

Adobe Launch Exercises August 2018

Contents

Exercise 1 - Log in 5 Mins	6
Part 1 - Create a Property	7
Preliminaries - Best Practices for Planning Properties	7
The Similarity vs Dissimilarity Factors to Consider When Planning Properties	7
So What Exactly is a Property?	8
Exercise 2 - Create a property 7 Mins	9
Part 2 – Setting up Adapters (Where)	11
Benefits of the Launch Adapter	11
<i>Did you know...(more about Akamai)</i>	11
Why choose Hosting By Adobe (Akamai) for library files?	11
Can I avoid errors in case of Akamai unavailability?	11
Self-managed adapter	11
Exercise 3: Adapters (Hosted by Adobe – Akamai) 3 Mins	13
Just for Grins.. Exercise: Adapters (SFTP)	14
Part 3 - Setting Up Your Environments	15
Exercise 4A – Setting up a New Environment (Development) 3 Mins	15
Exercise 4B – Setting up a New Environment (Staging) 3 Mins	17
Exercise 4C – Setting up a New Environment (Production) 2 Mins	19
Part 4 - Install Extensions (formerly aka DTM Tools)	21
Exercise 5: Add a New Extension (Core)	22
List of Valid Report Suite IDs	23
Tracking Server	23
Adobe Analytics Extension	24
Configure the Adobe Analytics extension	24
Library Management	24
Exercise 5A - Adobe Analytics Extension 4 Mins	25
Exercise 5B - Experience Cloud ID Service Extension 2 Mins	28
Experience Cloud Organization ID	28
Part 5 - Data Elements	30
Data element usage	30
Types of data elements	31
JavaScript Variable	31
DOM Attribute	31
Cookie	31
URL Querystring Parameter	31
Custom Script	31

Built-in data elements	32
Create a data element.....	33
Exercise 6A: Data Element (Java Script Variable – Page Name) 2 Mins	34
Exercise 6B: Data Element (URL Query String Parameter – Internal Campaign) 2 Mins.....	35
Exercise 6C: Data Element (URL Query String Parameter – Internal Search Term) 1 Min	36
Part 6 – Rules.....	37
Events (If)	37
Actions (Then)	37
Conditions, Events and exceptions (if)	37
Core Extension Rules Action Types	38
Experience Cloud ID Service Extension Rules Action Types	38
How to Create Rules.....	40
Exercise 7: Rules (Page Load Rule) 5 Mins.....	41
Part 7 - Test in your Development Environment.....	46
Libraries and builds	46
Publish a build to Development	47
Publishing	47
Publishing Process	48
Libraries	48
Exercise 8 - Create a library 2 Mins.....	49
Development Exercise 1 Min	50
Submitted Exercise 1 Minute	51
Approved Exercise 1 Minute.....	52
Published Exercise 1 min.....	53
Add to a library	55
Remove from a library	55
Manage library changes	56
Active Library	56
Builds	56
Promote to production	57
Approval Workflow	57
Permissions	57
Library States	57
Development	58
Submitted	58
Approved	58
Published	58

Adobe LAUNCH - Cheat Sheet —

Information scraped from many sources and Edited – but not authored by Elsa A. Wright

ADOBE LAUNCH - INTRODUCTION

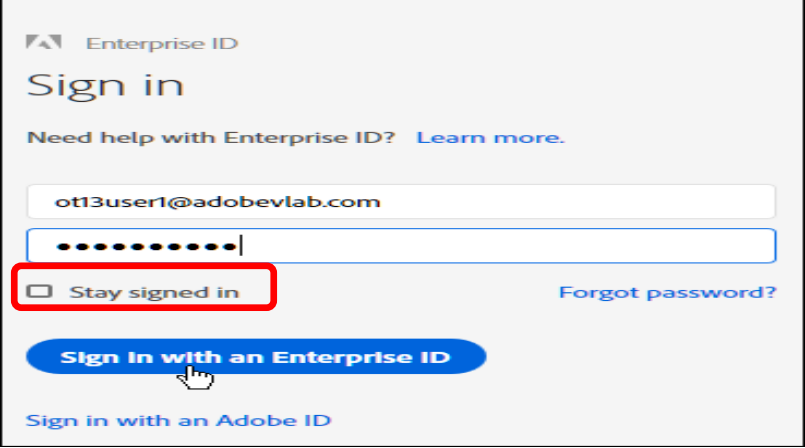
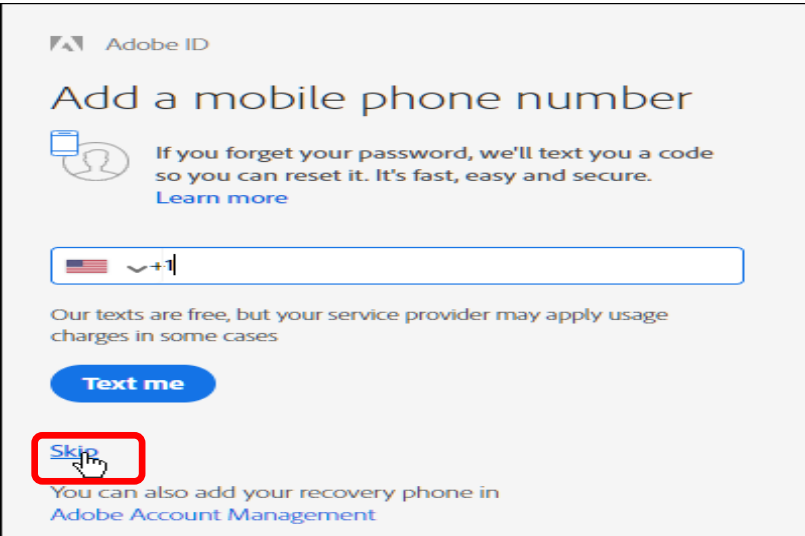
Launch is the next-generation of Adobe's tag management technology, built on the Adobe Cloud Platform. It is built from the ground up to support an open and sustainable ecosystem where anyone can build their own integrations that Adobe customers can deploy to their sites. It is an API-first application, so anything you can do through the UI you can also do programmatically through an API.

The basic Launch workflow:

1. Log in.
 2. Create a Property. (Adapter and Environment)
 3. Install Extensions.
 4. Create Data Elements and Rules.
 5. Test in your Development Environment.
 6. Promote (Publish) to Production.
-

Exercise 1 - Log in 5 Mins

- 1.1 <http://launch.adobe.com>
(It's recommend to use Incognito Window in Chrome)
- 1.2 Sign in with Experience Cloud
ID: **ot13user1@adobeqlab.com** through **ot13user20@adobeqlab.com**, with password **conversion**
- 1.3 Deselect Stay signed in
- 1.4 Click on **Skip** if the Phone # screen appears.
Ignore the addition of a mobile number (click **SKIP**)

<p>1.1 http://launch.adobe.com (It's recommend to use Incognito Window in Chrome)</p> <p>1.2 Sign in with Experience Cloud ID: ot13user1@adobeqlab.com through ot13user20@adobeqlab.com, with password conversion</p> <p>1.3 Deselect Stay signed in</p>	
<p>1.4 Click on Skip if the Phone # screen appears. Ignore the addition of a mobile number</p>	

1. Login via in the **Experience Cloud** --> <https://marketing.adobe.com>
2. Go to → the Activation page
3. Click on **Launch**.

Part 1 - Create a Property

Once you're in Launch, the first thing you'll want to do is create a property.

A property is basically a container that you fill with **extensions, rules, data elements, and libraries** as you deploy tags to your site. Many people create a property for each website (or group of closely related sites) where they want to deploy the same set of tags.

Preliminaries - Best Practices for Planning Properties

Each implementation can be very different in Launch, with a wide variety of data-collection needs, variable usage, extensions, third-party tags, other systems and technologies, people, teams, geographic regions, and so on. Using the flexible User Management features and properties, you can create a configuration that matches your workflow and processes.

If the scenarios you are tracking, the data you are collecting, the extensions you are deploying, and the variables you are setting are similar across all or some of your domains and subdomains, it is easier to have those domains and subdomains grouped into the same property in Launch. If those are unique for each domain and subdomain, it is easier to have those domains and subdomains in their own property. If you choose to group domains and subdomains in a single property now, you can always change your mind and later create several web properties.

The Similarity vs Dissimilarity Factors to Consider When Planning Properties

Data

- For all of your websites, is the data you are going to collect very similar, somewhat similar, or unique?
- If the data you need to collect is similar across websites, it might make sense to group those sites into one property to avoid duplicating rules or copying rules from one property to another.
- If your data collection needs are unique for each site, it might make sense to separate those sites in their own properties. This method lets you control the data collection more specifically for each site, without using large amounts of conditional logic in custom scripts.
- For example, if 80 percent of the data you are collecting across your sites is the same, or similar, it makes sense to group those sites together into the same property. If the data is unique for each site, it makes sense to put each site into its own property.

Variables

- Similar to data, for all of your websites, are the variables you are going to set in your Analytics and other extensions very similar, somewhat similar, or unique?
- For example, if eVar27 is used for the same source value across all of your websites, it might make sense to group those sites together so you can set those common variables across your sites in just one property.
- If your variable usage is unique for each site, it might make more sense to separate those sites into their own properties so you can control the variables more specifically for each site without using large amounts of conditional logic in custom scripts.

Extensions, Tags, and Systems

- Are the extensions, tags and systems you are going to deploy through Launch very similar, somewhat similar, or unique?
- If the extensions, tags, and systems you are going to deploy through Launch are very similar across your sites, you might want to include those sites in the same property.
- If you are deploying Adobe Analytics on only one site, and your other extensions and tags are also unique to certain sites, you might want to create separate properties for those sites so you can control those specific extensions in one property.
- For example, If you are deploying Adobe Analytics, Target, and the same 3rd-party tags and extensions across your sites, that is a reason to group sites together.

People

- For the individuals, teams, and organizations that are working in Launch, will they need access to all of your websites, some of them, or just one of your domains or sub-domains?

The User Management features allow you to assign different roles to different people for all of your properties, or on a per-property basis. If someone has sufficient rights, that person can perform administrative actions across all the properties in that Launch

Company. All the other roles can be assigned on a per-property basis. You can even hide a property from certain users (non-admins) by not giving them any role in that property.

So What Exactly is a Property?

A property is a collection of **rules**, **data elements**, configured **extensions**, **environments**, and **libraries**.

There is *only **one publish embed code per property***.

A property **can be any grouping of one or more domains and subdomains**. You can manage and track these assets similarly. For example, suppose that you have multiple websites based on one template, and you want to track the same assets on all of them. You can apply one property to multiple domains.

The left side of the screen shows the companies in your organization. This is particularly useful if you manage multiple accounts. Select a company to see the properties and audit logs for that company.

Each property is shown in the Properties list.

The Properties list shows the following information:

- Property name
- Status

Click a property to see an overview of that property. The overview shows any activity performed on the property. It also lists the **metrics** and **extensions** for the property.

Exercise 2 - Create a property 7 Mins

Let's Create a Property

Note: Only users with sufficient rights can create a property.

2.1 Navigate to your company page, and then click **New Property**.

2.2 Fill in the fields:

Name: username and date (mo/dd/year).

Example: **User3 07/31/18**

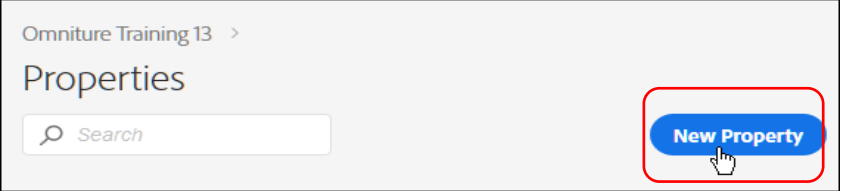
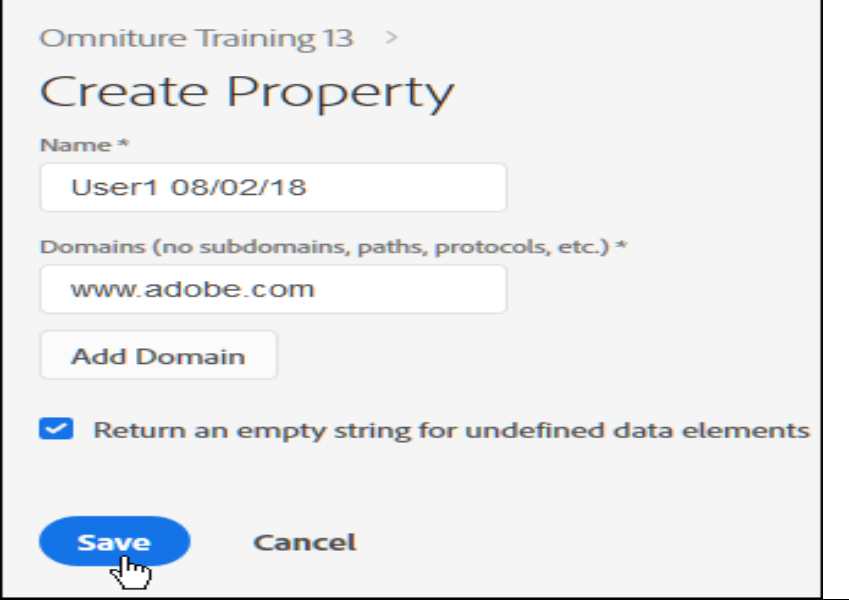
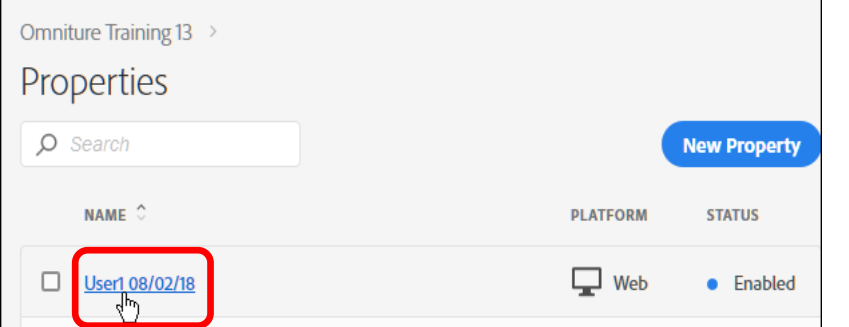
2.2 Domain: www.adobe.com

2.3 Select Return an empty string for undefined data elements

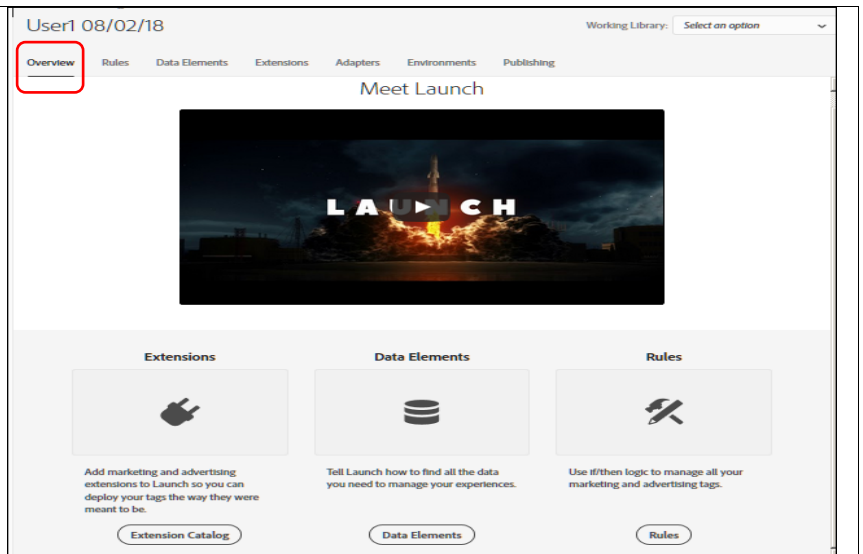
2.4 Click **Save**.

Note: The Core Extension is automatically installed into the new property. You should see your new property in the **Properties** list.

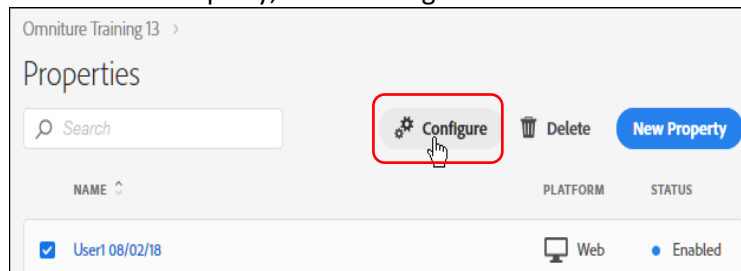
2.6 Click on your new property which will navigate you to the **Overview tab** in the main UI, so you can begin working with it

2.1 Navigate to your company page, and then click New Property .	
2.2 Fill in the fields: Name: username and date (mo/dd/year). Example: User1 08/02/18 2.3 Domain: www.adobe.com 2.4 Select Return an empty string for undefined data elements 2.5 Click Save . Note: The Core Extension is automatically installed into the new property. You should see your new property in the Properties list.	
2.6 Click on your new property	

2.7 Click on the **Overview** tab in the main UI, so you can begin working with it

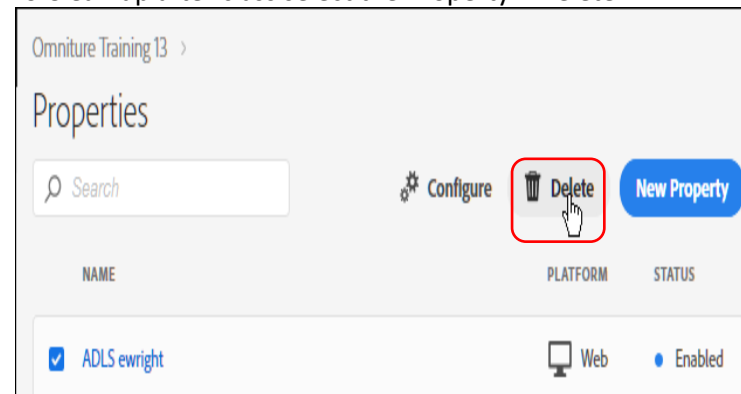


To rename a Property, select Configure



1. In the Properties list, select the property you want to update.
2. Click **Configure**

To Clean up after class select the Property > Delete



[CAUTION: Deleting a Property From Launch]

Note: **Property removal cannot be reversed.**

The requestor must be an admin-level user. This request cannot be undone.

1. In the Properties list, select the property you want to delete. You can select multiple properties to delete.
2. Click **Delete**, and then confirm the removal of the property.

Part 2 – Setting up Adapters (Where)

When you create a build, Launch delivers that build to a location determined by the adapter assigned to the environment. You can choose to have Launch manage that location for you or to manage it yourself.

Benefits of the Launch Adapter

Next you must decide **where you want your embed codes to reside**. If you choose to have **Launch** manage the hosting for your build, **Launch** delivers the build to the Akamai Edge CDN (Content Delivery Network). Akamai is the largest CDN and provides:

- Improved security e.g., via DDos mitigation (Distributed Denial of Service attack is a malicious threat to make an Internet-based service unavailable by overwhelming it with traffic from multiple sources in attempt to cause the network or server to crash).
- Reduced bandwidth cost (via distributed server clustering)
 - the server clusters never *store* your data...only cache it temporarily
- Enhanced data availability via hardware redundancy (RAID-esque)
- Reduced bounce by virtue of faster page loading performance

The embed code references the delivery location of these files so a browser can retrieve the files as needed.

This is the easiest, quickest, way to create an adapter for your environments.

