Adobe Experience Platform Bootcamp Deep Dive Edition

DOCUMENTATION AND LABS

Contents

Adobe Experience Platform
Activating the Real-Time Customer Profile

Adobe Experience Platform Bootcamp Deep Dive Edition

Pre-Requisites

- Profile Enabled
- Data Loaded
- Previous DEP Labs completed

Introduction

In this Lab you will build using one of the core capabilities of the Adobe Experience Platform, Real Time CDP Segments and Activating to Destinations.

Segments and Destinations get their requirements from the business. Segments are the first place where we get to see the ramifications of our decisions made during Schema design. They are also the first place we will see how well we captured requirements during the design phase.

Destinations are the second place where we see how some of the denormalization we did comes into play.

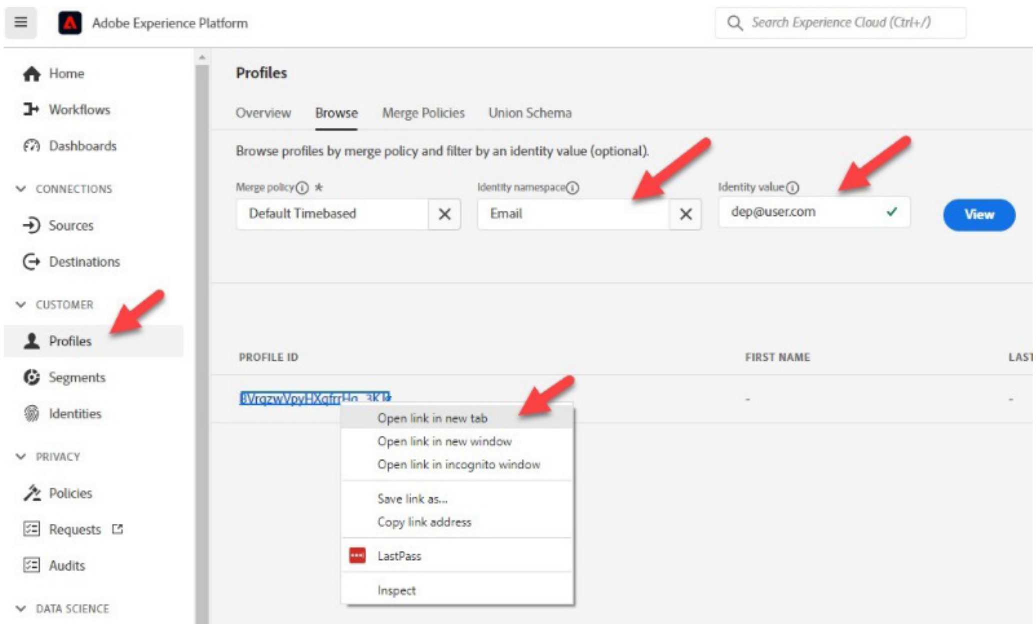
Building the Real Time Customer Profile

Inspecting the Profile

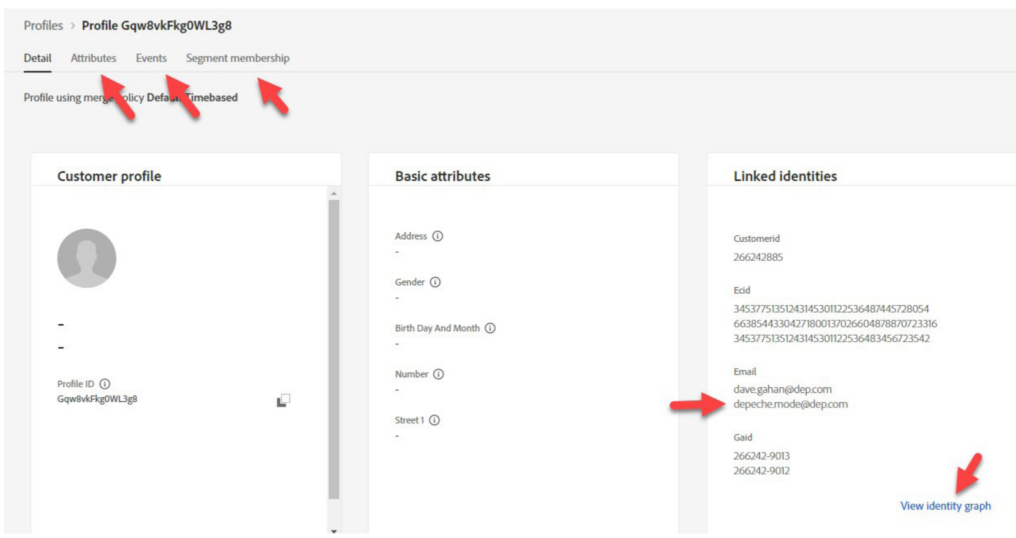
Before we dive into how RTCDP pulls together the Profile, let's take a look at its pieces.

Profile

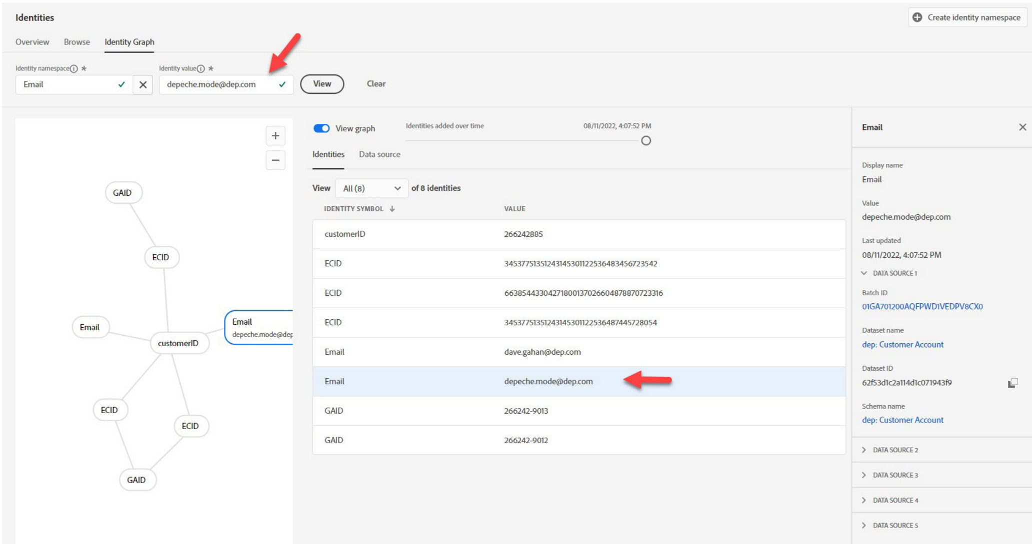
1. Open the Profile Viewer
   a. Put the Email address in of one of the Customers (who placed an order), you loaded in during the previous exercise.
      i. I'm going to use depeche.mode@dep.com
   b. Click View and open that Profile in a new tab



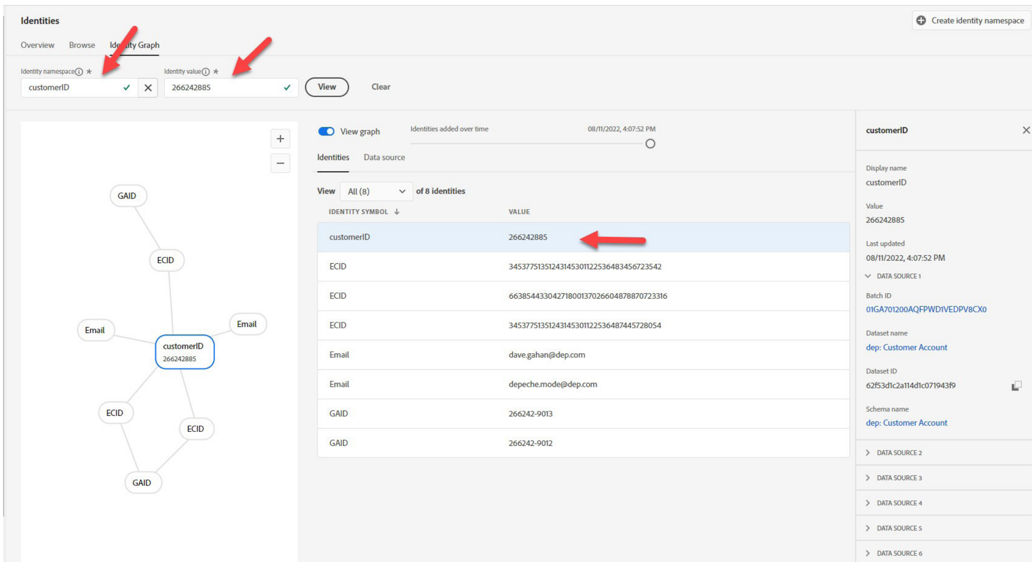
2. Click through the various tabs to see the resulting profile, attributes, events and segment membership. On the Detail tab, under Linked Identities, click on the View Identity Graph



3. Click on View Graph if it doesn't show up and you should see this:

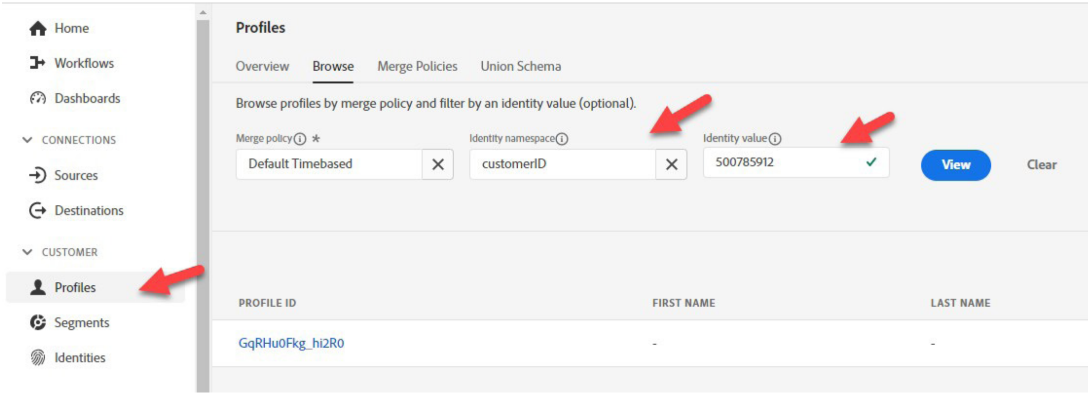


a. Here we see all the identities that are related to the customer we looked up using the namespace specified
b. Copy the Customer ID
c. Change the namespace to Customer ID and paste in the Customer ID. You still get the same results



