Launch by Adobe

Reference Architecture Guide for Basic Web

by Adobe Customer Solutions

Version 1.1/March 27, 2018

Table of Contents

Reference Architecture Overview

Platform Scope

This Reference Architecture Guide contains basic setup best practices for implementing the Adobe Experience Cloud in a website using Launch by Adobe. As a companion to this guide, a Reference Implementation using these best practices can be found at:

http://agslaunchdemo.businesscatalyst.com/basic/

Other Launch Reference Architectures

Additional Reference Implementation Guides will show you how to build upon these basic steps to do things like:

1. Implement more robust features of the Experience Cloud in a standard website
2. Implement the Experience Cloud in a Video player
3. Implement the Experience Cloud in Accelerated Mobile Pages (AMP)
4. Implement the Experience Cloud in Single Page Applications (SPAs) including those using the React and Angular frameworks

Guide Version History:

v1.1 March 27, 2018—updated to include the new server-side forwarding option in the Analytics extension

v1 March 13, 2018—initial version

Goal

The goal is of this document is to serve as an introductory guide to Launch by Adobe so that you can implement a basic instance of the Adobe Experience Cloud and publish your implementation. Along the way, there will be callouts to help you understand the similarities and differences between Launch and DTM, so that you will be better informed if you are migrating from DTM to Launch.

After completing this you will be able to:

- Create a Launch Property with required Adaptors and Environments
- Add the mbed codes to a website
- Create data elements
- Add basic versions of the Adobe Experience Cloud extensions
- Create rules to fire basic requests for the various Adobe Experience Cloud solutions
- Understand the Publishing workflow in Launch and deploy this basic Property to its Production environment

The end result will be a Launch property that implements:

1. Experience Cloud Id service
2. Adobe Analytics with a global beacon and pageName variable
3. Adobe Target with global mbox and pageName parameter
4. Audience Manager via the Audience Manager Module for Analytics (Server-side forwarding)

Assumptions

This guide assumes:

- Your Marketing Cloud Organization has been provisioned for Launch and you have full rights.
- You have a working knowledge of our current tag management solution, DTM. While experience with DTM is not required to benefit from this Guide, there are many feature comparisons to DTM contained herein.

Tools you will need

This Guide will use all of the latest tools for Debugging and Quality Assurance (QA).

Experience Cloud Debugger Chrome Extension

The Experience Cloud Debugger is a Chrome extension with debugging features for all SaaS-based Experience Cloud Solutions. Get the extension here: https://chrome.google.com/webstore/detail/adobe-experience-cloud-de/ocdmogmohccmeicdhlhhgepeaijenapj

Launch Command Chrome extension

This is a non-Adobe extension to switch embed codes. Get the extension here:

https://chrome.google.com/webstore/detail/launch-command/nkjhamgjeocefocmpbcjfmohkjgildki

Later in the Guide we will add Embed Codes to the extension.

Browser Developer Tools

The tools above are both Chrome extensions. If you prefer to use a different browser, make sure you have Developer Tools installed (some browsers require a separate download).

The Reference Site

Bookmark the reference site so you can refer to it later. You can create your own property in your own Launch account and use Launch Command to have this URL load your embed codes. Alternatively, you can copy the source code, replace the embed codes with your own and upload it to your own server: http://agslaunchdemo.businesscatalyst.com/basic/

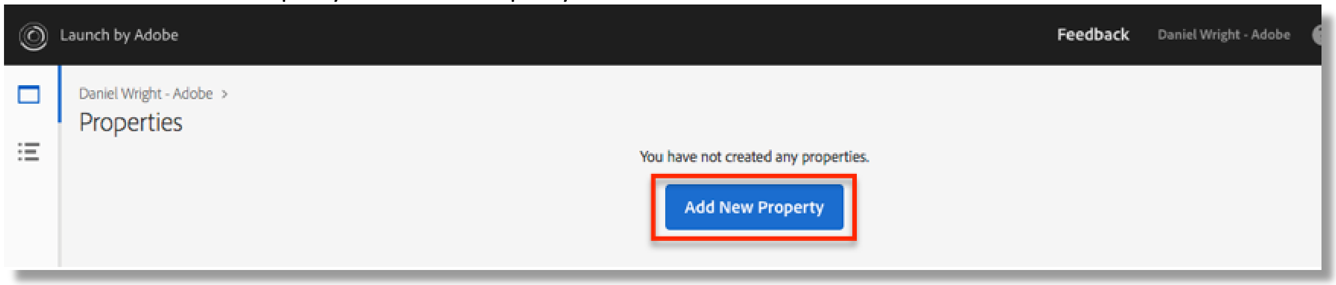
General Known Platform Limitations

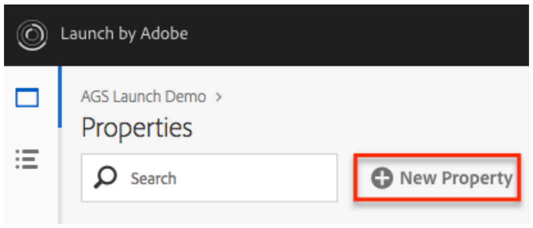
None

General Launch Configuration & Settings

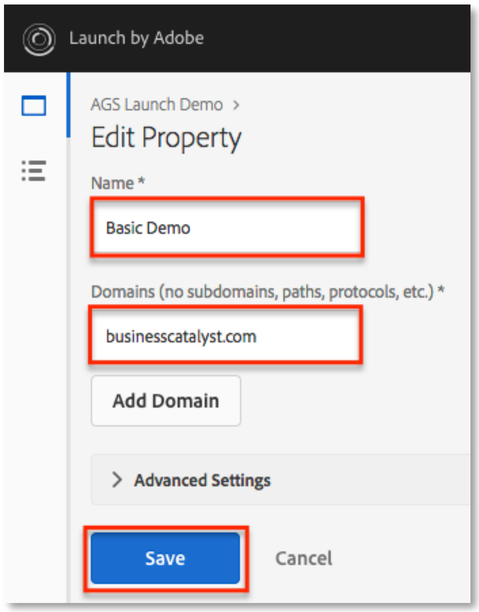
Configuration 1: Create a Property

1. Click the “Add New Property” or “New Property” button:





2. Name your property.
3. Enter the domain or domains where you will be implementing the property.
4. Don’t change any of the Advanced Settings.
5. Click “Save”



Name *: Basic Demo

Domains (no subdomains, paths, protocols, etc.) *: businesscatalyst.com

Your new property should display on Property list page. Click on the name of your property to open the Overview screen and then click on the Extensions tab. Note that the “Core Extension” was automatically added when you created the Property. The “Core” extension provides many of the capabilities that you are used to from DTM, such as the ability to fire actions at specific events, add custom code snippets and so forth.

What is similar to DTM?

Launch and DTM properties both have these options:

- Name
- Additional domains
- Return empty string for undefined Data Elements
- Tracking Cookie Name, with “sat_track” as the default
- Anchor Delay, with 100ms as the default
- Properties can be deleted
- Properties on the list page can be filtered by Property Name

What is different from DTM?

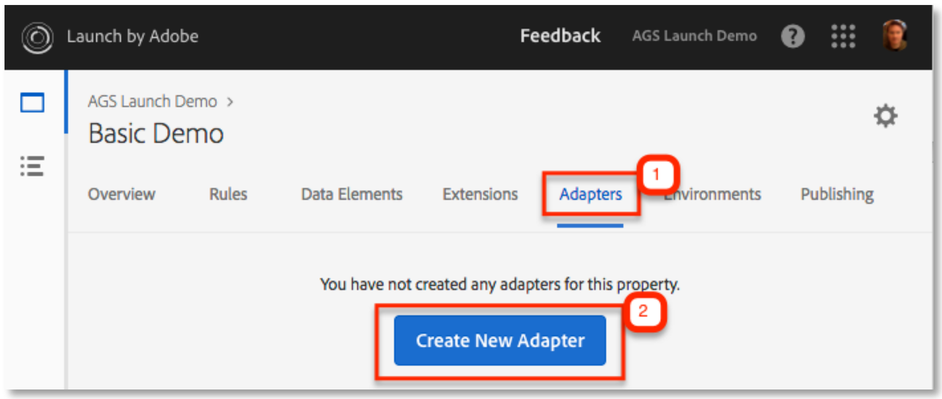
1. “URL” has been changed to “domain” and only accepts a domain name
2. Launch properties do not have these options:
    a. Allow Multi-Rule Approvals (similar capabilities are available in Launch’s publishing process)
    b. Enable Selective Publish (similar capabilities are available in Launch’s publishing process)
    c. Tag Timeout
    d. Properties in Launch cannot currently be disabled (ETA is TBD)
    e. Properties on the list page cannot currently be filtered by domain name
    f. Properties on the list page cannot currently be filtered by the property id (use the Experience Cloud Debugger to look up a property name)

Configuration 2: Create an Adapter

Adapter is a new term in Launch, although the concept existed in DTM. An “Adapter” refers to the destination(s) where you want to deploy your Launch libraries—on Adobe’s Akamai instance or on your own web servers. To make environments more flexible in Launch, this level of control was broken out as its own setting. For example, you could choose to host to your Production libraries on your own domain (self-hosting allows for security checks, automated testing, version control, TTL, caching, etc), but use Adobe’s Akamai instance for development environments since it doesn’t require any IT work. The Akamai Adapter can also be used to create a zip archive of your library, which you can then retrieve via the Launch API.

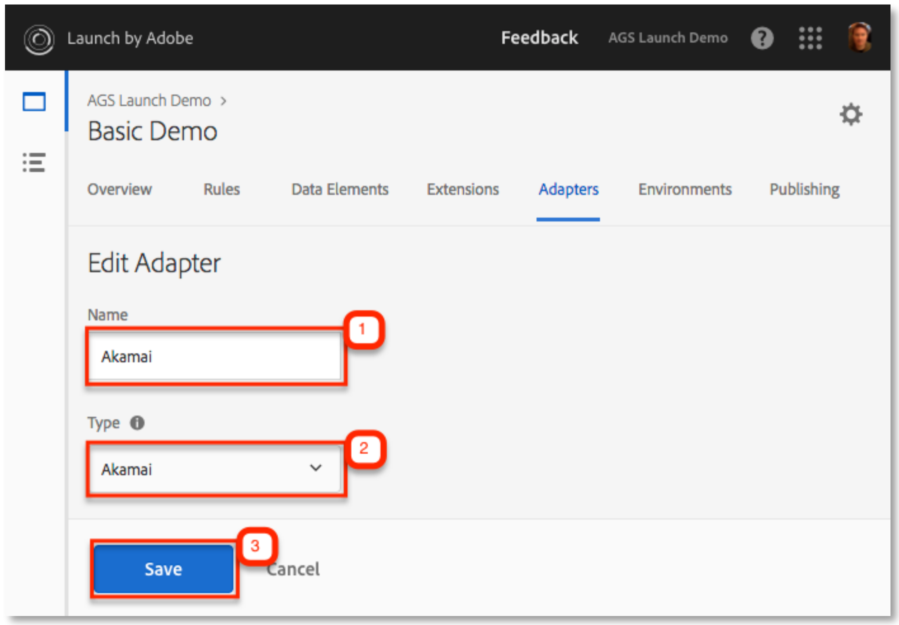
This reference architecture uses the Akamai Adapter.

Click “Adapters” in the Property navigation, then click “Create New Adapter:”



You have not created any adapters for this property.

Name the Adapter—"Akamai" is a good name—using the "Akamai" type from the dropdown and click "Save":



Name: Akamai

Type: Akamai

What is similar to DTM?

- Options to host your libraries through Akamai or on your own server

What is different from DTM?

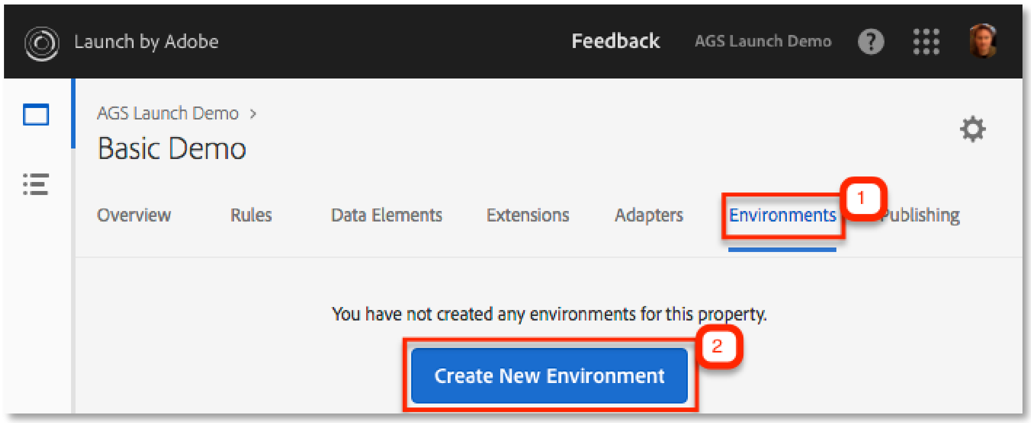
- No insecure FTP options
- SFTP option uses an encrypted private key instead of a password
- SFTP has no separate HTTP/HTTPS path sections
- SFTP has no "Enable relative hostnames for staging and production library hosting" checkbox

Configuration 3: Create Environments

In DTM, there were only two environments—Staging and Production. Launch introduces a Development environment and allows you to have *multiple* Development environments.

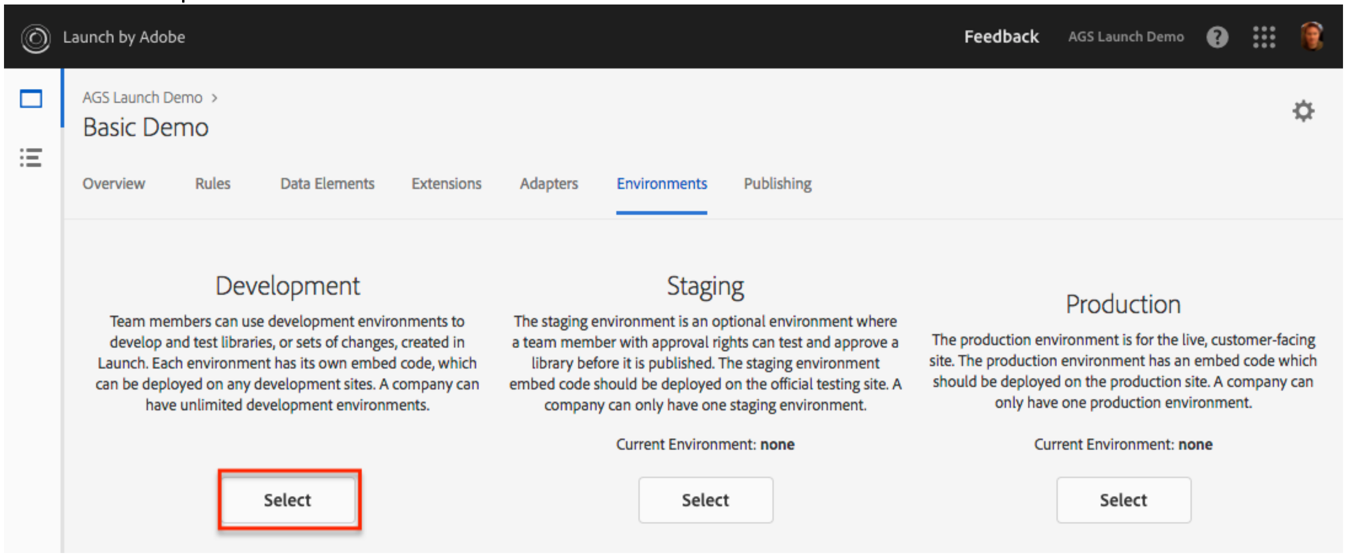
For example, the Analytics implementation team can work in their own development environment, while the Target implementation team can work in their own development environment. They can work independently and merge their code later in the publishing process.

1. Click “Environments” in the top navigation and then click “Create New Environment”



You have not created any environments for this property.

2. Under “Development” click “Select”:



Development

Team members can use development environments to develop and test libraries, or sets of changes, created in Launch. Each environment has its own embed code, which can be deployed on any development sites. A company can have unlimited development environments.

Staging

The staging environment is an optional environment where a team member with approval rights can test and approve a library before it is published. The staging environment embed code should be deployed on the official testing site. A company can only have one staging environment.