Did you know...(more about Akamai)

Akamai is the primary external hosting option. Akamai provides you with a reliable hosting experience and is the simplest option to implement. Akamai provides the greatest third-party infrastructure dependencies, such as DNS lookup, faster load times, and faster round-trip delivery times.

- Why choose Akamai hosting for library files?
- Can I avoid errors in case of Akamai unavailability?
- Akamai cache control headers

Why choose Hosting By Adobe (Akamai) for library files?

When you choose the Hosting By Adobe (Akamai) option, it means that your Launch library files are served to your visitors from Adobe's Edge CDN, which runs on Akamai architecture.

Akamai is robust when serving content to a global, high-volume audience of Web visitors. Akamai runs redundant networks of load-balanced, geo-optimized nodes to serve content as quickly as possible to visitors wherever they are located throughout the world.

More specifically...

- *Akamai runs more than 137,000 servers in 87 countries within more than 1,150 networks.*
- *In terms of redundancy, Akamai does not just route from one server to another, Akamai routes from one node of servers to another node of servers as-needed. In other words, each node consists of multiple servers for redundancy within a node, so a box going down is not an issue because the other boxes in the node take over.*
- *If a node goes down, Akamai serves from the next closest one, with the same cached content.*
- *Nodes are dynamically selected based on visitor location, traffic load, and other factors so content is consistently served from the best local node for each visitor.*

Can I avoid errors in case of Akamai unavailability?

No. Launch can do nothing from the client side if the library is unavailable. However, it is extremely unlikely that Akamai would be unavailable.

Self-managed adapter

There are a number of reasons to choose to host your own build files.

- Organizations have a number of security and legal requirements that might make cloud-hosting a less-desirable or infeasible option.
- You can reduce the number of required DNS lookups by hosting your own files.
- You might prefer to have more control over Edge locations, caching, and so on.

To meet these requirements, Launch allows you to push your completed builds to an SFTP destination. Launch connects to an SFTP site using an encrypted key. To use this connection type:

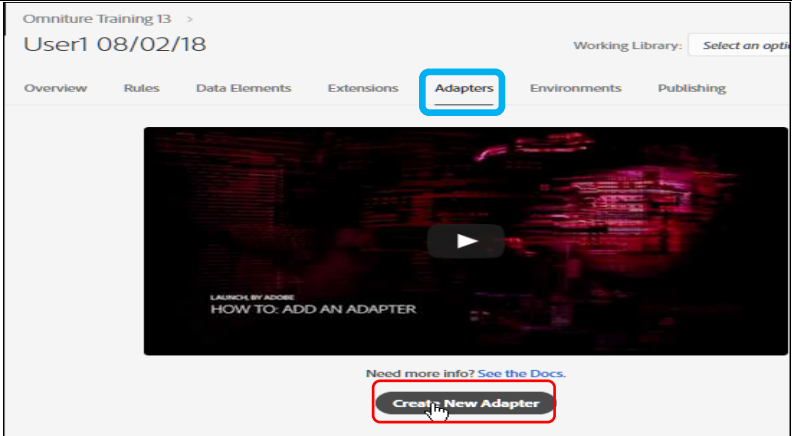
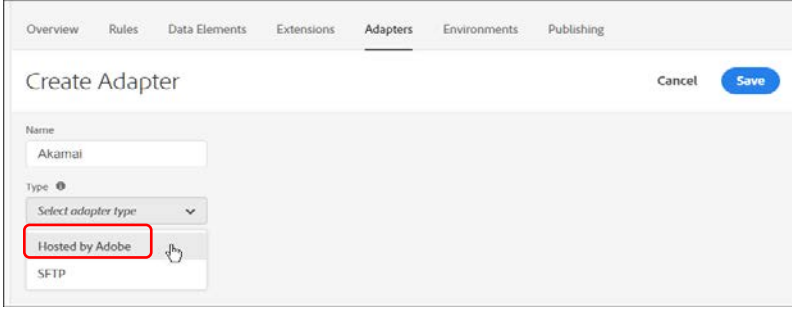
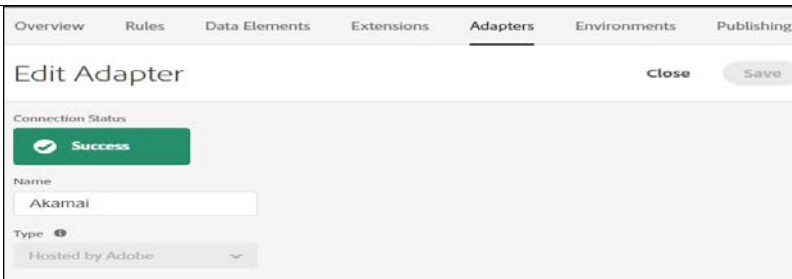
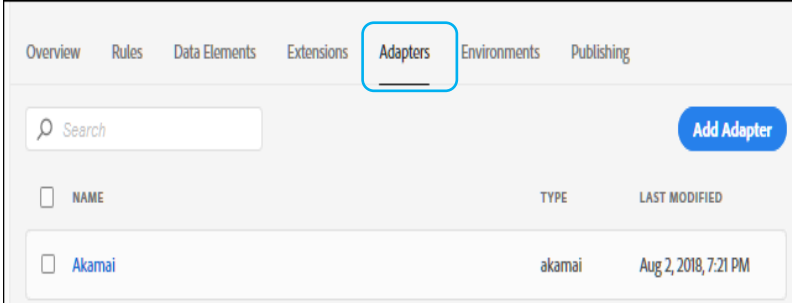
- Your SFTP server must generate a key
- You must provide the username and encrypted private key during the adapter setup process

If you select an SFTP adapter for your environment, there is an additional path variable that you'll need to provide to the environment. This path field is the HTTP path to where your build files are stored relative to your web site. This field is required because the different files in the build reference one another. Launch needs to know where they will be so the files can reference each other properly.

Exercise 3: Adapters (Hosted by Adobe – Akamai) 3 Mins

Create an Akamai Adapter

- 3.1 Click on the Adapters tab
- 3.2 **Create New Adapter.**
- 3.3 **Name** the adapter. Example: Akamai
- 3.4 Select **Hosted by Adobe (Akamai)** as the adapter type.
- 3.5 Click **Save**.

<div>3.1 Click on the Adapters tab</div> <div>3.2 Create New Adapter.</div>	
<div>3.3 Name the adapter. Example: Akamai</div> <div>3.4 Select Hosted by Adobe (Akamai) as the adapter type.</div> <div>3.5 Click Save.</div>	
	
	

Just for Grins.. Exercise: Adapters (SFTP)

1. Open the Adapters tab.
2. Create the new Adapter.
3. Name your adapter (e.g., Self-Hosted).
4. Select SFTP as the adapter type.
5. Enter the host, path, port, sftp username, and encrypted private key.
6. Click **Cancel**.

Bottom line, as you proceed through your Approval and Workflow process promoting code to different environments, Adapters enable you to control exactly where those different files go. Also, you can have multiple SFTP adapters (e.g., Staging and Production) - by simply naming them appropriately.

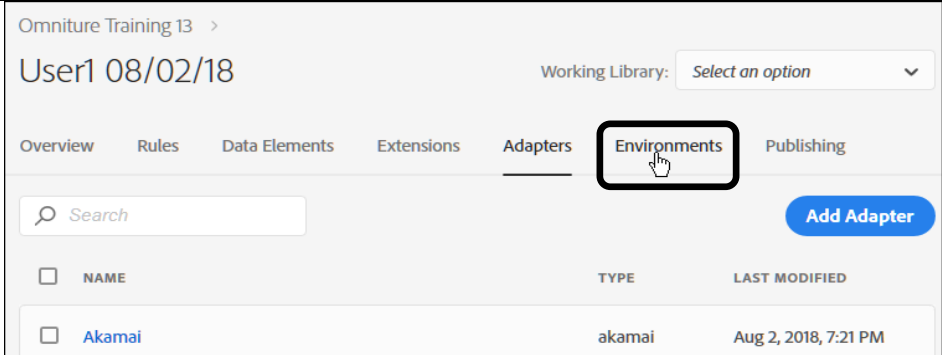
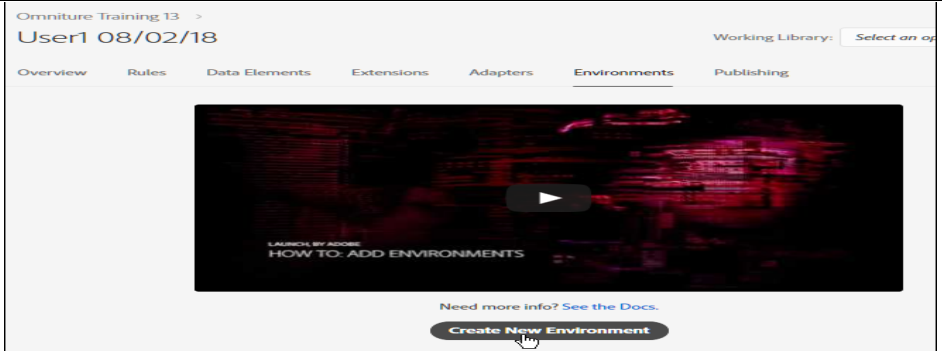
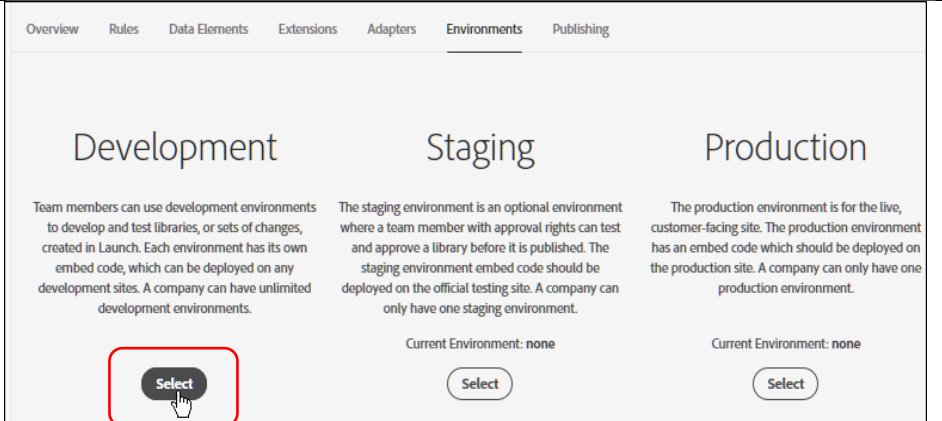
Part 3 - Setting Up Your Environments

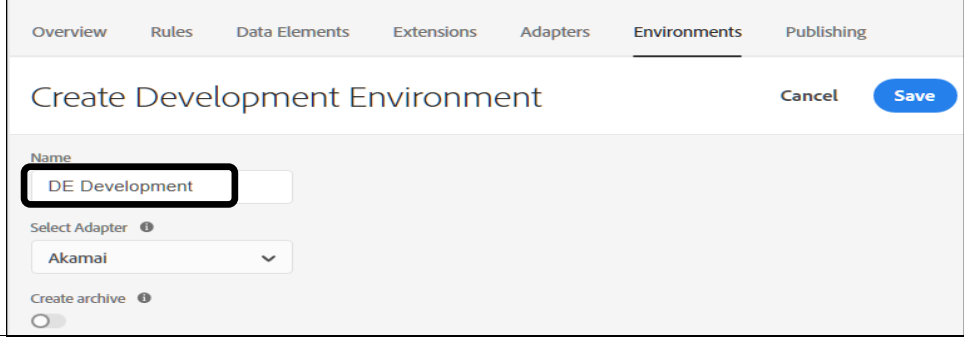
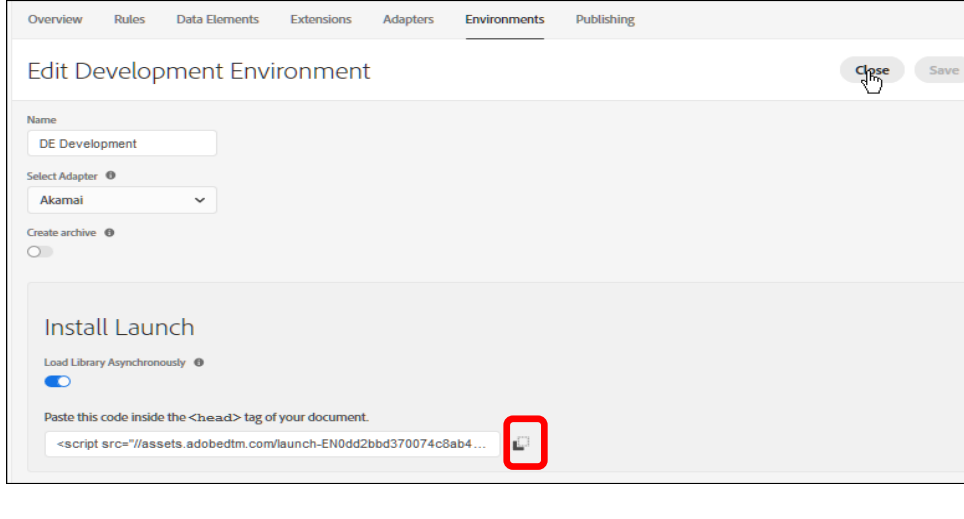
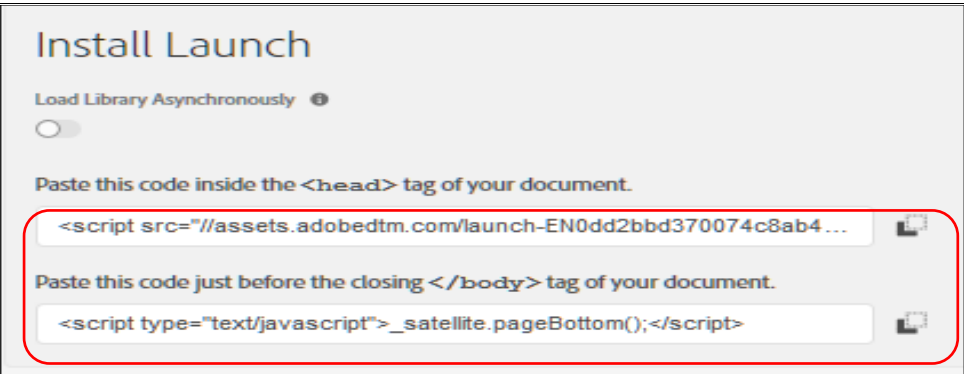

Exercise 4A – Setting up a New Environment (Development) 3 Mins

Setting up Your Environments – Determining **Exactly Where** Your Code Bases Will Reside

1. Click on Environments
2. Click **Create New Environment**
3. Select Production DE PROD
4. Give it a name
5. Select an adapter (Akamai)
6. Don't link to DTM embed code
7. Don't create an archive
8. Select Save
9. Click Close

Notice you snippet code which must be pasted into the Head tag of all pages that are being tracking on your website

4.1 Click on Environments	
4.2 Click Create New Environment	
4.3 Select Development Note: Unlimited number of Development Environments	

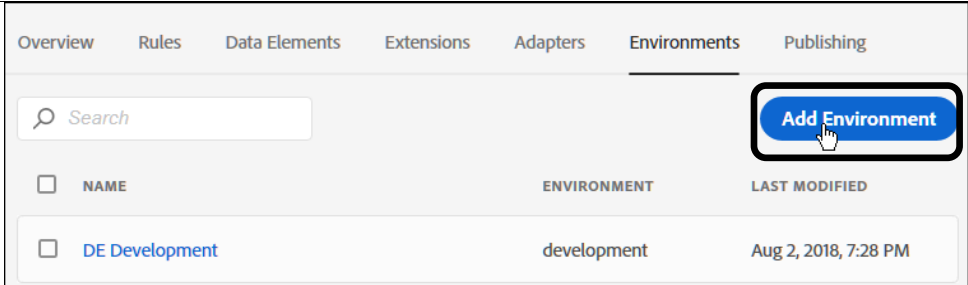
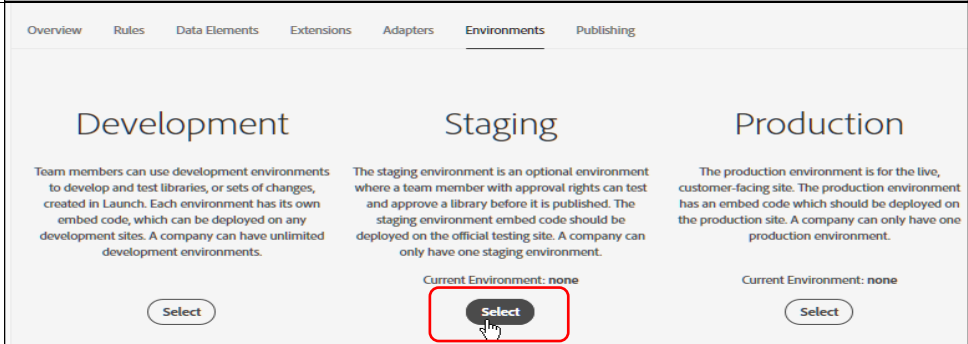
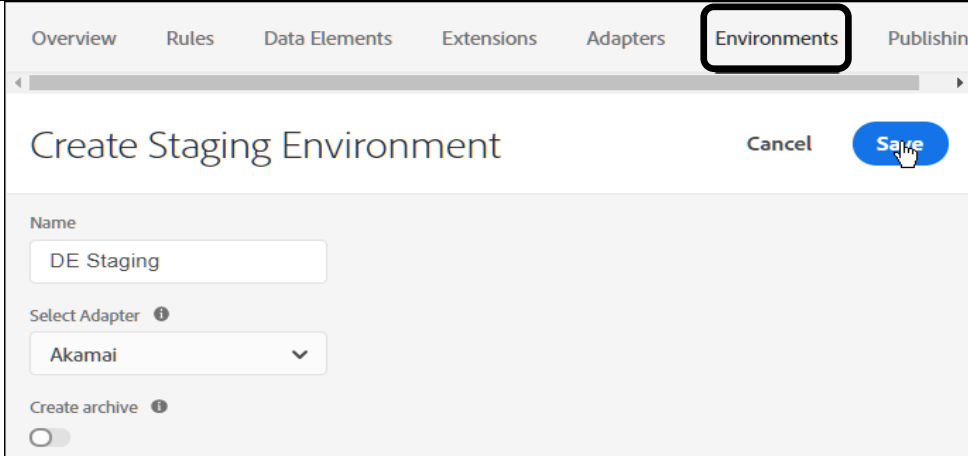
<p>4.4 Give it a name (DE Development)</p> <p>4.5 Select an adapter (Akamai)</p> <p>4.6 Don't create an archive</p> <p>4.7 Select Save</p>	
<p>4.8 Notice you snippet code which must be pasted into the Head tag of all pages that are being tracking on your website</p> <p>4.9 Click Close</p> <p>Note: If you are not doing anything synchronous on the page then you only need header code to access your library. The default is Asynchronously.</p>	
<p>Note: If you are doing something synchronous then you would need both the header & footer code.</p>	
	

Exercise 4B – Setting up a New Environment (Staging) 3 Mins

Setting up Your Environments – Determining **Exactly Where** Your Code Bases Will Reside

1. Click on Environments
2. Click **Add Environment**
3. Select Staging DE Staging
4. Give it a name
5. Select an adapter (Akamai)
6. Don't create an archive
7. Select Save
8. Click Close

Notice you snippet code which must be pasted into the Head tag of all pages that are being tracking on your website

4.1 Click on Environments	
4.2 Click Add Environment	
4.3 Select Staging Note: Only 1 Staging and 1 Production Environment are allowed.	
4.4 Give it a name (DE Staging)	
4.5 Select an Adapter (Akamai)	
4.6 Don't create an archive	
4.7 Select Save	

4.8 Notice you snippet code which must be pasted into the Head tag of all pages that are being tracking on your website. Click **Close**

Note: If you are not doing anything synchronous on the page then you only need header code to access your library. The default is Asynchronously.