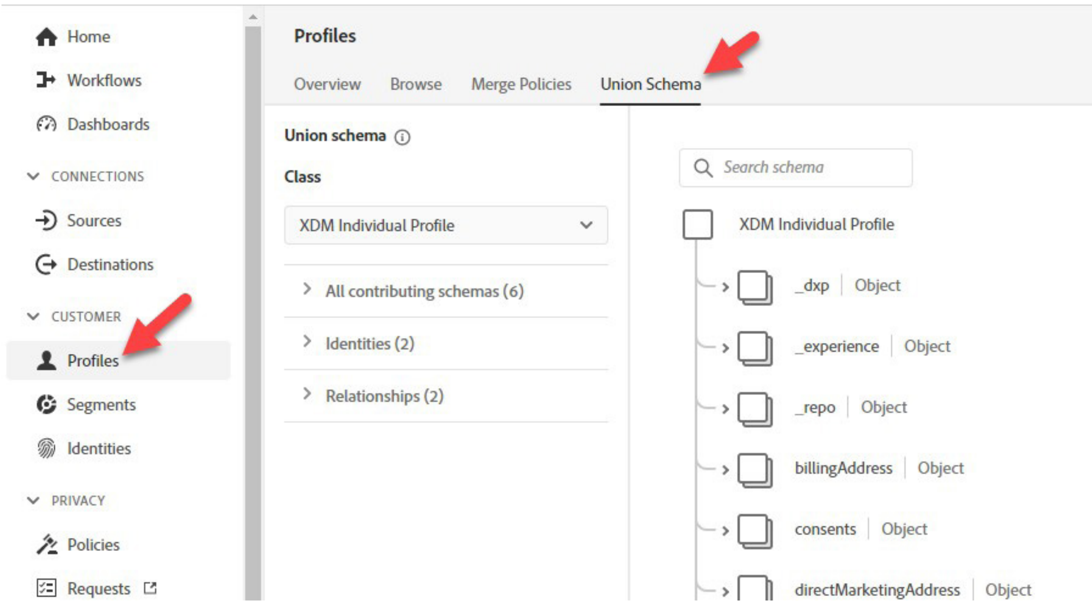
d. This is because they both refer to the same cluster

4. Does this mean the same thing will happen with the Profile Viewer? Let's Try
    a. Go back to the Profile tab on the left rail
    b. Change the namespace and value
    c. You will see they both refer to the same Profile



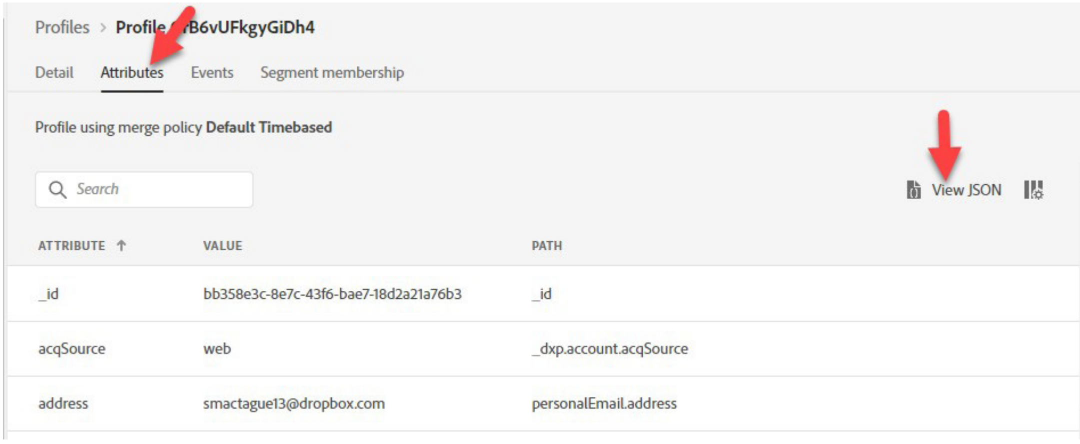
Mapping this back to all the Schemas we created

1. Go to the Profile Viewer and open the Union Schema



2. Click through different areas under _devbc and you will see how all our Schemas are pulled together in one view
    a. This is how the Profile will look as it pulls together all the datasets
3. Search for Profile depeche.mode@dep.com in the Profile Viewer and open that Profile

4. Go to the Attributes tab and click View JSON



a. Let's see how fields show up that came from your Field Groups you added to the Customer Account Schema

i. Scroll down to "entity"

1. Look for these:

a. "entity": {
"billingAddress": { "postalCode": "44511", "city": "Youngstown",
"state": "OH",
"street1": "5143 Dennis Point"
},

i. This came from [Personal Contact Details](#) Field Group

1. Notice not all fields on the Field Group are on the Profile, only the ones sent in on the payload

b. "consents": {
"marketing": {
"sms": {
"val": "y"
},
"email": {
"val": "y"
}
}
},

i. This came from [Consent and Preference Details](#) Field Group

ii. Scroll down to _devbc

1. Look for this:

a. "plan": {
"planID": "m1",
"name": "basic"

}

b. This came from a Custom dep: Plan Details Field Group

c. How about the Aggregates, where are they?

i. Also under _devbc

1. e.g.

a. “aggregates”: {
“rollingSixMonthAvgMonthlyDataUsage”: 30,
“rollingSixMonthTotalDataUsage”: 400
},

b. This came from a different Schema dep: Customer Aggregates and a Custom Field Group dep: Aggregates

d. Can we see identities on the Profile Viewer?

i. Look for identityMap at the bottom

1. You should see something like this:

2. “identityMap”: {

“ecid”: [
{
“id”: “77258571638552156584254557533435172565”
}
],
“email”: [
{
“id”: “pbruneau18@icio.us”
},
{
“id”: “smactague13@dropbox.com”
}
],
“customerid”: [
{
“id”: “333074888”
}
]
},

This is a representation of all the identities we have passed in (regardless if we used the identityMap or a field with an identity Descriptor

Note: there is no reference to "primary" identity even though we flagged it on the schema or data (depending on how we passed it in)

i. The reason is two-fold
   1. The identity Graph only cares about relationships between identities. Each identity is treated the same. A is related to B and it does not matter if it was via a Primary identity.
   2. The Profile Service only stores fragments against the Primary identity. It does not treat other identities as special.

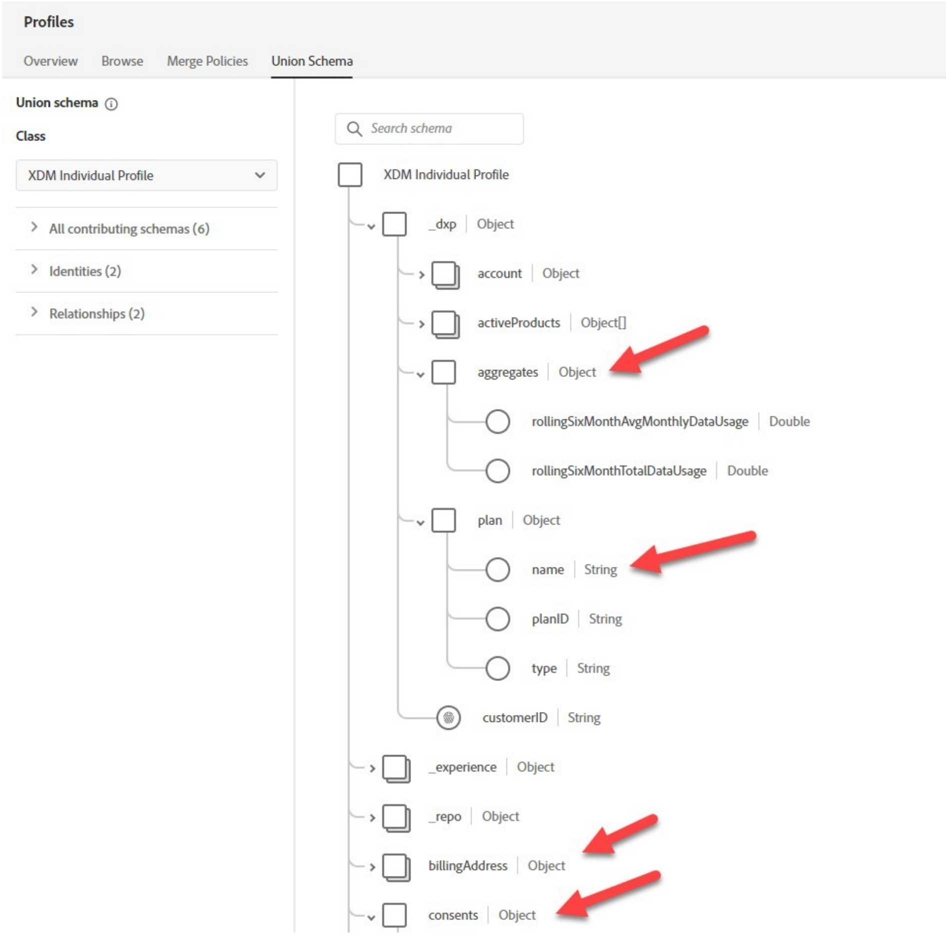
Note: Did you notice the IdentityMap has two email addresses?
Before looking, how many email addresses do you think are on the Profile? When we built our Schema, we only selected one "personalEmail.address".
Go look and you will find:

```
"personalEmail": {
                "address": depeche.mode@dep.com
                 },
```

What is going on?

- Identity Graph is constantly adding values to itself keeping every value it encounters
- Profile overwrites field values as it encounters them
- When we flag a field with a Descriptor as an identity, it is still a field for Profile
- For these fields, the new value is added to the ID Graph and the new value overwrites the previous value in Profile

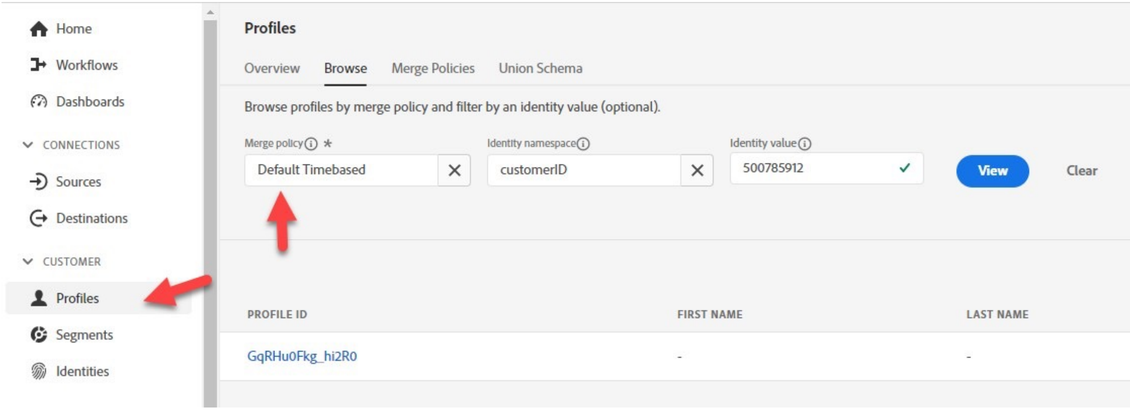
5. If you go back and look at the Union Schema View we can see how the pieces of Schemas and Field Groups all come together



a. Even though we have different Schemas, Field Groups and datasets, Profile brings it all together for us

