you have configured in Launch
- Ensure the mBox call sets mboxPC and mboxSession consistently on each view. mboxPC will remain same for a browser until you delete the cookie or is inactive for 14 days by default. mboxSession will change if you are inactive for 30 minutes in a browser.



```
ame
AM id?d_visid_ver=2.4.0&d_fieldgroup=MC&d_rtbd=json&d…C5490A495E5A%40AdobeOrg&d_nsid=0&ts=1520866921062
AM dest5.html?d_nsid=0
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…5FC4491D239D48&gender=male&age=35&isBirthDay=true
AM ibs:dpid=411&dpuuid=WiB2RwAABV4PhdGt&d_uuid=59688975649209907811782480311662085751
OM s14144210607710?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM s19385842629167?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…er=male&age=35&supportneeded=true&isBirthDay=true
```

```
× Headers  Preview  Response  Timing
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebK
  Gecko) Chrome/64.0.3282.186 Safari/537.36
▼ Query String Parameters     view source     view URL encoded
  mbox: target-global-mbox
  mboxSession: c3c2e740d13044d182a145229b2222fc
  mboxPC: c3c2e740d13044d182a145229b2222fc.17_60
  mboxPage: 57fbbaa798ca46ab8daebfad73ca310b
  mboxVersion: 1.2.0
  mboxCount: 2
  mboxTime: 1520854066646
```

- Ensure your mbox and in-mbox profile parameters are sending correct key/value pairs



```
/bVss|m2V|demdex/

Name
AM id?d_visid_ver=2.4.0&d_fieldgroup=MC&d_rtbd=json&d…C5490A495E5A%40AdobeOrg&d_nsid=0&ts=1520866921062
AM dest5.html?d_nsid=0
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…5FC4491D239D48&gender=male&age=35&isBirthDay=true
AM ibs:dpid=411&dpuuid=WiB2RwAABV4PhdGt&d_uuid=59688975649209907811782480311662085751
OM s14144210607710?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM s19385842629167?AQB=1&ndh=1&pf=1&callback=s_c_il[1…mcorgid=68825AEC5A0DC5490A495E5A%40AdobeOrg&AQE=1
OM json?mbox=target-global-mbox&mboxSession=c3c2e740d…er=male&age=35&supportneeded=true&isBirthDay=true

7 / 41 requests | 7.4 KB / 16.4 MB transferred | Finish: 25.8 min | DOMContentLoaded: 505 ms | Load: 3.40 s
```

```
▼ Query String Parameters
  mbox: target-global-mbox
  mboxSession: c3c2e740d13044d182a145229b2222fc
  mboxPC: c3c2e740d13044d182a145229b2222fc.17_60
  mboxPage: 57fbbaa798ca46ab8daebfad73ca310b
  mboxVersion: 1.2.0
  mboxCount: 2
  mboxTime: 1520854066646
  mboxHost: launchreactjs.businesscatalyst.com
  mboxURL: http://launchreactjs.businesscatalyst.com/#/products
  mboxReferrer: http://launchreactjs.businesscatalyst.com/
  browserHeight: 150
  browserWidth: 1425
  browserTimeOffset: -240
  screenHeight: 900
  screenWidth: 1440
  colorDepth: 24
  mboxMCGVID: 59096726190386200561759274937750025170
  mboxAAMB: RKhpRz8krg2tLO6pguXWp5olkAcUniQYPHaMWWgdJ3xzPWQmdj0y
  mboxMCAVID:
  mboxMCGLH: 7
  mboxMCSDID: 0D0A174779E60A06-7A9BB7206ACAB2BE
  gender: male
  age: 35
  supportneeded: true
  isBirthDay: true
```

Validate Target Activity Authoring and Delivering.

- Setup an A/B activity on you SPA view using Adobe Target
- You can use all VEC actions available inside the locations you have tagged.
- Activate the campaign using QA parameters.
- Test the pages to ensure that the Activity is delivering the offer code and applying it.

Adobe Audience Manager (AAM) Tagging & Configuration

Overview

There are two ways to implement AAM code, server-side forwarding and client-side DIL.

Server-Side Forwarding (SSF) – If clients have Adobe Analytics this solution forwards Adobe Analytics data to AAM on the back end, allowing for one less pixel on the page. This also enables key integration features and conforms with our best practices for AAM code implementation and deployment.