Note: If you are doing something synchronous then you would need both the header & footer code.

Edit Staging Environment

Close

Name
DE Staging

Select Adapter

Akamai

Create archive

☐

Install Launch

Load Library Asynchronously

☒

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

```
<script src="//assets.adobedtm.com/launch-EN46f2c38922414dc684 ...
```

Install Launch

Load Library Asynchronously

☐

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

```
<script src="//assets.adobedtm.com/launch-EN46f2c38922414dc684 ...
```

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

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

Overview Rules Data Elements Extensions Adapters Environments Publishing

Search

Add Environment

☐

NAME

ENVIRONMENT

LAST MODIFIED

☐

DE Staging

staging

Aug 2, 2018, 7:36 PM

☐

DE Development

development

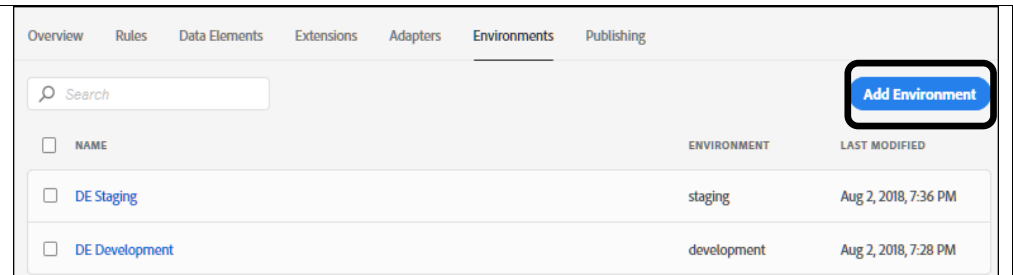
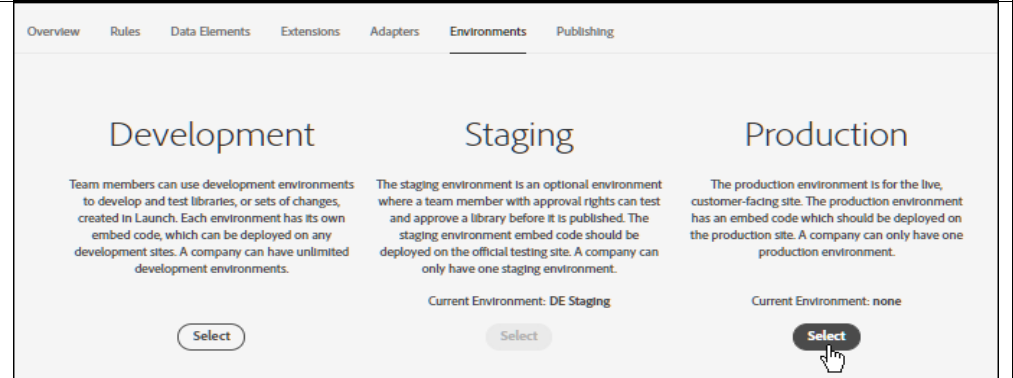
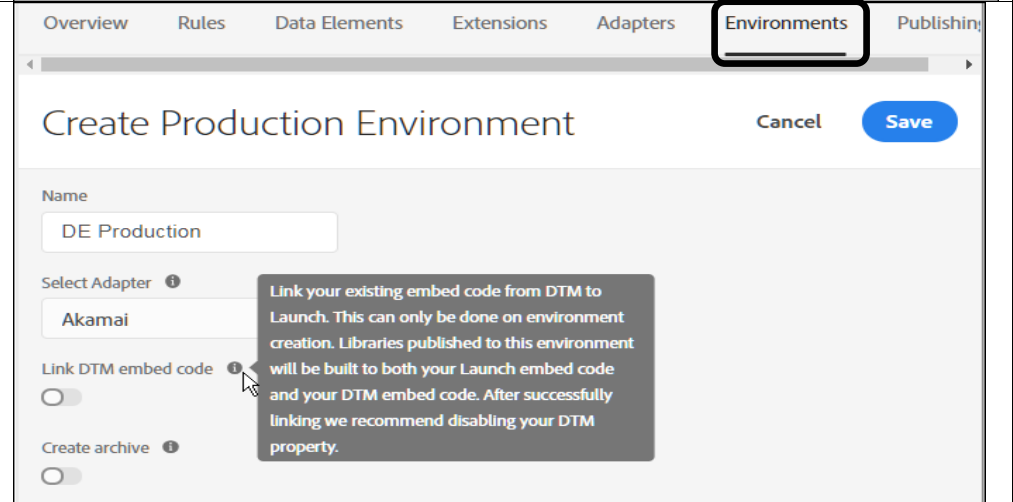
Aug 2, 2018, 7:28 PM

Exercise 4C – Setting up a New Environment (Production) 2 Mins

Setting up Your Environments – Determining **Exactly Where** Your Code Bases Will Reside

1. Click on Environments
2. Click **Add Environment**
3. Select Production DE Production
4. Give it a name
5. Select an adapter (Akamai)
6. Don't link to DTM embed code
7. Don't create an archive
8. Select Save
9. Click Close

Notice you snippet code which must be pasted into the Head tag of all pages that are being tracking on your website

<p>4.1 Click on Environments</p> <p>4.2 Click Add Environment</p>	 <p>The screenshot shows the 'Environments' tab in the top navigation bar. Below the navigation bar is a search bar and a table with columns: NAME, ENVIRONMENT, and LAST MODIFIED. The table lists two environments: 'DE Staging' (staging, Aug 2, 2018, 7:36 PM) and 'DE Development' (development, Aug 2, 2018, 7:28 PM). A red box highlights the 'Add Environment' button in the top right corner.</p>
<p>4.3 Select Production</p> <p>Note: Only 1 Staging and 1 Production Environment are allowed.</p>	 <p>The screenshot shows the 'Environment Selection' page with three options: Development, Staging, and Production. Each option has a description and a 'Select' button. The 'Production' button is highlighted with a red box. Below the buttons, it says 'Current Environment: DE Staging' and 'Current Environment: none'.</p>
<p>4.4 Give it a name (DE Production)</p> <p>4.5 Select an adapter (Akamai)</p> <p>4.6 Don't link to DTM embed code</p> <p>4.7 Don't create an archive</p> <p>4.8 Select Save</p>	 <p>The screenshot shows the 'Create Production Environment' form. The 'Name' field is 'DE Production'. The 'Select Adapter' dropdown is set to 'Akamai'. The 'Link DTM embed code' and 'Create archive' checkboxes are disabled. A red box highlights the 'Save' button. A tooltip is visible over the 'Link DTM embed code' checkbox, stating: 'Link your existing embed code from DTM to Launch. This can only be done on environment creation. Libraries published to this environment will be built to both your Launch embed code and your DTM embed code. After successfully linking we recommend disabling your DTM property.'</p>

4.9 Notice you snippet code which must be pasted into the Head tag of all pages that are being tracking on your website. Click **Close**.

Note: If you are not doing anything synchronous on the page then you only need header code to access your library. The default is Asynchronously.

Note: If you are doing something synchronous then you would need both the header & footer code.

In the next Exercise we will add two Extensions.

Edit Production Environment

Close **Save**

Name
DE Production

Select Adapter
Akamai

Create archive
☐

Install Launch

Load Library Asynchronously ☒

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

```
<script src="//assets.adobedtm.com/launch-EN0c68cd6...">
```

Install Launch

Load Library Asynchronously ☐

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

```
<script src="//assets.adobedtm.com/l...">
```

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

```
<script type="text/javascript">_satelli...
```

Overview Rules Data Elements Extensions Adapters Environments Publishing

Search

Add Environment

<input type="checkbox"/>	NAME	ENVIRONMENT	LAST MODIFIED
<input type="checkbox"/>	DE Production	production	Aug 2, 2018, 7:46 PM
<input type="checkbox"/>	DE Staging	staging	Aug 2, 2018, 7:36 PM
<input type="checkbox"/>	DE Development	development	Aug 2, 2018, 7:28 PM

Part 4 - Install Extensions (formerly aka DTM Tools)

Extensions add core functionality to Launch, and are one of its core features. An extension is an integration built by Adobe (or an Adobe partner) that adds new and potentially endless options for the tags that can be deployed to your websites.

All new properties come with the Core extension installed. This extension is built by the Launch team to provide a robust default set of data element types for your data layer and event types for your rules. Most actions you will want to perform (get an ECID, send Adobe Analytics beacons, load the Target global mbox, etc) will come from extensions that you install from the catalog.

But what makes Launch unique among tag management systems is that these extensions can be built by anyone.

For example:

- If you need to drop a Facebook remarketing pixel on your site - check out the extension that Facebook built.
- If you want the same for Twitter or Linked In - use those extensions.
- If you need to run a survey - consider Question Pro or Foresee.
- If you need to manage privacy and consent from your end users to comply with GDPR - consider Evidon and Trust Arc.
- If you need more granular insight into the behavior of individual users on your site - consider Clicktale.

Exercise 5: Add a New Extension (Core)

The **Core Extension** is the starting point for your new extension and provides:

- Default **Event**
- Default **Conditions** and **Exceptions**
- Default **JavaScript**

These defaults are the basis for the **custom rules** you'll build to create your extension.

When creating or editing elements, you can save and build to your **active library**. **This immediately saves your change to your library and executes a build.** The status of the build is displayed. You can also create a new library from the Active Library drop down.

Configure an extension

Mouse over an installed extension and click Configure.

Note: Some extensions do not require configuration and do not offer configuration options.

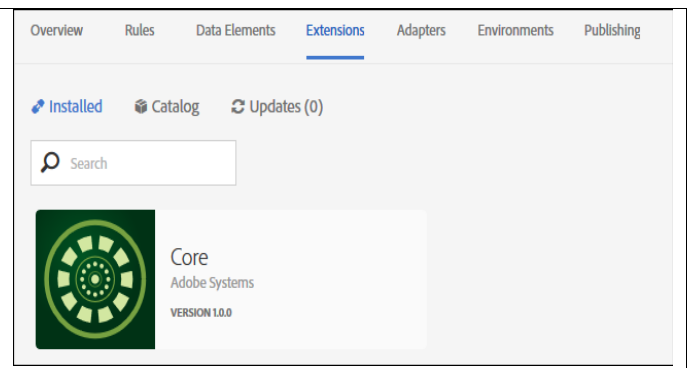
Each configurable extension has unique options. Refer to [Extensions Reference](#) online.

Add a new extension

A common use of extensions is to create integrations with other applications. In the previous version of Launch, known as Dynamic Tag Management, extensions were called *tools*.




- 5.1 From a property's overview page, open the **Extensions** tab.
- 5.2 Notice that the default Adobe Core extension has been already added.
- 5.3 Select an extension.
- 5.4 If the extension exists, select it from the extensions catalog.
Mouse over an extension in your list to configure or disable it.
- 5.5 Add other extensions from the catalog if they are not currently in your list.

- 5.1 From a property's overview page, open the **Extensions** tab.
- 5.2 Notice that the default Adobe Core extension has been already added.
- 5.3 Select an Extension.
- 5.4 If the extension exists, select it from the extensions catalog.
Mouse over an extension in your list to configure or disable it.
- 5.5 Add other extensions from the catalog if they are not currently in your list.



List of Valid Report Suite IDs

For example: outrainjjscit25site03 outrainjjscit25global

 Edit Settings ▾	 Create New ▾
 Report Suite Id ▾	Site Title ▾
outrainjjscit25global	JJ SCIT Global
outrainjjscit25site01	JJ SCIT 01
outrainjjscit25site02	JJ SCIT 02
outrainjjscit25site03	JJ SCIT 03
outrainjjscit25site04	JJ SCIT 04
outrainjjscit25site05	JJ SCIT 05
outrainjjscit25site06	JJ SCIT 06
outrainjjscit25site07	JJ SCIT 07
outrainjjscit25site08	JJ SCIT 08
outrainjjscit25site09	JJ SCIT 09
outrainjjscit25site10	JJ SCIT 10
outrainjjscit25site11	JJ SCIT 11
outrainjjscit25site12	JJ SCIT 12

For example: outrainjjscit13site11 outrainjjscit13global

[Edit Settings](#)

[Create New](#)

Report Suite Id	Site Title	Report Suite Id
clicktest	Activity Map (Demo)	clicktest
geo1xxpnwb2bdemo	we.B2B	geo1xxpnwb2bdemo
geo1xxpnwfsidemo	we.FINANCE	geo1xxpnwfsidemo
geo1xxpnwmediademo	we.MEDIA	geo1xxpnwmediademo
geo1xxpnwretaildemo	we.RETAIL	geo1xxpnwretaildemo
geomaster1	Geometrix Retail	geomaster1
jjesquirejjesquiretrain	Geometrix Training	jjesquirejjesquiretrain
outrainjj13	JJ Esquire Training 13	outrainjj13
outrainjjscit13global	JJ SCIT Global	outrainjjscit13global
outrainjjscit13site11	JJ SCIT 11	outrainjjscit13site11
sctestscriptdtmdemo	DTM Demo	sctestscriptdtmdemo
sctestscriptrolluptest	RollupTest	sctestscriptrolluptest
sctestscriptrptest	Geo Practice Site	sctestscriptrptest
sctestscriptscarrt	Scarr Training	sctestscriptscarrt
vlab1usweb2b	we.B2B	vlab1usweb2b
vlab1uswefinance	we.Finance	vlab1uswefinance
vlab1uswehealth	we.Health	vlab1uswehealth
vlab1uswemedia	we.Media	vlab1uswemedia
vlab1usweretail	we.Retail	vlab1usweretail

Tracking Server

dcrookston.d1.sc.omtrdc.net

Adobe Analytics Extension

Use this reference for information about configuring the Adobe Analytics extension, and the options available when using this extension to build a rule.

Configure the Adobe Analytics extension

This section provides a reference for the options available when configuring the Adobe Analytics extension.

If the Adobe Analytics extension is not yet installed, open your property, then click Extensions > Catalog, hover over the Adobe Analytics extension, and click Install.

To configure the extension, open the Extensions tab, hover over the extension, and then click Configure.

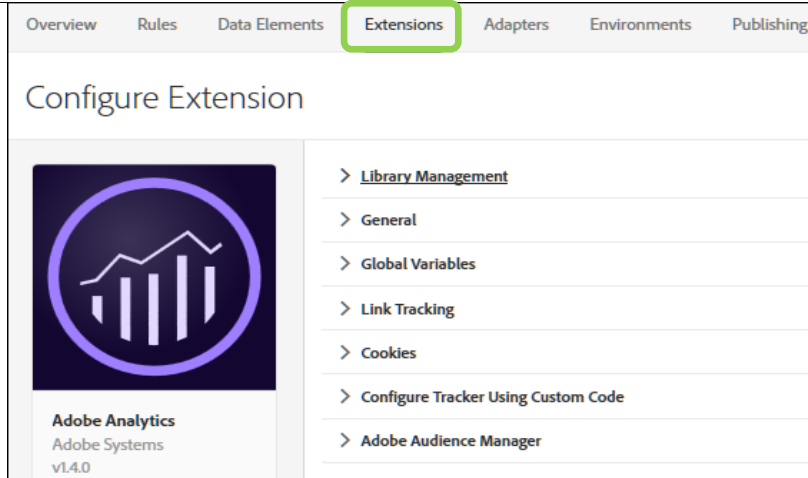
Library Management

Select an option from the Library Management section of the configuration page. The following configuration options are available:

<p>Option 1 - Manage the library for me Report Suites Specify one or more report suites for each of the following environments:</p> <ul style="list-style-type: none">• Development• Staging• Production	<div><p>Library Management</p><p><input checked="" type="radio"/> Manage the library for me</p><p>Report Suites ⓘ</p><div><p>Development Report Suites ⓘ</p><div><input type="text" value="outrainjscit25site03"/> ⓘ X</div><div><input type="text" value="outrainjscit25global"/> ⓘ X</div><div>Add Another</div></div><div><p>Staging Report Suites ⓘ</p><div><input type="text" value="outrainjscit25site03"/> ⓘ X</div><div><input type="text" value="outrainjscit25global"/> ⓘ X</div><div>Add Another</div></div><div><p>Production Report Suites ⓘ</p><div><input type="text" value="outrainjscit25site03"/> ⓘ X</div><div><input type="text" value="outrainjscit25global"/> ⓘ X</div><div>Add Another</div></div></div> <p><input type="checkbox"/> Make tracker globally accessible ⓘ</p>	
---	--	--

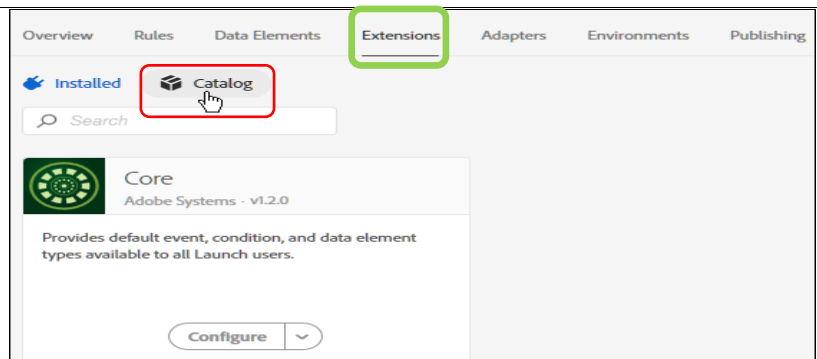
Exercise 5A - Adobe Analytics Extension 4 Mins

An Adobe Analytics Extension has many components.

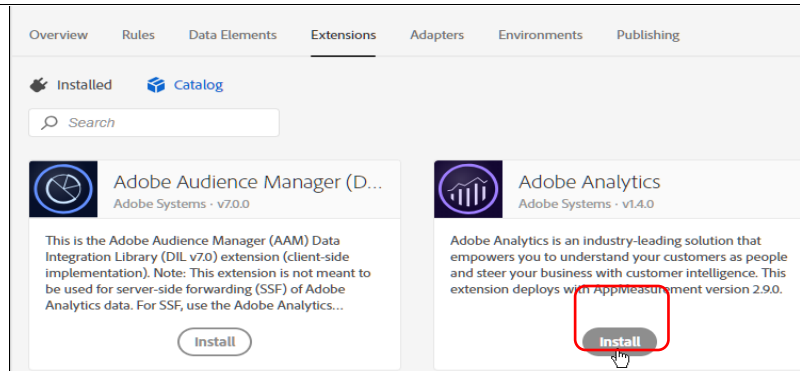


5.1 From a property's overview page, open the **Extensions** tab.

5.2 Click Extensions > Catalog



5.3 Hover over the **Adobe Analytics** extension, and click **Install**.



5.4 Open **Library Management** > Type the name of the Report Suite for Development, Staging and Production.

outrainjscit13site01

Click Add Another
outrainjscit13global

Note: Add Another for multi-suite tagging.

The screenshot shows the 'Library Management' section. At the top, there's a toggle for 'Manage the library for me'. Below it, the 'Report Suites' section is divided into three columns: 'Development Report Suites', 'Staging Report Suites', and 'Production Report Suites'. Each column contains two text input fields with the names 'outrainjscit13site01' and 'outrainjscit13global', each followed by a trash icon and an 'X' icon. Below these fields are 'Add Another' buttons. At the bottom, there is a checkbox labeled 'Make tracker globally accessible'.