Merge Policy

1. What is this Merge Policy? We see it in the Profile Viewer when we search for a Profile.



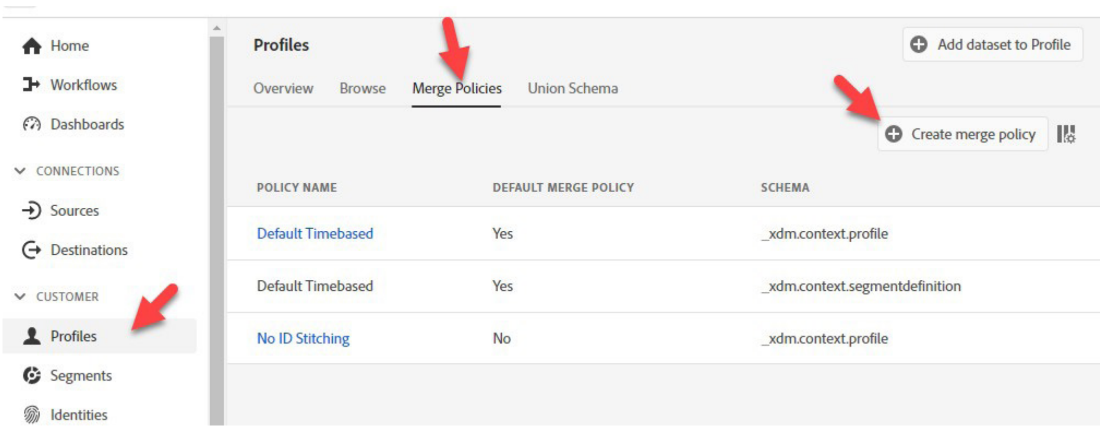
2. To best understand how Merge Policies work with the ID Graph, we are going to create one and use it.

Note: Remember a Merge Policy does two things

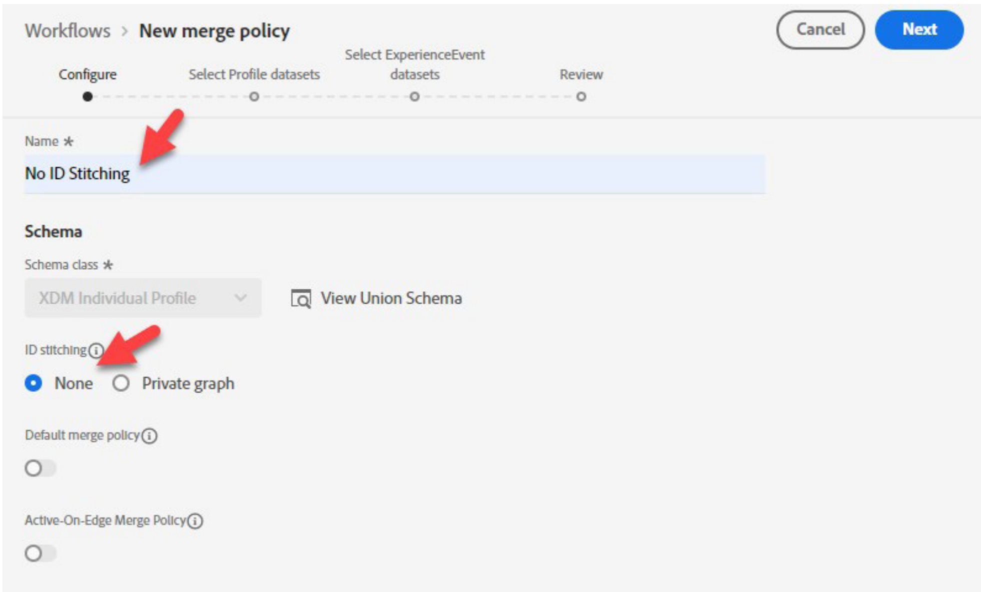
i. Allows us to specify an ID Graph or No ID Graph
   1. Link identities based on the data we passed in or ignore them

ii. Tells us how to resolve field conflicts when a field may come from multiple datasets (but all have the same field, e.g. three datasets all have first name, which one do I use?)
   1. Merge Method (i.e Field Conflict Resolution)
      a. Timestamp Precedence
         i. Most recent/current value is the value to use
      b. Dataset Precedence
         i. Use a particular dataset if it has the value, a different one if it doesn't
      c. Note: Many times we design around this so that we don't need to use a Custom Merge Policy that uses Dataset Precedence for this
         i. Rather than having multiple datasets record the same field, we give them unique names
         ii. e.g.
            1. First Name - CRM
            2. First Name - Loyalty
            3. First Name - Web Form
         iii. This allows a marketer to choose which datasource+field rather than the system choosing one based on a ruleset they may not understand and possibly choosing pieces of the Profile from one source and other pieces from another.
         iv. For our design we do not need to resolve any field conflicts

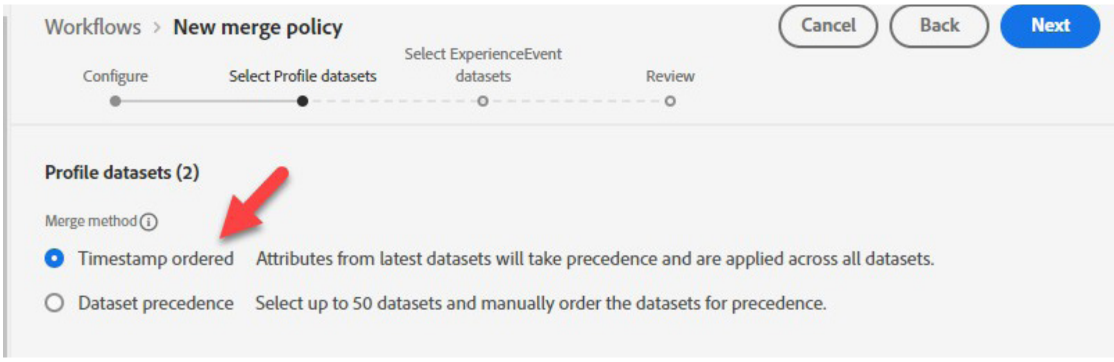
3. Why use No Stitch?
    a. Let's create a Merge Policy that doesn't use the ID Graph so we can see how it behaves
    b. Go to the Merge Policy tab on the right rail
    c. Create a New Merge Policy