Client-Side DIL – This solution covers anyone who does not have Adobe Analytics. DIL code (Data Integration Library Code) sends data directly from the web page into AAM. This is also utilized for clients who have Google Analytics.

Server-side or Client-side Instructions:

Server-side forwarding

Server-side forwarding is implemented on this demo site because it also has Adobe Analytics. The instructions on how it was implemented, and how you can implement server-side forwarding on your site can be found under the "Basic Setup with Launch by Adobe" section here - https://helpx.adobe.com/experience-manager/kt/integration/using/launch-reference-architecture-guides.html. Please complete the steps in that document to implement server-side forwarding of data from Adobe Analytics to Adobe Audience Manager.

Client-side

Client-side implementation options are presented to you below, in case that better represents your environment.

Prerequisites & Product-Specific Setup – AAM Only – Client Side

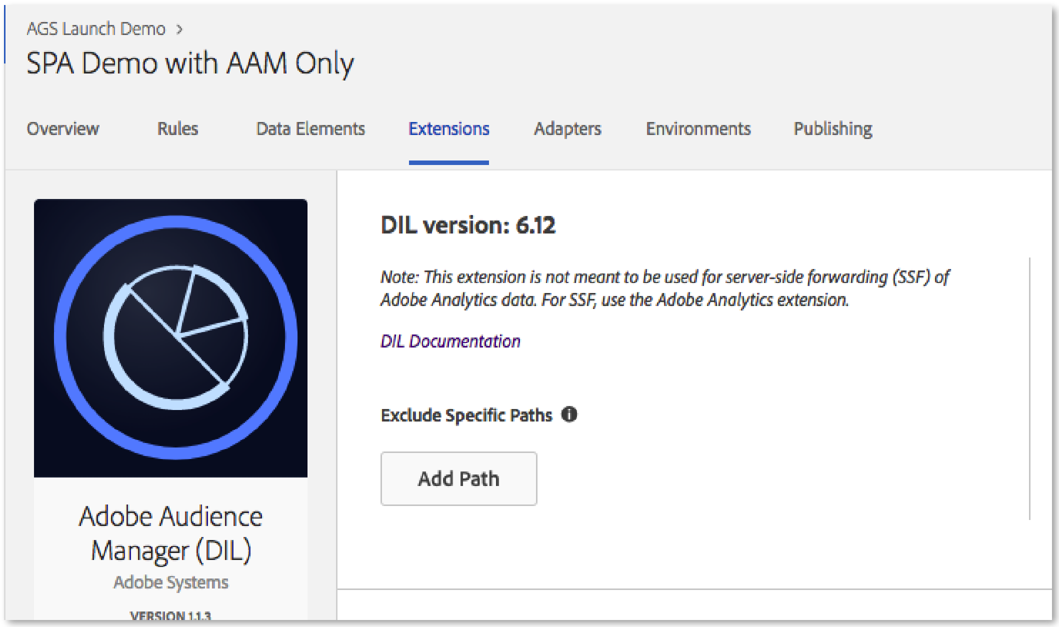
Client-Side DIL – this solution covers anyone who has neither Adobe Analytics nor Google Analytics. DIL code (Data Integration Library Code) sends data from the web page into AAM directly. Hence, there is no need to enable any server-side forwarding.

Which Extension Do I Configure in Launch?

In this case, we will configure and use the Adobe Audience Manager extension and will pull variables out of the page's data layer to be sent directly to AAM.

Steps to set up AAM DIL in Launch

1. Add the extension for Adobe Audience Manager


AGS Launch Demo >

SPA Demo with AAM Only

Adobe Audience Manager (DIL)