5.5 Under **General** >
Select Preset > UTF-8
Currency Code > USD – United States Dollar
Enter the name of the **Tracking Server** and
SSL Tracking Server as
dcrookston.d1.sc.omtrdc.net

Note: Character Set and Currency Code

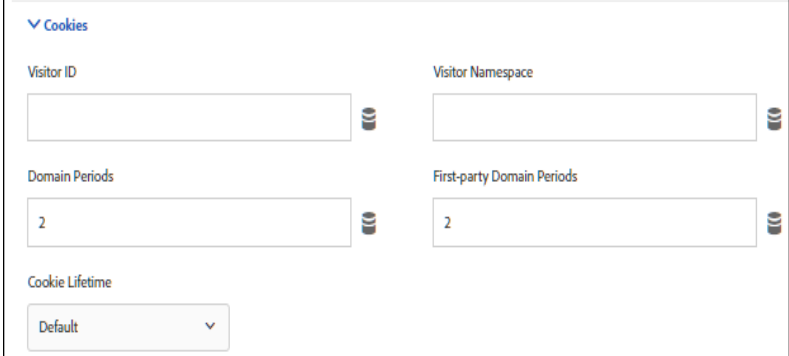
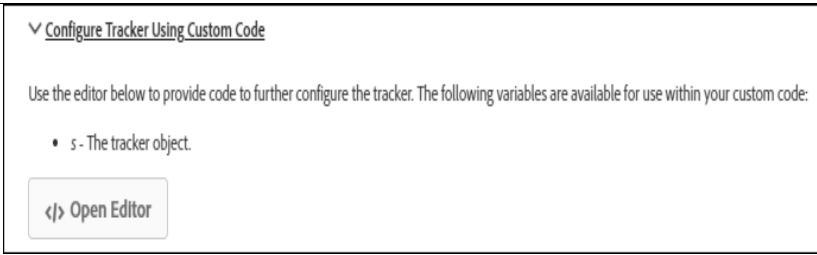
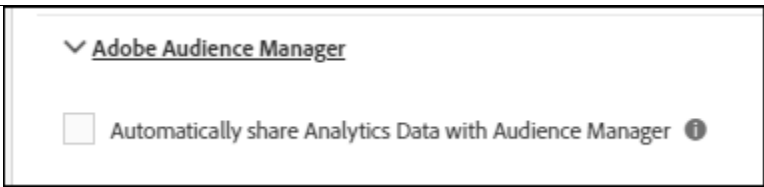
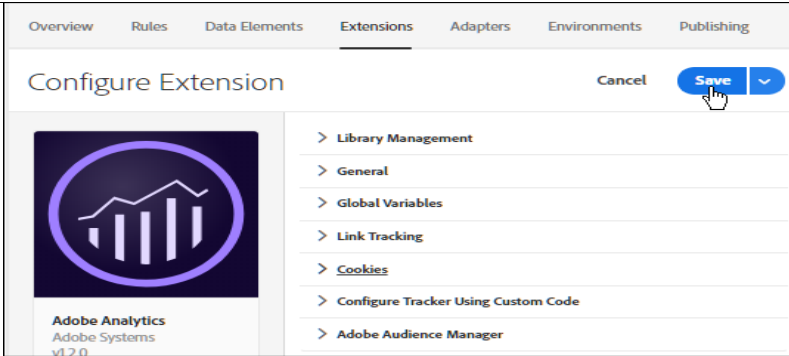
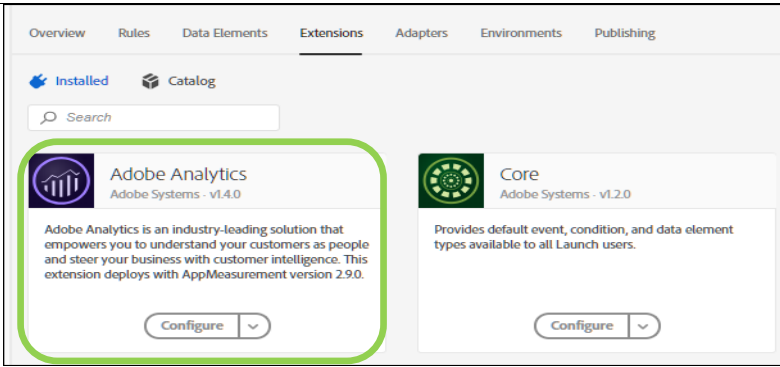
The screenshot shows the 'General' configuration section. It includes a checkbox for 'Enable EU compliance for Adobe Analytics'. Below this, there are two main sections: 'Character Set' and 'Currency Code'. Both have radio buttons for 'Preset' and 'Custom'. The 'Preset' options are 'UTF-8' for Character Set and 'USD - United States Dollar' for Currency Code. Below these, there are text input fields for 'Tracking Server' and 'SSL Tracking Server', both containing the value 'dcrookston.d1.sc.omtrdc.net'.

5.6 **Link Tracking** > Enable ClickMap

5.7 Select **Track download links**

5.8 Select **Track Outbound Links**

The screenshot shows the 'Link Tracking' configuration section. It starts with a checked checkbox for 'Enable ClickMap'. Below this is the 'Downloads' section, which includes a checked checkbox for 'Track download links' and a 'Download Extensions' section with a grid of file type buttons (doc, docx, eps, jpg, png, svg, xls, ppt, pptx, pdf, xlsx, tab, csv, zip, txt, vsd, vxd, xml, js, css, rar, exe, wma, mov, avi, wmv, mp3, wav, m4v). Each button has an 'X' icon. To the right of the grid is a 'Save' button. Below the Downloads section is the 'Outbound Links' section, which includes a checked checkbox for 'Track outbound links'. It has two text input fields: 'Track' and 'Never Track', each with a trash icon and a 'Save' button. At the bottom, there is a checkbox for 'Keep URL Parameters'.

<p>5.9 Cookies > Domain Periods (2) and First Party Domain Period (2)</p>		
		
		
<p>5.9 Save</p>		
		

Exercise 5B - Experience Cloud ID Service Extension 2 Mins

Use this reference for information about configuring the Experience Cloud ID extension, and the options available when using this extension to build a rule.

Use this extension to integrate the Experience Cloud ID Service with your property. With the Experience Cloud ID Service, you can create and store unique and persistent identifiers for your site visitors.

Configure the Experience Cloud ID extension

This section provides a reference for the options available when configuring the Experience Cloud ID extension.

If the Experience Cloud ID extension is not yet installed

5.B1 Open your property, then click Extensions > Catalog

5.B2 Hover over the Experience Cloud ID extension

5.B3 Click Install.

To configure the extension, open the Extensions tab, hover over the extension, and then click Configure.

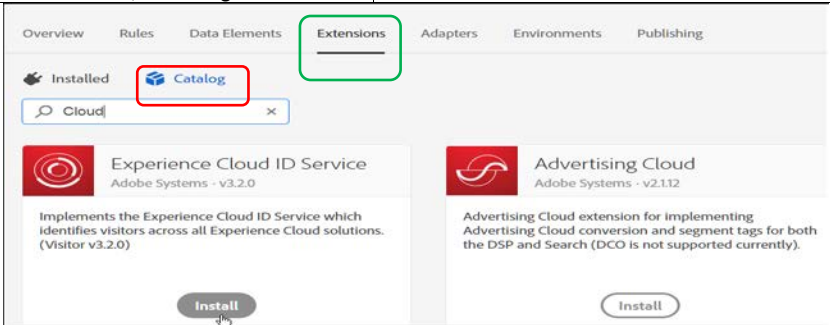
http://ats-sb.adobe.com/j/site/include/downloads/whitepapers/marketing_orgs.txt

Dave Post Omniture Training 6 0F6D57EB548B254F0A4C98A5@AdobeOrg	Elsa Wright Omniture Training 13 D8A9C8A1548A2C630A4C98BC@AdobeOrg	Dwight Smith Omniture Training 18 12C2C703548B1BD90A4C98A1@AdobeOrg
Alex Clark Omniture Training 19 220FC8A1548B68C60A4C98BC@AdobeOrg	Tamika Gordon Omniture Training 20 D68B525A548A299D0A4C98C6@AdobeOrg	Chris Valentine Omniture Training 21 1BD7D441548B6C490A4C98A2@AdobeOrg
Staci Hunter Omniture Training 24 23392CA7548B6B7E0A4C98A7@AdobeOrg	Enrico Ciarna Omniture Training 25 1A051042548B27540A4C98A4@AdobeOrg	Emily South Omniture Training 29 1834C8A1548B3D9F0A4C98BC@AdobeOrg

5B1. Open your property, then click Extensions > Click on **Catalog** and search for Cloud

5B2. Hover over the **Experience Cloud ID** extension

5B3. Click **Install**.



The following configuration options are available:

Experience Cloud Organization ID

The ID for your Experience Cloud Organization. Your ID is a 24-character, alphanumeric string followed by @AdobeOrg. If you do not know this ID, contact Customer Care. This should be auto populated with the following value

D8A9C8A1548A2C630A4C98BC@AdobeOrg

Exclude specific paths

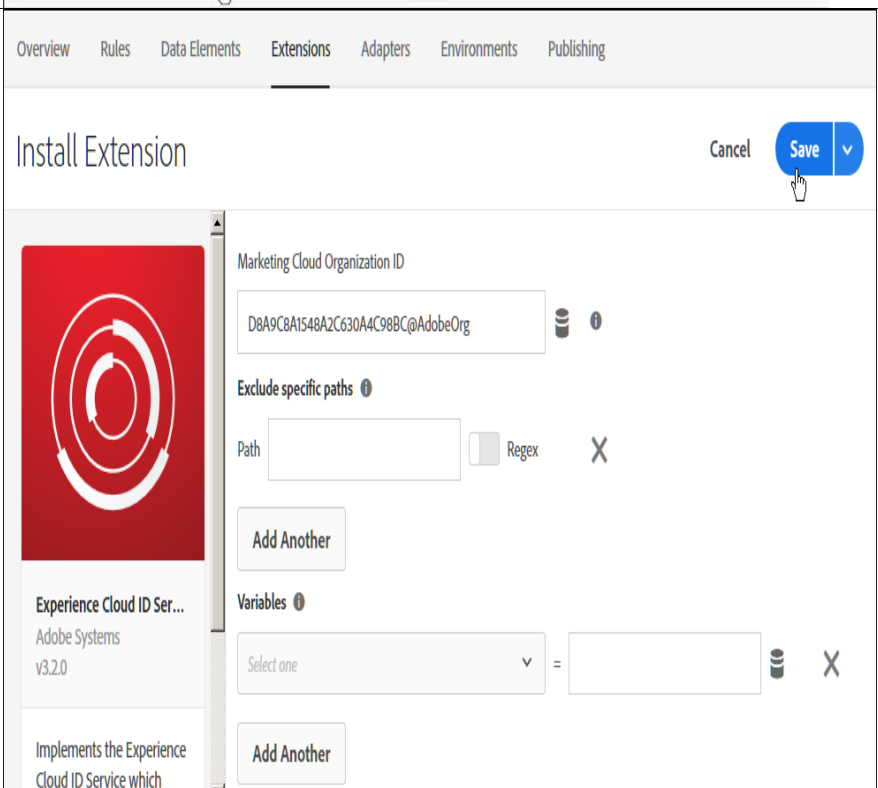
The Experience Cloud ID does not load if the URL matches any of the specified paths. (Optional) Enable Regex if this is a regular expression.

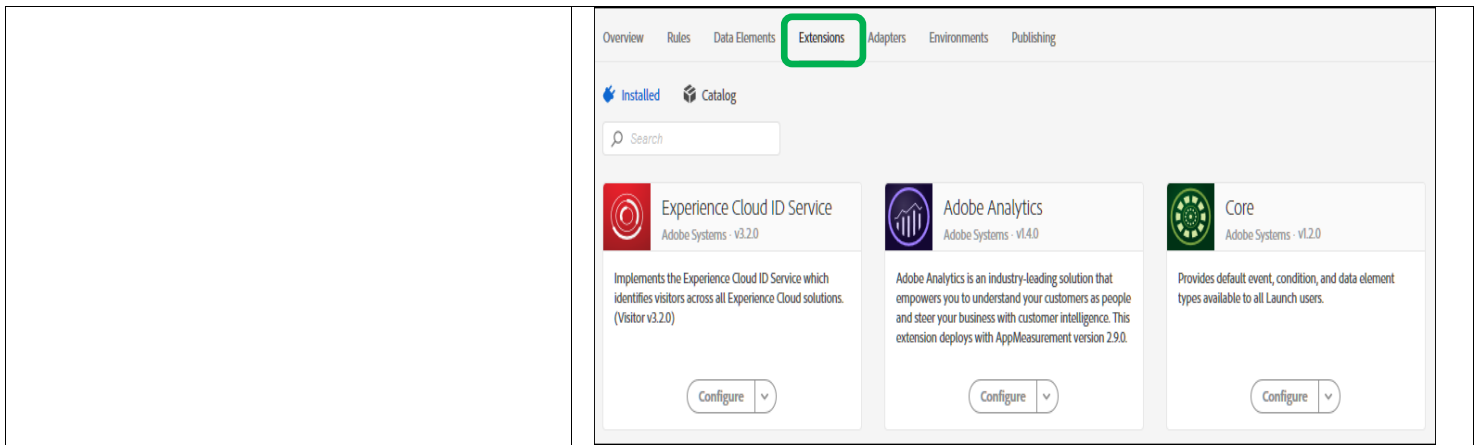
Variables

Set name-value pairs as Experience Cloud ID instance properties. Use the drop-down to select a variable, then type or select a value. For information about each variable, refer to the [Experience Cloud ID Service documentation](#).

Experience Cloud ID extension action types

This section describes the action types available in the Experience Cloud ID extension.





Open Editor

Enables you to insert core AppMeasurement code. This code is populated automatically when using the **automatic configuration method**.

Note: The validator used in the Launch code editor is designed to identify issues with developer-written code. Code that has gone through a minification process--such as the AppMeasurement.js code downloaded from the Code Manager--might be falsely flagged as having issues by the Launch validator, which can usually be ignored.

Set the following report suites on tracker

If you select this option, specify one or more report suites for each of the following environments:

- Development
- Staging
- Production

Tracker is accessible on the global variable named

Specify a global variable.

Part 5 - Data Elements

Data elements are pointers to the information that you want to collect and send to different places on your page:

- A defined data layer in JSON
- DOM elements
- Cookies
- Session and local storage
- Just about everything else

Once defined in a data element, you can use the element anywhere throughout Launch for any extension. (See [Data Elements](#).)

Data elements are the building blocks for your data dictionary (or data map). Use data elements to collect, organize, and deliver data across marketing and ad technology.

A single data element is a variable whose value can be mapped to query strings, URLs, cookie values, JavaScript variables, and so on. You can reference this value by its variable name throughout Launch. This collection of data elements becomes the dictionary of defined data that you can use to build your rules (events, conditions, and actions). This data dictionary is shared across all of Launch for use with any extension you've added to your property.

Important: Changes do not take effect until they are published.

Use data elements as widely as possible throughout rule creation to consolidate the definition of dynamic data and to improve the efficiency of your tagging process. You define data rules once and then use them in multiple places.

The concept of reusable data elements is very powerful and you should use them as best practice.

Data elements are populated with data when they are used in rules or when manually called in a script. At a high level, you:

1. [Create a data element](#), if you haven't done so already.
2. Use the data element in a [rule](#) or a custom script.

For an introductory video, see [Data elements](#).

Data element usage

In Rules

You can use data elements in the rule editing interface by using the search box to find the name of your data element.

In Custom Script

You can use data elements in custom scripts by using the `_satellite` object syntax:

```
_satellite.getVar('data element name');
```

Types of data elements

Data Element types are determined by the extension. There is no limit to the types that can be created.

The following sections describe the types of data elements available in the Core extension. Other extensions use other types of data elements.

JavaScript Variable

Any available JavaScript object or variable can be referenced **using the path field**.

When you have JavaScript variables, or object properties in your markup, and you want to collect those values in Launch to use with any of your extensions or rules, one way to capture those values is to use Data Elements in Launch. This way, you can refer to the Data Element throughout your Rules, and if the source of the data ever changes, you only need to change your reference to the source (the Data Element) in one place in Launch.

For example, let's say your markup contains a JavaScript variable called `Page_Name`, like this:

```
<script>
  //data layer
  var Page_Name = "Homepage"
</script>
```

When you create the Data Element in Launch, simply provide the path to that variable.

If you use a data collector object as part of your data layer, simply use dot notation in the Path to reference the object and property you want to capture into the Data Element, like `_myData.pageName`, or `digitalData.pageName`, etc.

Example:

```
window.document.title
```

DOM Attribute

Any element value can be retrieved, such as a div or H1 tag.

Example:

```
CSS Selector Chain:
id#dc logo img
Get the value of:
src
```

Cookie

Any available domain cookie can be referenced in the cookie name field.

Example:

```
cookieName
```

URL Querystring Parameter

Specify a single URL parameter in the URL Parameter field.

Only the NAME parameter is necessary, and any special designators like "?" or "=" should be omitted.

Example:

```
contentType
```

Custom Script

Custom JavaScript can be entered into the UI by clicking Open Editor and inserting code into the editor window.

A return statement is necessary in the editor window in order to indicate what value should be set as the data element value. If a return statement is not included, the default value or an empty string will be returned as the data element value.

Example:

```
var pageType = $('div.page-wrapper').attr('class').split(")[1];  
if (window.location.pathname === '/') {  
  return 'homepage';  
} else {  
  return pageType;  
}
```

Built-in data elements

If you used any of the following data elements in the past, you must create custom data element in Launch:

- URI
- Protocol
- Hostname

Create a data element

Data elements are the building blocks for rules. Data elements let you create a data dictionary (or data map) of commonly used items on a page, regardless of where they originate (query strings, URLs, or cookie values) for any object that is contained on your site.

1. From a Property page, open the Data Elements tab, then click Create New Data Element.
2. Name the data element.
3. Select an extension and type.
The available data element types are determined by the extension. For information about the types available with the Launch Core extension, refer to [Types of data elements](#).
4. Provide any requested information about the chosen type in the fields provided.
5. (Optional) Enter a default value.
If you do not provide a value, no value is sent. Some people choose to enter something like "none" or "n/a" so they can determine what is sent if there isn't a value. Different solutions deal with an empty variable differently. This creates consistency even if a value doesn't exist.
6. Select whether to force a lowercase value and whether to remove line breaks and spaces.
7. Select duration.

The available choices are:

- None
 - The value is not stored.
- Page view
 - The value is held in a JavaScript variable until the page is refreshed or a new page is loaded.
 - Can be created and set in scripts using `_satellite` object syntax:
`_satellite.setVar('data_element_name')`
- Session
 - Values persist in the browser's session storage until the browser tab is closed.
 - Available throughout the site visit.
- Visitor
 - The value is stored indefinitely in the browser's local storage.

8. Click Save.

When creating or editing elements, you can save and build to your [active library](#). This immediately saves your change to your library and executes a build. The status of the build is displayed. You can also create a new library from the Active Library drop down.

Exercise 6A: Data Element (Java Script Variable – Page Name) 2 Mins

6A1. From a Property page, open the **Data Elements** tab, and then click **Create New Data Element**.

6A2. Name the data element (**Page Name**).

6A3. Select an extension (**Core**) and type (**JavaScript Variable**).

Note: The available data element types are determined by the extension. For information about the types available with the Launch Core extension, refer to [Types of data elements](#).

6A4. Provide any requested information about the chosen type in the fields provided.