d. Name = No ID Stitching
e. Select ID Stitching = None
f. Next



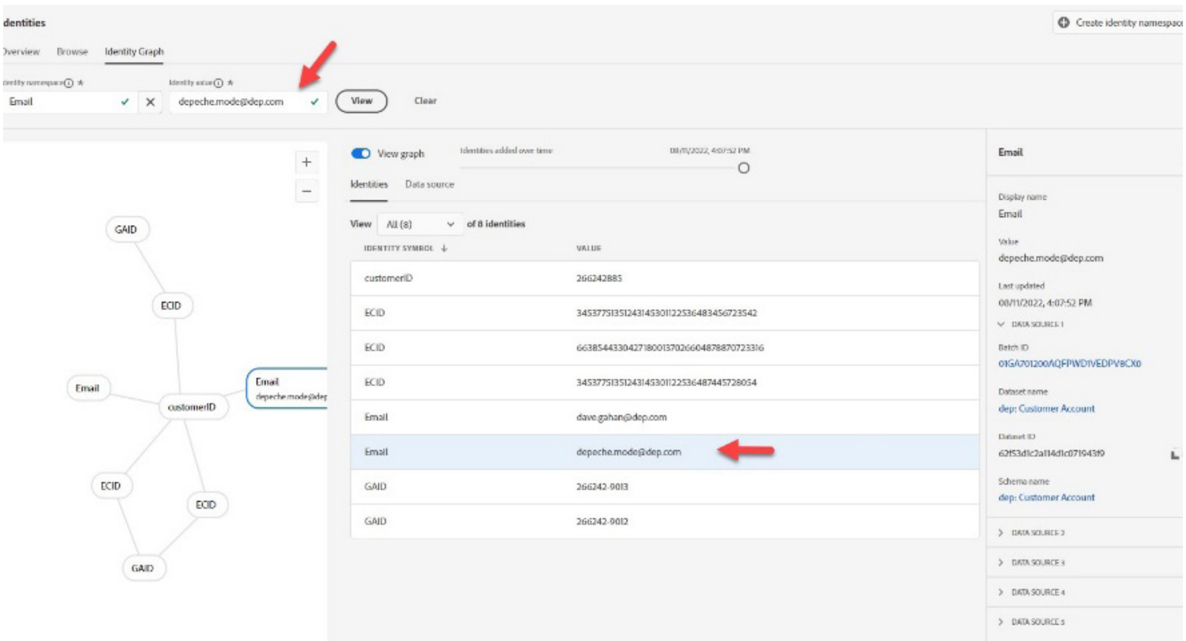
g. Select Timestamp



h. Next and Save

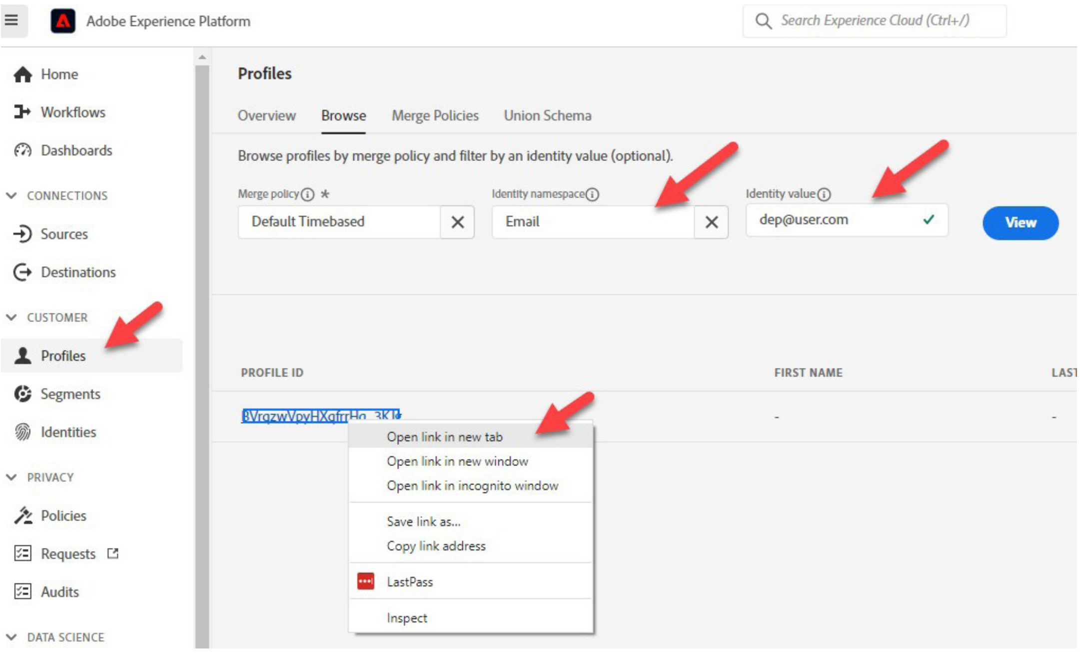
ID Graph

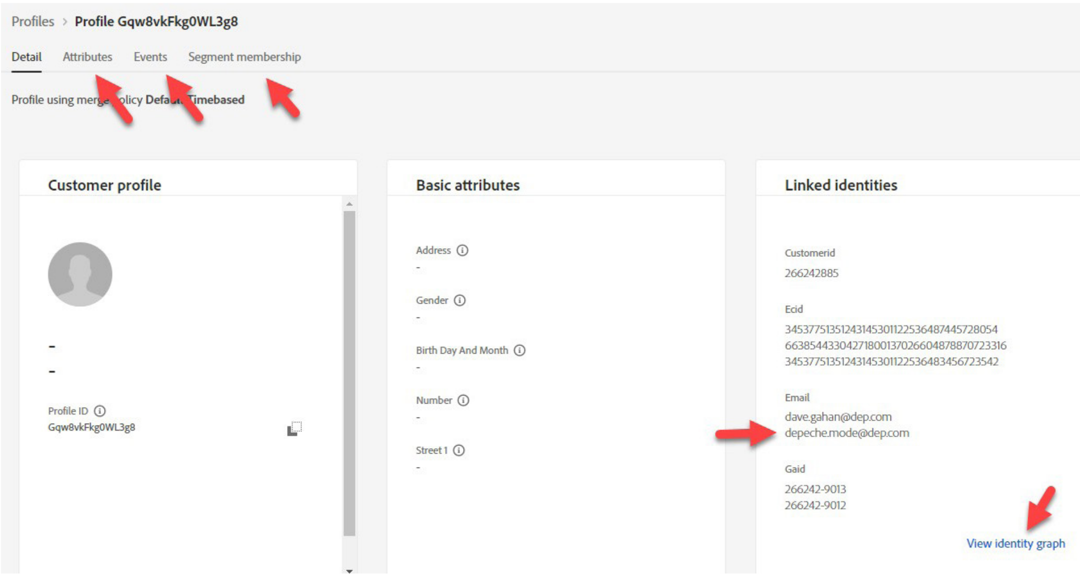
1.Remember we saw in the ID Graph Viewer the relationship between identities



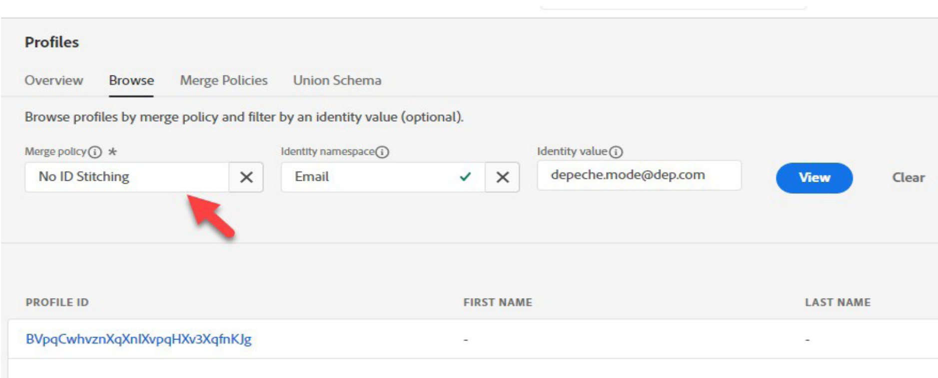
2. The best way to see the ID Graph in action is to see the Profile Viewer

a. Open the Profile viewer for the Customer in 1a in a new tab

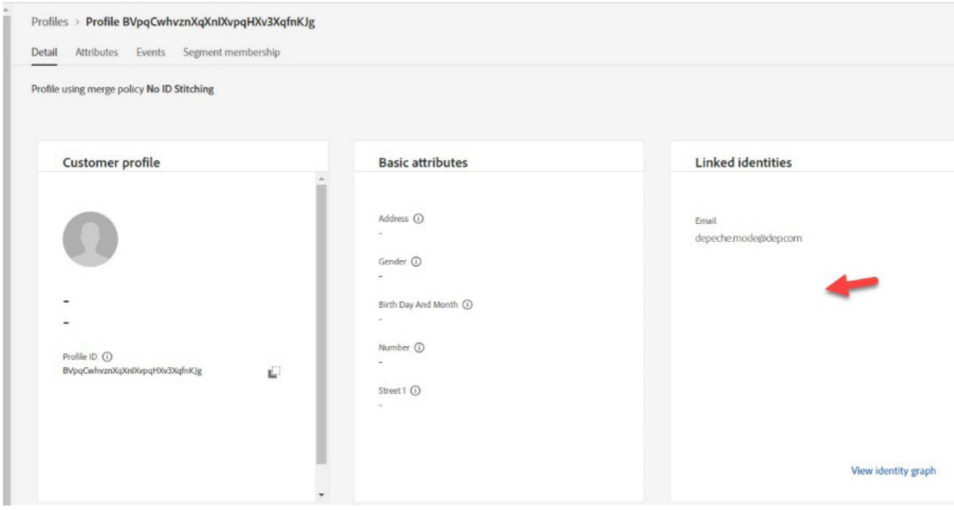




b. Go back to the original tab and do another search for the same Customer using the No ID Stitching Merge Policy



c. Open the Profile in a new tab, look at the identities, what do you see that is different?



i. Some of the identities are missing

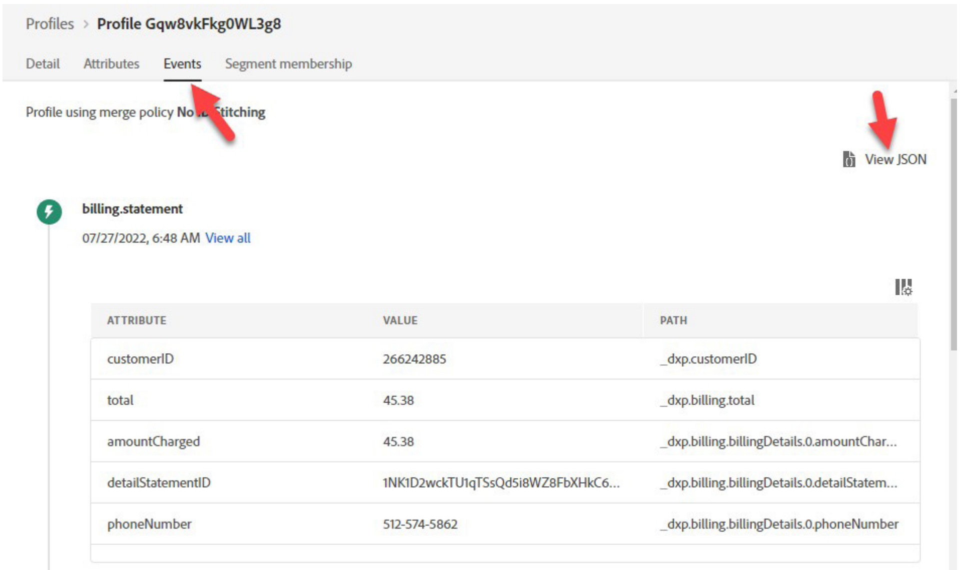
1. Why? because we said to not use the ID Graph and Profile only stores data and thus “remembers” one identity, the Primary identity

d. Look at the Attributes

i. Everything looks fine

e. Look at the Events

i. Seems fine but look click on View JSON



ii. What you will notice is the only events that show up are the ones that have Email as the primary identity

i. Go back to the Profile Search and change the namespace to Customer ID and the value to your Customer ID

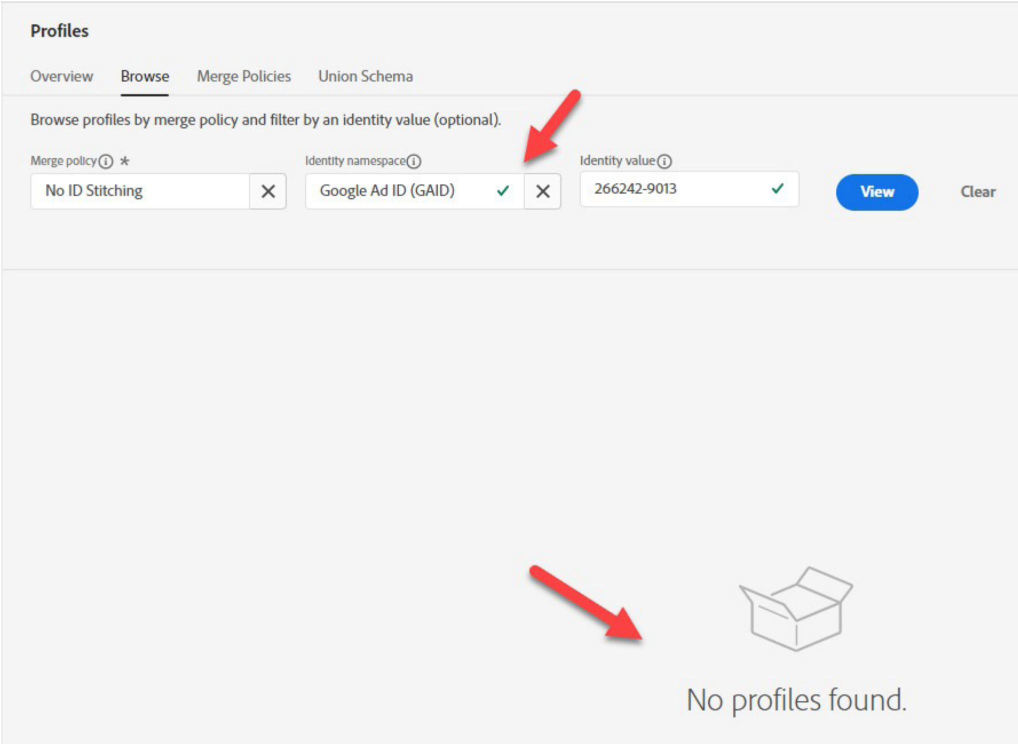
j. Notice anything about the Attributes?

i. There are none, why? Because we load attributes using Email as our primary identity

k. How about Events?

i. What you will notice is the only events that show up are the ones that have Customer ID as the primary identity

l. Do a search now on any of the GAID identities



m. No Profiles are found

What is going on? Why are no profiles found?

i. The ID Graph allows us to traverse it to get to all the Profile fragments regardless of the identity searched/used

ii. However, Profile stores and keeps data ONLY against the Primary identity

1. If we pass in an Event with two identities,

a. Profile only stores the Event using the Primary identity

b. Profile only uses that Primary identity to retrieve that fragment

iii. In our example, we have no Profile or Events stored with that GAID as a Primary Identity

iv. We do have one stored with an Email and other fragments with Customer ID

v. Thus, we can find the Profile (fragment) using the Email or Customer ID, but cannot with GAID