Adobe Systems

VERSION 1.1.3

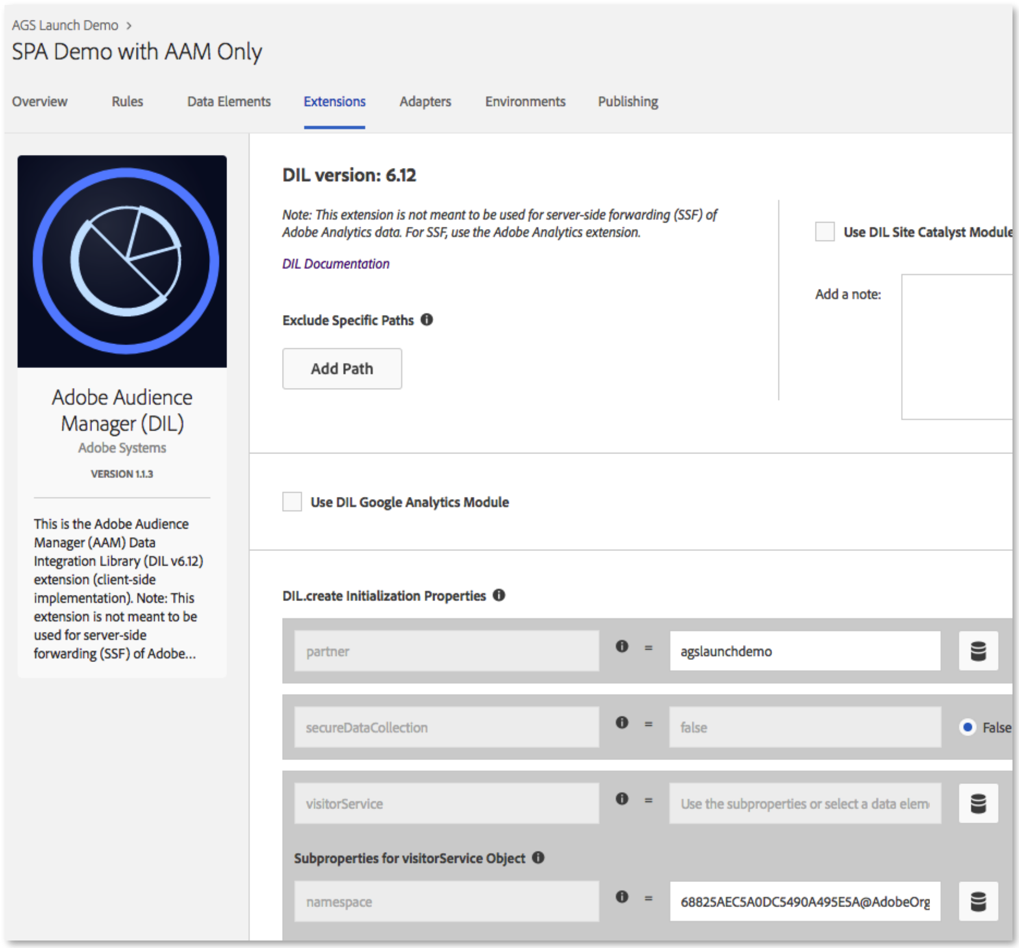
DIL version: 6.12

Note: This extension is not meant to be used for server-side forwarding (SSF) of Adobe Analytics data. For SSF, use the Adobe Analytics extension.

DIL Documentation

Exclude Specific Paths

2. Within the configure extension there is nothing to add to the configuration once the extension is enabled. Verify the partner ID and the OrgID (these will be automatically added, and you should not need to change them).


AGS Launch Demo >

SPA Demo with AAM Only

Overview | Rules | Data Elements | Extensions | Adapters | Environments | Publishing

Adobe Audience Manager (DIL)
Adobe Systems
VERSION 1.1.3

This is the Adobe Audience Manager (AAM) Data Integration Library (DIL v6.12) extension (client-side implementation). Note: This extension is not meant to be used for server-side forwarding (SSF) of Adobe...

DIL version: 6.12

Note: This extension is not meant to be used for server-side forwarding (SSF) of Adobe Analytics data. For SSF, use the Adobe Analytics extension.