Enter the Path to Variable (**digitalData.page.pageName**)

6A5. (Optional) Enter a default value.

If you do not provide a value, no value is sent. Some people choose to enter something like "none" or "n/a" so they can determine what is sent if there isn't a value. Different solutions deal with an empty variable differently. This creates consistency even if a value doesn't exist.

6A6. Select **Force a lowercase value**

6A7. Select **Clean Text** to remove line breaks and spaces.

6A8. Select **Duration**.

The available choices are:

a. None

The value is not stored.

b. Page view

- The value is held in a JavaScript variable until the page is refreshed or a new page is loaded.
- Can be created and set in scripts using `_satellite` object syntax: `_satellite.setVar('data_element_name')`

c. Session

Values persist in the browser's session storage until the browser tab is closed. Available throughout the site visit.

d. Visitor

The value is stored indefinitely in the browser's local storage.

6A9. Click **Save**.

When creating or editing elements, you can save and build to your [active library](#). This immediately saves your change to your library and executes a build. The status of the build is displayed. You can also create a new library from the Active Library drop down.

Important: Changes do not take effect until they are published.

JSON Path Name - `digitalData.page.pageName`

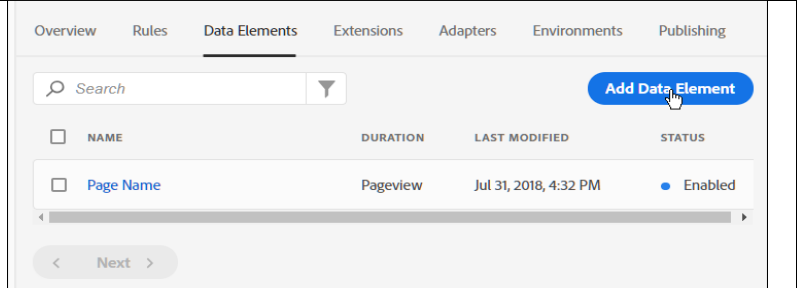
Path to Variable: JSON Path Name (conversion of JSON hierarchy into a dot structure).

The screenshot shows the 'Create New Data Element' dialog box. At the top, there are tabs for Overview, Rules, Data Elements, Extensions, Adapters, Environments, and Publishing. The 'Data Elements' tab is selected. The dialog has a title bar with 'Cancel' and 'Save' buttons. The main form contains the following fields and options:

- Name:** A text input field containing 'Page Name'.
- Path to variable:** A text input field containing 'digitalData.page.pageName'.
- Extension:** A dropdown menu with 'Core' selected.
- Data Element Type:** A dropdown menu with 'JavaScript Variable' selected.
- Default Value:** A text input field with the placeholder 'Enter a Default Value'.
- Force lowercase value:** A checkbox that is checked.
- Clean text:** A checkbox that is checked.
- Storage Duration:** A dropdown menu with 'Pageview' selected.

Exercise 6B: Data Element (URL Query String Parameter – Internal Campaign) 2 Mins

6B1. From a Property page, open the Data Elements tab, then click **Create New Data Element**.



NAME	DURATION	LAST MODIFIED	STATUS
Page Name	Pageview	Jul 31, 2018, 4:32 PM	Enabled

6B2. Name the data element (**Internal Campaign**).

6B3. Select an Extension (**Core**) and Data Element Type (**Query String Parameter**).

Note: The available data element types are determined by the extension. For information about the types available with the Launch Core extension, refer to [Types of data elements](#).

6B4. Provide any requested information about the URL Query String Parameter Name (**intcmp**).

6B5. (Optional) Enter a default value.

If you do not provide a value, no value is sent. Some people choose to enter something like "none" or "n/a" so they can determine what is sent if there isn't a value. Different solutions deal with an empty variable differently. This creates consistency even if a value doesn't exist.

6B6. Select **Force a lowercase value**

6B7. Select **Clean text** to remove line breaks and spaces.

6B8. Select **Duration**.

The available choices are:

a. **None**

The value is not stored.

b. **Page view**

- The value is held in a JavaScript variable until the page is refreshed or a new page is loaded.
- Can be created and set in scripts using `_satellite` object syntax:
`_satellite.setVar('data_element_name')`

c. **Session**

Values persist in the browser's session storage until the browser tab is closed.
Available throughout the site visit.

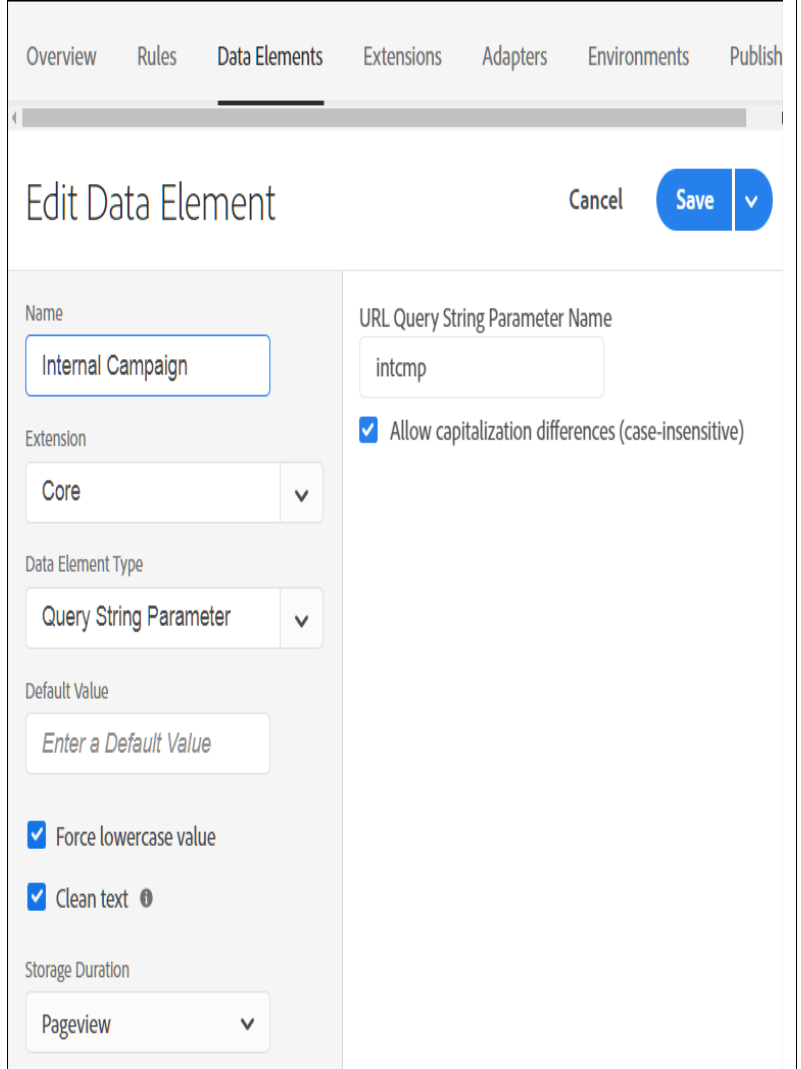
d. **Visitor**

The value is stored indefinitely in the browser's local storage.

6B9. Click **Save**.

When creating or editing elements, you can save and build to your active library. This immediately saves your change to your library and executes a build. The status of the build is displayed. You can also create a new library from the Active Library drop down.

Important: Changes do not take effect until they are published.



Edit Data Element [Cancel] [Save]

Name: Internal Campaign

Extension: Core

Data Element Type: Query String Parameter

URL Query String Parameter Name: intcmp

☒ Allow capitalization differences (case-insensitive)

Default Value: Enter a Default Value

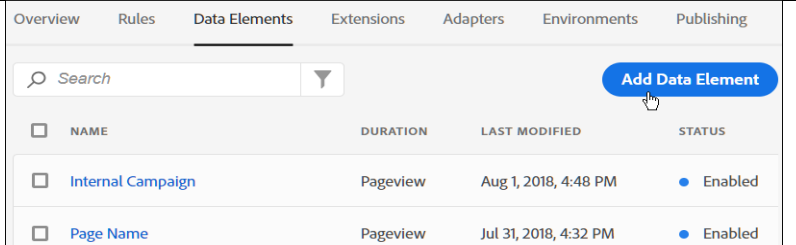
☒ Force lowercase value

☒ Clean text ⓘ

Storage Duration: Pageview

Exercise 6C: Data Element (URL Query String Parameter – Internal Search Term) 1 Min

6C1. From a Property page, open the Data Elements tab, then click **Create New Data Element**.



NAME	DURATION	LAST MODIFIED	STATUS
<input type="checkbox"/> Internal Campaign	Pageview	Aug 1, 2018, 4:48 PM	● Enabled
<input type="checkbox"/> Page Name	Pageview	Jul 31, 2018, 4:32 PM	● Enabled

6C2. Name the data element (**Internal Search Term**).

6C3. Select an Extension (**Core**) and Data Element Type (**Query String Parameter**).

Note: The available data element types are determined by the extension. For information about the types available with the Launch Core extension, refer to [Types of data elements](#).

6C4. Provide any requested information about the **URL Query String Parameter Name (q)**

6C5. (Optional) Enter a default value.

If you do not provide a value, no value is sent. Some people choose to enter something like "none" or "n/a" so they can determine what is sent if there isn't a value. Different solutions deal with an empty variable differently. This creates consistency even if a value doesn't exist.

6C6. Select **Force a lowercase value**

6C7. Select **Clean text** to remove line breaks and spaces.

6C8. Select **Duration**.

The available choices are:

e. **None**

The value is not stored.

f. **Page view**

- The value is held in a JavaScript variable until the page is refreshed or a new page is loaded.
- Can be created and set in scripts using `_satellite` object syntax:
`_satellite.setVar('data_element_name')`

g. **Session**

Values persist in the browser's session storage until the browser tab is closed.
Available throughout the site visit.

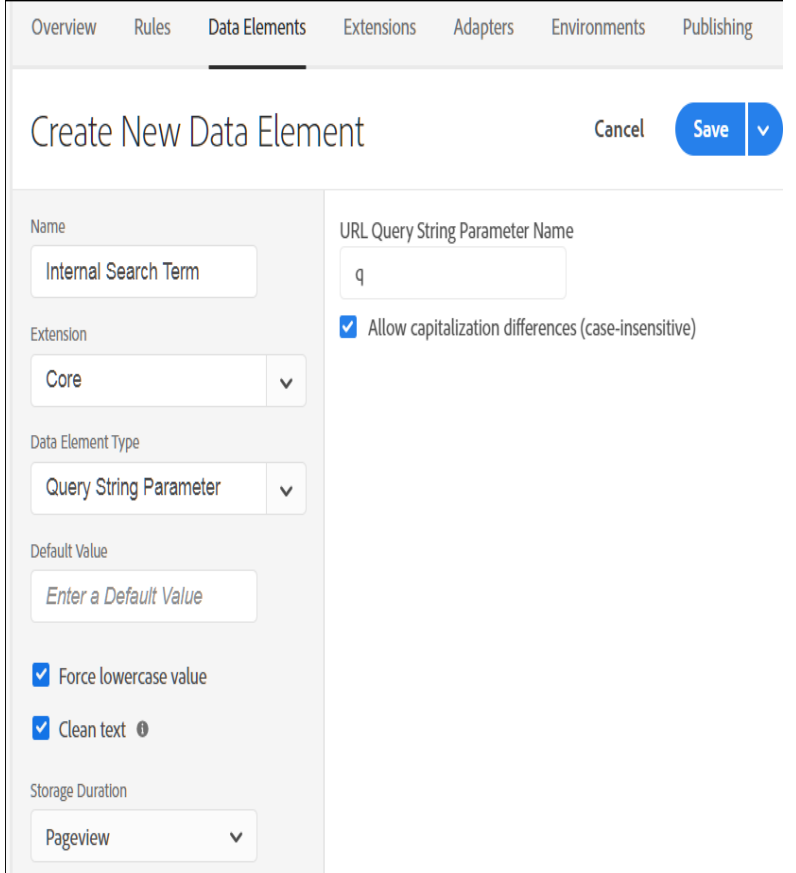
h. **Visitor**

The value is stored indefinitely in the browser's local storage.

6C9. Click **Save**.

When creating or editing elements, you can save and build to your [active library](#). This immediately saves your change to your library and executes a build. The status of the build is displayed. You can also create a new library from the Active Library drop down.

Important: Changes do not take effect until they are published.



Create New Data Element

Cancel Save

Name: Internal Search Term

Extension: Core

Data Element Type: Query String Parameter

Default Value: Enter a Default Value

☒ Force lowercase value

☒ Clean text ⓘ

Storage Duration: Pageview

URL Query String Parameter Name: q

☒ Allow capitalization differences (case-insensitive)

Part 6 – Rules

Rules are at the logical core of your implementation and control the what, when, where, and how of all the tags on your site. Define an event, set conditions and exceptions, and then define the actions and order. Finally, publish your changes to see the results. For more information, see [Rules](#).

Launch is a rule-based system. It looks for user interaction and associated data. When the criteria outlined in your rules are met, the rule triggers the extension, script, or HTML you identified.

Build rules to integrate the data and functionality of marketing and ad tech that unifies disparate products into a single solution.

Events are where the majority of interactions on sites take place. You can measure and react to these interactions in real-time, without the need for JavaScript.

Events (If): The event is what you want the rule to look for. This is defined by choosing an event, any applicable conditions, and any exceptions.

Actions (Then): Triggers occur after a rule's events take place and all conditions are satisfied. A rule in Launch can trigger as many discrete actions as you want, and you can control the order in which these actions occur. For example, a single rule for an e-commerce Thank You page can trigger your analytics tools and third-party tags from a single rule. There is no need to create separate rules for each extension or tag.

You can add more event types. Multiple events are joined with an OR, so the rule's conditions will be evaluated if any of the events are met.

Important: Changes do not take effect until they are [published](#).

Conditions, Events and exceptions (if)

Events, with any conditions and exceptions, are the *If* portion of a rule.

If a specified event occurs, the conditions and exceptions are evaluated, then the specified actions take place if needed.

- **Conditions**

Narrow the event by configuring any conditions that must be true for an event to trigger the rule.

- **Events**

Specify one or more events that must take place to trigger the rule. Multiple events are joined by an OR. Any of the specified events will trigger the rule.

- **Exceptions**

Specify any exceptions that would keep the rule from firing, even if the events and conditions are met.

The events that are available **depend on which extensions are installed**. For information about the events in the Core extension, see [Core extension event types](#).

Note: DTM provided page load, event-based and direct call event types. This is no longer true in Launch. Rather than event types, Launch extensions configure different individual events.

Only events are required. **The conditions and exceptions are optional.**

Actions (then)

Actions are the Then portion of a rule. When an event is triggered, if conditions evaluate to true and exceptions evaluate to false, the actions are performed. You can drag and drop actions to order them as desired.

The action, or *then*, part of the formula determines what happens when the event takes place and all conditions and exclusions are met.

Core Extension Rules Action Types

This section describes the action types available in the Core extension when creating rules.

Custom Code

Provide the code that runs after the event is triggered and conditions are evaluated.

1. Name the action code.
2. Select the language used to define the action:
 - o JavaScript
 - o HTML
3. Select whether to execute the action code globally.
4. Click Open Editor.
5. Edit the code, then click Save.

The screenshot shows the 'Action Configuration' dialog for the 'Core' extension. On the left, 'Extension' is set to 'Core' and 'Action Type' is 'Custom Code'. The 'Name' field contains 'Core - Custom Code'. On the right, 'Language' is set to 'JavaScript' (selected with a blue dot), and 'HTML' is unselected. There is an unchecked checkbox for 'Execute globally' and an 'Open Editor' button with a code icon. The top right has 'Cancel' and 'Keep Changes' buttons.

Experience Cloud ID Service Extension Rules Action Types

Action Types

1. Set Customer IDs
Set one or more customer IDs.
2. Enter the integration code.
The integration code should contain the value set up as a data source in Audience Manager or Customer Attributes.
3. Select a value.
The value should be a user ID. Data elements are most suitable for dynamic values like IDs from a client-specific internal system.
4. Select an authentication state.
Available options are:
 - Unknown
 - Authenticated
 - Logged out
5. (Optional) Click Add to set more customer IDs.
6. Click Keep Changes.

The screenshot shows the 'Action Configuration' dialog for the 'Experience Cloud ID Service' extension. 'Extension' is set to 'Experience Cloud ID Service' and 'Action Type' is 'Set Customer IDs'. The 'Name' field contains 'Experience Cloud ID Service - ...'. The main configuration area has three columns: 'Integration Code', 'Value', and 'Auth State'. The 'Integration Code' and 'Value' fields are empty, separated by an equals sign. The 'Auth State' dropdown is set to 'Unknown'. There are vertical ellipsis icons below the 'Integration Code' and 'Value' fields, and an 'Add Another' button. The top right has 'Cancel' and 'Keep Changes' buttons.

Rule ordering

Rule ordering allows you to control the order of execution for rules that share an event.

It is often important to have your rules fire in a specific order. Examples: (1) you have several rules that conditionally set Analytics variables and you need to make sure that the rule with Send Beacon goes last. (2) You have a rule that fires Target and another rule that fires Analytics and you want the Target rule to run first.

Adobe highly recommends that you order your rules with positive numbers between 1 and 100 (default is 50). Simpler is better. Remember you have to maintain your order.

Rules with page bottom or page top event

- **JavaScript:** The JavaScript is embedded in the main Launch library. The custom script is wrapped in a script tag and written to the document using document.write. If the rule has multiple custom scripts, they're written in order.
- **HTML:** The HTML is embedded in the main Launch library. document.write is used to write the HTML to the document. If the rule has multiple custom scripts, they're written in order.

Rules with any other event

- **JavaScript:** The JavaScript is loaded from the server as regular text, wrapped in a script tag, and added to the document using Postscribe. If the rule has multiple JavaScript custom scripts, they will be loaded in parallel from the server, but executed in the same order that was configured in the rule.
- **HTML:** The HTML is loaded from the server and added to the document using Postscribe. If the rule has multiple custom HTML scripts, they will be loaded in parallel from the server, but executed in the same order that was configured in the rule.

Delete Resources

You can delete data elements, rules, and extensions.

Select the resource you want to delete. Then click Delete.

Prepare a resource for deletion

Before you delete a resource, you must make sure it is in a state where it can be deleted.

Resources exist in different states and they depend on one another. In some cases, you must resolve conflicts before you can delete a resource. In other cases, you can delete the resource, but you have to update any other resources that are dependent on the one you delete.

For example:

- Deleting a data element affects the behavior of any rules that reference the data element.
- Deleting an extension affects any data elements and any rules that include components provided by the extension.

How to Create Rules

Create a rule by specifying what actions occur if a condition is met.