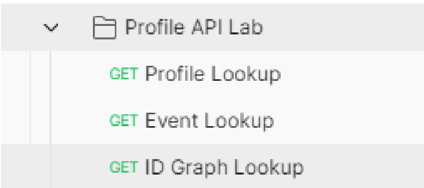
vi. And we do not know the two are related, because we said not to use the Identity Graph

Have some extra time and want to experiment… try:

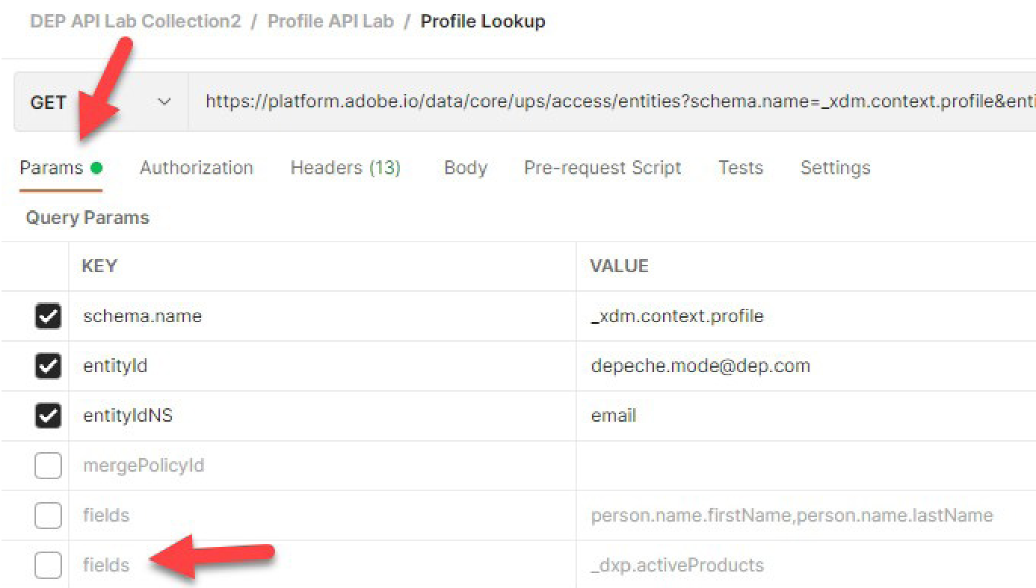
- Searching other profiles in the UI you know of that have two identities
  - See how some Events are stored against one fragment but not the other
  - See how some Profile attributes are stored against one fragment but not the other
  - Go to the Profile you have open in a tab and find or add this to the URL
  - mergePolicyId=[Your No ID Stitching Merge Policy]

Profile API

1.Open up Postman and go to the Profile API folder



2. Run the Profile Lookup
   a. Notice how the entire Profile comes back in the result
   b. Click on the Params tab and select the fields



   c. Run it again and you will notice this returns fewer fields
   d. Notice we can specify the Merge Policy

3. Run the Event Lookup
    a. Notice how tons of Events come back in the result
    b. Click on the Params and select some of them
        i. fields = returns fewer fields
        ii. property = filters which Events to return
        iii. orderby = descending
        iv. limit = limits how many Events to return
4. Run the ID Graph Lookup
    a. This returns just the identities in the ID Graph

> These APIs can mostly be done in the UI as well and thus you can use that.
>
> However, the ability to filter, limit and order Events is not in the UI and very handy for troubleshooting.

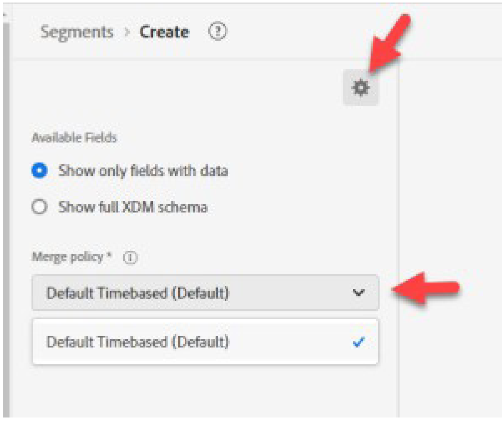
Break/Wait Here

How Does RTCDP Pull It All Together in Segments?

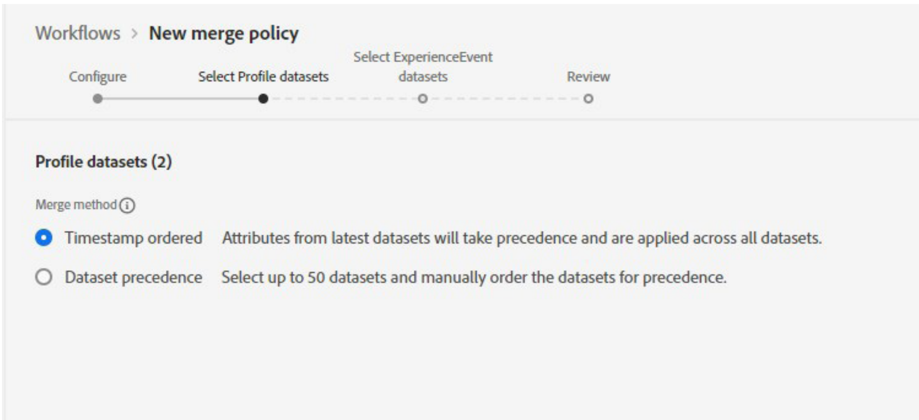
Segments run on the Real-Time Customer Profile. Everything we have done before this is design for the Profile. So let's walk through how these worlds come together before designing any Segments:

Pre-Read (understanding how the Profile comes together for Segmentation)

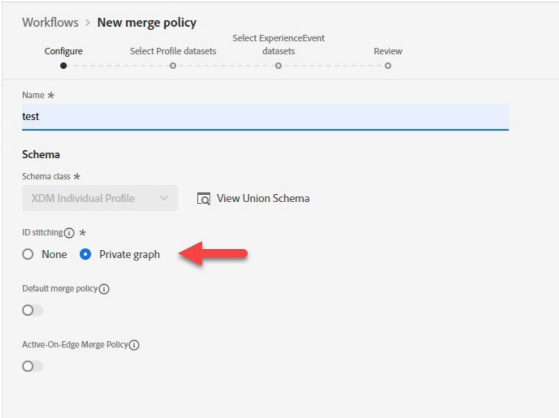
1. The ID Graph tells the engine which identities are related
   a. e.g. CRM ID = 123 is related to ECID = ABC
2. The Segment defines which Merge Policy to use and the filter criteria
   a. e.g. Use the default Merge Policy
   b. e.g. Use the No Stich Merge Policy



3. The Segments provide a rule/criteria
   a. e.g. Event Type = Purchase
4. The Merge Policy tells Profile Service if it should use an ID Graph, which datasets to use and how to resolve any naming conflicts (time or dataset precedence)
   a. e.g. Use timestamp precedence
   b. e.g. Use Loyalty Dataset if data exists, but CRM Dataset if it doesn't



c. e.g. Ignore the ID Graph (No Stitch)



5. The Profile Service brings together the various pieces/fragments based on what the Segment is looking for (Event Type = Purchase) and the Merge Policy specified (using the ID Graph or not). It will lookup each fragment using the primary identity (it stored each fragment with the primary identity as the key).
   a. e.g. Segment Use the default Merge Policy (timestamp precedence) and Event Type = Purchase
      i. Profile ignores the CRM dataset since it isn't needed for any filter
      ii. Profile brings in the Events needed for filtering
      iii. Profile overlays any conflicting fields based on most recent timestamp
6. Segmentation uses the final Union View of the Profile to filter based on the rules in the Segment
   a. e.g. Event Type = Purchase
7. Segmentation writes the results back to the Profile on the Segment Membership
   a. This will include the status and timestamp of when they enter or exit the segment
   b. Note: Edge Segmentation only updates the Edge Profile. The Segment will also be run as a Streaming Segment on the Hub and that will write to the Profile on the Hub
   c. Note: The result of a Segment is only captured if you enter or exit a segment, not if you don't qualify for it
   d. Note: Since the result is stored on the Profile, it loses context of what made it qualify for the Segment.
      i. e.g. If a Segment is has at least one Event Type = Purchase, the result is yes or no for the Profile. It does not store which Event qualified it for the Segment.
8. Triggers will fire based on Segment Membership being updated
   a. e.g. Destination or AJO may use this Segment

Segments Scope/Focus is on One Customer Profile at a Time

Every Segment is focused on a single Customer. Which means underneath the covers it automatically does two things:

1. Applies the filter to each Customer Profile one at a time
   a. No need to add a Group By Customer clause to prevent collapsing counts/events/etc. across Customers
2. Prevents you from building Segments that compare Customers to each other
   a. e.g. Segment of Top 10 Revenue Generating Customers
      i. Note: You could build an offline query to calculate this and store a number 1-N and build a Segment on that
   b. e.g. Segment of Highest Ranking Propensity to Convert Customers
      i. Note: You could build an offline query to calculate this and store a number 1-N and build a Segment on that

Break/Wait Here

Use Cases

These are the use cases that Marketing has decided they want to build:

1. **Social Media Activation: Nurturing Warm Audiences**
   Activate all profiles who have visited an iPhone 13 product page and no order exists for an iPhone 13 or do not have an active iPhone 13 Activate all profiles who have visited a Pixel 6 product page and no order exists for a Pixel 6 and do not have an active Pixel 6
2. **Unlock New Opportunities: Upsell Ultimate Phone Plan**
   Find all customers who have a total billing data usage in the last 6 months >140GB, a rolling 6 month avg. monthly data usage of >=20GB and they do not have an ultimate phone plan.

   Activate into Facebook / Google and Direct Mail channels.

For each Use Case we will break it down in these areas:

1. Analysis
   a. This is taking the business requirements and understanding what is missing or asking clarifying questions
2. Approach
   a. This is taking what we know and laying out an approach for building
3. Build
   a. This is doing some discovery on what fields are available and can address the requirements
   b. Then building the Segments

Use Case #2 (Social Media Activation: Nurturing Warm Audiences)

Activate all profiles who have visited an iPhone 13 product page and no order exists for an iPhone 13 or do not have an active iPhone 13 Activate all profiles who have visited a Pixel 6 product page and no order exists for a Pixel 6 and do not have an active Pixel 6

Analysis Tasks

Analyze the above and write down:

1. What fields you think are needed to address this use case?
2. Does the segment need to be Streaming?
3. What are the ramifications of Streaming when Events being used in the Segment come in at different times?
4. How do we know what "active" means?
5. What other information would you like to know?

Remember: When we get requirements from the business stakeholders, they tend to be incomplete, use another terminology and make assumptions without knowing it. It is your job to bring as much of that to the surface and guide them to something that can be done.