DIL Documentation

Exclude Specific Paths

Add Path

Use DIL Site Catalyst Module

Add a note:

Use DIL Google Analytics Module

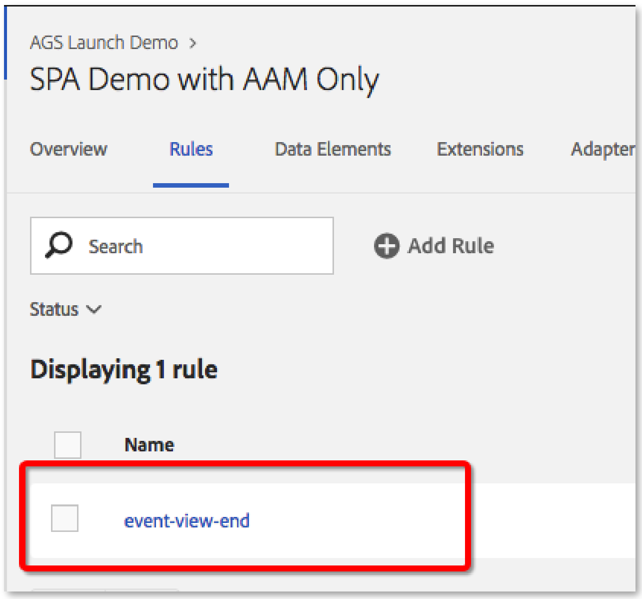
DIL.create Initialization Properties

partner	=	agslaunchdemo
secureDataCollection	=	false ● False
visitorService	=	Use the subproperties or select a data elem