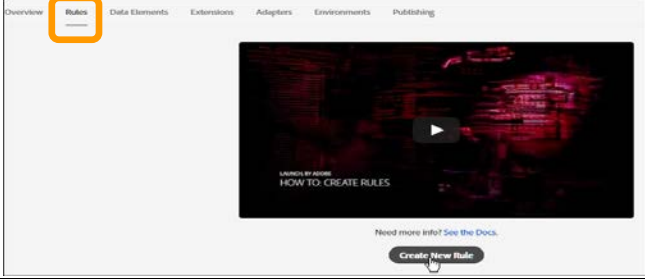

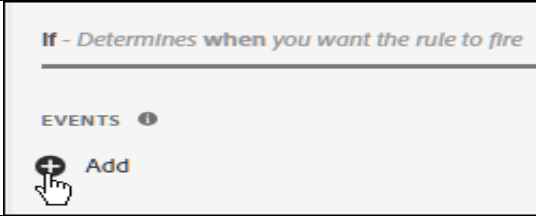
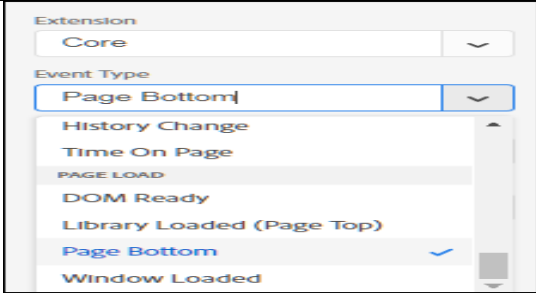
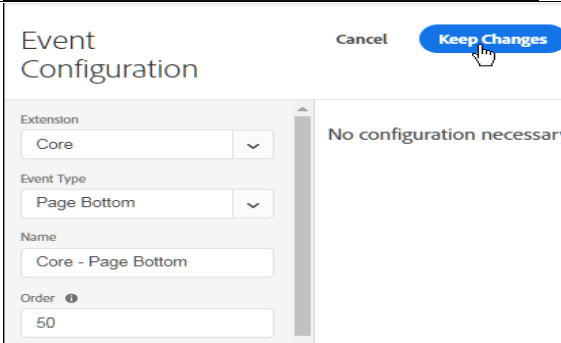
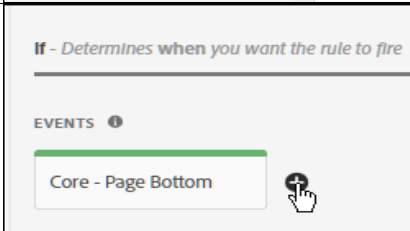
1. Open the Rules tab, and then click Create New Rule.
2. Name the rule.
3. Click the **Events** Add icon.
4. Choose your extension and one of the event types available for that extension, and then configure the properties for the event.
Note: The available event types are defined by the extension. Some events don't have any properties that need to be configured.
5. Set the Order parameter, and then click Keep Changes.
Default order for all rule components is 50. If you want one to run sooner, give it a number lower than 50.
 - Order of execution is order of numbers. 1 comes before 3. 3 comes before 10. 10 comes before 100, etc.
 - Rules that have the same order run in no particular order.
 - Rules are fired in order, but do not necessarily finish in the same order. If Rule A and Rule B share an event, and you assign order so that Rule A comes first, then if Rule A does something asynchronously, there's no guarantee that Rule A finishes before Rule B starts.
If you want it to run later, give it a number higher than 50. For more information about ordering, see [Rule ordering](#).
6. Click the **Conditions** Add icon, then choose a condition type and configure the properties for your condition. Then Click Keep Changes.
You can add other conditions. Multiple conditions are joined with an OR. The rule's exceptions will be evaluated if any of the events are met, along with their conditions.
7. Click the **Exceptions** Add icon, then choose an exception type and configure the properties for your exception. Then Click Keep Changes.
You can add other exceptions. Multiple exceptions are joined with an OR. The rule's actions will be evaluated if any of the events are met, along with their conditions and exceptions.
8. Click the **Actions** Add icon, then choose your extension and one of the action types available for that extension, configure the properties for the action, then click Keep Changes.
9. Review your rule, and then click Save.

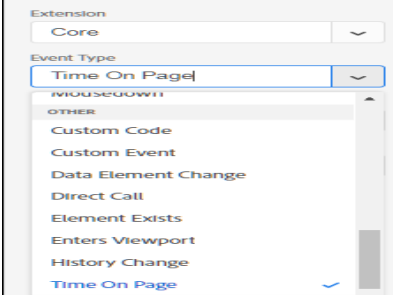
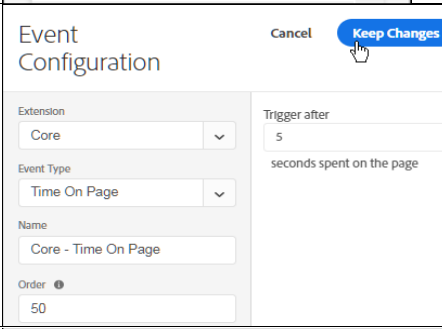
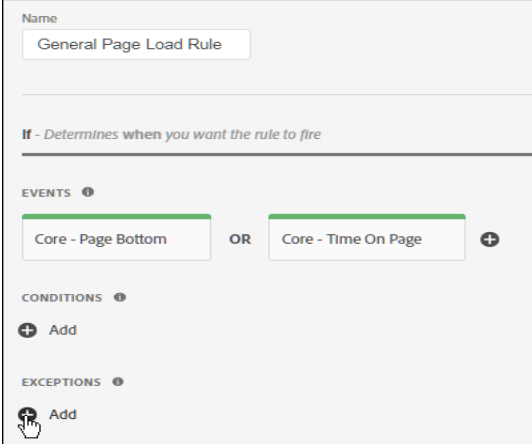
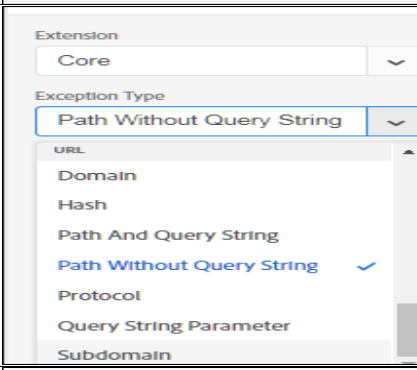
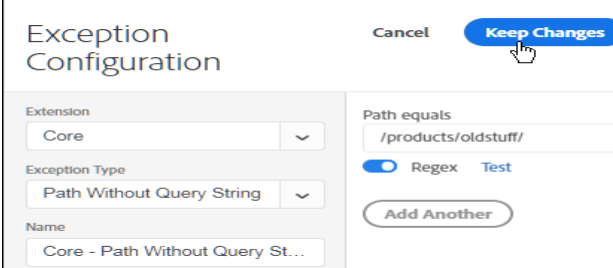
Later, when you publish, you'll add this rule to a library and deploy it.

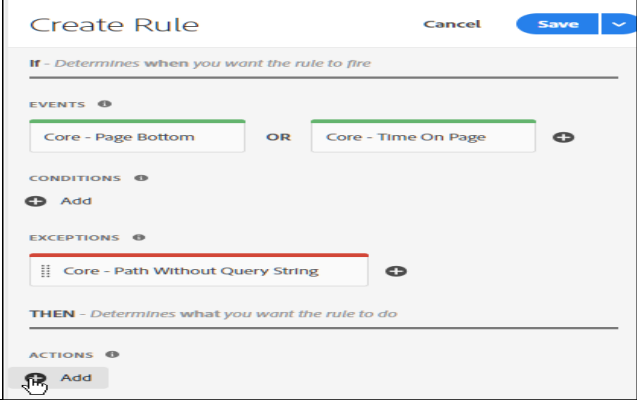
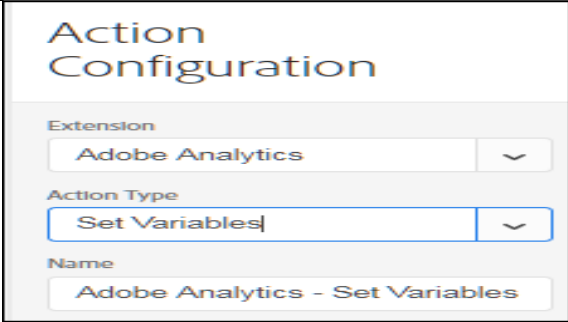
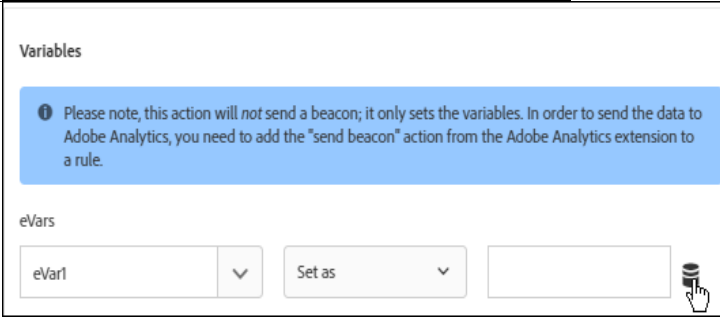
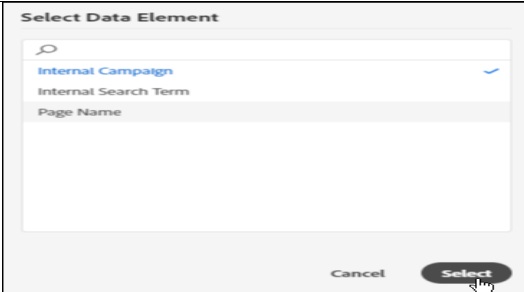
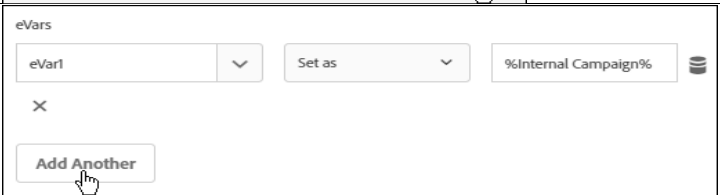
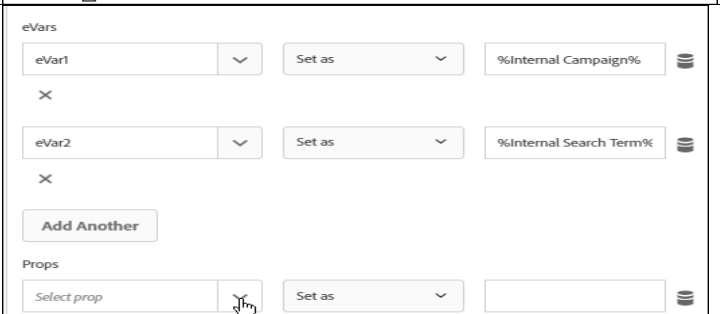
When you edit or update an existing rule, a new version of the rule is created, which you can then deploy.

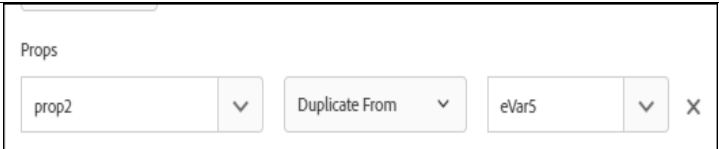
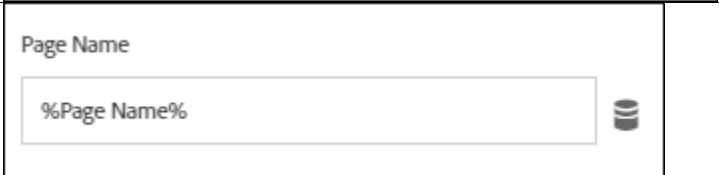
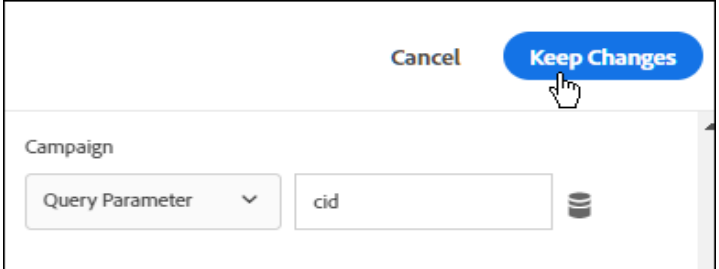
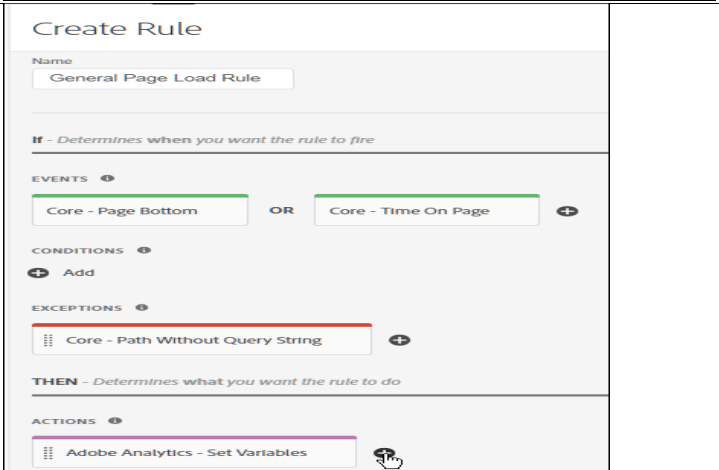
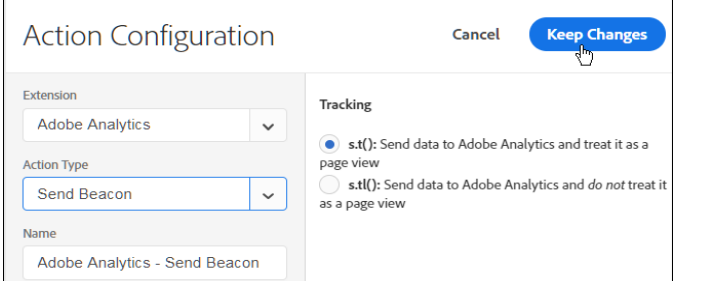
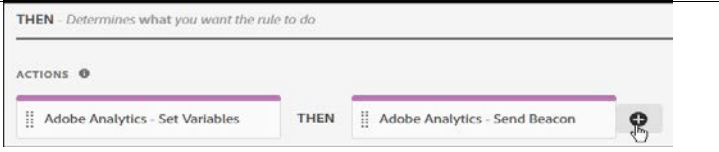
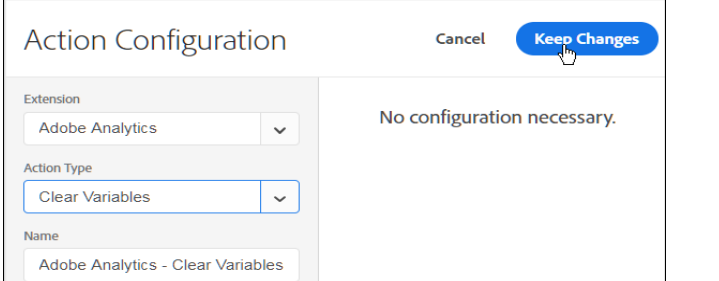
When creating or editing rules, you can save and build to your active library. This immediately saves your change to your library and executes a build. The status of the build is displayed.

Exercise 7: Rules (Page Load Rule) 5 Mins

7.1 Open the Rules tab, and then click Create New Rule .	
7.2 Name the rule (General Page Load Rule).	
7.3 Click the Events Add icon.	
7.4 Choose your extension (Core) and one of the event types (Page Load -> Page Bottom) available for that extension. Note: The available event types are defined by the extension. Some events don't have any properties that need to be configured.	
7.5 Set the Order parameter, and then click Keep Changes . Default order for all rule components is 50 . Note: If you want one to run sooner, give it a number lower than 50. Order of execution is order of numbers. 1 comes before 3. 3 comes before 10. 10 comes before 100, etc. Rules that have the same order run in no particular order. Rules are fired in order, but do not necessarily finish in the same order. If Rule A and Rule B share an event, and you assign order so that Rule A comes first, then if Rule A does something asynchronously, there's no guarantee that Rule A finishes before Rule B starts. If you want it to run later, give it a number higher than 50. For more information about ordering, see Rule ordering .	
7.6 Click the Events Add icon. You can add other conditions. Multiple Events are joined with an OR .	

<p>7.7 Choose your extension (Core) and one of the event types (Other -> Time on Page) available for that extension</p>		
<p>7.8 Configure the properties for the event. Trigger after 5 seconds spent on the page.</p> <p>7.9 Set the Order parameter (50) and then click Keep Changes.</p>		
<p>7.10 Click the Exceptions Add icon</p>		
<p>7.11 Choose your extension (Core) and one of the Exception Types (URL -> Path Without Query String).</p>		
<p>7.12 Configure the Path equals (/products/oldstuff/) for your exception. Then Click Keep Changes.</p> <p>Note: You can add other exceptions. Multiple exceptions are joined with an OR. The rule's actions will be evaluated if any of the events are met, along with their conditions and exceptions.</p>		

7.13 Click the Actions Add icon	
7.14 Choose your extension (Adobe Analytics) and one of the action types (Set Variables) available for that extension, configure the properties for the action, and then click Keep Changes . Note: The available action types are defined by the extension. You can add more event types. Multiple events are joined with an OR, so the rule's conditions will be evaluated if any of the events are met.	
7.15 Under Variables > eVars select eVar1 Set as %Internal Campaign% .	
7.16 You can click on the Database Icon to select from the available Data Elements.	
7.17 Under eVars select Add Another.	
7.18 Under Variables > eVars select eVar2 Set as %Internal Search Term% .	

7.19 Under Props select prop2 Duplicate From eVar5	
7.20 Page Name select %Page Name% .	
7.21 Under Props select prop2 Duplicate From eVar5 and then click Keep Changes .	
7.22 Click the Actions Add icon to add another Action.	
7.23 Choose your extension (Adobe Analytics) and one of the action types (Send Beacon) available for that extension, configure the properties for the action (Tracking > s.t() : Send data to Adobe Analytics and treat it as a pageview , and then click Keep Changes .	
7.24 Click the Actions Add icon to add another Action .	
7.25 Choose your extension (Adobe Analytics) and one of the action types (Clear Variables) and then click Keep Changes .	