Approach

For this use case we are going to break it down into multiple Segments:

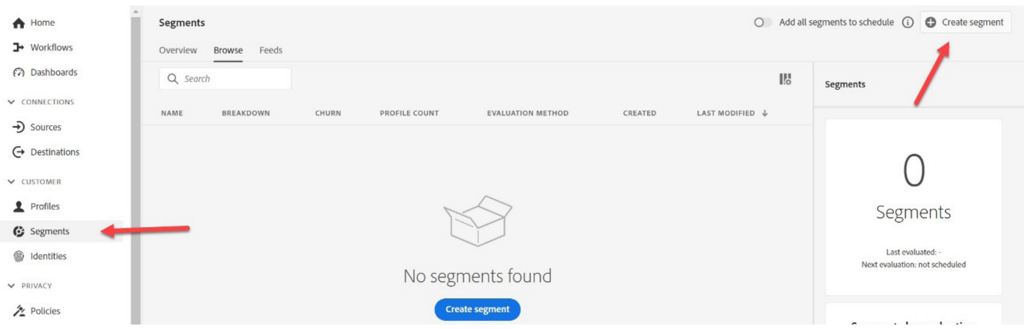
1. No Order exists iPhone/Pixel
2. No Active iPhone/Pixel
3. Visited iPhone/Pixel & No Order exists iPhone/Pixel & No Active iPhone/Pixel

Build Tasks

Segment #1 (Placed Order iPhone 13)

Pre-Work (breaking down the Segment and finding the fields)

Let us start by creating our first segment. It is composed of many pieces that we need to incorporate. Click on segments on the left rail and create Segment button in the top right.



We are going to break this Use Case up into pieces and solve them with multiple Segments. The reason for this is we are trying to make this a Streaming Segment and two things are preventing this:

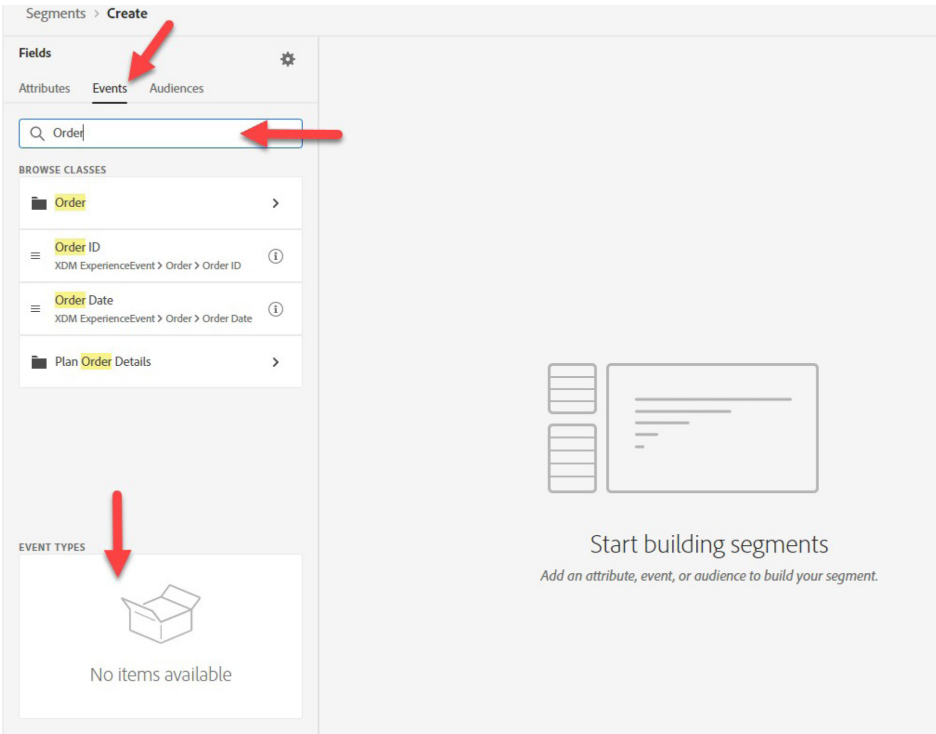
1. The exclude clause "no order exists for iPhone 13/Pixel 6"
2. The exclude clause "no active iPhone 13/Pixel 6" We will go over the ramifications of this at the end.

We will go over the ramifications of this at the end.

The first part of our segment is to look for "no order exists for an iPhone 13"

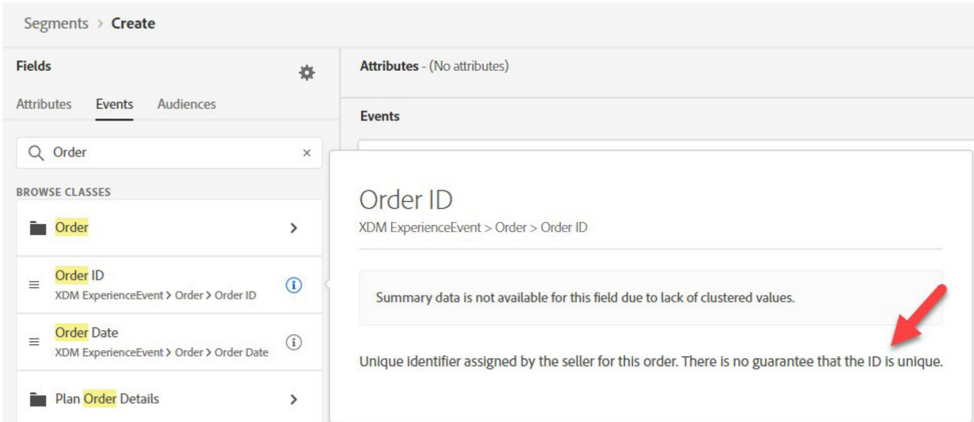
1. Where do we find the field to filter on?

    a. Imagine we are a new Marketer to AEP and did not design the Schema
    b. Do a search for "Order" in the Events tab on the left rail



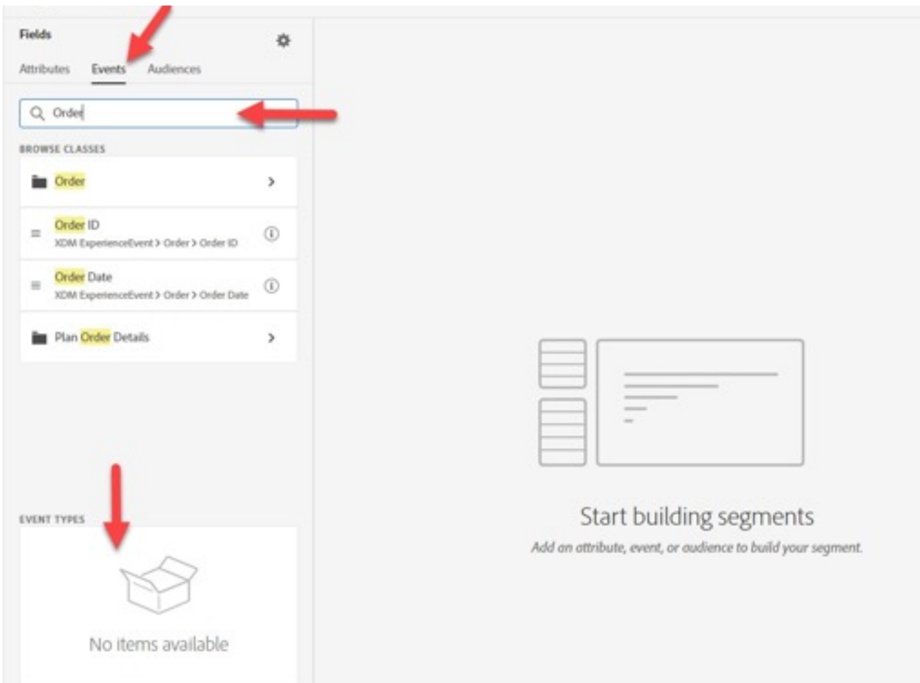
    c. We get:
        i. An Order folder
        ii. A Plan Order Details
        iii. But no Order Events
    d. Which one do we choose?
        i. Event Types
            1. No Order Events show up on the left rail under Event Types
        ii. Browse Classes
            1. Looking at/hovering over the folder here doesn't tell/show us anything

2. Click on the Order folder and drill down, nothing looks like an Order Event, but it does have attributes of an Order. Close but not quite what we are looking for.
    i. Click on the "i" for each attribute to learn more about it

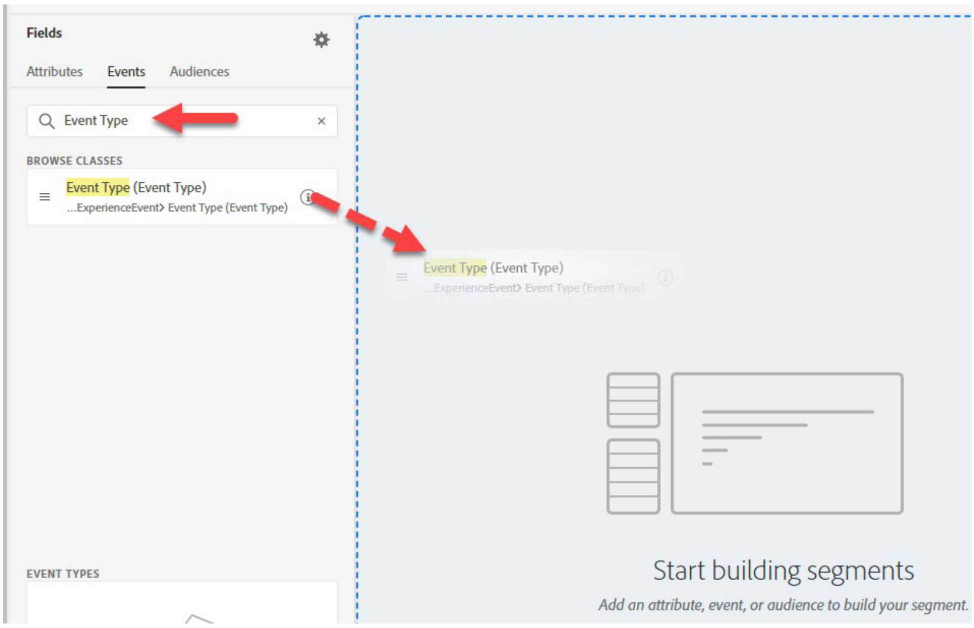


Note: there is no "i" for the Order "folder". Even though our description has been populated, it does not have it and this may be a source of confusion for your Marketer as they may try and use that or want to know what it is.
Note: Summary data will only show if the value is present in more than 2% of merged profiles. This also drives any autocomplete when filtering on a String.