Subproperties for visitorService Object

namespace	=	68825AEC5A0DC5490A495E5A@AdobeOrg

3. Create one Rule, name the rule “event-view-end”


AGS Launch Demo >

SPA Demo with AAM Only

Overview | Rules | Data Elements | Extensions | Adapter

Search

Add Rule

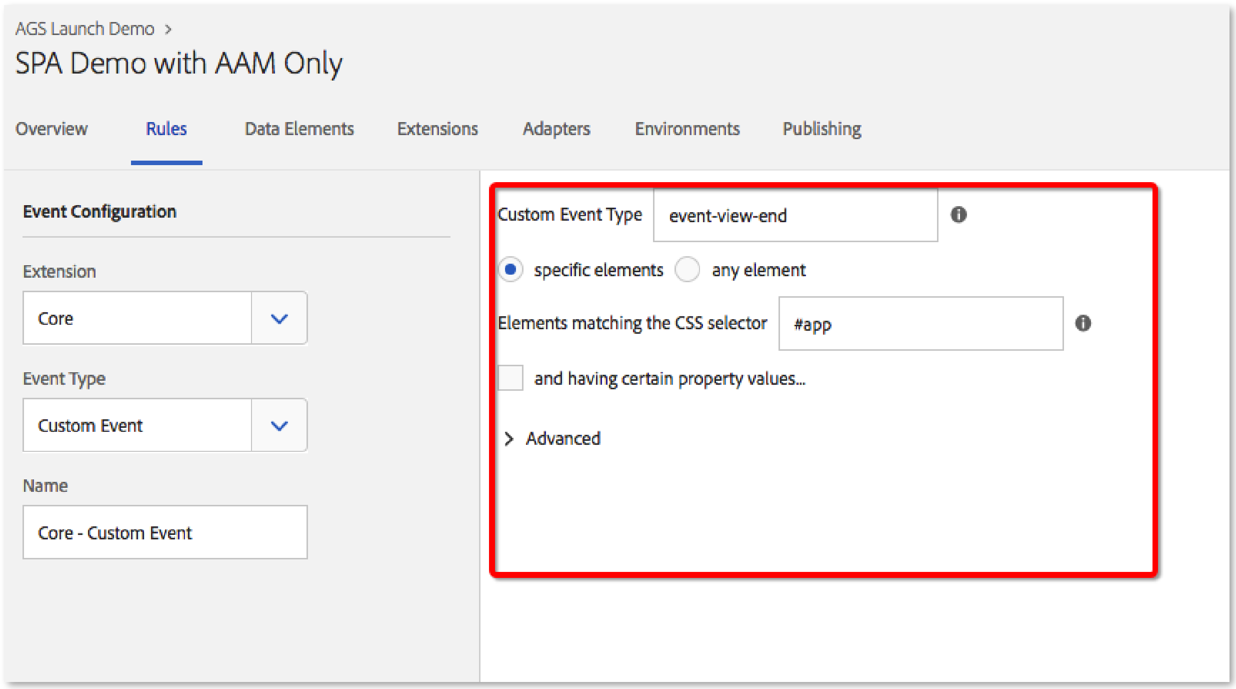
Status

Displaying 1 rule

Name

event-view-end

4. Add and event within Rules and select Extension “Core”, Action Type “Custom Event”, Name “Core – Custom Event”


AGS Launch Demo >

SPA Demo with AAM Only

Event Configuration

Extension: Core

Event Type: Custom Event

Name: Core - Custom Event

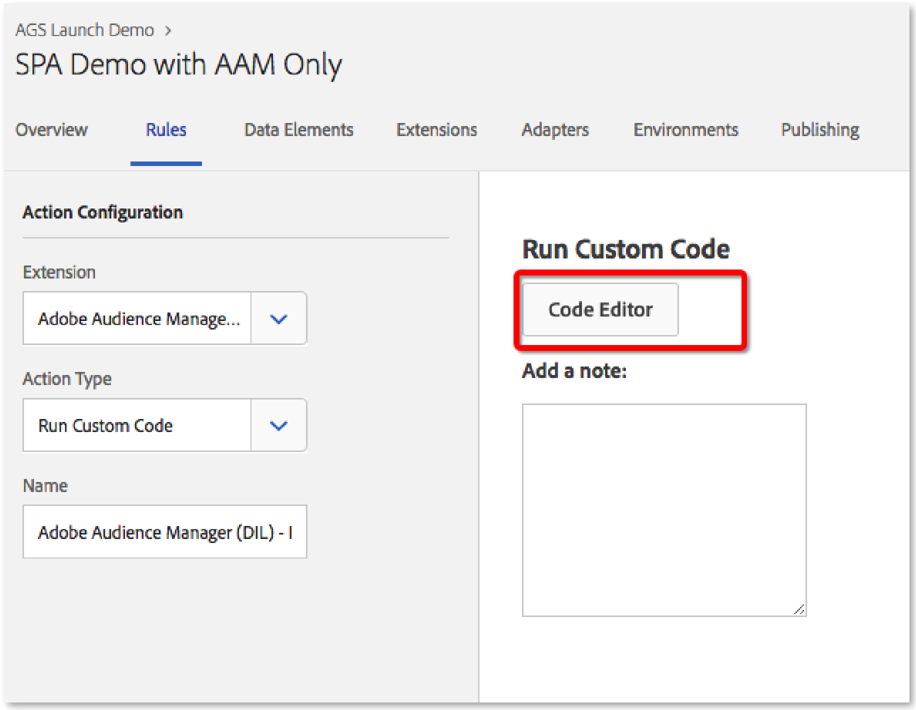
Custom Event Type: event-view-end

● specific elements ○ any element

Elements matching the CSS selector: #app

☐ and having certain property values...

5. Add and Action within the Rules and select Extension “Adobe Audience Manager”, Action Type “Run Custom Code”, Name “Audience Manager (DIL) – Run Custom Code” select “Code Editor”


AGS Launch Demo >

SPA Demo with AAM Only

Action Configuration

Extension: Adobe Audience Manage...