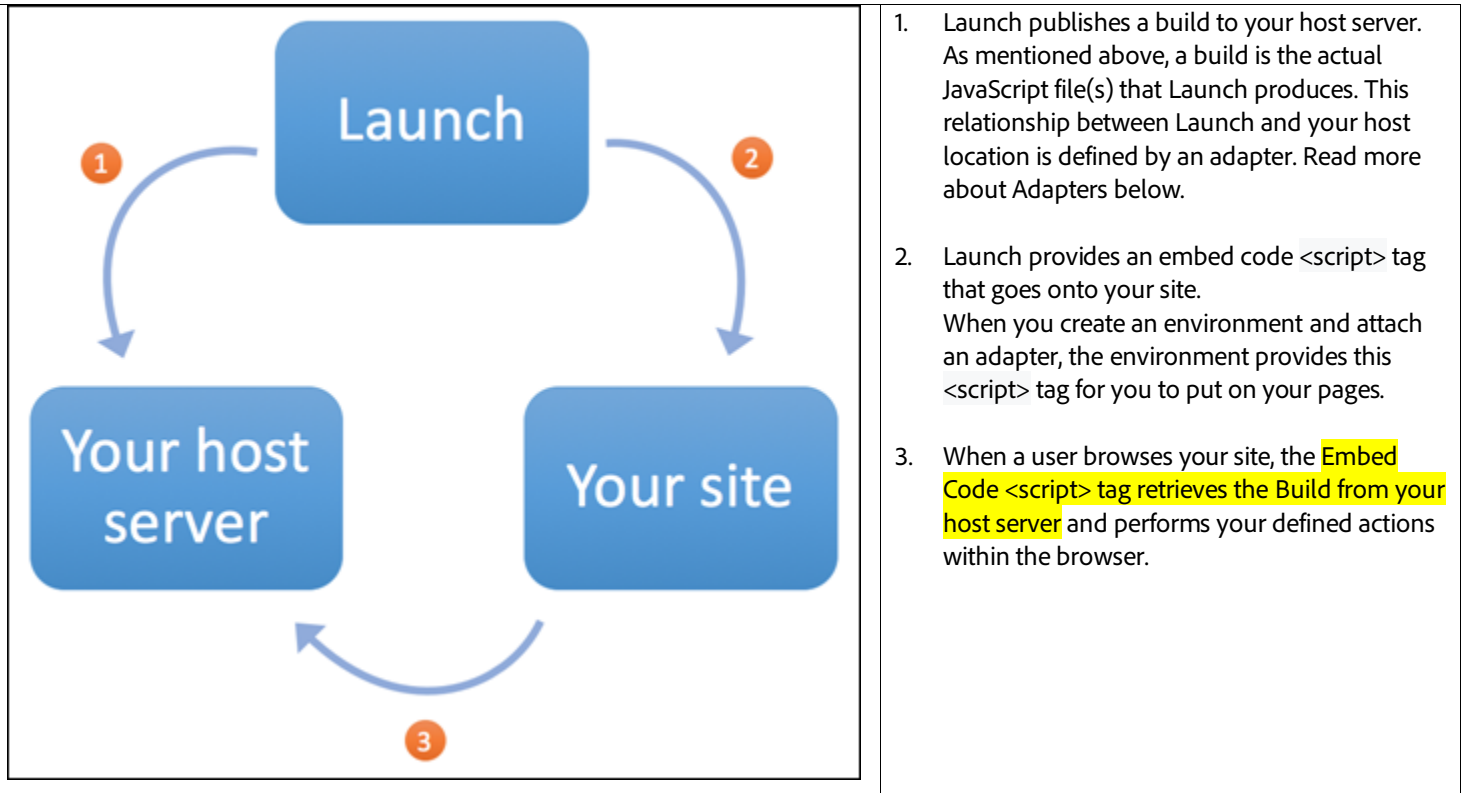
	<div><div>ACTIONS 0</div><div><div>Adobe Analytics - Set Variables</div><div>THEN</div><div>Adobe Analytics - Send Beacon</div><div>THEN</div><div>Adobe Analytics - Clear Variables</div></div></div>
7.26 Review your rule, and then click Save .	<div><div>Create Rule</div><div>CancelSave</div><div>Name</div><div>General Page Load Rule</div><div>If - Determines when you want the rule to fire</div><div>EVENTS 0</div><div>Core - Page BottomORCore - Time On Page</div><div>CONDITIONS 0</div><div>Add</div><div>EXCEPTIONS 0</div><div>Core - Path Without Query String</div><div>THEN - Determines what you want the rule to do</div><div>ACTIONS 0</div><div>Adobe Analytics - Set VariablesTHENAdobe Analytics - Send BeaconTHENAdobe Analytics - Clear Variables</div></div>
Later, when you publish, you'll add this rule to a library and deploy it.	<div><div>OverviewRulesData ElementsExtensionsAdaptersEnvironmentsPublishing</div><div>Search</div><div>Add Rule</div><div><div><input type="checkbox"/></div><div>NAME</div><div>LAST MODIFIED</div><div>STATUS</div></div><div><div><input type="checkbox"/></div><div>General Page Load Rule</div><div>Aug 1, 2018, 8:41 PM</div><div>Enabled</div></div></div>

Part 7 - Test in your Development Environment

Libraries and builds

Nothing in Launch is published automatically. Each set of changes you make is encapsulated into a library. Each library you create automatically inherits anything upstream (developed, submitted, approved, or published) as a baseline, so all you need to do is define the changes you'd like to make. This library serves as the blueprint for a build. A build is the actual set of JavaScript files that are deployed and used.

To make sense of that process, there are a couple relationships between Launch, your web page, and your hosting location that you need to understand.



Publish a build to Development

Now that you understand the basic components, the publishing process should make more sense. You need to:

1. Create an **Adapter**.
2. Create a **Dev Environment** using the adapter you created.
3. Deploy the **Embed Code** from your Dev Environment to your Dev test site.
4. Create a **Library** and assign it to the Dev environment you created.
5. **Build** your library.

Publishing

Publishing is all about getting your **extensions**, **data elements**, and **rules** to work for you, collecting the data you want them to collect and providing the experience you want to provide for your users. Launch gives you control over the entire publishing process.

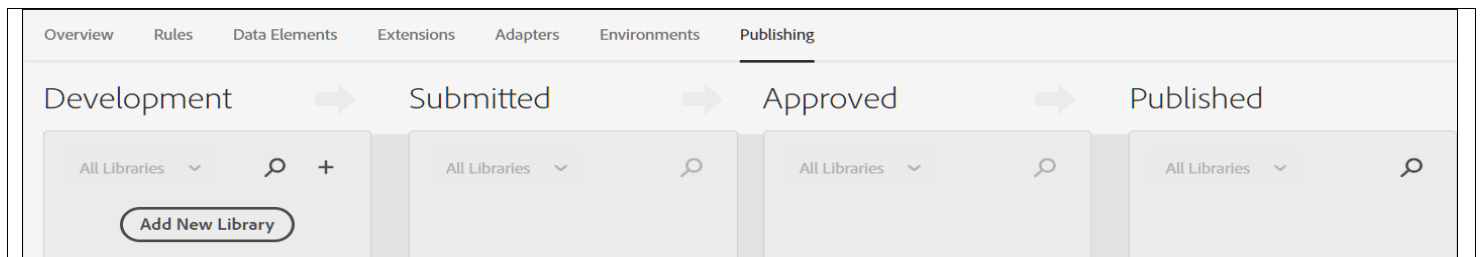
There are a few components and relationships that are important to understand in order to take full advantage of the publishing workflow. These are:

- **Adapters**
An **adapter** represents the connection details for the environment to deliver the build. You can choose to let Launch manage the hosting of your build, or you can provide information for your own host servers.
- **Environments**
An **environment** is a set of deployment instructions that tells Launch what format you'd like your build in and where you'd like that build delivered.
- **Embed Code**
The **embed code** is the set of script tags that you place within the HTML on your site. These tags tell each browser where to retrieve the build. The embed code is attached to an environment and can change when you make changes to your environment configuration.
- **Libraries**
A **library** is a **set of instructions** for how **extensions**, **data elements**, and **rules** will interact with one another and with your website. Libraries are compiled into builds. A library can contain as many changes as you are comfortable making or testing at once.
- **Builds**
A **build** is the **actual set of files containing the code** that is *delivered to each user's browser* when that user views your site.

Publishing Process

The publishing process consists of:

1. **Creating** and editing libraries.
2. **Testing** the functionality of those libraries where you need to test them.
3. **Deploying** those libraries to your production site.



For example, if you create a new checkout event, create a revenue data element related to that event, and make a change to the Adobe Analytics extension configuration to support the new event and data element, you can save them all at once to a checkout library, then test, approve, and publish them as a group.

Libraries

A library is a set of instructions for how **extensions**, **data elements**, and **rules** will interact with one another once they've been deployed.

When creating a library, you will specify the changes you want to make to your library.

At **build time**, these changes are combined with everything that has been **submitted, approved, or published** in previous libraries.

Libraries contain the addition or removal of:

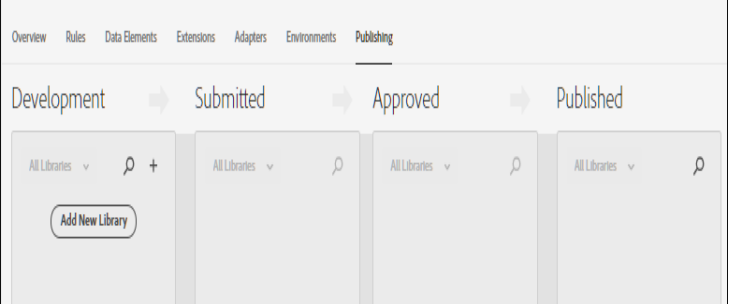
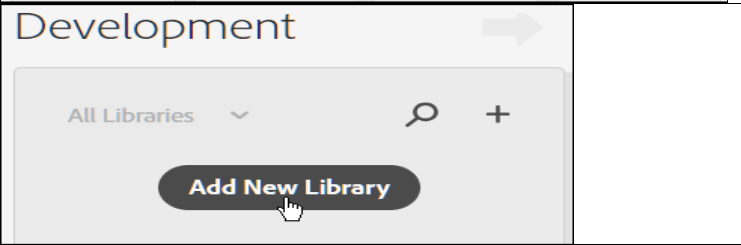
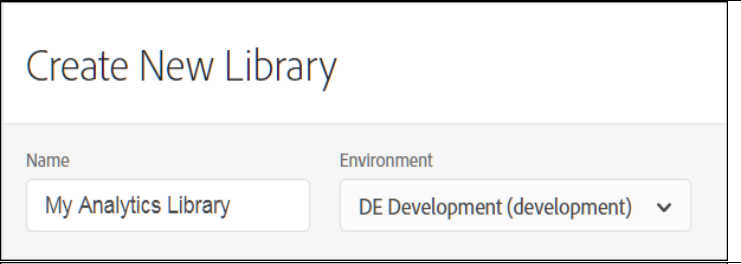
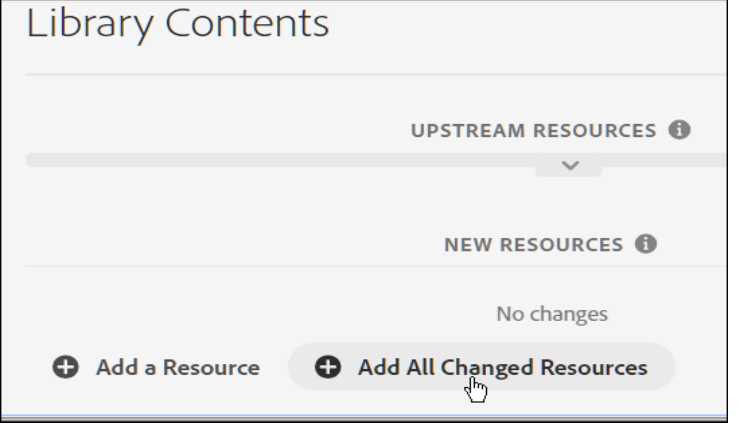
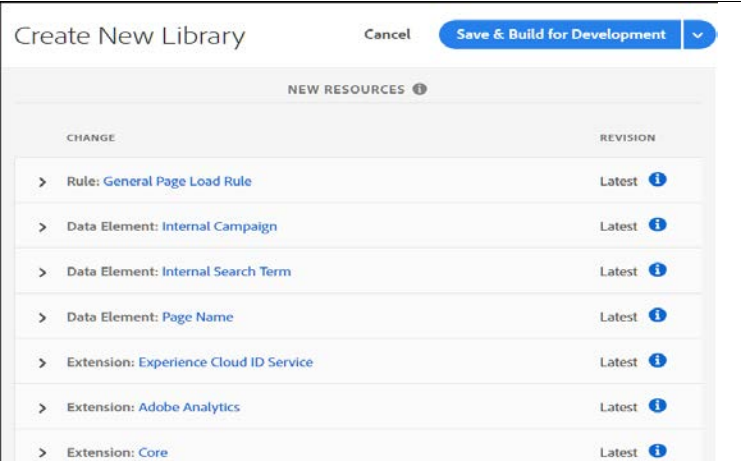
- Extension configuration
- Data Elements
- Rules

Libraries must be assigned to an environment before they can be compiled into a build.

Libraries are **approved or rejected** as a **whole**. You cannot approve or reject individual items within a library.

A library moves between several environments as it makes its way through the publishing workflow.

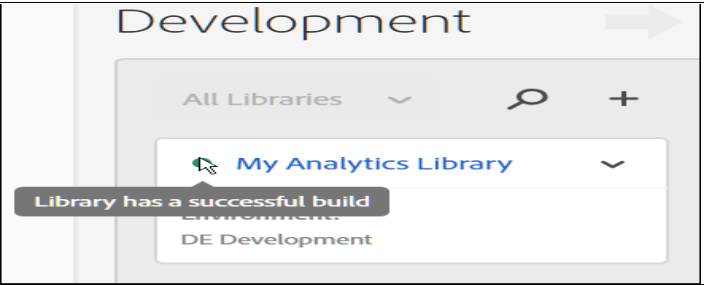
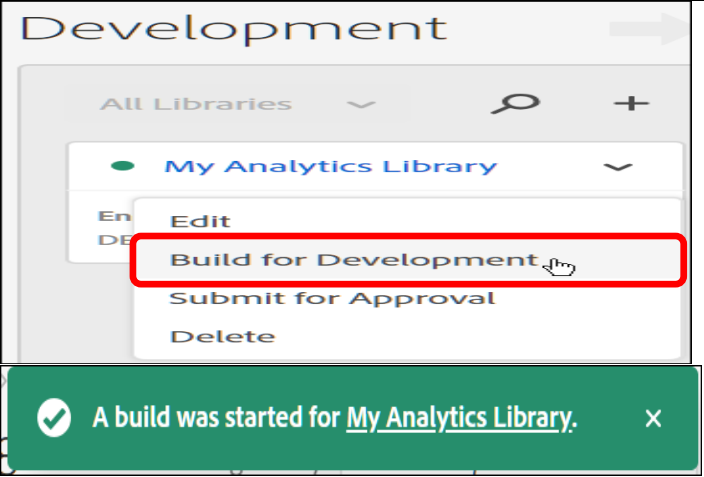
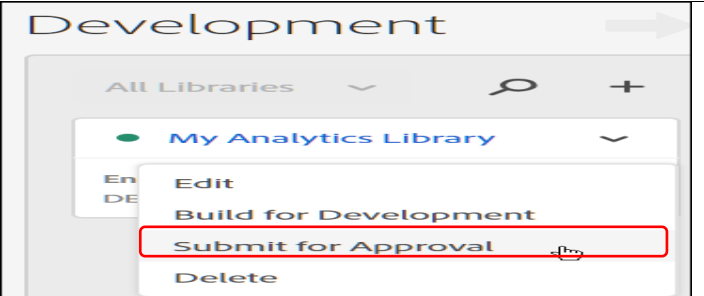
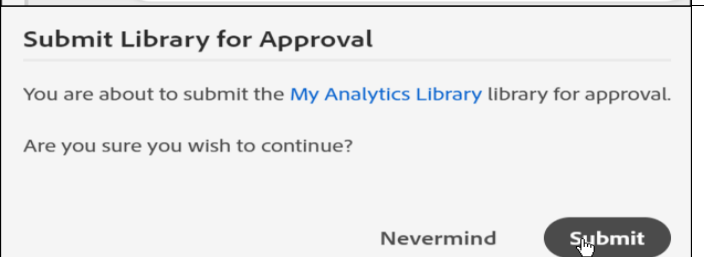
Exercise 8 - Create a library 2 Mins

<p>1. Open the Publishing tab.</p> <p>The Publishing page lists the Dev libraries and provides the means to submit them for approval, move them to staging, or publish them to production.</p>	
<p>2. Click Add New Library.</p>	
<p>3. Name the library (My Analytics Library).</p> <p>4. Assign the library to a Dev environment (DE Development).</p>	
<p>5. Add a change to the library. To add an item, click Add a Change, then choose the items you want to add. Any item that has been edited or deleted is available to add to the chosen library.</p> <p>You can add the following to your library:</p> <ul style="list-style-type: none"> Extension configurations Data elements Rules <p>6. To add any resources that have changed, click Add All Changed Resources.</p>	
<p>7. Click Save or Save and Build for Development. Deploying compiles a build and deploys it to the assigned environment.</p>	

Development Exercise 1 Min

Once a library is created, use the drop down menu for that library to select one of the following options:

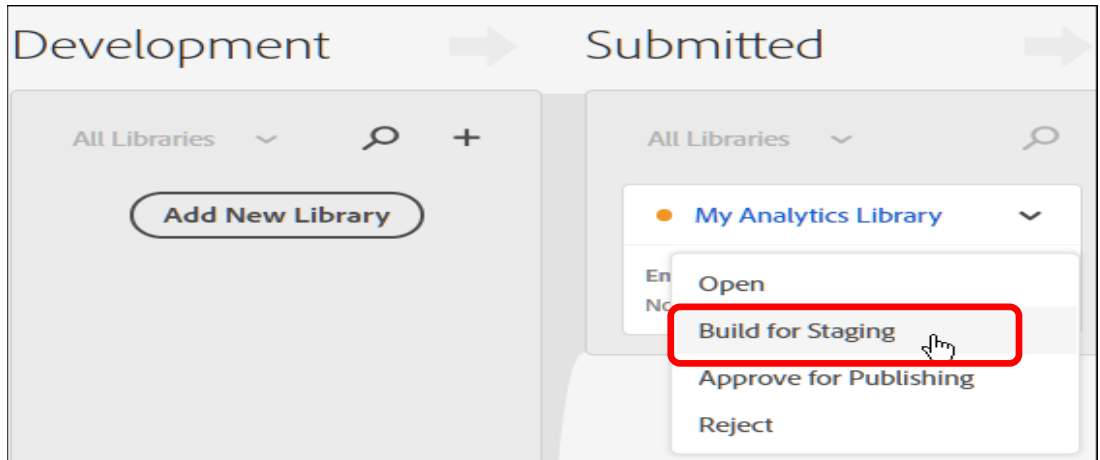
- Edit - Use the library Edit screen to add or remove components from a library.
- **Build for Development** - Create a build for the library. The build is compiled and deployed to the environment the library is assigned to. This step fails if the library has not been assigned to an environment.
- Submit for Approval - Unassigns the library from its development environment and moves the library to the submitted column for an Approver to work on.
- Delete - Remove that library from the publishing process.

	
1. Select Build for Development	
2. Select Submit for Approval	
3. Click Submit	

Submitted Exercise 1 Minute

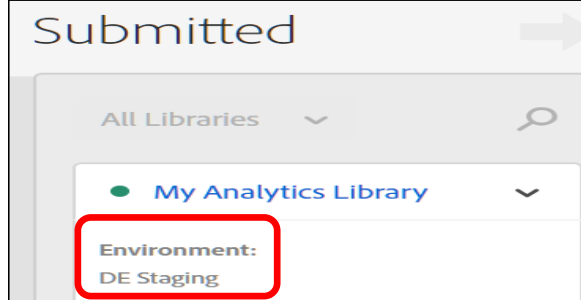
- Open
View the contents of the library. **Changes are not allowed**. If changes are needed, the library *should be rejected* so changes can be made in Development.
- **Build for Staging**
Assigns the library to the **staging environment** and deploys it.
- Approve for Publishing
Moves the library to the Approved column for a Publisher to publish
- Reject
Unassigns the library from the staging environment and moves the library to the Development column for changes.

1. Select **Build for Staging**

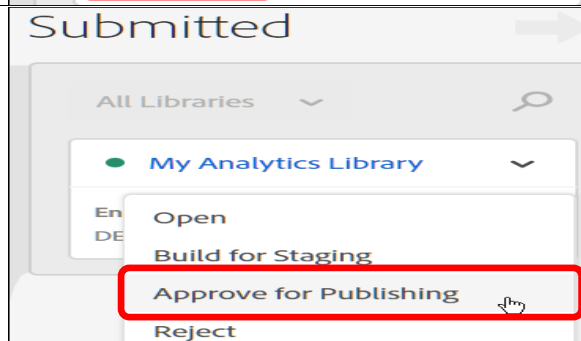


✓ A build was started for My Analytics Library.

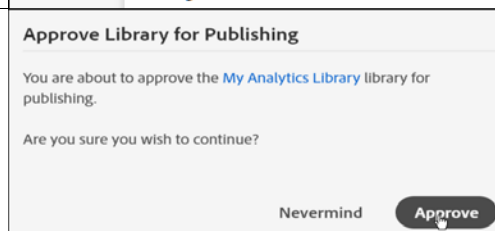
Note that it automatically places the library in the Staging Environment.