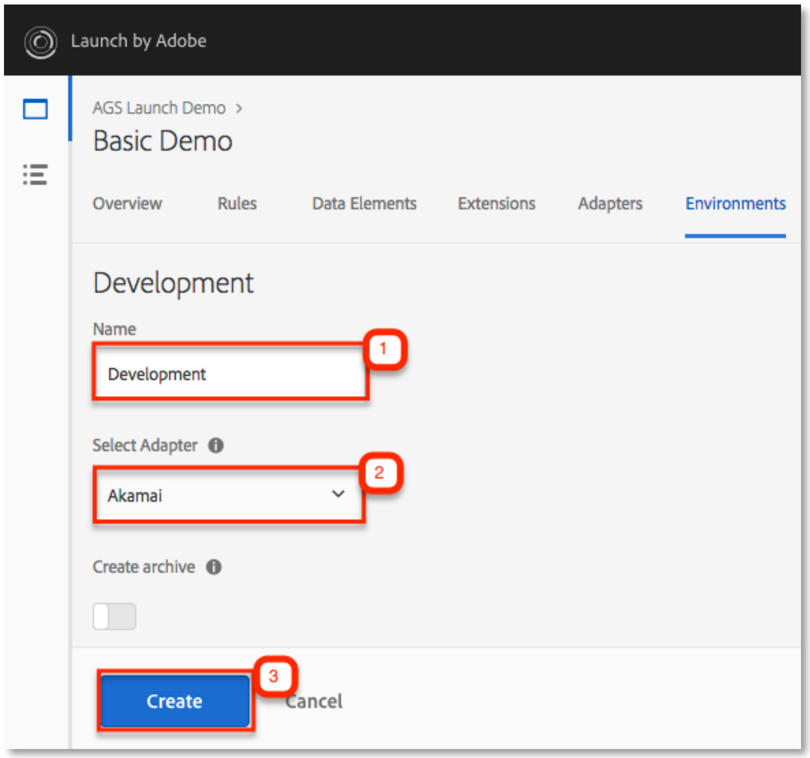
Current Environment: **none**

Production

The production environment is for the live, customer-facing site. The production environment has an embed code which should be deployed on the production site. A company can only have one production environment.

Current Environment: **none**

3. Name your environment select your "Akamai" adaptor, and then click "Create":



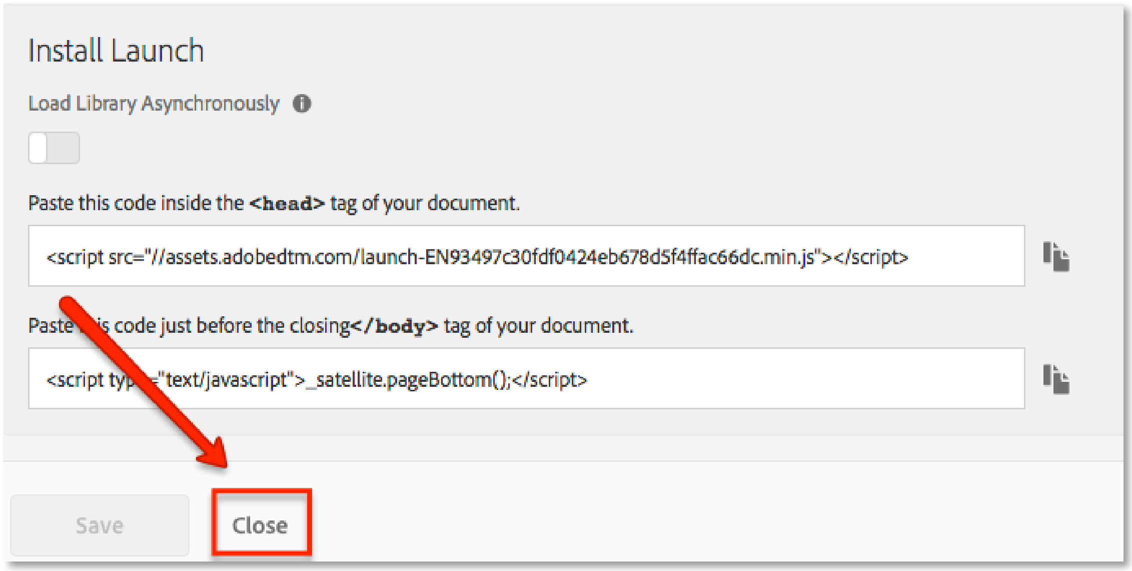
Name: Development

Select Adapter: Akamai

The Create archive option is a feature used for self-hosting, which we will not be using in this reference architecture.

Note that as soon as you select the Adapter and hit the create button, the embed code for that environment is created and displayed. Each environment will have its own unique embed code. An asynchronous loading option was recently added to the Launch UI, but is not being used in this reference architecture.

4. Click "Close"



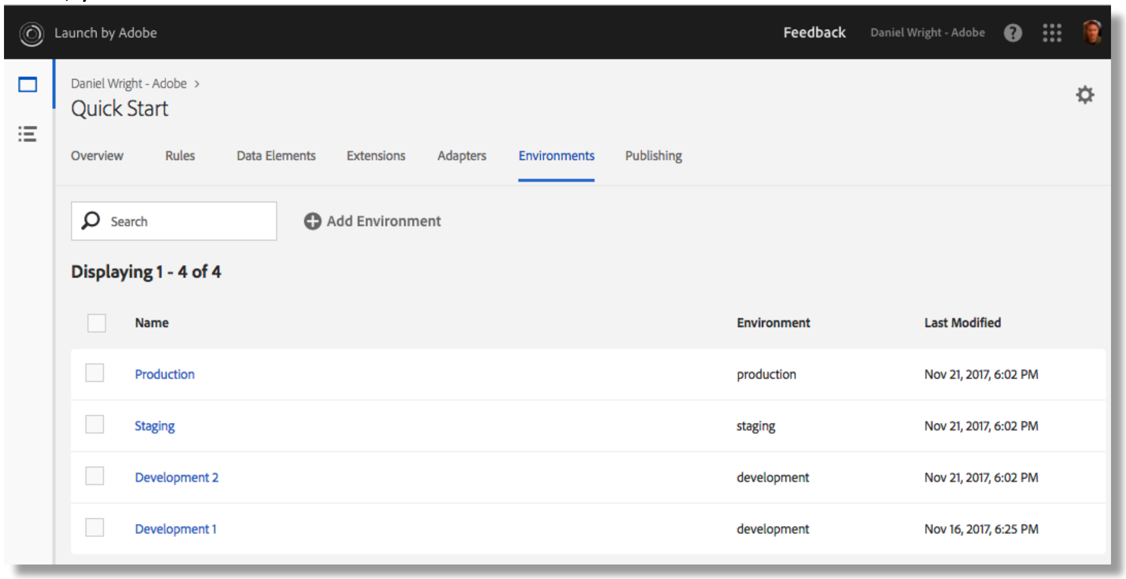
Paste this code inside the **<head>** tag of your document.

```
<script src="//assets.adobedtm.com/launch-EN93497c30fdf0424eb678d5f4ffac66dc.min.js"></script>
```

Paste this code just before the closing **</body>** tag of your document.

```
<script type="text/javascript">_satellite.pageBottom();</script>
```

5. Add an additional development environment, a staging environment and a production environment. Once you are done, you should have four Environments:



Displaying 1 - 4 of 4

Name	Environment	Last Modified
Production	production	Nov 21, 2017, 6:02 PM
Staging	staging	Nov 21, 2017, 6:02 PM
Development 2	development	Nov 21, 2017, 6:02 PM
Development 1	development	Nov 16, 2017, 6:25 PM

What is similar to DTM?

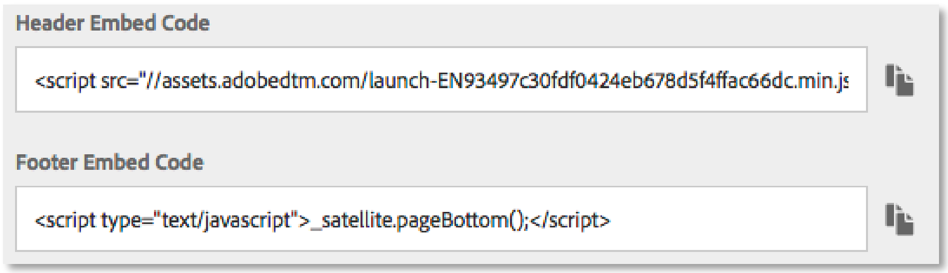
- Footer embed code for synchronous deployments is exactly the same
- URLs will return 404 errors until the first change has been made and built to the environment

What is different from DTM?

- DTM Switch can't be used to switch you between Staging and Prod environments—the new Launch Command tool should be used for this purpose.
- DTM Embed URLs are slightly different from Launch Embed URLs. Migration tools are being built so existing DTM embed codes can be transferred to Launch properties, so that customers will eventually be able to migrate without having to update their page code.
- An Asynchronous embed code option is available

Configuration 3: Implement the Launch Embed Codes

For each environment using a synchronous deployment, there are two embed codes, the Header Embed Code and the Footer Embed Code which need to be implemented:



Header Embed Code

```
<script src="//assets.adobedtm.com/launch-EN93497c30fdf0424eb678d5f4ffac66dc.min.js
```

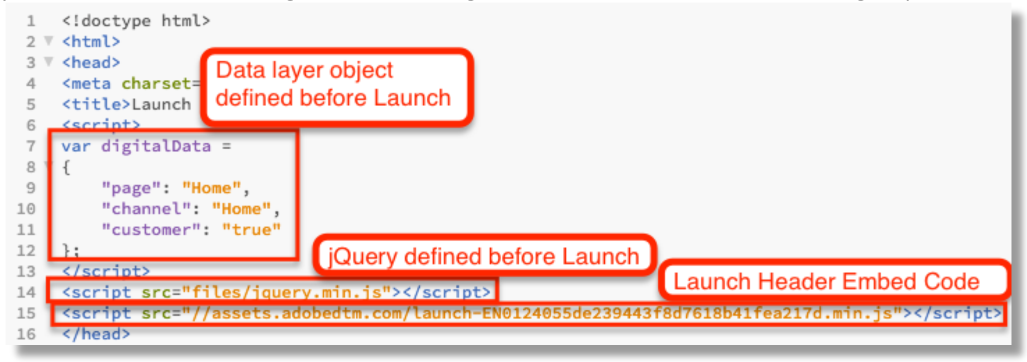
Footer Embed Code

```
<script type="text/javascript">_satellite.pageBottom();</script>
```

Implement the Header Embed Code

The Header Embed Code should be implemented in the <head> element of all HTML pages. Many customers have one or several template files which control the <head> globally across the site.

The best practice is to define it before the Header Embed Code and populate it with all of the attributes needed to populate variables in Analytics, Target, and other marketing solutions. If you are unable to define the data layer before the embed code, you will be limited with what you can do in Target, Customer Attributes, and Analytics. Additionally, if you use a JavaScript helper library like jQuery—and already implement it in the <head>--load it before your Embed Code if you would like to leverage it when using custom code in Launch and Target:|


Data layer object defined before Launch
jQuery defined before Launch
Launch Header Embed Code

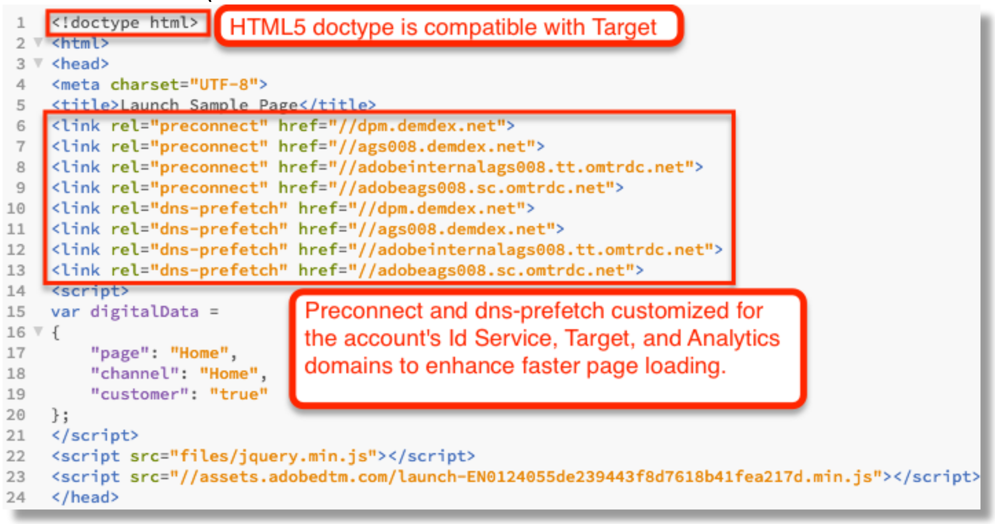
```
<!doctype html>
<html>
<head>
<meta charset=
<title>Launch
<script>
var digitalData =
{
    "page": "Home",
    "channel": "Home",
    "customer": "true"
};
</script>
<script src="files/jquery.min.js"></script>
<script src="//assets.adobedtm.com/launch-EN0124055de239443f8d7618b41fea217d.min.js"></script>
</head>
```

Implement the Embed Code of your production environment on your page.

If you are updating your <head> to implement Launch, consider these other best practices:

1. Use the HTML5 doctype to support your Target implementation
2. As an optional best practice use preconnect and dns-prefetch to improve the page load time

Here is a summary what this looks like, with preconnect and dns-prefetching subdomains customized to the Adobe AGS008 account (note that this does not have to be done for the assets.adobedtm.com domain):

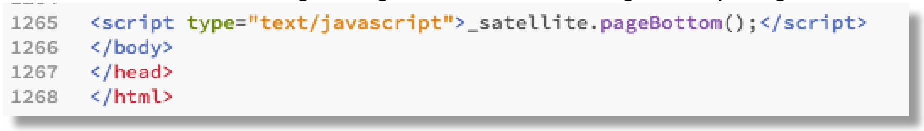

HTML5 doctype is compatible with Target
Preconnect and dns-prefetch customized for the account's Id Service, Target, and Analytics domains to enhance faster page loading.

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Launch Sample Page</title>
<link rel="preconnect" href="//dpm.demdex.net">
<link rel="preconnect" href="//ags008.demdex.net">
<link rel="preconnect" href="//adobeinternalags008.tt.omtrdc.net">
<link rel="preconnect" href="//adobeags008.sc.omtrdc.net">
<link rel="dns-prefetch" href="//dpm.demdex.net">
<link rel="dns-prefetch" href="//ags008.demdex.net">
<link rel="dns-prefetch" href="//adobeinternalags008.tt.omtrdc.net">
<link rel="dns-prefetch" href="//adobeags008.sc.omtrdc.net">
<script>
var digitalData =
{
    "page": "Home",
    "channel": "Home",
    "customer": "true"
};
</script>
<script src="files/jquery.min.js"></script>
<script src="//assets.adobedtm.com/launch-EN0124055de239443f8d7618b41fea217d.min.js"></script>
</head>
```

See also: https://w3c.github.io/resource-hints/

Implement the Footer Embed Code

The Footer Embed Code goes right before the closing </body> tag:

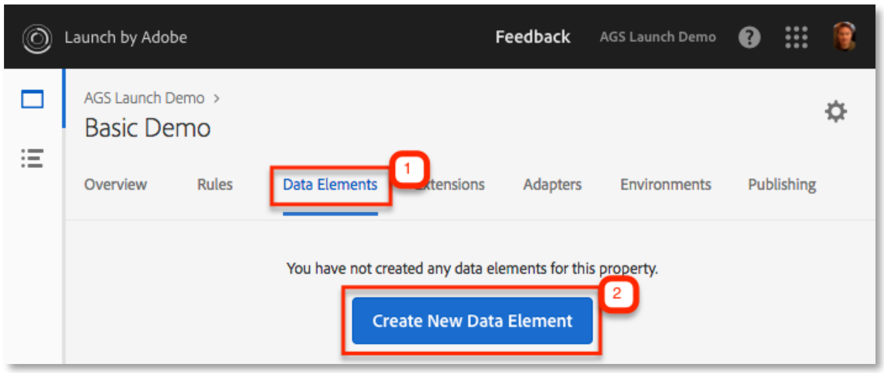