Action Type: Run Custom Code

Name: Adobe Audience Manager (DIL) - I

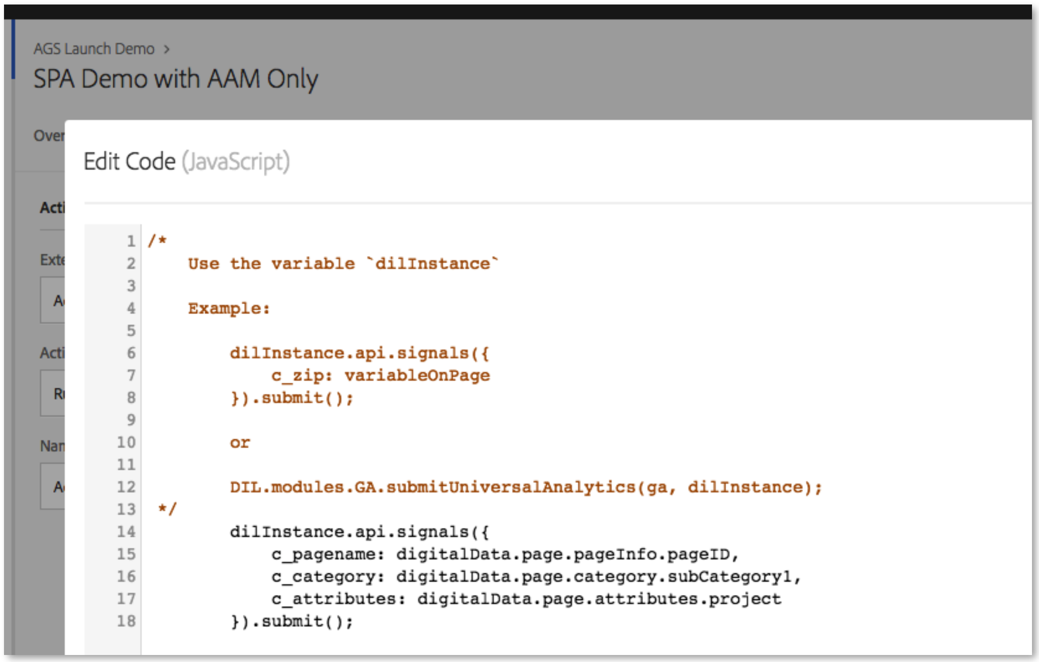
Run Custom Code

Add a note:

6. Paste Code into the code editor that will pull information off your page, presumably from a data layer. Use the example code to know how to set your custom code. In the following example, we are pulling data from a “digitalData” data layer. You can use the format to pull data from your data layer or from other variables on your page.

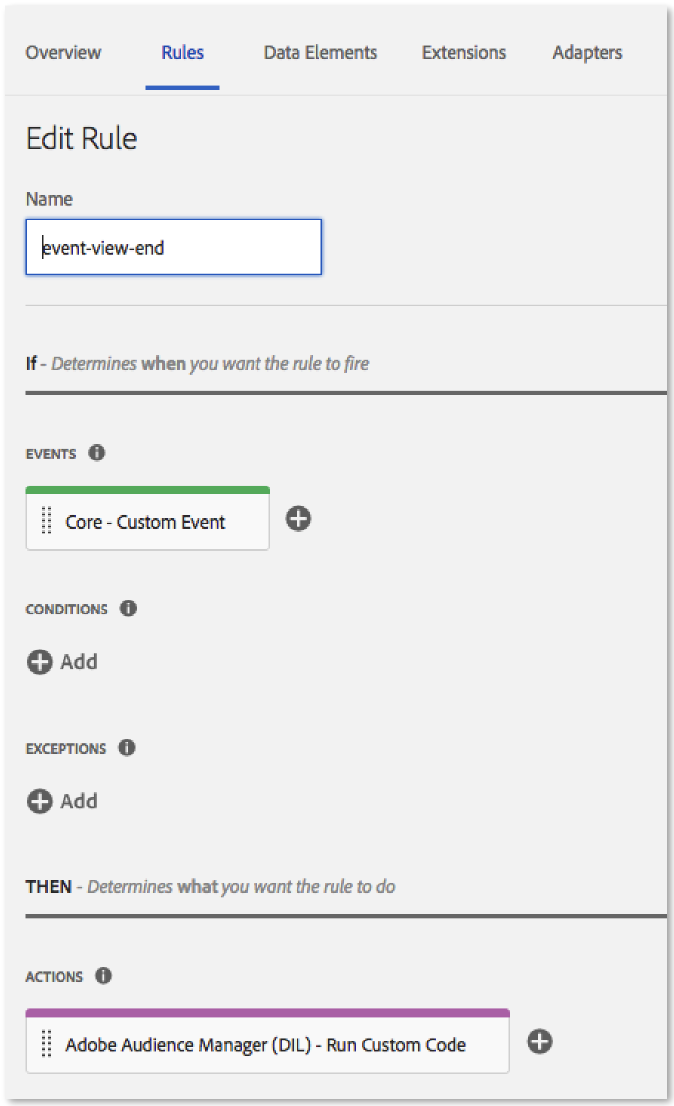
```
/*
    Use the variable `dilInstance`

    Example:

        dilInstance.api.signals({
            c_zip: variableOnPage
        }).submit();

        or

        DIL.modules.GA.submitUniversalAnalytics(ga, dilInstance);
 */
        dilInstance.api.signals({
            c_pagename: digitalData.page.pageInfo.pageID,
            c_category: digitalData.page.category.subCategory1,
            c_attributes: digitalData.page.attributes.project
        }).submit();
```