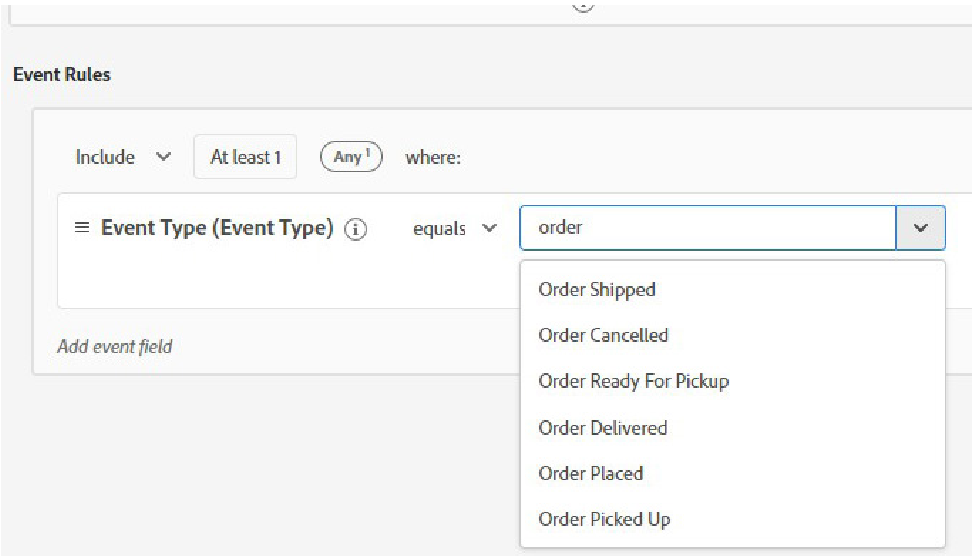
3. Where do we see the Order since we don't see any Order Events in the Event Type?



4. Remember that every Event has an Event Type, let's search on Event Type and filter on it



a. Drag that onto the canvas
b. Type in Order



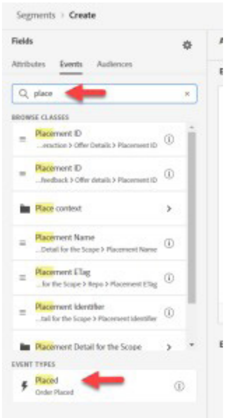
c. Now we see six different Event Types of Order
d. Which one tells us "no order exists"

i. This is a business decision, for now we are going to use Order Placed

> **Note:** Remember back to when we extended the Order Schema Event Type. We added the values we now see in the Segment drop down 1. When we did this every value in the drop down should show up as an Event in the Event Type (both in the drop down and as a card on the left rail)

e. So why isn't these Order Event types showing up in the left rail?

i. Type in "Place" instead of Order in the search



i. Now we see it
ii. Type in a few others "Cancel", "Delivered"
iii. They all show up

> Why?
> 1. We decided to use the naming convention in Event Type as order.placed. Should we have used placed.order?
> 2. **Note:** When you put your cursor over the "i" on the Event, you won't get any additional info

5. We have seen two ways now to filter on Order Type
   a. Using the Event Card
      i. Benefit: The Event Card shows the name of the Event Type
   b. Using the Event Type field
      i. Benefit: Allows selecting multiple Event Types (e.g. "Order Picked Up" or "Order Delivered") if we wanted to include multiple types in our criteria

Now we have decided on our approach, but our Marketer knows from their training that more than one data sources was loaded:

- **Orders** (captured by the Order system across all channels)
- **Web** (client side tracking of what people are clicking on, including Orders placed on the site)
- **eCommerce** (captured by the eCommerce system on the site)

Which source should we use?
They all logically represent the same event "Order Purchase". But they physically stored in different systems. How do we know which to use? The best way is to look at the descriptions on each Schema object and each field to know.

> Descriptions should have relevant info to help make these decisions, such as:
>
> 1. Where is the data coming from?
> 2. What does it contain or not contain?
> 3. What is the latency?
> 4. Has any system been designated "source of truth"?
> 5. Are there any nuances we need to take into consideration?

For us, we want to use Order Purchase, but keep in mind, depending on our use case we could have had the following requirements, that may influence which source we pull from:

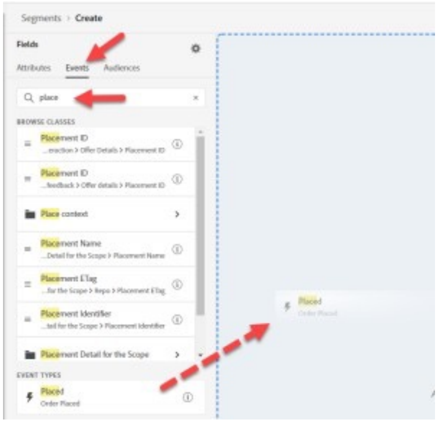
- Onsite purchases in last 30 minutes
- Orders placed and not Cancelled
- Orders picked up within 1 day of being Ready

> Optional thought exercise, imagine we placed a single Order on our site today (remember that Order is recorded by all three systems):
>
> 1. How many events would be counted if we filtered on Order Date = today?
> 2. How many events would be counted if we filtered on Shipping Method = overnight (assuming they chose this)
> 3. How should we address this (Segment or Data Model)?

After doing some analysis, we are going to go with the Orders Event of Event Type="order.placed". We want to ensure our segment using the source of truth at the tradeoff of speed (the web data streams in with each click while the Order goes through some processing before sent). Plus in the future we may want to exclude those who Cancelled and that could be done through any channel.

Start Work (actually build the segment)

1. Create a new Segment
2. Search on "placed" on Events tab
3. Drag Placed Event onto the canvas

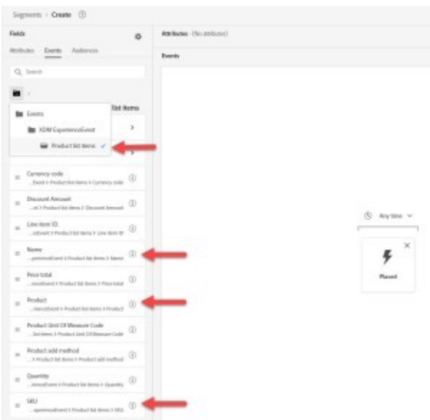


4. Click on the Placed Card
5. Clear "placed" from the Search on the left rail and drill down into the XDM Experience Event -> Product List items folder

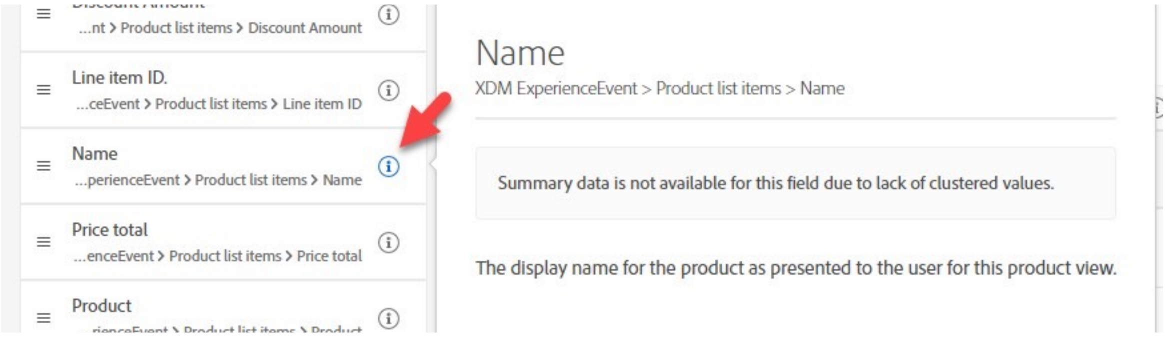
> A common confusion for your Marketer would be to use Device here (since we will filter on iPhone). Again, another reason for good descriptions.

a. We are looking for something that we can filter on Product
b. Notice we have three options

i. Name
ii. Product
iii. SKU



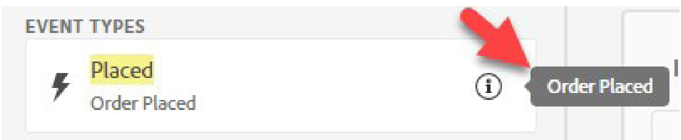
b. Click on the “i” to see the description for all three



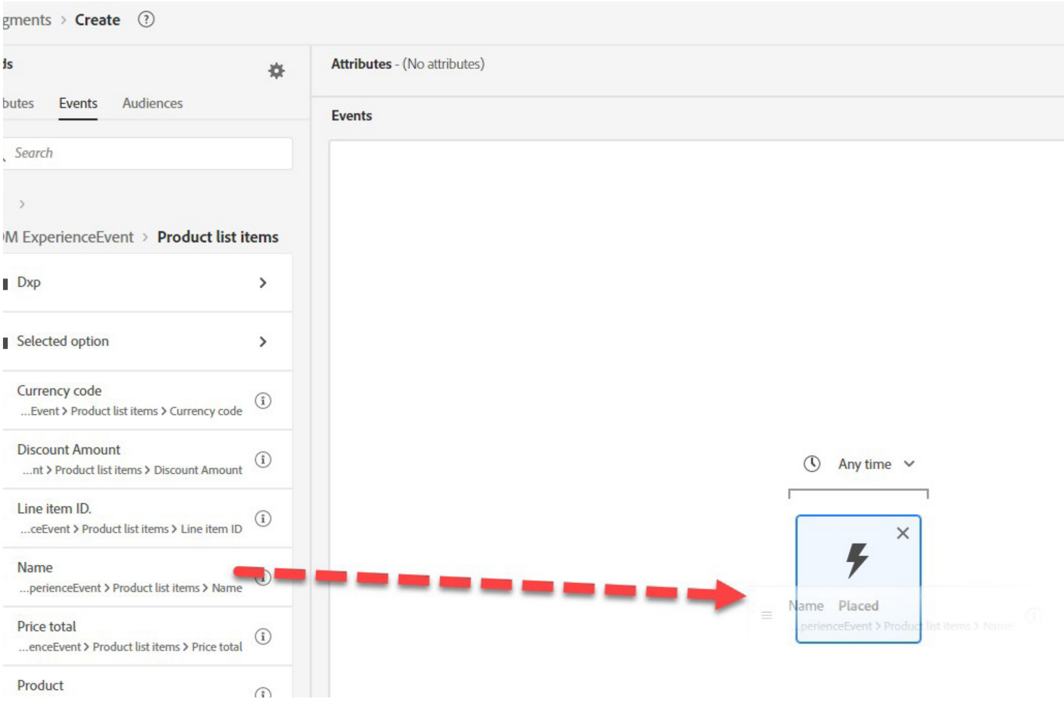
c. Name looks like it is probably the right one, but we don’t know. For now, let’s select it

- **Note:** you can change the descriptions for any OOTB fields, so update or even hide the fields not being used to reduce confusion for your users. These OOTB descriptions may not make sense in your industry/business.
- A good description might even contain examples
    i. e.g. Name Description = The display name for the product as presented to the user for this product view. For example: iPhone 13, Pixel 6
    ii. e.g. SKU Description = Stock keeping unit (SKU), the unique identifier for a product defined by the vendor. For example: iP13, Pix6
    iii. e.g. product Description = The XDM identifier of the product itself. For example: 123, 456