```
<script type="text/javascript">_satellite.pageBottom();</script>
</body>
</head>
</html>
```

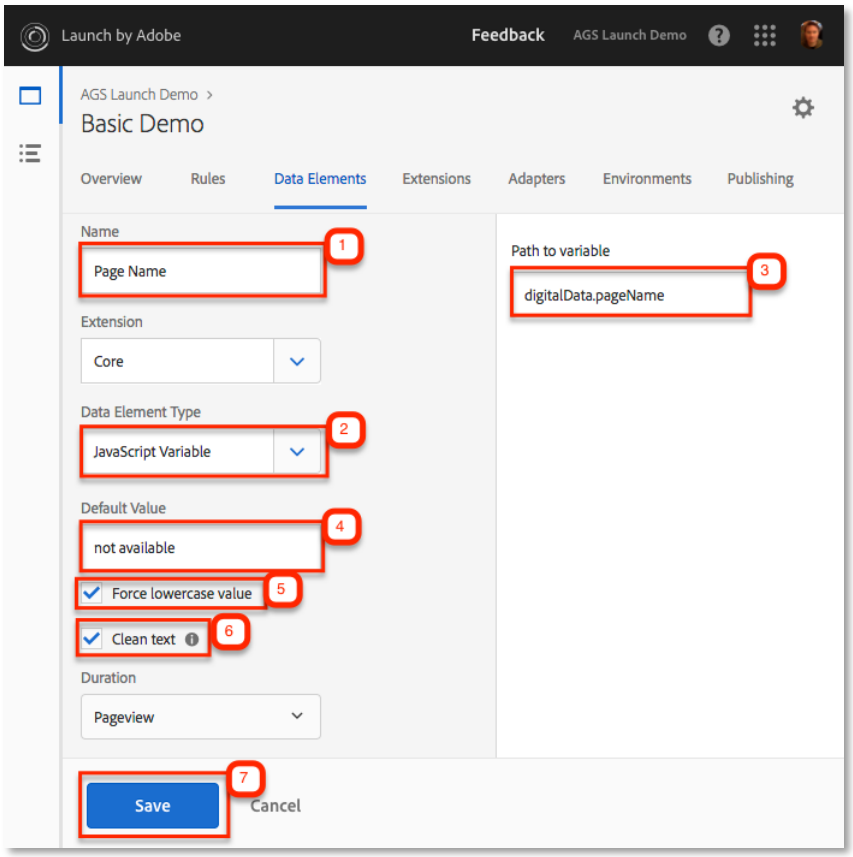
Configuration 4: Data Element for Page Name

Data elements are Launch's version of a data layer. They can store values from your own data layer object, cookies, local storage objects, query string parameters, page elements, meta tags, etc. In this exercise, we will create a data element for Page Name, which we will use later for our simple Target and Analytics implementations.

1. Click "Data Elements" in the top navigation and then click the "Create New Data Element" button:



2. Name the data element "Page Name"
3. Use the Core Extension
4. Use the JavaScript Variable Data Element type to read the property "digitalData.pageName"
5. Use "not available" as the Default Value
6. Use the "Force lowercase value" and "Clean text" options" to standardize the case and remove extraneous spaces
7. Use the "Pageview" duration (since this value will typically be different on every page)
8. Save the data element



Name: Page Name
Extension: Core
Data Element Type: JavaScript Variable
Default Value: not available
✓ Force lowercase value
✓ Clean text
Duration: Pageview
Path to variable: digitalData.pageName

What is similar to DTM?

- All DTM data element types are available
- The main options—Name, Default Value, Force lowercase, cleanText, and duration—are the same
- Sub-options of specific element types are also the same

What is different from DTM?

- Data Elements have their own top navigation element and are no longer under "Rules"
- New data element types—Local Storage, Session Storage, Page Info, and Random Number
- Names of Data Element types are different. Here is a full list of the element types in Launch with their corresponding DTM names in parentheses ():
  - Cookie (Cookie)
  - Custom Code (Custom Script)
  - DOM Attribute (CSS Selector)
  - Query String Parameter (URL Parameter)
  - JavaScript Variable (JS Object)
- The DOM Selector Tool used to select CSS paths for DOM Attributes was discontinued
- Data element capabilities *can be extended with Extensions.* For example, the ContextHub extension allows you to add data elements using features of the extension.

Experience Cloud Id Service Configuration & Tagging

Overview

This exercise will guide your through all of the steps required to implement the Experience Cloud ID Service, which will set a common visitor id across all solutions to power integrations.

The Experience Cloud ID Service extension consists of two main parts:

1. The extension configuration where the settings of VisitorAPI.js are managed
2. A rule action-type called "Set Customer IDs" which can send multiple customer ids to Adobe solutions for integrations with Customer Relationship Management (CRM) systems

Extensions in Launch

Since this is the first extension you will be adding, a few concepts should be explained. "Tools" in DTM have been replaced by "Extensions". While on the surface they will feel very similar, there are some subtle, yet extremely powerful differences to be aware of.

Launch is an open and extendable platform. Whereas the Activation team built all of the Adobe and 3rd-party Tools for DTM, Launch allows the Adobe solution teams and 3rd parties to *build their own extensions*. You can even create your own extensions and share them to the extension marketplace.

Think of Launch as the OS and extensions as the App Store. Extensions on the Launch platform allow Launch to evolve with the digital marketing landscape without Adobe bottlenecks. DTM had eight Tools. Launch is brand new yet already has twenty-two extensions! We expect dozens if not hundreds more to be created over the coming years. No other tag management vendor has similar capabilities.

Keep in mind that 3rd party extensions will not be vetted by Adobe. It is up to the customer to thoroughly vet and accept responsibility for any 3rd party extensions used in a Launch implementation. At present, the only way to vet an extension is to add it to a development library and review it.

Extensions can extend rule actions, rule events, and even data elements—there are many ways in which extensions can modify how you do things up Launch.

The Core extension is preinstalled in every new property and provides all of the rule actions, event types, and data element types you are familiar with from DTM.

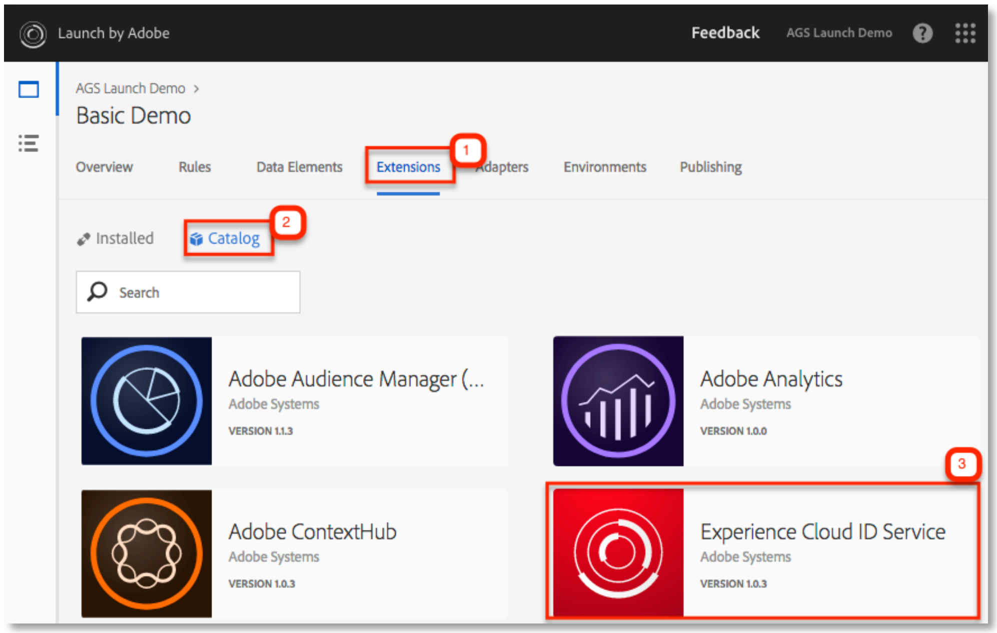
Extensions are versioned and each Adobe extension is attached to a specific version of the solution library. In many instances, a migration to Launch will require an updating of solution libraries. At the time of writing, these are the current versions:

1. Experience Cloud ID Service extension v3.1.1 comes with VisitorAPI.js v3.1.0
2. Target extension v0.4.2 comes with at.js v1.2.3
3. Analytics extension v1.2.0 comes with AppMeasurement.js v2.8.0
4. Audience Manager extension v1.1.3 comes with DIL v6.12

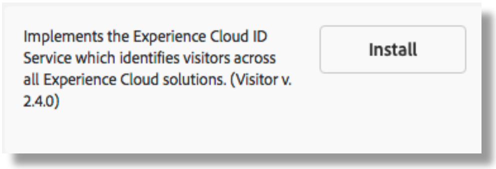
Another key difference between Launch Extensions and DTM Tools is that ***Launch Extensions need to be used with Rules in order to fire <u>most</u> marketing pixels***. For example, in order to fire an Analytics page view call or target-global-mbox, you must use a Rule to instruct Launch to do so. This is an important difference that you will need to keep in mind, and will make more sense as we continue setting up the property.

Configuration 1: Add the Experience Cloud Id Service Extension

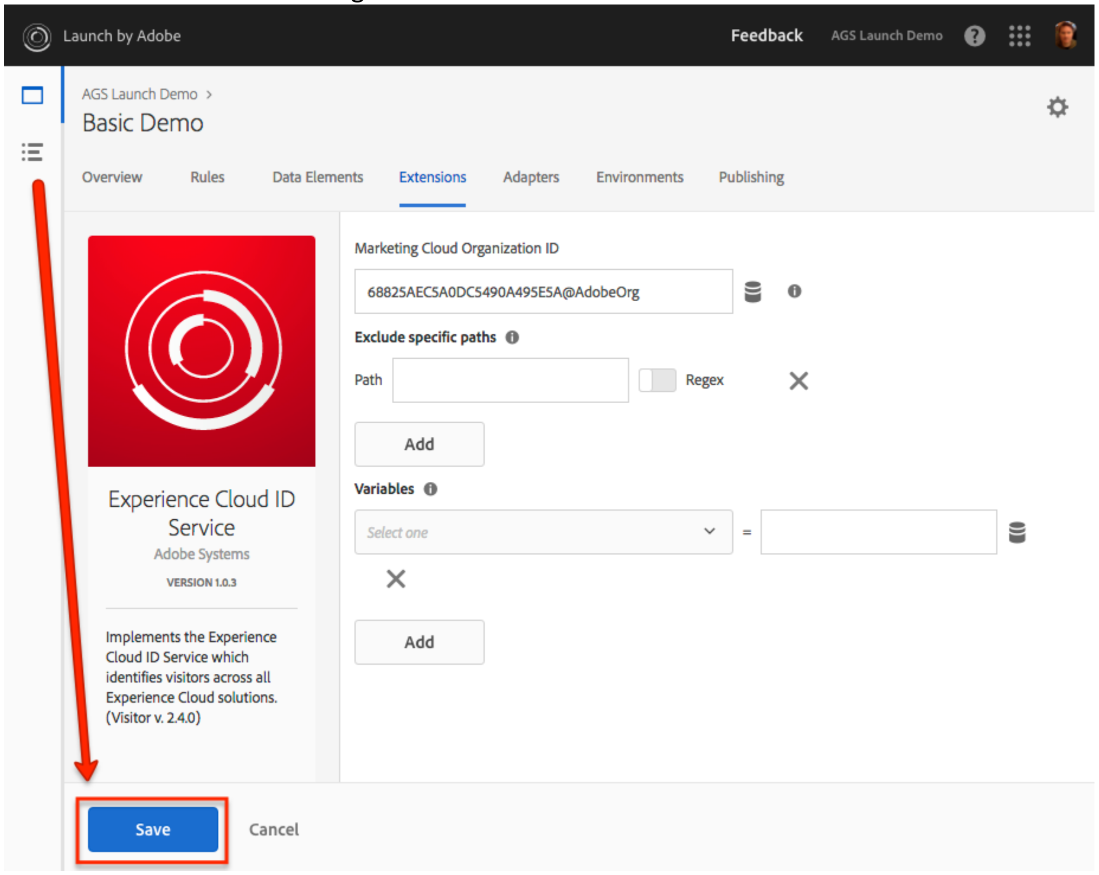
1. Go to the Extensions Catalog Screen and mouse over the Experience Cloud ID Service extension



2. Mouse over the extension, notice that the VisitorAPI.js version it uses is exposed (currently v2.4.0), and click "Install"


Implements the Experience Cloud ID Service which identifies visitors across all Experience Cloud solutions. (Visitor v. 2.4.0)

3. Leave all of the default settings and click “Save”



Marketing Cloud Organization ID: 68825AEC5A0DC5490A495E5A@AdobeOrg

Exclude specific paths

Path | Regex

Variables

Implements the Experience Cloud ID Service which identifies visitors across all Experience Cloud solutions. (Visitor v. 2.4.0)

What is similar to DTM?

- Marketing Cloud Organization ID is automatically detected
- The Marketing Cloud Organization ID can be overwritten or loaded from a data element
- The Visitor ID is retrieved automatically—you don’t need a rule and action in order to request a Visitor ID
- ID Service configurations can be set on the configuration screen

What is different from DTM?

- You can exclude the ID Service from loading on specific URL paths.
- ID Service configurations are available as predefined dropdown options