2. Select **Approve for Publishing**



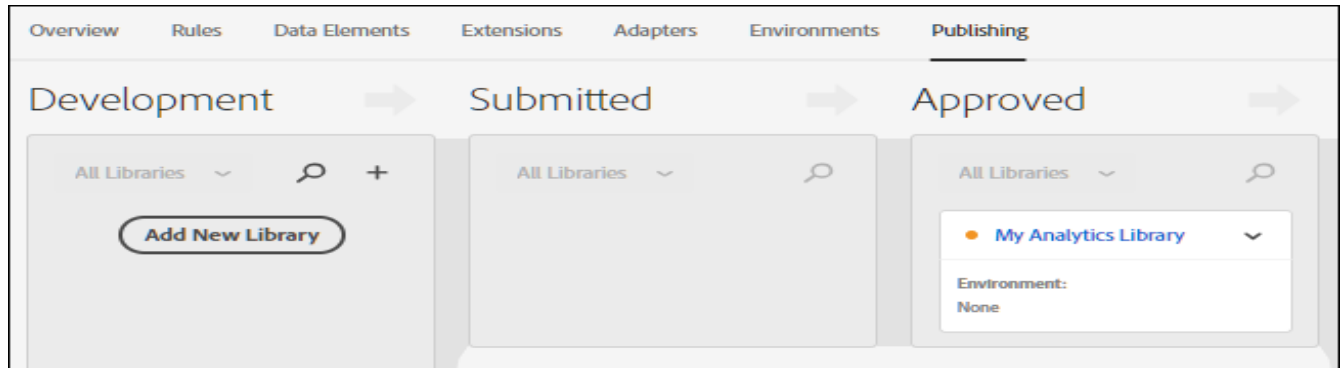
3. Click **Approve**



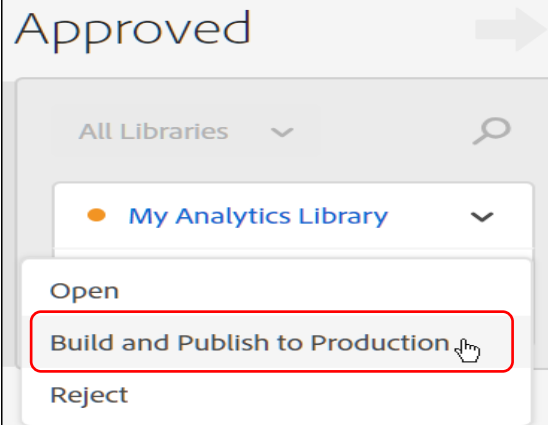
Approved Exercise 1 Minute

- Open
View the contents of the library. **Changes are not allowed.** If changes are needed, the library should be rejected so changes can be made in Development.
- **Build and Publish to Production**
Unassigns the library from the staging environment, assigns the library to the **production environment**, and deploys it. Important: When this option is selected, your library becomes live in your production environment, so be sure before you click the button.
- Reject

Unassigns the library from the staging environment and moves the library to the Development column for changes.



1. Select **Build and Publish to Production**

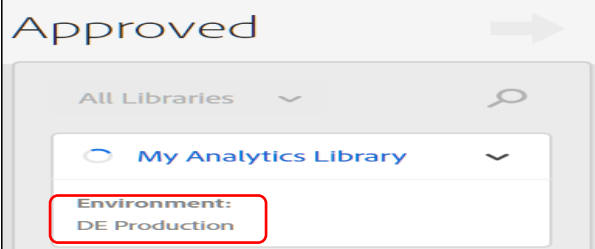


2. Click **Publish**

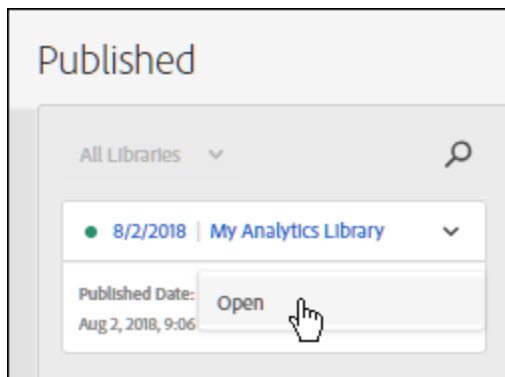
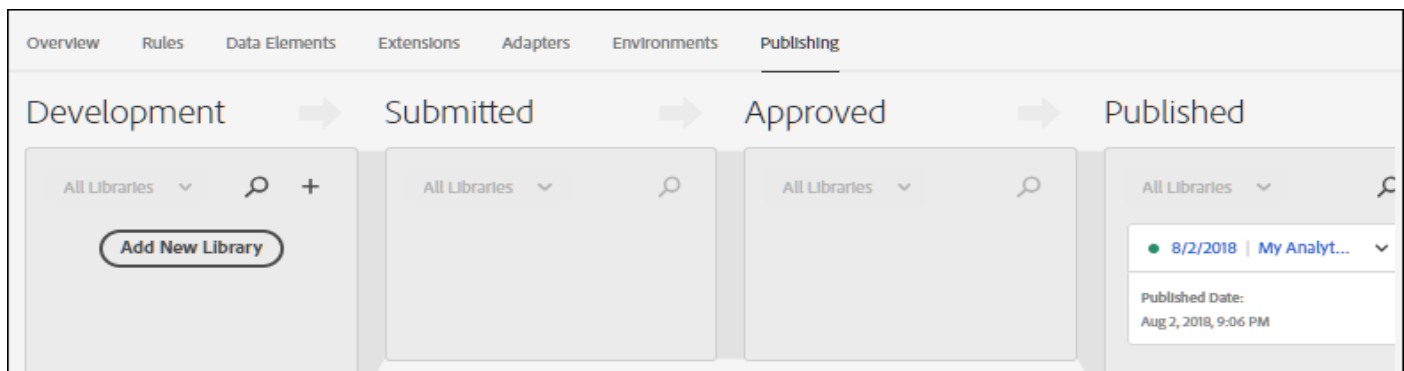


✓ A build was started for My Analytics Library.

Note: The Environment is updated to Production



Published Exercise 1 min



The library is live in the production environment. This column shows which libraries have been published and their publish dates. You can look at these libraries, but **you cannot make changes**. If you want to change what's in your production environment, you'll need to create a new library and push it through the approval process.

ClickClose

Omniture Training 13 >

User1 08/02/18

Working Library:

Select an option

Overview

Rules

Data Elements

Extensions

Adapters

Environments

Publishing

Edit Library

Close

Status

Published

Name

My Analytics Library

Environment

None

Activity Log

Build (Aug 2, 2018, 9:06 PM)

succeeded

Build (Aug 2, 2018, 9:04 PM)

succeeded

Build (Aug 2, 2018, 8:57 PM)

succeeded

Library Contents

UPSTREAM RESOURCES

NEW RESOURCES

CHANGE	REVISION	STATUS
> Data Element: Internal Campaign	Revision 1 (Latest)	● Enabled
> Data Element: Internal Search Term	Revision 1 (Latest)	● Enabled
> Data Element: Page Name	Revision 1 (Latest)	● Enabled
> Rule: General Page Load Rule	Revision 1 (Latest)	● Enabled
> Extension: Experience Cloud ID Service	Revision 1 (Latest)	● Enabled
> Extension: Adobe Analytics	Revision 1 (Latest)	● Enabled
> Extension: Core	Revision 1 (Latest)	● Enabled

Add to a library

1. Install the **extensions** you want to add.
2. Create the **data elements** and **rules** you want to add.
3. Open the **Publishing** tab.
4. Select the **library** you want to change, then click Edit.
5. Use the extensions, data elements, and rules buttons to select the items you want to add to the library.
6. Save.

Changes to the library are shown in the Library Contents change log.

Note: Data elements can depend upon extensions. Rules can depend on both data elements and extensions. If you do not include all the necessary components in your library, the build will fail at build time and you'll need to add the necessary components before you complete a successful build. A future release will check dependencies when making changes to a library.

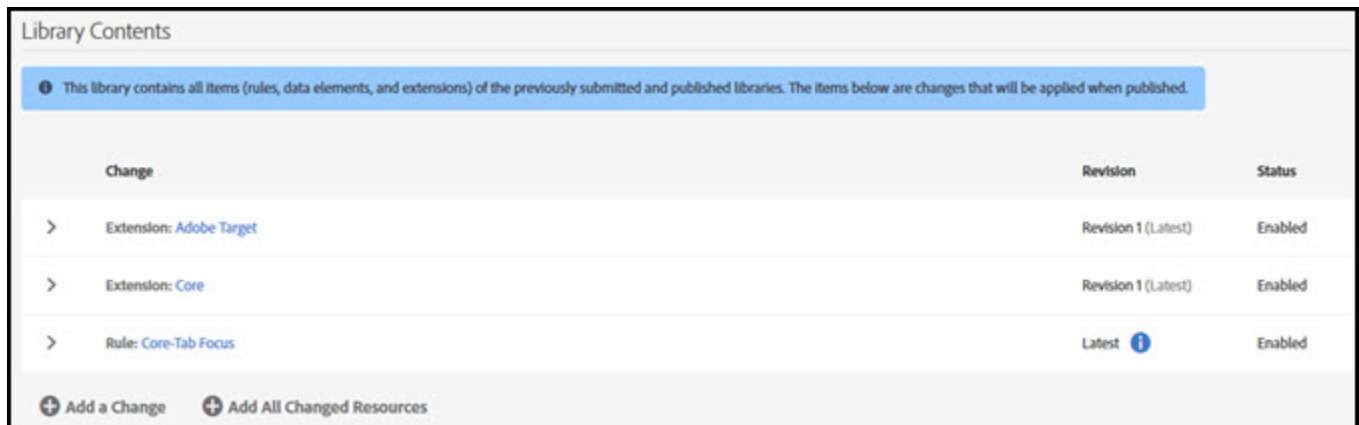
Remove from a library

To remove something from a library, you must **deactivate it** and then **publish** the deactivated state.

1. Disable the **extensions** you want to remove, along with any **data elements** and **rules** that depend on those extensions.
2. Disable the **data elements** and **rules** you want to remove.
3. Open the **Publishing** tab.
4. Select the **library** you want to change.
5. Use the extensions, data elements, and rules buttons to select the disabled items you want to remove from the library.
6. Save.

Manage library changes

1. Click on a library and select **Edit** to view library changes. All changes are shown in the **Library Contents** list.

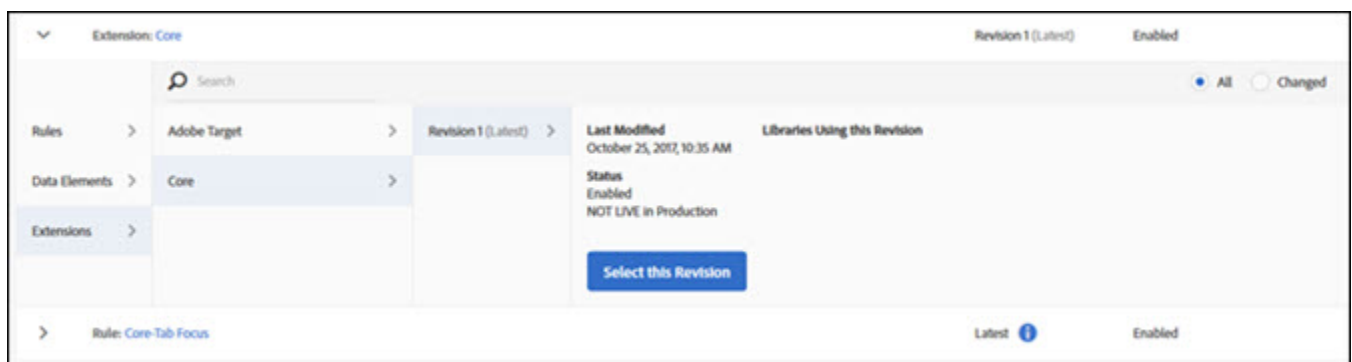


The screenshot shows the 'Library Contents' page. At the top, a blue banner states: 'This library contains all items (rules, data elements, and extensions) of the previously submitted and published libraries. The items below are changes that will be applied when published.' Below this is a table with three columns: 'Change', 'Revision', and 'Status'. The table lists three items: 'Extension: Adobe Target' (Revision 1 (Latest), Enabled), 'Extension: Core' (Revision 1 (Latest), Enabled), and 'Rule: Core-Tab Focus' (Latest, Enabled). At the bottom, there are two buttons: 'Add a Change' and 'Add All Changed Resources'.

Change	Revision	Status
> Extension: Adobe Target	Revision 1 (Latest)	Enabled
> Extension: Core	Revision 1 (Latest)	Enabled
> Rule: Core-Tab Focus	Latest	Enabled

+ Add a Change + Add All Changed Resources

2. Click a change to view and select a revision.



The screenshot shows the 'Extension: Core' page. The top bar indicates 'Revision 1 (Latest)' and 'Enabled'. Below this is a search bar and a filter toggle for 'All' (selected) and 'Changed'. The main content area is divided into two sections. On the left, there is a sidebar with 'Rules', 'Data Elements', and 'Extensions'. The 'Rules' section is expanded, showing 'Adobe Target' and 'Core'. The 'Core' rule is selected, showing its details: 'Last Modified: October 25, 2017, 10:35 AM', 'Status: Enabled', and 'NOT LIVE in Production'. A 'Select this Revision' button is visible. At the bottom, there is a 'Rule: Core-Tab Focus' entry with 'Latest' and 'Enabled' status.

3. Select whether to show all items or changed items.
4. Click the revision, and then click **Select Revision**.
5. Click either Add a Change or Add All Changed Resources.

Active Library

Libraries encapsulate a set of changes you'd like to make to your deployed code. Active Library makes this easier, allowing you to rapidly iterate through changes and see the impact.

You can save new and existing extensions, data elements, and rules directly to the library you're working on and, if desired, immediately kick off a build. You can also create a new library from the Active Library drop down.

1. Create a new **library**.
2. Go to **Extensions**, **Data Elements**, or **Rules**
3. Select your **Active Library**.
4. Make your changes, then save and **build the library**.
5. **Test** your changes, and repeat these steps as

Builds

A build is the set of files containing all the code that runs on your website.

It is a composite of the changes you specified within a library, as well as everything that has been submitted, approved, or published before it.

The build consists of one or more JavaScript files that reference each other. These files are delivered to your hosting location using the environment and adapter that you have chosen for the library. The embed code that you deploy on your site points to this same location so the files can load in a browser when a user accesses your site.

Promote to production

Once you've tested your build in your dev environment, the promotion process is pretty straightforward. Before you try it out, make sure to create your stage and production environments and put the embed codes in the necessary places. (You can reuse existing adapters.)

Promoting a library all the way through to production will typically require coordination among different people with the appropriate rights.

1. A Developer (someone with the Develop right) submits the library, which moves the library to the Submitted state.
2. An Approver (someone with the Approve right) can build the library to the stage environment and can approve it after testing. This moves the library to the approved state. **Only one library can be submitted and approved at a time.**
3. A Publisher (someone with the Publish right) can build the library to the production environment.

You can assign all these rights to a single person.

For more information about the different states and options available during the publishing process, refer to the Approval Workflow below.

Approval Workflow

The Approval workflow refers to the process of creating libraries, testing builds, and approving them for production. The available actions depend on the state of the library and the level of permission you have.

Permissions

There are three levels of permission that are important for the approval workflow:

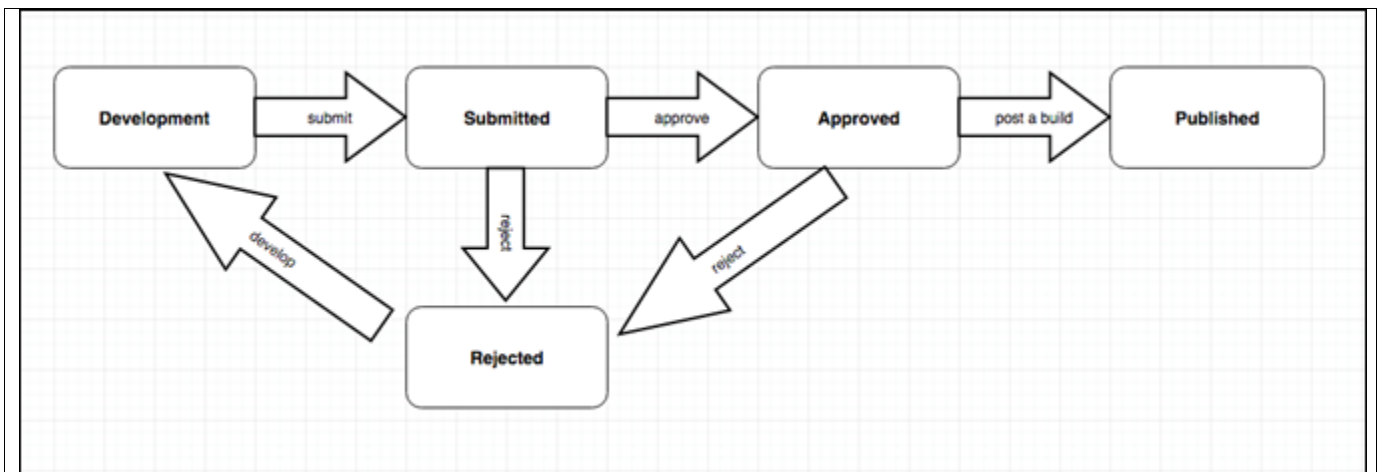
- **The develop right:** Includes the ability to create libraries, build for development, and submit for approval
- **The approve right:** Includes the ability to build for staging and approve
- **The publish right:** Includes the ability to publish an approved library

The rights are not inclusive. For a single person to perform the workflow from start to finish, that person must be granted all three rights within a given property.

Library States

There are four basic states that a library can be in. Specific actions must be taken to move a library between these states. These four states are represented as columns within the Publishing tab.

- Development
- Submitted
- Approved
- Published



Development

Libraries are created in Development. Any changes to a library must be made while the library is in Development. When development and testing are completed, the library is submitted.

Available actions for a library in Development state are:

- Edit

Use the library Edit screen to add or remove components from a library.

- Build for Development

Create a build for the library. The build is compiled and deployed to the environment the library is assigned to. This step fails if the library has not been assigned to an environment.

- Submit for Approval

Unassigns the library from its development environment and moves the library to the submitted column for an Approver to work on.

Submitted

An Approver tests the library in a staging environment. When testing is completed, the library is approved or rejected. Rejected builds go back to Development so changes can be made before the approval flow starts over again.

Available actions for a library in Submitted state are:

- Open

View the contents of the library. **Changes are not allowed**. If changes are needed, the library *should be rejected* so changes can be made in Development.

- Build for Staging

Assigns the library to the staging environment and deploys it.

- Approve for Publishing

Moves the library to the Approved column for a Publisher to publish

- Reject

Unassigns the library from the staging environment and moves the library to the Development column for changes.

Approved

The library is waiting to be published. A Publisher can publish or reject the library. Rejected builds go back to Development so that changes can be made before the approval flow begins again.

Available actions for a library in Approved state are:

- Open

View the contents of the library. **Changes are not allowed**. If changes are needed, the library should be rejected so changes can be made in Development.

- Build and Publish to Production

Unassigns the library from the staging environment, assigns the library to the production environment, and deploys it.

Important: When this option is selected, your library becomes live in your production environment, so be sure before you click the button.

- Reject

Unassigns the library from the staging environment and moves the library to the Development column for changes.

Published

The library is live in the production environment. This column shows which libraries have been published and their publish dates. You can look at these libraries, but **you cannot make changes**. If you want to change what's in your production environment, you'll need to create a new library and push it through the approval process.