7. Confirm you code looks like this in the editor:



```
/*
    Use the variable `dilInstance`

    Example:

        dilInstance.api.signals({
            c_zip: variableOnPage
        }).submit();

        or

        DIL.modules.GA.submitUniversalAnalytics(ga, dilInstance);
 */
        dilInstance.api.signals({
            c_pagename: digitalData.page.pageInfo.pageID,
            c_category: digitalData.page.category.subCategory1,
            c_attributes: digitalData.page.attributes.project
        }).submit();
```

8. The completed rule for “Insert Custom Variable will have Events “Core – Library Loaded (Page Top) and Actions “Adobe Audience Manager(DIL) – Run Customs Code”



Edit Rule

Name: event-view-end

If - *Determines* **when** *you want the rule to fire*

EVENTS

Core - Custom Event

CONDITIONS

Add

EXCEPTIONS

Add

THEN - *Determines* **what** *you want the rule to do*

ACTIONS

Adobe Audience Manager (DIL) - Run Custom Code

Build your Library

9. Create a new library in the Development 1 environment
10. Add the changes to the Analytics extension and build the library
11. Open the Demo site
12. Make sure that Launch Command is on and set to load the Development 1 embed code

Publish your implementation of AAM DIL

Approve and Publish the library to Production

Testing Considerations & Steps

You have two options here for testing and validation:

1. If you have followed these steps above on your own site and your own Launch property, you can validate your site
2. If you want to see this use case on our demo site, you can use the Launch Command extension for Chrome to see it implemented

To Validate AAM Only on Your Property

Assumptions:

1. For this step, we will assume that you have implemented as described above, and that you have pushed your library through to production
2. We will also assume that you have place the production embed code on your site

Validation:

1. Open a debugger on your machine (E.g. Charles, or the Chrome debugger, etc)
2. To see the AAM hit, filter the debugger to "demdex"
3. Run/refresh your page
4. In the debugger, you should have a hit that has "event" in it. Select that hit
5. You should be able to see the variables that were pulled from your page in the AAM hit

To See AAM Only on the AAM Only - SPA Demo Site

For this, you should use the Launch Command extension for Chrome

1. Open the Launch Command Extension
2. Type "Adobe SPA Demo - AAM Only" – Production" in the first box

   Paste the following embed code for the environment into the second box
   `<script src="//assets.adobedtm.com/launch-EN94c632ce757541e183fe3ae50af84cd9.min.js"></script>`

3. Check the debugger for the call by searching for "demdex" call (make sure the page is refreshed). You should see the hits described above in the call.
4. Verify that there is an AAMUUID in the response

Integrations

Integration Consideration 1: Audience Sharing

Audience Sharing with the People Core Service requires that you implement the Experience Cloud ID Service and the Audience Manager Module for Analytics, which we have already done. The Experience Cloud ID Service extension for Launch will guarantee that the ID Service always loads in the correct order to set the appropriate MID parameter in the Analytics and Target calls to ensure proper visitor stitching on Adobe's backend. This will enable you to share audiences from Analytics, Audience Manager, and the Audience Library to Target and Analytics.

Note: If you have just enabled the SSF in the interface, as described at the top of this section, it will take up to four hours for it to start working. Be patient.

Integration Consideration 2: Analytics for Target (A4T)

In addition to implementing the Experience Cloud Id Service, A4T requires that Target loads before Analytics. If you fire the 'event-view-start' and 'event-view=end' events as mentioned in this guide. Adobe Launch will load and execute these calls in order. You should always see matching supplemental data ids (SDIDs) in both calls-these ids stitch hits together in the Analytics processing for A4T. Refer the validation section in the guide to know how to validate the SDIDs

Integration Consideration 3: Customer Attributes

In addition to implementing the Experience Cloud Id Service, Customer Attributes requires that you set Customer Ids via the

Id Service. Although we are not doing so in this reference architecture, the Customer Ids need to be set before Target and Analytics load. You need to ensure that you have set the Customer ID action on top of the page before firing the 'event-view-start' event. The Standard Web Reference Architecture Guide demonstrates how to do this and is available from here: Additional Reference Implementation Guides.