- There is no freeform option for ID Service configurations
- Customer IDs are not set in the extension configuration—they are now set in rules
- Analytics and Marketing Cloud tracking server configurations are in the Variables dropdown (these are not needed if you have already migrated your visitors to the Id Service)
- No option to “Automatically request Visitor ID,” which is no longer necessary
- You must use the VisitorAPI.js version that comes with the extension version

Adobe Target Configuration & Tagging

Overview

In this exercise, we will implement Adobe Target with a global mbox including a custom parameter.

The Adobe Target extension supports client-side implementations using Target's library for the modern web, at.js—the library of choice for the majority of Target customers. Customers still using Target's older library, mbox.js, should upgrade to at.js in order to use Launch.

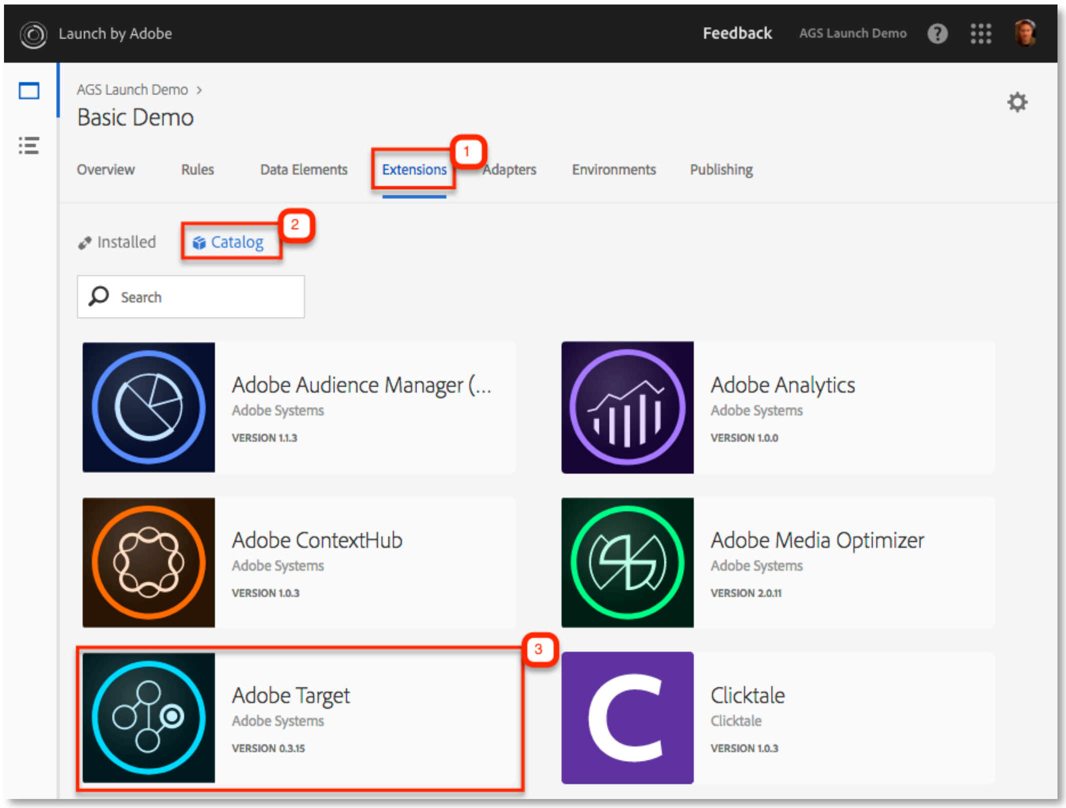
The Target extension consists of two main parts:

1. The extension configuration, which manages some of the core at.js library settings
2. Rule actions to do the following:
   a. Load Target—loads the at.js library
   b. Add Params to All Mboxes
   c. Add Params to Global Mbox
   d. Fire Global Mbox

Some async options, such as "Load Target Async" and "Fire Global Mbox" async are currently available in the UI, but are not used in this Reference Architecture.

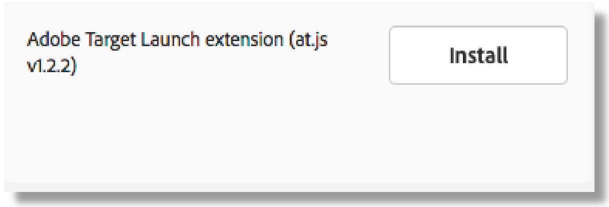
Configuration 1: Add the Target Extension

1. Go to Extensions->Catalog and position your mouse over the Adobe Target extension

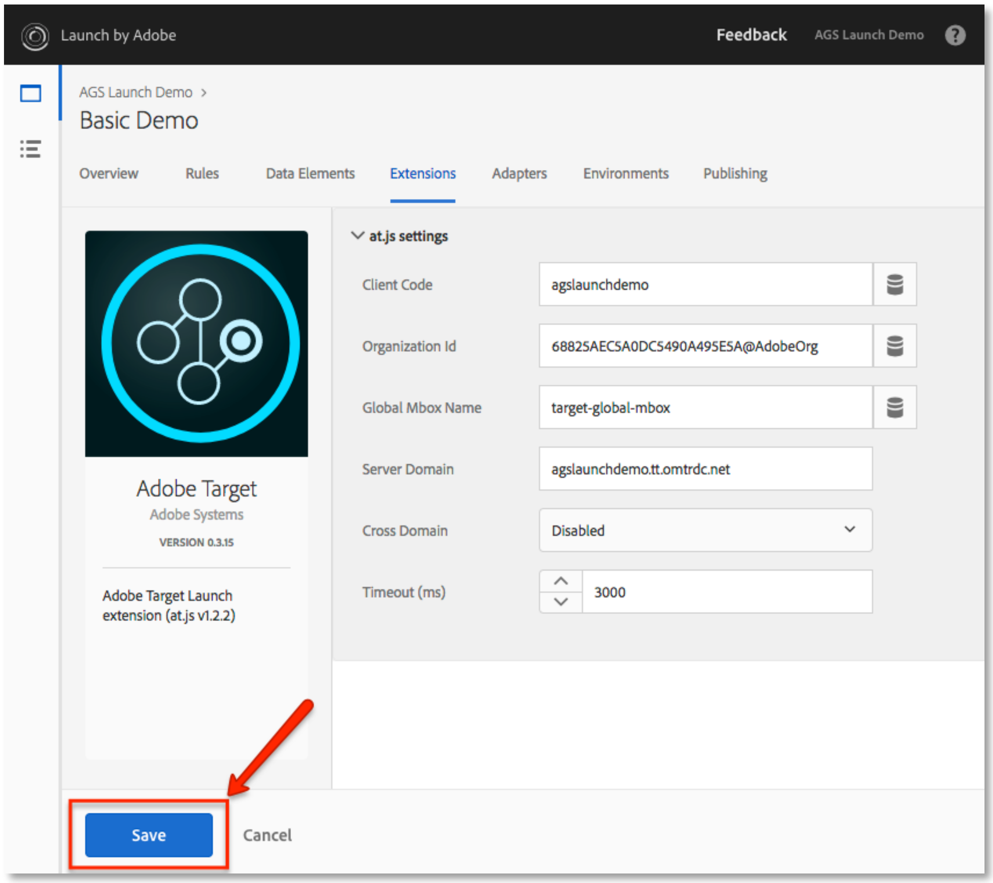


2. Note that this will expose the at.js version which the extension currently uses

3. Click "Install" to add the extension


Adobe Target Launch extension (at.js v1.2.2)

4. Note that when you add the extension, it will import many of your at.js settings from the Target UI, as pictured below, although the Timeout will always be 3000ms. Leave the default settings.
5. Click "Save"



at.js settings

Client Code	agslaunchdemo
Organization Id	68825AECSA0DC5490A495E5A@AdobeOrg
Global Mbox Name	target-global-mbox
Server Domain	agslaunchdemo.tt.omtrdc.net
Cross Domain	Disabled
Timeout (ms)	3000

Configuration 2: Add a Rule to fire the global mbox

Rules Overview

Since this is this is the first rule we are setting up, a few core concepts should be explained. ***In Launch, rules are required in order to fire <u>most</u> marketing pixels***. For example, in order to fire the target-global-mbox, we must use a Rule to instruct Launch to do so.

The Rule builder has been dramatically redesigned and rebuilt in Launch. Some of the main changes are:

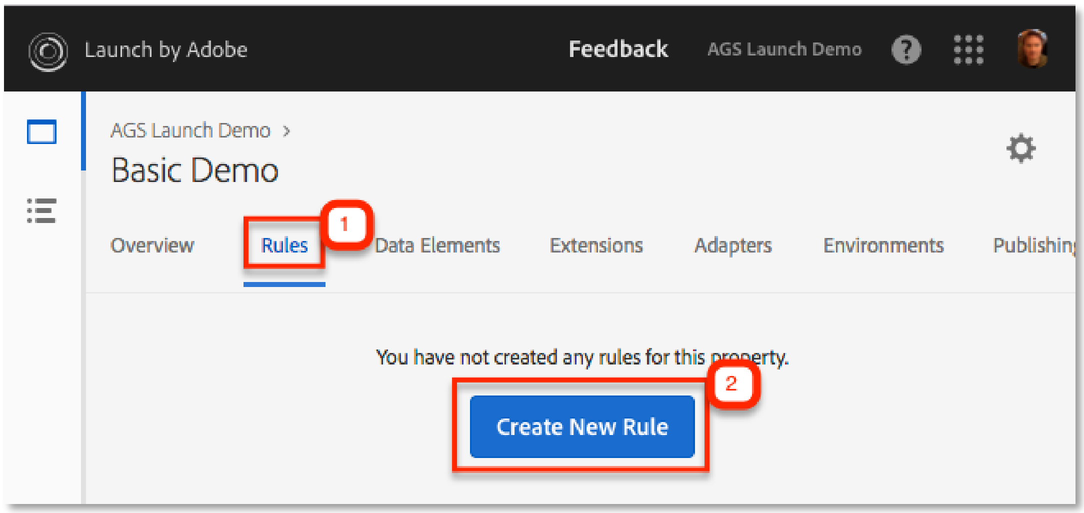
- There is just one Rule builder. Things like "Page Bottom", "Click", and "Direct Call" are all just event-types in the Rule builder. This makes it much easier to update a rule, should you need to change the trigger from, say, a DOM Ready event to a custom event.
- There is a new "Custom Code" event-type

- Extensions can add new event types to the Rule builder. For example, the Target extension could eventually add built-in support for its at.js custom events, so custom code wouldn't be needed to use this feature.
- Extensions can add new actions to the Rule builder, reducing issues by deprecating reliance on custom code.
- Rules are required to fire requests associated with most marketing tools. This will require a mindset adjustment, especially for things like setting Customer IDs, firing Analytics beacons, and firing the global mbox.

Add the global mbox rule

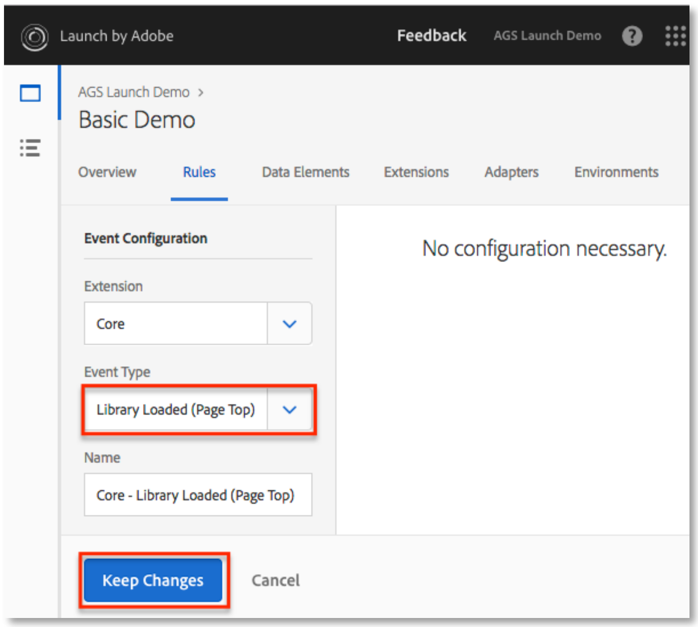
Target needs to load at the top of the page in order to present different visitor experiences without flicker. In this section, we will load the Target library—at.js—as well as fire the target-global-mbox with a parameter containing the Page Name data element.

1. Go to the Rules in the top navigation and then click "Create New Rule"



You have not created any rules for this property.

2. Name the rule, e.g. "All Pages – Top"
3. Under "Events," click "Add." The event is what triggers the rule to fire.
4. Use the "Library Loaded (Page Top)" event from the Core Extension and click the "Keep Changes" button. Note that you could enter a different name besides the auto-generated one ("Core – Library Loaded (Page Top)") , if you prefer.



Event Configuration

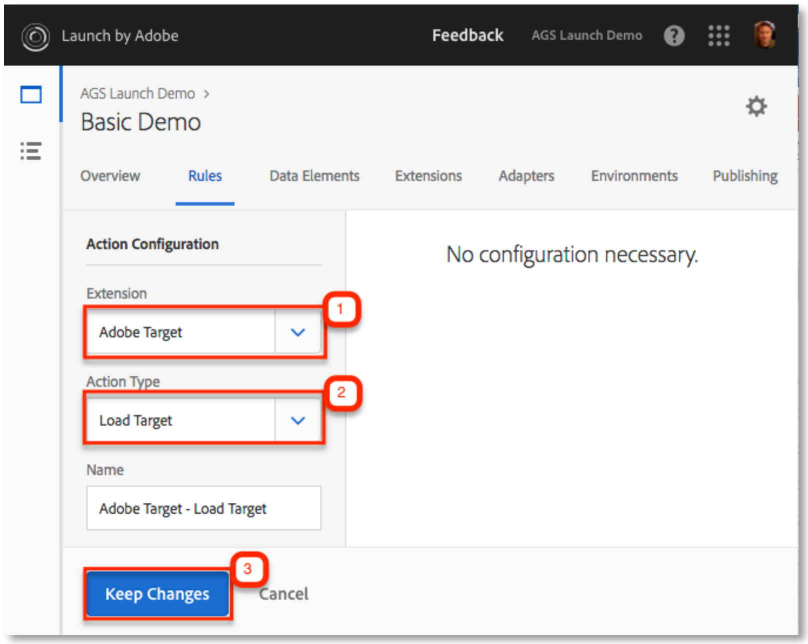
Extension: Core

Event Type: Library Loaded (Page Top)

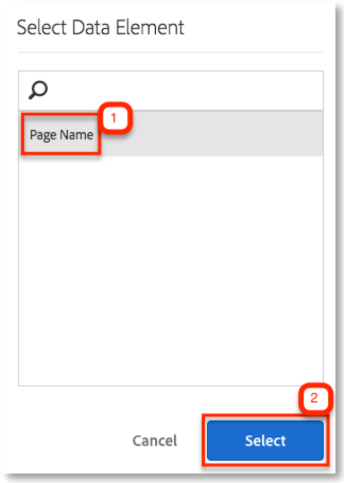
Name: Core - Library Loaded (Page Top)

No configuration necessary.

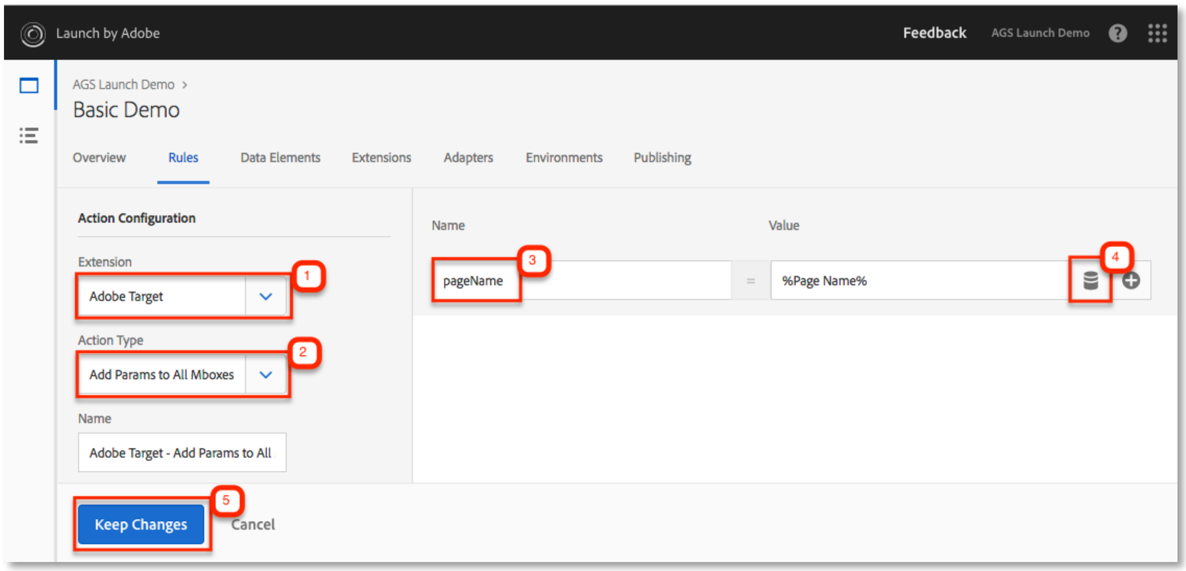
5. Since we want this rule to fire on all pages of the site, we will leave the Conditions and Exceptions empty. Note that Conditions and Exceptions add restrictions and exclusions based on a large variety of options including, URLs, data element values, date ranges, and more.
6. Under Actions, click “Add.” Actions are what you want to happen when the event fires under the right conditions
7. Select “Adobe Target” as the Extension
8. Select the “Load Target” as the action
9. Click “Keep Changes”



10. Add a second action by clicking the ⊕ icon
11. Select “Adobe Target” as the Extension
12. Select the “Add Params to All Mboxes” as the action
13. Add a parameter called “pageName” and use the icon to open the “Select Data Element” dialog
14. Select “Page Name” and then click the “Select” button in the dialog



15. Click “Keep Changes”


Launch by Adobe

Feedback

AGS Launch Demo

AGS Launch Demo >

Basic Demo

Overview Rules Data Elements Extensions Adapters Environments Publishing

Action Configuration

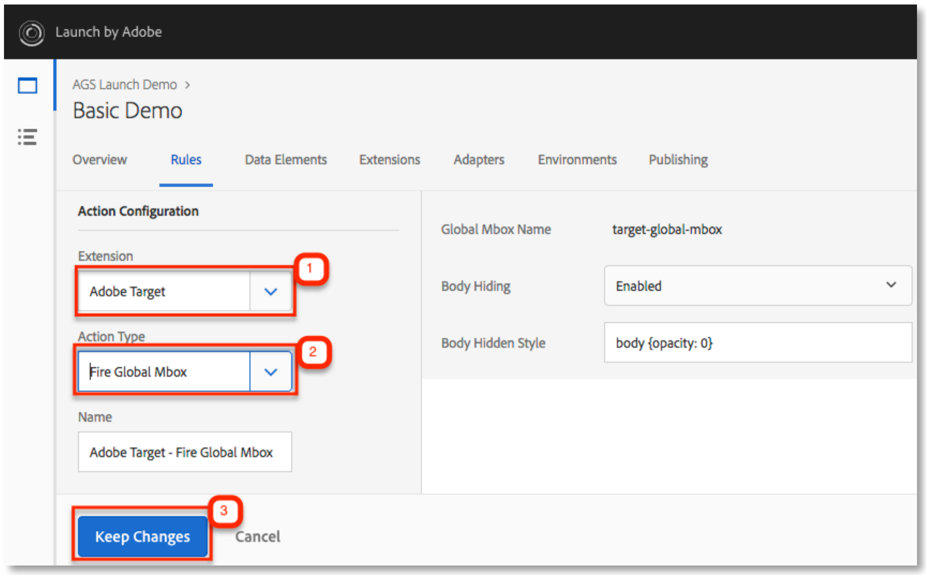
Extension: Adobe Target

Action Type: Add Params to All Mboxes

Name: Adobe Target - Add Params to All

Name	Value
pageName	%Page Name%

16. Add a third action
17. Select “Adobe Target” as the Extension
18. Select the “Fire Global Mbox” as the action (note that the pre-hiding options for typical websites are pre-selected)
19. Click “Keep Changes”


Launch by Adobe

AGS Launch Demo >

Basic Demo

Overview Rules Data Elements Extensions Adapters Environments Publishing

Action Configuration

Extension: Adobe Target

Action Type: Fire Global Mbox

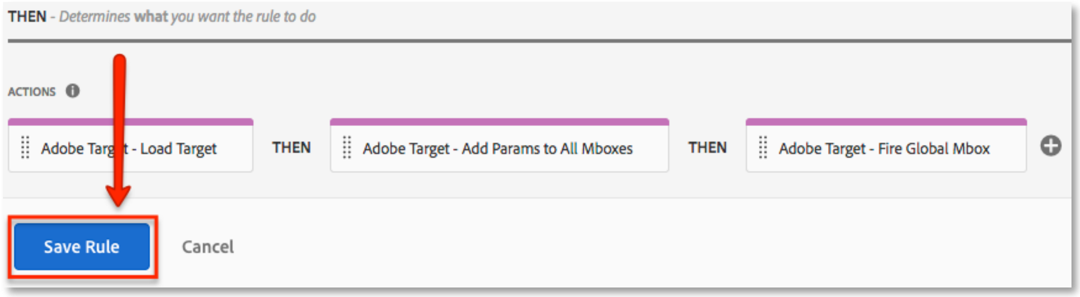
Name: Adobe Target - Fire Global Mbox

Global Mbox Name: target-global-mbox

Body Hiding: Enabled

Body Hidden Style: body {opacity: 0}

20. Verify that your rule now has three actions
21. Click “Save Rule” to save your changes


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

ACTIONS

Adobe Target - Load Target THEN Adobe Target - Add Params to All Mboxes THEN Adobe Target - Fire Global Mbox

What is similar to DTM?

The Target extension has been completely refactored for Launch. In general, you still can:

- Manage the Target library (as long as you use at.js)
- Fire the global mbox (needs to be set in Rules)
- Add global mbox parameters (need to be set in Rules)

What is different from DTM?

- Natively supports at.js instead of mbox.js
- You must use the at.js version that comes with the extension
- The at.js library is packaged with the main Launch library--Document.write is not used and no warnings in Chrome will appear related to the loading of at.js
- You don't have to load Target on all pages. You can control when the library loads and when the global mbox fires in the rule configuration
- Some settings (Client Code, Organization Id, Global Mbox Name, Server Domain, and Cross Domain) are imported from the at.js settings in the Target UI. Other settings (Timeout, Library Header and Footer) are not.
- Default mbox timeout is lower than many clients have currently and does not import from the Target UI at.js setting. (3 seconds instead of 5 seconds)
- Library Header and Footer sections need to be managed in separate "custom code" actions of the Core extension
- Additional mboxes can be fired using getOffer()/applyOffer() or trackEvent() in separate "custom code" actions of the Core extension
- No support for wrapping mboxes (might be possible with custom code)
- Parameters can be added to just the global mbox or *all mboxes* in the Rule builder by using the "Add Params to Global Mbox" or "Add Params to All Mboxes," respectively

Adobe Analytics Configuration & Tagging

Overview

In this exercise, we will implement Adobe Analytics with a global page view beacon with the Page Name variable.

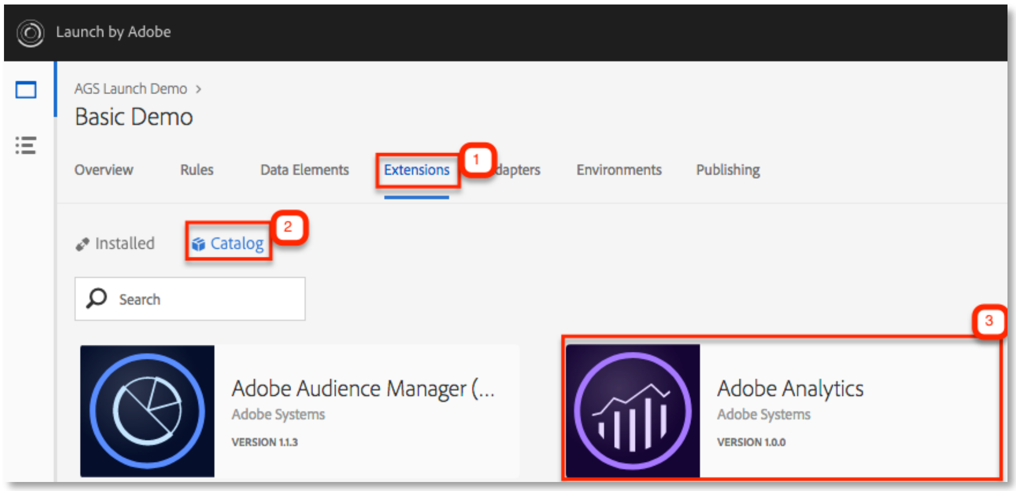
The Adobe Analytics extension supports client-side Analytics implementations using AppMeasurement.js. The Analytics extension is one of the few Adobe extensions which allows the customer to completely replace the underlying solution library with whatever you want.

Prerequisites & Product-Specific Setup

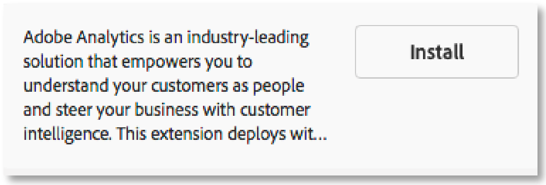
The report suites you are going to use in your property should be defined before you configure the extension.

Configuration 1: Add the Analytics Extension

1. Go to Extensions->Catalog and position your mouse over the Adobe Analytics extension



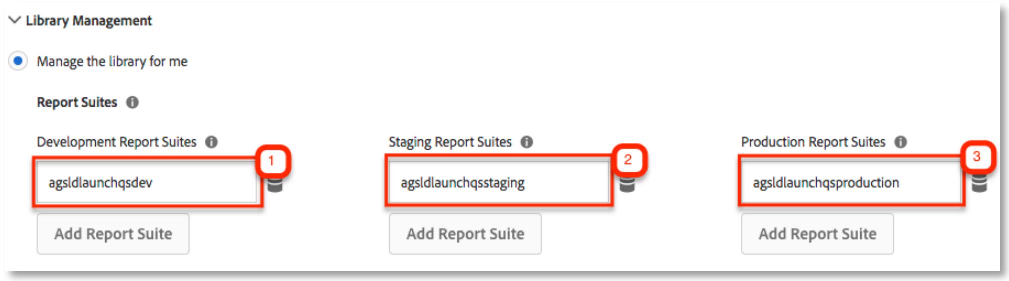
2. Note that this will expose the AppMeasurement.js version which the extension currently uses (you might need to hover over the description text for a moment to see it)
3. Click “Install” to add the extension



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

4. Leave all of the default settings, except for the following (in the Adobe Audience Manager (AAM) Configuration & Tagging section we will switch to a custom AppMeasurement.js library):

a. Library Management: Specify the report suites you would like to use:



Library Management

● Manage the library for me

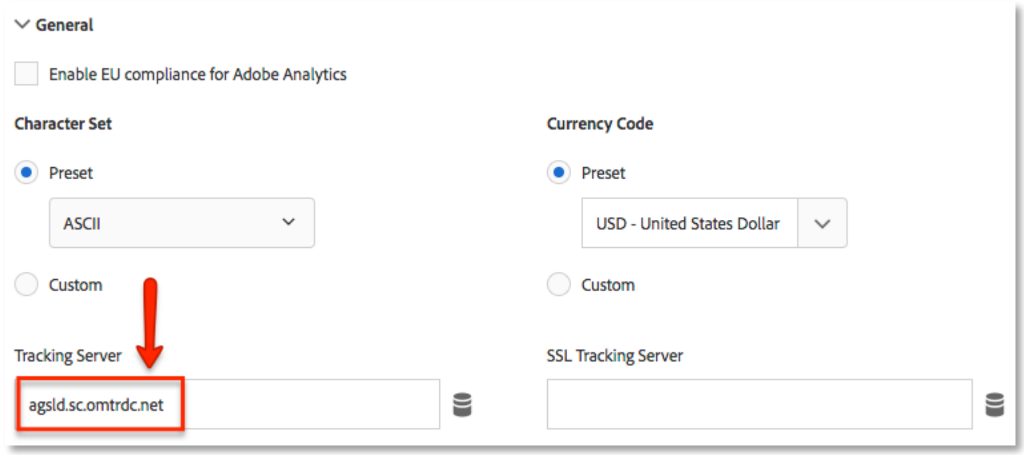
Report Suites

Development Report Suites: agsldlaunchqsdev

Staging Report Suites: agsldlaunchqsstaging

Production Report Suites: agsldlaunchqsproduction

b. General->Tracking Server: Enter your tracking server, e.g. “agsld.sc.omtrdc.net.” Enter your SSL Tracking Server if your site supports https:



General

☐ Enable EU compliance for Adobe Analytics

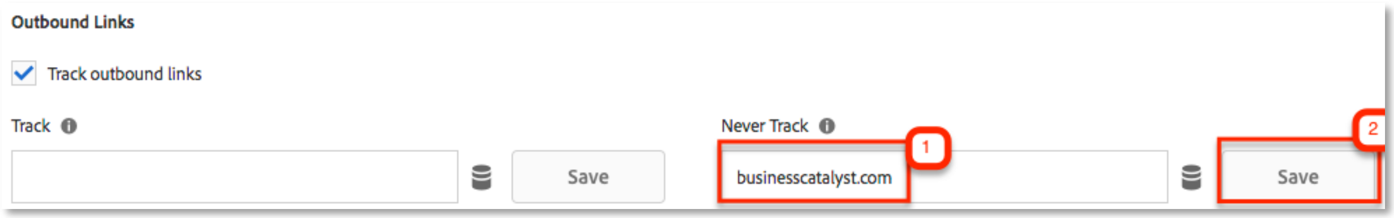
Character Set: ● Preset — ASCII ○ Custom

Currency Code: ● Preset — USD - United States Dollar ○ Custom

Tracking Server: agsld.sc.omtrdc.net

SSL Tracking Server:

5. In the Link Tracking section, enter your domain in the “Never Track” section and click the “Save” button



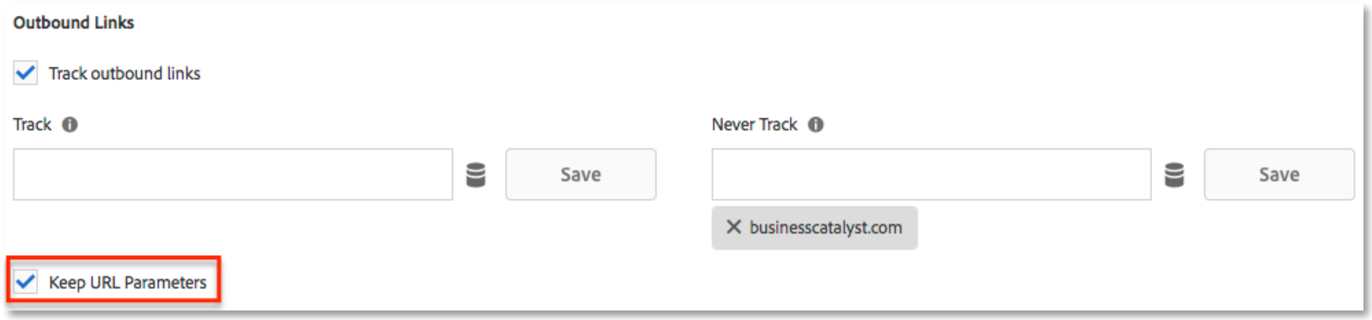
Outbound Links

✓ Track outbound links

Track:

Never Track: businesscatalyst.com

6. Check the box for “Keep URL Parameters”



Outbound Links

✓ Track outbound links

Track:

Never Track: × businesscatalyst.com

✓ Keep URL Parameters

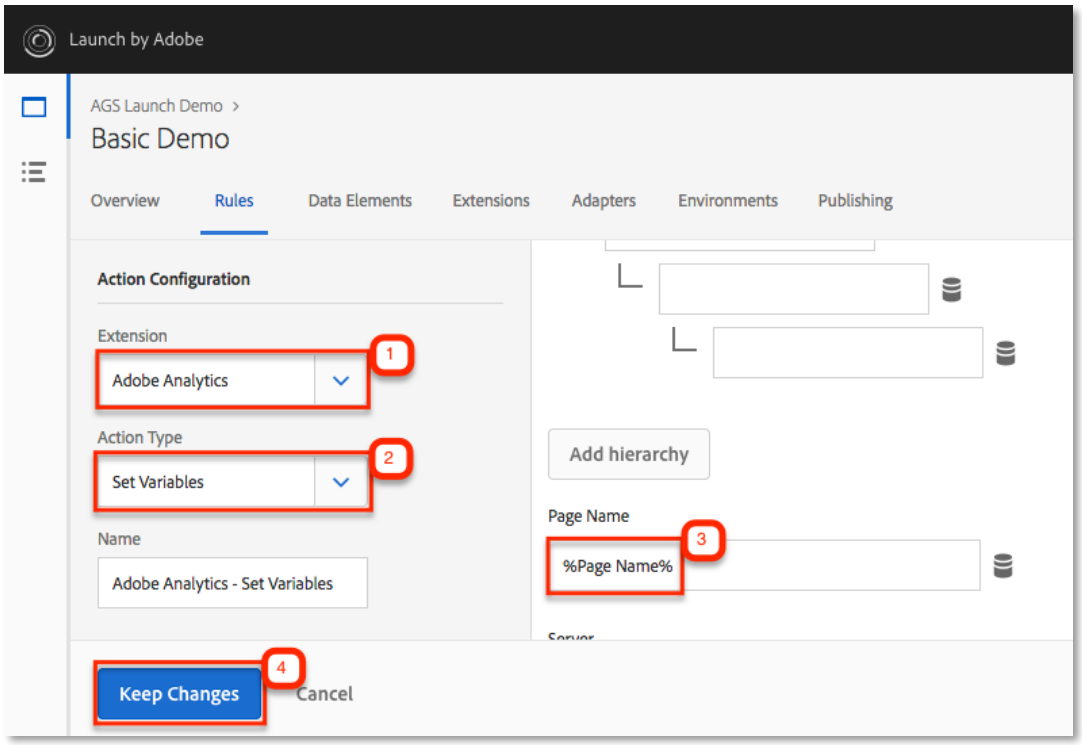
7. Click “Save” to save the extension

Global variables can be set in the extension configuration or in rule actions. Either way is fine. One "gotcha" to be aware of is that the extension configuration can only read data elements that have been defined before the Launch embed codes, which is different from DTM. The reference implementation sets the “Page Name” variable using a rule action.

Configuration 2: Add Rules to set variables and fire the Analytics beacon

Now we are going to create a rule to set the “Page Name” variable and fire the Analytics beacon. The best practice is to fire the Analytics beacon at the bottom of every page of a website:

1. Create a new rule called “All Pages - Bottom”
2. Select “Page Bottom” as the event
3. Do not specify a Condition or Exclusion so the rule will fire on all pages
4. Add an Adobe Analytics Extension->Set Variables action to report the “Page Name” data element as the page name. Many other variables can be set via this action as well:



Action Configuration

Extension: Adobe Analytics

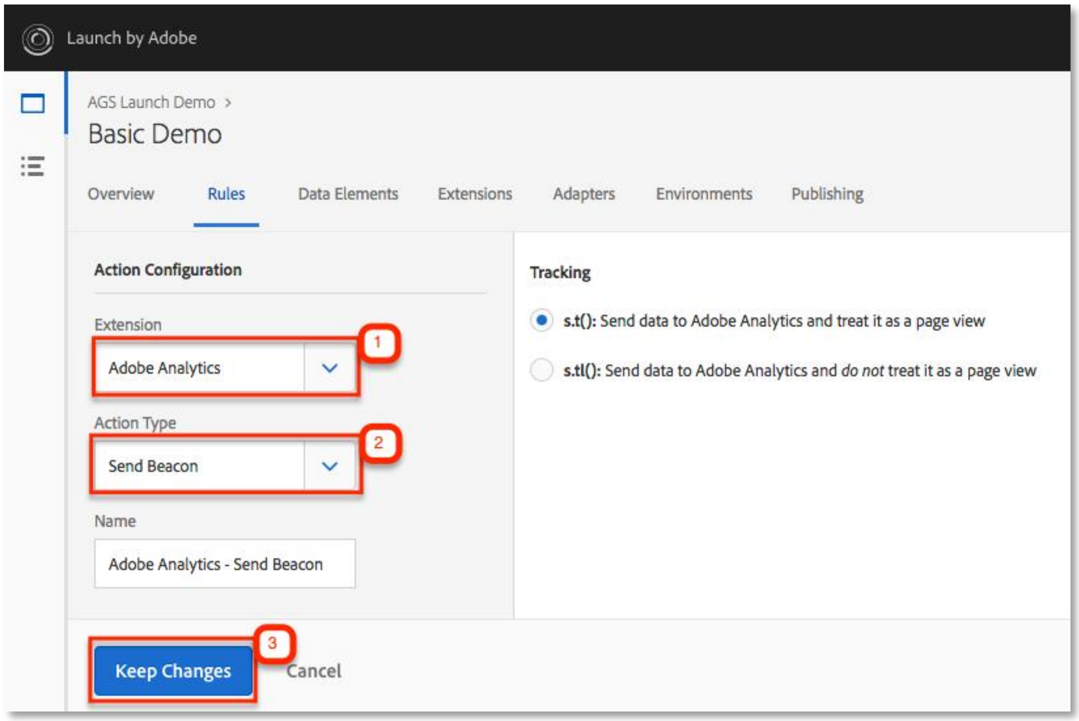
Action Type: Set Variables

Name: Adobe Analytics - Set Variables

Page Name: %Page Name%

5. Add an Adobe Analytics Extension->Send Beacon action

6. Leave Tracking set to s.t(). Note that if you wanted to make an s.tl() call in a click-event rule you could do that using the Send Beacon action, as well:



Action Configuration

Extension: Adobe Analytics

Action Type: Send Beacon

Name: Adobe Analytics - Send Beacon

Tracking

- [✓ 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

7. Save the rule

What is similar to DTM?

- Almost all options are the same as in DTM, although some of the options are grouped differently.

What is different from DTM?

- Report Suites don't auto-populate—you have to know the specific report suite ids and manually enter them
- The managed library option ships with a specific version of AppMeasurement.js
- Multiple Analytics instances like having two Analytics "Tools" in DTM is not supported
- The Audience Manager Module for server-side forwarding can be added by simply checking a box (no need to use a custom library). See Adobe Audience Manager (AAM) Configuration & Tagging for details.
- General: New options for *Custom* Character Set and Currency Code
- General: Data Center option has been removed
- Global Variables: New options for Server, State, Zip
- Global Variables: data elements must be defined before the Launch header embed code for variables set in the extension configuration (variables set in Rules can come from data elements set before the event)
- Custom Code: Access to the s. object has been added
- Custom Code: Option to load before or after the UI settings has been removed
- Some options have been rearranged and should be in more intuitive locations
- Since Marketing Cloud authentication is used to access Launch, Web Services authentication was removed

Adobe Audience Manager (AAM) Configuration & Tagging

Overview

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

1. **Server-Side Forwarding (SSF)** – If customers have Adobe Analytics this solution forwards Adobe Analytics data to AAM on Adobe's backend, allowing for one less pixel on the page. This enables key integration features and conforms with our best practices for AAM code implementation and deployment. Server-side forwarding is implemented in this Reference Implementation because Adobe Analytics is also implemented.
2. **Client-Side DIL** – This solution covers customers who do 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.

This guide provides basic information on how to set up a server-side forwarding implementation. For a complete description and requirements list for server-side forwarding, please review the documentation as well, so that you are familiar with how it works, what is required, and how to validate.

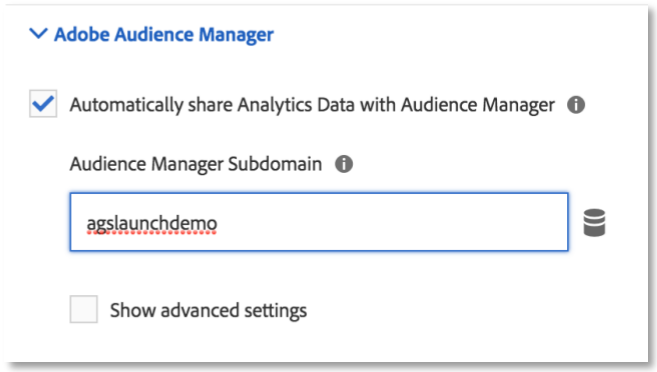
Configuration 1: Enable Server-Side Forwarding in the Adobe Analytics Admin Console

Launch will contain the code associated with SSF and the response from AAM back to the page, but *SSF also needs to be enabled on the back end,* in the Adobe Analytics Admin Console.

As stated on this page of the Help documentation, after logging into Analytics go to Admin > Report Suites > (select report suites) > Edit Settings > General > Server-Side Forwarding. Enable SSF for the report suite.

Configuration 2: Enable Server-Side Forwarding in Launch

1. Go to Extensions > Installed and click to configure the Analytics extension.
2. Expand the "Adobe Audience Manager" section
3. Check the box to "Automatically share Analytics Data with Audience Manager"—this will add the Audience Manager Module to the AppMeasurement.js implementation.
4. Add your "Audience Manager Subdomain" (also known as the "Partner Name," "Partner ID," or "Partner Subdomain")



☑ Automatically share Analytics Data with Audience Manager

Audience Manager Subdomain: agslaunchdemo

☐ Show advanced settings

5. Save these changes to the Analytics extension

Server-side forwarding code is now implemented!

Note: If you have just enabled the SSF in the interface, as described in the "Configuration 1: Enable Server-Side Forwarding in the Adobe Analytics Admin Console", it will take up to four hours for it to start working. Be

Integrations

Integration Consideration 1: Audience Sharing

Audience Sharing with the People Core Service requires that you implement the Experience Cloud ID Service and Audience Manager Server-Side Forwarding,, 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.

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 a global mbox in a "Library Loaded (Page Top)" rule and the Analytics page view beacon in a "Page Bottom" rule as configured in this reference architecture, you will satisfy this requirement. You should always see matching supplemental data ids (SDIDs) in both calls—these ids stitch hits together in the Analytics processing for A4T.

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. When using a "Library Loaded (Page Top)" rule to fire the target-global-mbox, be sure to use the "Set Customer Id" action before all of the Target actions. The Standard Web Reference Architecture Guide demonstrates how to do this and is available from here: Additional Reference Implementation Guides.

Validate and Publish your implementation

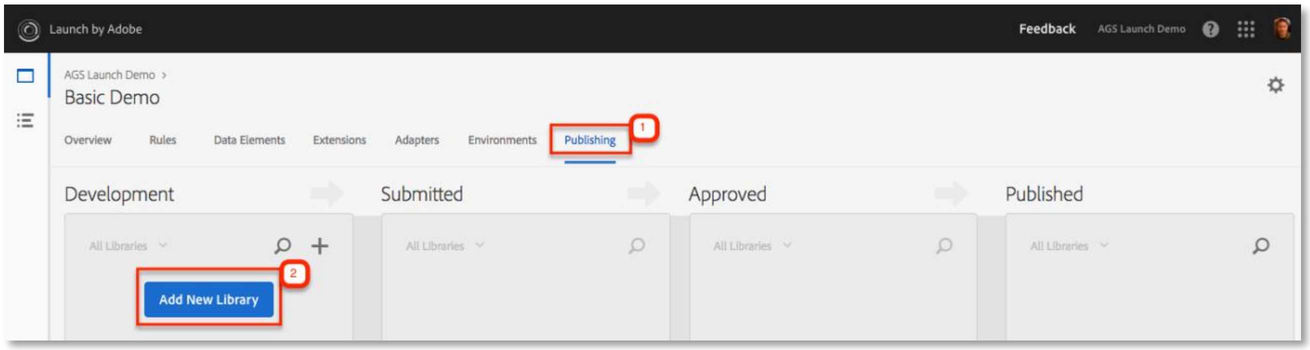
Overview

Perhaps the most dramatic change from DTM to Launch is the Publishing process. The new publishing process was designed for enterprise-level support. Because Launch offers unlimited development environments, more work is required to push changes through the publishing process. Here are some key changes to be aware of:

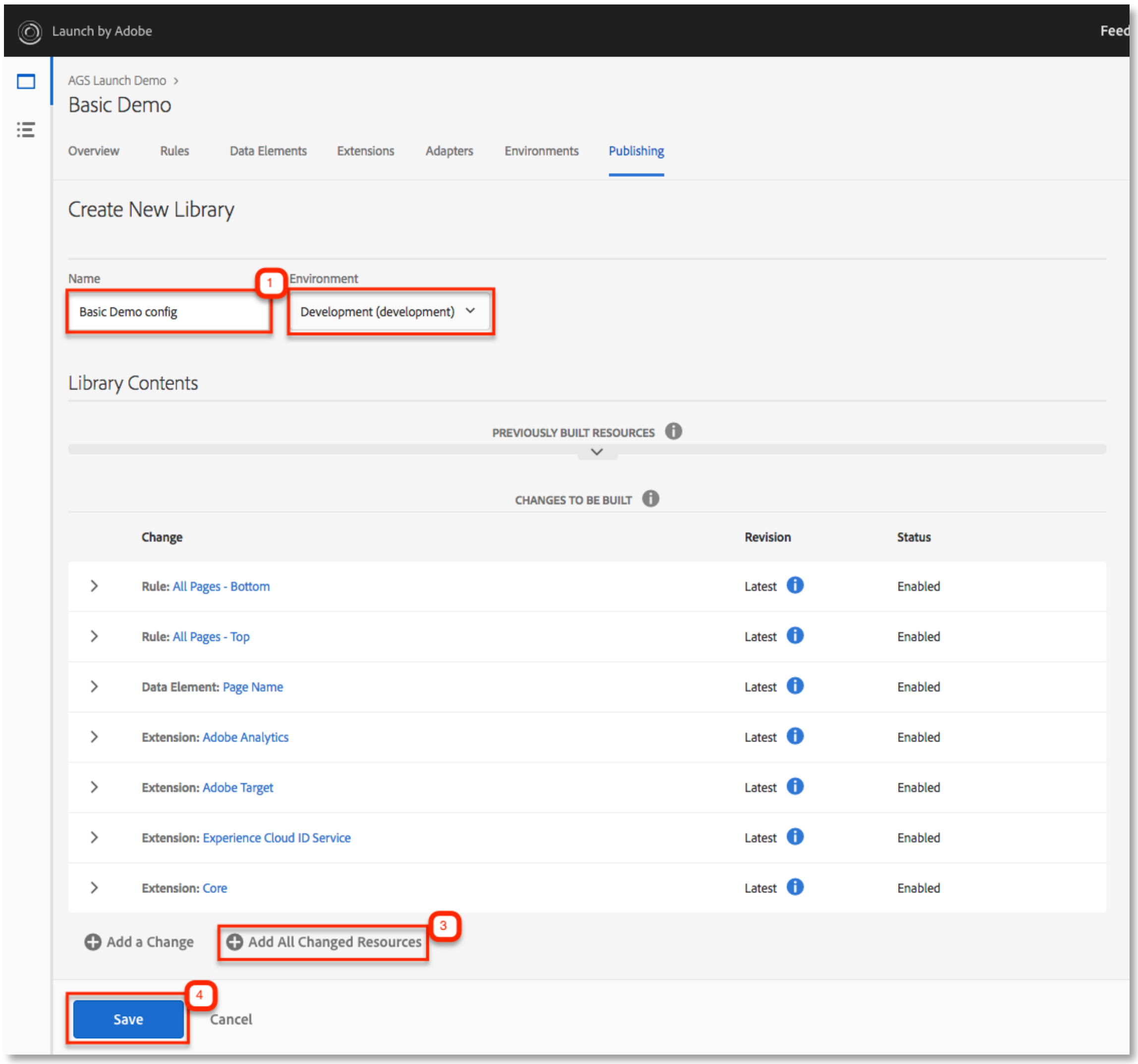
- DTM automatically saved any change directly to the Staging environment. Launch can be used to automatically save changes to any of your Development environments.
- Because Launch was designed to meet the requirements of large enterprises with multiple teams using Launch simultaneously, it requires more steps to get code onto the Staging environment. To do so, you must:
    - Add the change to a Library
    - Build the Library in a Development environment
    - Validate your changes in the Development environment
    - Submit to the Staging environment
    - Build the Library in the Staging environment

Create a library in your Development environment

1. Go to “Publishing” in the top navigation and click the “Add New Library” button



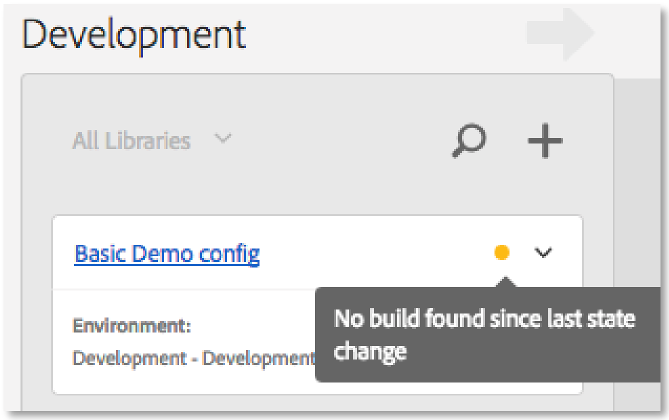
2. Name the library, e.g. “Basic Demo config”
3. Assign it to an environment, e.g. “Development”
4. Click “Add All Changed Resources”—this will add all of the changes we just made to the library. In multi-user scenarios, you could then remove the changes that weren’t yours or otherwise don’t belong in your library.
5. Save the library



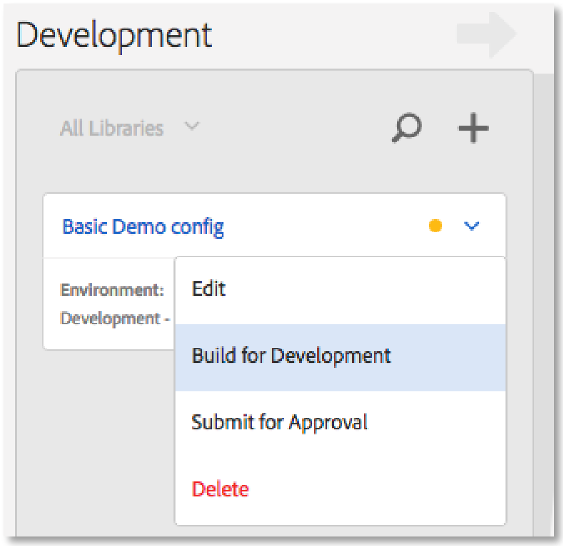
Name: Basic Demo config
Environment: Development (development)

Change	Revision	Status
Rule: All Pages - Bottom	Latest	Enabled
Rule: All Pages - Top	Latest	Enabled
Data Element: Page Name	Latest	Enabled
Extension: Adobe Analytics	Latest	Enabled
Extension: Adobe Target	Latest	Enabled
Extension: Experience Cloud ID Service	Latest	Enabled
Extension: Core	Latest	Enabled

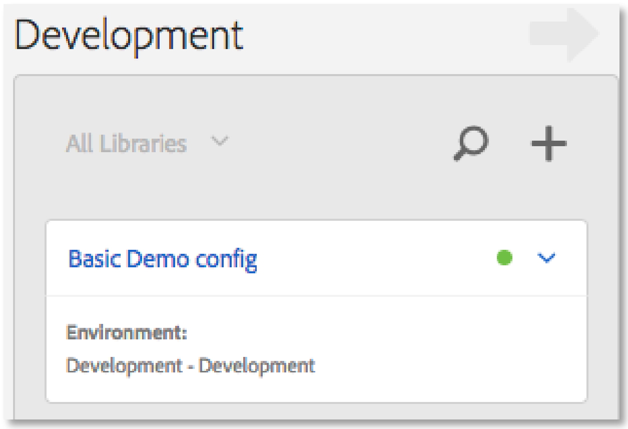
6. At this point your library has been saved, but needs to be "built" before you can validate your changes. The yellow dot icon indicates that the library has not been built:


Development
All Libraries
Basic Demo config
Environment:
Development - Development
No build found since last state change

7. Open the dropdown and select "Build for Development"



8. A green dot icon will appear when the library has built. You are now ready to validate your changes!



Validate your changes using Launch Command and the Experience Cloud Debugger

How do we know our Launch configuration is firing all of our marketing tags the way we want? You can use several new tools to validate your implementation—Launch Command and the Experience Cloud Debugger.

Configure the Launch Command Chrome Extension

The Launch Command Chrome Extension is not an official Adobe extension, but can be used to switch the embed codes implemented on the page. You can use it to do things like:

1. Make the page load a Development library instead of the Production library
2. Make the page load a Launch library instead of a DTM library
3. Make the page load a library from a different account

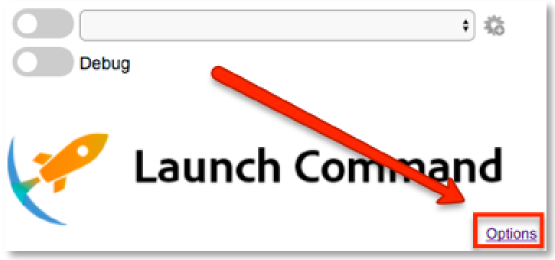
Other tools such as Requestly or Charles could be used to accomplish the same task. Also, Search Discovery's "DTM Switch" extension has been updated to support Launch and can be used to switch to Development and Staging libraries of the same property.

If you don't have the extension already installed, add it from the Chrome store: https://chrome.google.com/webstore/detail/launch-command/nkjhamgjeocefocmpbcjfmohkjgildki

Add Embed Code Environments to Launch Command

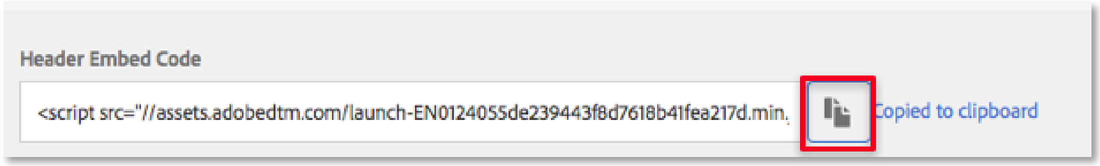
You can add the embed codes for all of the environments you just created to Launch Command and then enable the extension on the demo site so it will load your Launch embed codes instead of ours. You can then validate the setup of your property. Alternatively, you can save the demo site html file, hardcode your own embed codes, and publish it to your own webserver.

Instructions can be found within the extension itself by clicking on the "Options" link:



Here are step-by-step instructions to add the embed codes:

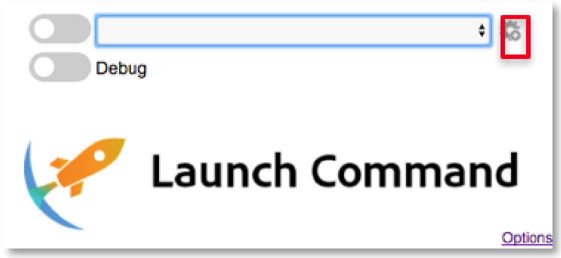
1. Open the "Development 1" environment and click the icon to copy the embed code to your clipboard



Header Embed Code

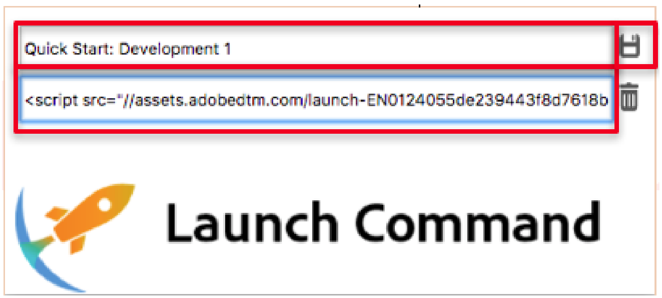
<script src="//assets.adobedtm.com/launch-EN0124055de239443f8d7618b41fea217d.min.

2. Open the Launch Command browser extension
3. Click the icon to configure a new embed code:



4. Enter "Quick Start: Development 1" as the Name

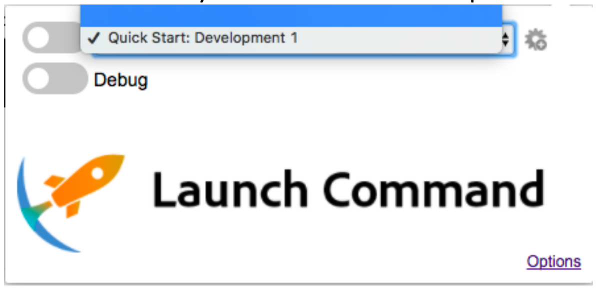
5. Paste the entire header embed code
6. Click the 💾 icon to save



Quick Start: Development 1

<script src="//assets.adobedtm.com/launch-EN0124055de239443f8d7618b

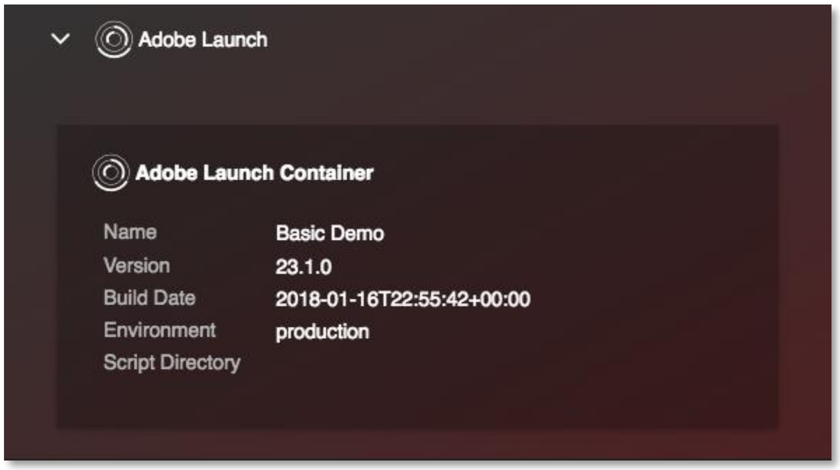
7. Switch back to the Launch UI
8. Open the "Development 2" environment and click the icon to copy the embed code to your clipboard
9. Open the Launch Command browser extension.
10. Click where it says "Quick Start: Development 1" to open the dropdown and select the empty option



11. Click the ⚙ icon again to enter a new embed code
12. Name the new environment "Quick Start: Development 2," paste its embed code, and save
13. Repeat these steps for the Staging and Production embed codes

Verify Launch Command Setup

1. Open the Demo Site URL: http://agslaunchdemo.businesscatalyst.com/basic/
2. Open the Experience Cloud Debugger extension
3. On the Summary tab of the debugger
4. Scroll to the Adobe Launch section--it should show that the Launch environment used is the "Basic Demo" production environment

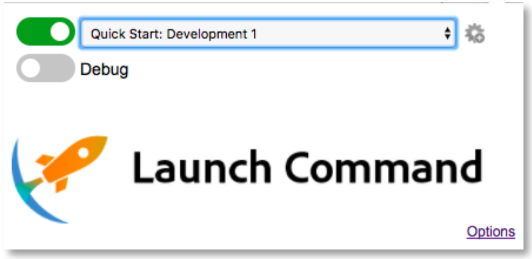


Adobe Launch Container

Name	Basic Demo
Version	23.1.0
Build Date	2018-01-16T22:55:42+00:00
Environment	production
Script Directory	

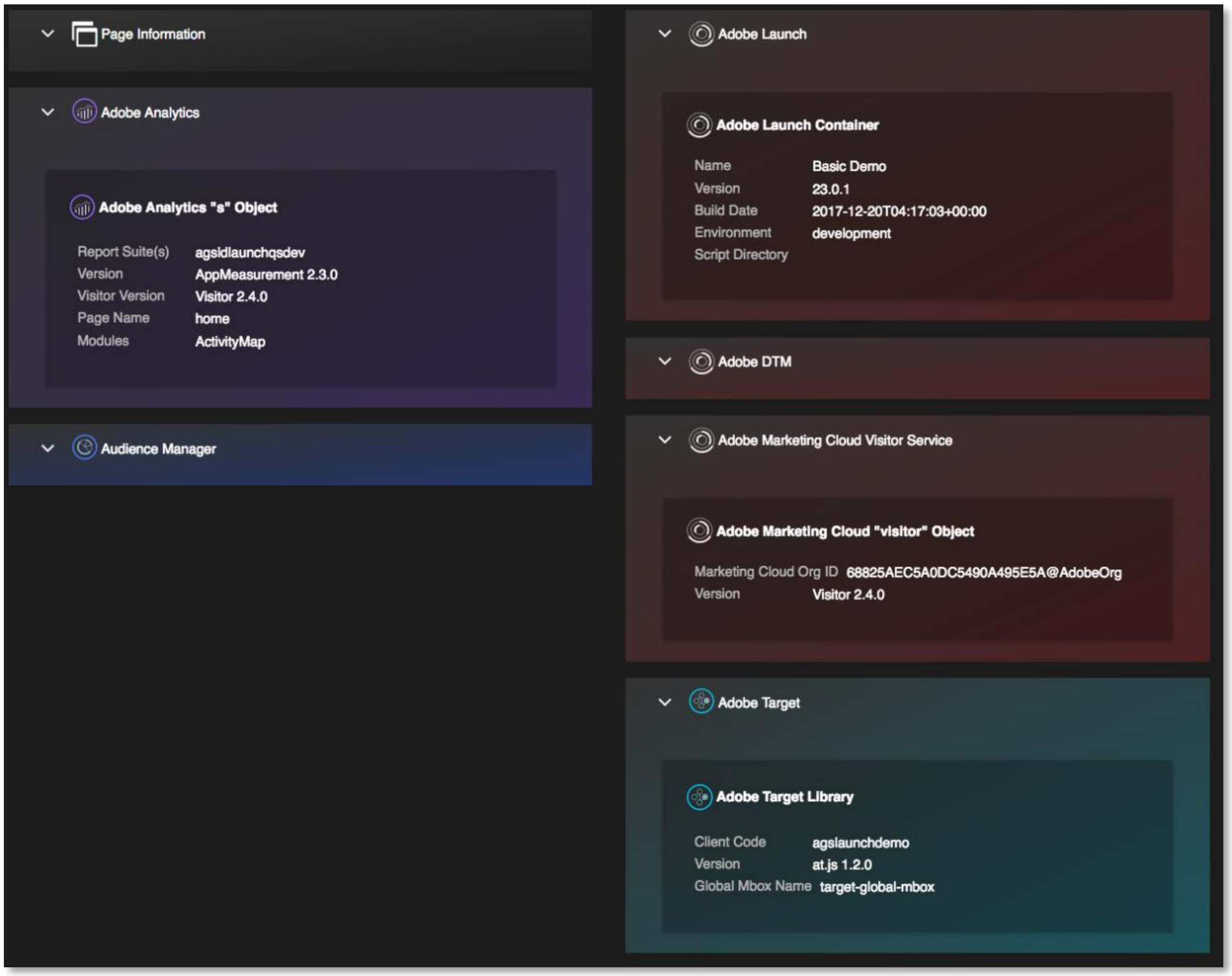
5. Open the Launch Command extension

6. Select the "Quick Start: Development 1" environment and toggle it on as pictured below:


Quick Start: Development 1
Debug

7. Reload the Demo site

8. On the Summary tab, you should see many details related to Launch (e.g. the property name and environment), Analytics (the library version, report suite, and other key variables), Visitor ID Service (e.g. Org Id and library version), and Target (e.g. client code, library version, and global mbox name). Verify that this matches what you are expecting to see:



Adobe Analytics "s" Object

Report Suite(s)	agsldlaunchqsdev
Version	AppMeasurement 2.3.0
Visitor Version	Visitor 2.4.0
Page Name	home
Modules	ActivityMap

Adobe Launch Container

Name	Basic Demo
Version	23.0.1
Build Date	2017-12-20T04:17:03+00:00
Environment	development
Script Directory	

Adobe Marketing Cloud "visitor" Object

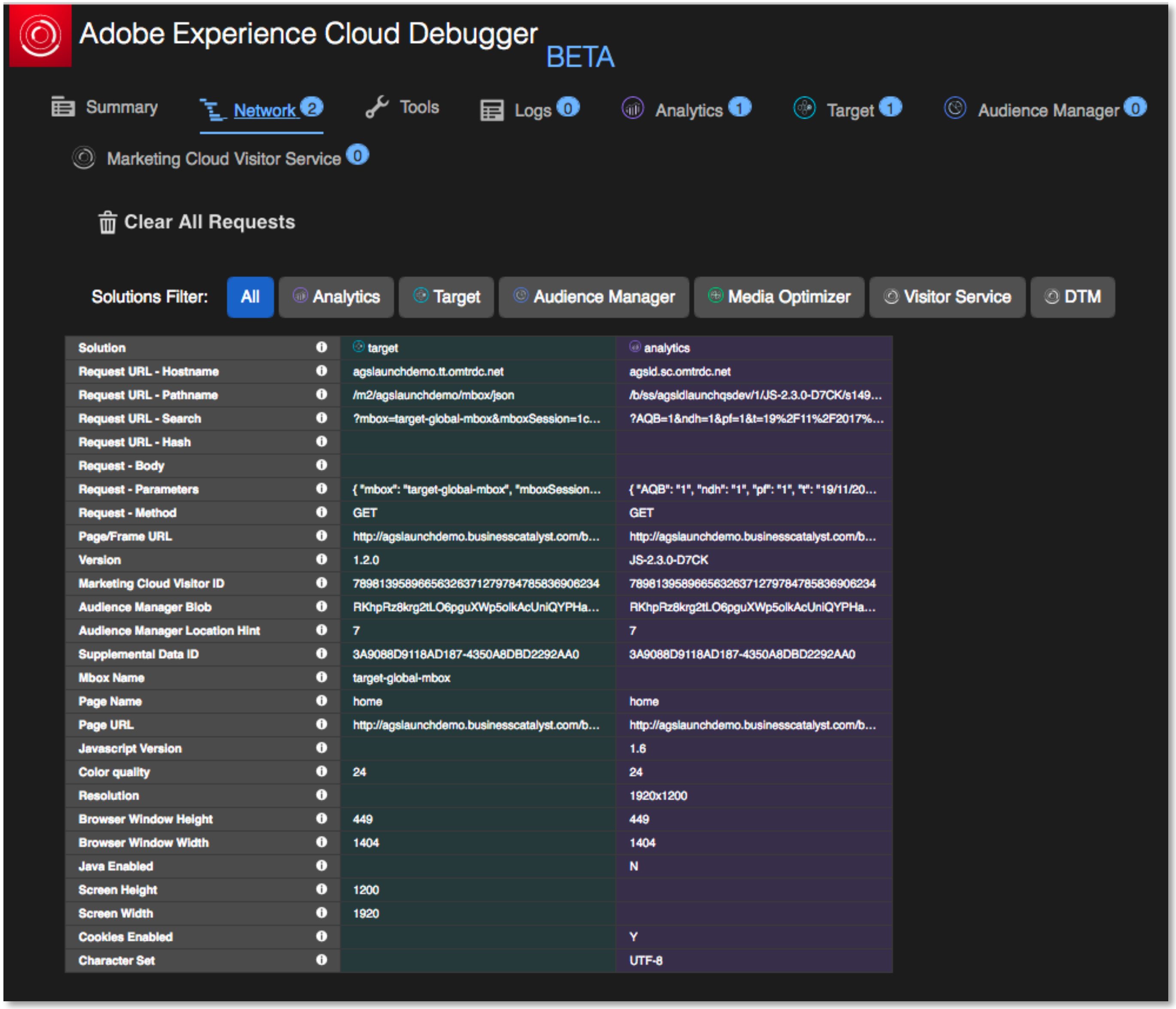
Marketing Cloud Org ID	68825AEC5A0DC5490A495E5A@AdobeOrg
Version	Visitor 2.4.0

Adobe Target Library

Client Code	agslaunchdemo
Version	at.js 1.2.0
Global Mbox Name	target-global-mbox

9. Switch to the "Network" tab of the Debugger. This will show you a running list all of the calls made to Adobe solutions since the Debugger was opened (note that the Debugger does a page reload when it first opens). Verify that the Target and Analytics requests fired with the expected parameters. You can also confirm that the Visitor

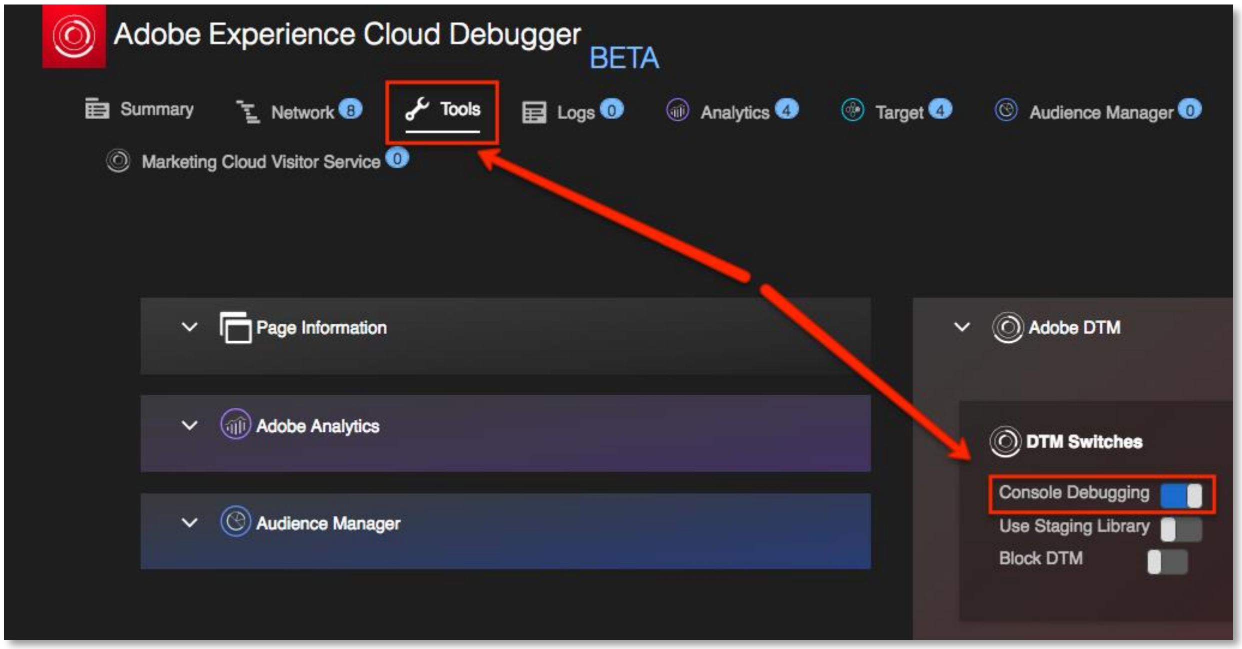
Ids match (required for Audience sharing) and that the Supplemental Data Ids match (required for A4T):


Adobe Experience Cloud Debugger BETA
Summary
Network 2
Tools
Logs 0
Analytics 1
Target 1
Audience Manager 0
Marketing Cloud Visitor Service 0
Clear All Requests
Solutions Filter: All Analytics Target Audience Manager Media Optimizer Visitor Service DTM

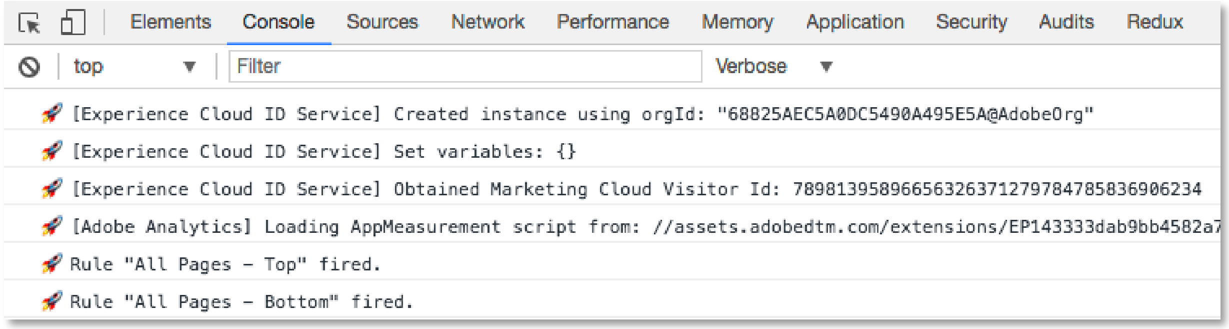
Solution	target	analytics
Request URL - Hostname	agslaunchdemo.tt.omtrdc.net	agsld.sc.omtrdc.net
Request URL - Pathname	/m2/agslaunchdemo/mbox/json	/b/ss/agsldlaunchqsdev/1/JS-2.3.0-D7CK/s149...
Request URL - Search	?mbox=target-global-mbox&mboxSession=1c...	?AQB=1&ndh=1&pf=1&t=19%2F11%2F2017%...
Request URL - Hash		
Request - Body		
Request - Parameters	{ "mbox": "target-global-mbox", "mboxSession...	{ "AQB": "1", "ndh": "1", "pf": "1", "t": "19/11/20...
Request - Method	GET	GET
Page/Frame URL	http://agslaunchdemo.businesscatalyst.com/b...	http://agslaunchdemo.businesscatalyst.com/b...
Version	1.2.0	JS-2.3.0-D7CK
Marketing Cloud Visitor ID	78981395896656326371279784785836906234	78981395896656326371279784785836906234
Audience Manager Blob	RKhpRz8krg2tLO6pguXWp5olkAcUniQYPHa...	RKhpRz8krg2tLO6pguXWp5olkAcUniQYPHa...
Audience Manager Location Hint	7	7
Supplemental Data ID	3A9088D9118AD187-4350A8DBD2292AA0	3A9088D9118AD187-4350A8DBD2292AA0
Mbox Name	target-global-mbox	
Page Name	home	home
Page URL	http://agslaunchdemo.businesscatalyst.com/b...	http://agslaunchdemo.businesscatalyst.com/b...
Javascript Version		1.6
Color quality	24	24
Resolution		1920x1200
Browser Window Height	449	449
Browser Window Width	1404	1404
Java Enabled		N
Screen Height	1200	
Screen Width	1920	
Cookies Enabled		Y
Character Set		UTF-8

Not all parameters will show in the Network tab (e.g. Customer Ids reported by the Id Service, Target, and Analytics requests). You can use the tabs for the individual solutions to confirm all parameters in a solution request.

10. Open the “Tools” tab and toggle on the DTM “Console Debugging” tool

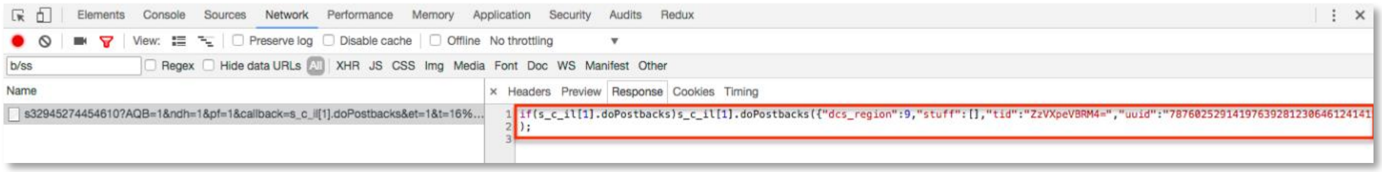


This will still turn on Console Debugging statements for Launch. Note that they are all prefixed with a 🚀 icon:



```
🚀 [Experience Cloud ID Service] Created instance using orgId: "68825AEC5A0DC5490A495E5A@AdobeOrg"
🚀 [Experience Cloud ID Service] Set variables: {}
🚀 [Experience Cloud ID Service] Obtained Marketing Cloud Visitor Id: 78981395896656326371279784785836906234
🚀 [Adobe Analytics] Loading AppMeasurement script from: //assets.adobedtm.com/extensions/EP143333dab9bb4582a7
🚀 Rule "All Pages - Top" fired.
🚀 Rule "All Pages - Bottom" fired.
```

11. With server-side forwarding of AAM, an AAM hit is not seen. Instead the “callback” parameter will be in the Adobe Analytics hit. The AAM response will now be in the AA hit, including a dcs_region at the beginning of it and a uuid at the end of it.



```
s32945274454610?AQB=1&ndh=1&pf=1&callback=s_c_il[1].doPostbacks&et=1&t=16%...
```

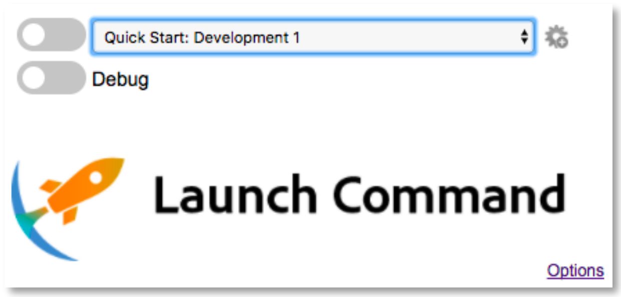
```
if(s_c_il[1].doPostbacks)s_c_il[1].doPostbacks({"dcs_region":9,"stuff":[],"tid":"ZzVXpeVBRM4=","uuid":"78760252914197639281230646124141
);
```

If there is an error “Invalid customer id _crm_id” a Customer Attributes (part of the Profiles & Audiences core service) or Audience Manager data source needs to be created, using this Integration Code (without the leading “_”).

If the response says “SUCCESS” in it, make sure the steps have been completed to configure the report suite and make sure it has been four hours since the server-side forwarding of the report suites has been configured.

12. Disable the Launch Command extension by toggling it off. We recommend always turning off the extension when you are done using it. When enabled, *it will replace every Launch or DTM embed code on every page of every*

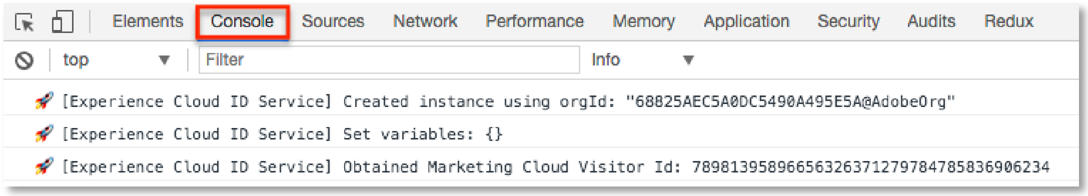
browser tab that you load. It will even change the embed codes within the Launch UI, basically implementing your Launch property in the Launch UI, which could lead to confusion. This is a known issue that will hopefully be fixed soon.



⚠ When the extension is enabled, it might replace all DTM and Launch embed codes in every page of every tab you load regardless of the appearance of the toggle. This is a known issue that the developer will hopefully fix soon.

⚠ Although Launch Command allows you to paste the entire header embed code, behind the scenes it *only replaces the URL* of the existing DTM or Launch library URL. It cannot be used to compare synch vs asynch embed codes.

Note that Launch Command also has a toggle for debugging: Debug . Just like with the DTM Switch extension, this will expose debugging statements in your browser console. Launch console statements are prefixed with a 🚀 icon:



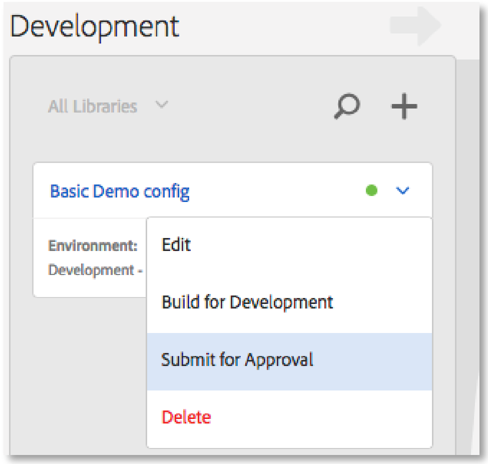
```
🚀 [Experience Cloud ID Service] Created instance using orgId: "68825AEC5A0DC5490A495E5A@AdobeOrg"
🚀 [Experience Cloud ID Service] Set variables: {}
🚀 [Experience Cloud ID Service] Obtained Marketing Cloud Visitor Id: 78981395896656326371279784785836906234
```

Publish your changes to Staging and Production environments

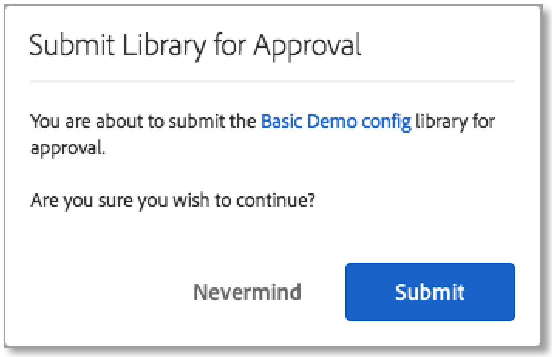
Publish to Staging

Once you have validated your changes in the Development environment, it is time to publish them to the Staging environment.

1. On the “Publishing” page, open the dropdown next to your library and select “Submit for Approval:”



2. Click the “Submit” button in the dialog:

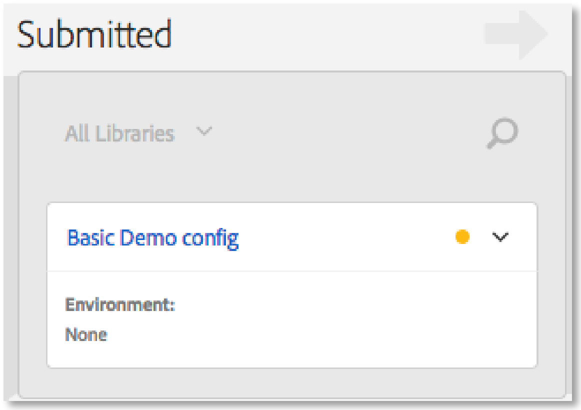


Submit Library for Approval

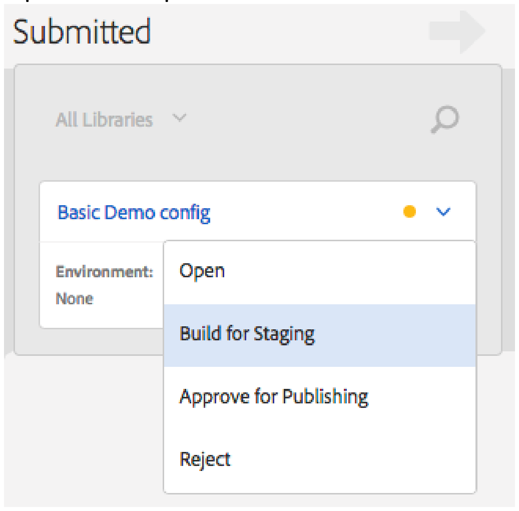
You are about to submit the Basic Demo config library for approval.

Are you sure you wish to continue?

3. Your library will now appear in the Submitted column in an unbuilt state:



4. Open the dropdown and select "Build for Staging:"

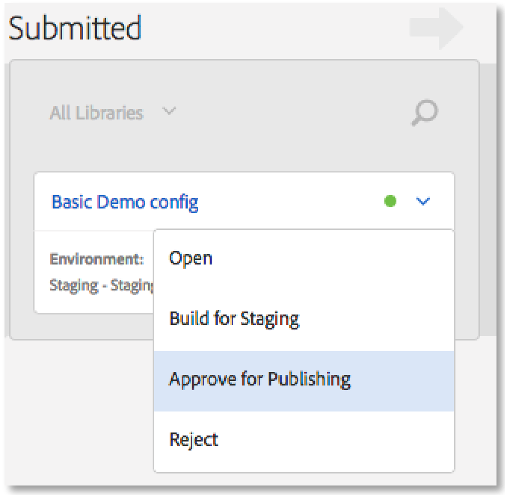


5. Once the green-dot icon appears, the library can be previewed in the Staging environment.

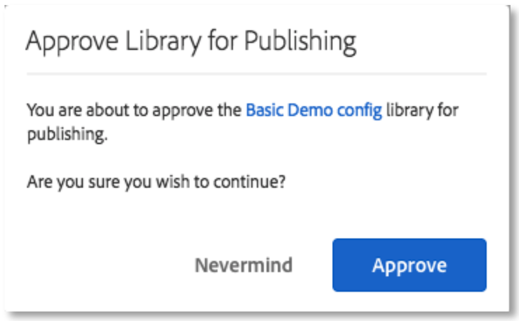
Once your QA team has signed off by reviewing the changes in the Staging environment it is time to publish to production.

Publish to Production

1. From the "Publishing" page, click "Approve for Publishing":



2. Click "Approve" in the dialog box:

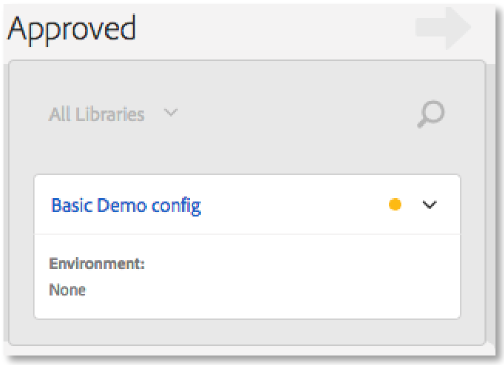


Approve Library for Publishing

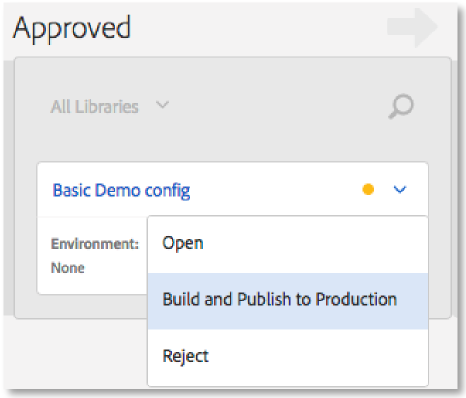
You are about to approve the Basic Demo config library for publishing.

Are you sure you wish to continue?

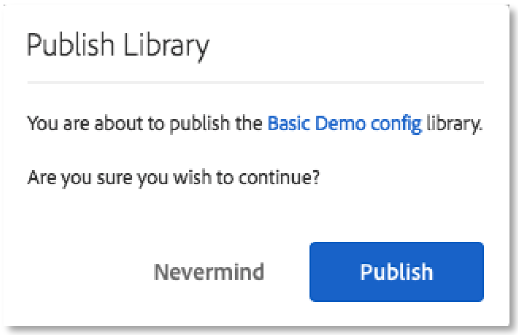
3. The library will now appear in the Approved column in the unbuilt state (yellow dot):



4. Open the dropdown and select “Build and Publish to Production:”



5. Click “Publish” in the dialog box:

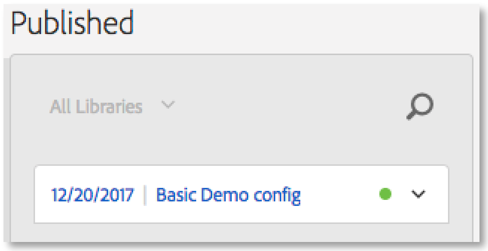


Publish Library

You are about to publish the Basic Demo config library.

Are you sure you wish to continue?

6. The library will now appear in the Published column:



At this point you should disable the Launch Command extension by toggling it off. You should see all of your changes in the production embed codes that are hard-coded in the page.

That’s it! You have now implemented the Experience Cloud using Launch!