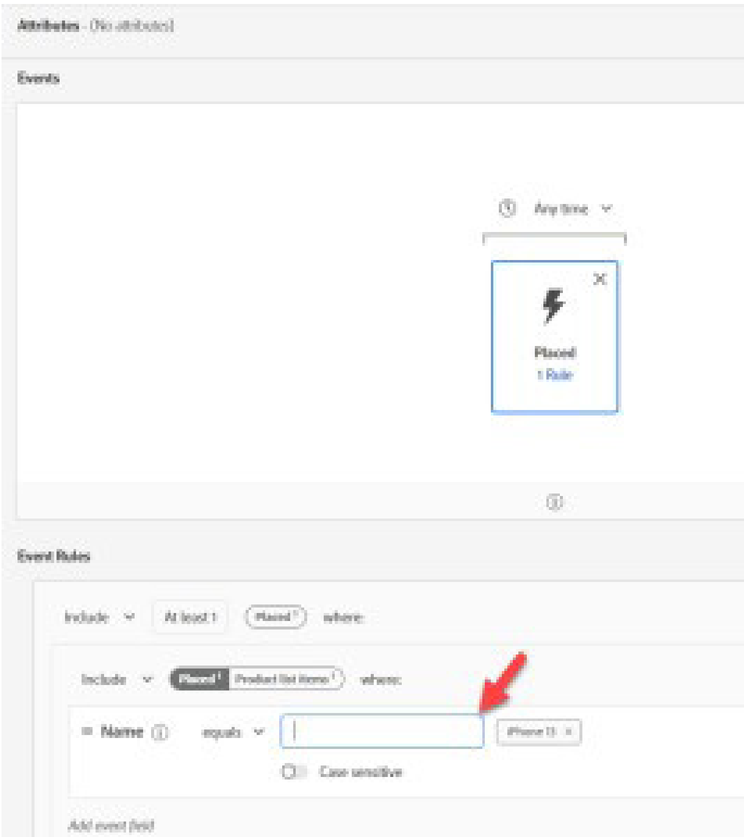
Note: the “i” over an Event Card has no additional description. Thus, make sure your Event Types are self documenting.



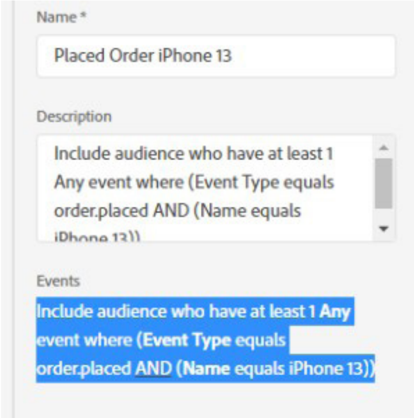
6. Drag the Name field onto the Placed Card



7. Add iPhone 13



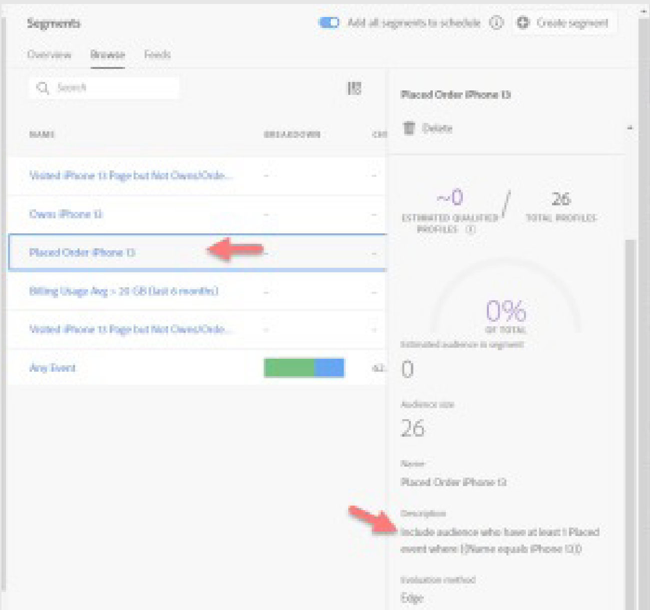
8. Copy the “pseudocode” version of the Segment Rule into the Description



Note: while this is a good practice it is not perfect, so **DO NOT FORGET TO UPDATE IT IF YOU CHANGE THE SEGMENT**

i. We do this because:

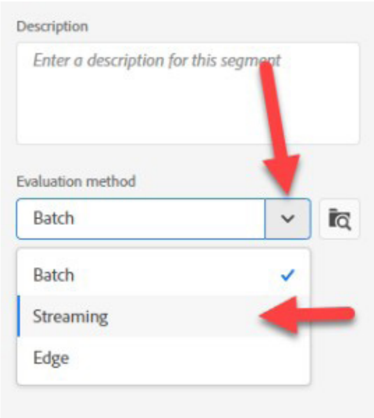
1. Descriptions follow the Segment across each downstream application
2. The Marketer can see the definition in the list of Segments without having to open each Segment



3. There is a common way of writing descriptions

ii. You can use your own method if you like

9. Change Segment type to Streaming



10. Save Segment as "**Placed Order iPhone 13**"

Go through the same steps but this time filter on "Pixel 6" and save the Segment as "Placed Order Pixel 6"

Containers

Notice the way we filter on Event Type and Name in the Product list items looks visually different. The reason for this is that Product list items is an Array data type Event Type is not. When filtering on an Array, a Container is created (called Product list items in our example).

Containers are a way of referencing an Event variable or Array element. You can read more about what the ramification of this is in this Blog, but for simplicity's sake, this allows you to specify if a single element in the array meets both conditions or the condition can be spread across two elements.

https://experienceleaguecommunities.adobe.com/t5/adobe-experience-platform-blogs/how-exactly-do-containers-work-in-aep-segmentation-a-deeper-look/ba-p/458780

Time Filters

While nothing is specified in the requirements, this Segment has a problem that we should go back and clarify with the business.

It has no time filter. What this means is that if someone placed an order a year or five years ago they would qualify for this. Always try to incorporate a method to ensure you don't fall into this trap or have to always update your Segments as the new version comes out.

If we add a time filter, how far back can we go before an Edge Segment becomes Streaming or even Batch?

Is Product stored in two places?

i. Note the different path naming convention and description. Compare it with the previous Segment

1. XDM Individual Profile > Dxp > Active Products > Product ID properties > Product Name
    a. Description: Name of the product.
2. XDM ExperienceEvent > Product list items > Name
    a. Description: The display name for the product as presented to the user for this product view.

ii. When we start storing the same value in different places for different reasons and purposes, we need to think through the ramifications to our Users and how the Profile will merge these (and how a Merge Policy will resolve this conflict if needed).

iii. Our current descriptions make it difficult for the Marketer to know which to use

Segment #2 (do not have an active iPhone 13)

Analysis Tasks

- How do we know someone does not "have an active iPhone 13"?
- To answer this question, we have some options: Include those who purchased an iPhone 13
- Include those who have Billing data for an iPhone 13
- Include those who have any web data coming from an iPhone 13 Any others?

In the end, this boils down to a business choice on who they want to market to. In our case, the company has deemed this so important we built a schema that defines Active Lines, so let's use that.

Note: Since **Active Lines** is an array stored on a Profile, this is going to select the owner of the account vs each individual owner of the device. Make sure the Marketing team is aware of and wants that. Otherwise, you might want a different approach.

1. Create a new Segment
2. On the **Attributes tab** in the left rail, navigate down to Product Name (or search for it)
   a. XDM Individual Profile > Dxp > Active Products > Product ID properties > Product Name
3. Drag Product Name onto the canvas

Side Thought, "couldn't we just filter on the Events, rather than having another field on Profile store the same thing"?

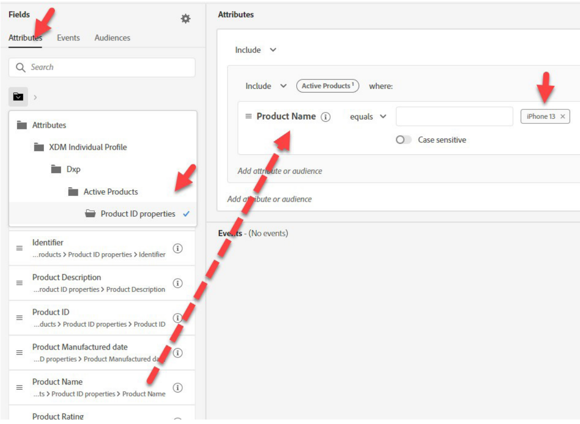
Yes we could, but we have to go into some business and technical nuances that make the Segments complex and introduce some challenges:

1. If we use the Purchase Event:
   a. What if they didn't buy from us, but have an active line?
   b. What if they purchased 2 years ago, my Segment has to look back N number of years and we only kept 1 year of Events on Profile?
2. The Billing Event seems like a better fit:
   a. But now the data is up to a month old.
   b. What if the last Billing Event was 2 years ago, this could include people who aren't customers
   c. What if my data load failed, my segment count might drop to zero if I'm only looking back one month in order to exclude old data
   d. Do we even capture the Device for a Billing Event? No, so we would have to change our data feed

In the end, we will have to make some trade offs for this Segment. If your heart is still set on using Events for this Segment, read this Blog about it:

https://experienceleaguecommunities.adobe.com/t5/adobe-experience-platform-blogs/how-to-capture-latest-experience-event-in-adobe-experience/ba-p/430941

4. Type iPhone 13
5. Copy the “pseudocode” version of the Segment Rule into the Description
6. Save as Batch Segment
7. Save Segment as “**Owns iPhone 13**”



Go through the same steps above for the Pixel 6.

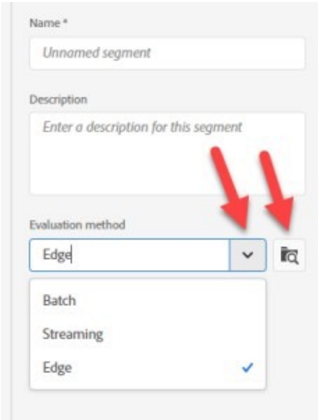
Rebuilding Segment #2 (Owns iPhone 13)

Marketing walked in today and gave us a requirement to have this Streaming and unfortunately the way we have this built is Batch. Let's fix that:

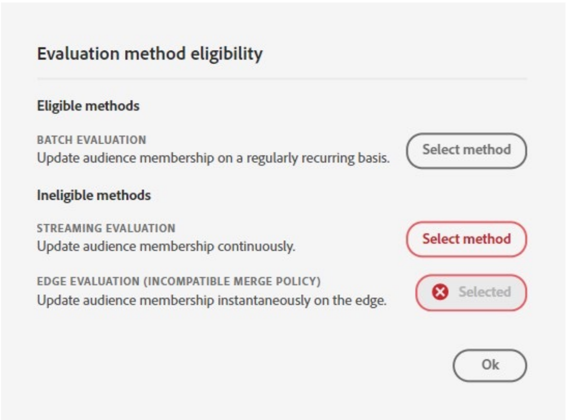
1. Delete the Owns iPhone 13 Segment

> **Note:**
> i. Today we cannot change the Evaluation Method in the UI
> ii. Any Segments that reference this Segment will also have to be deleted
> iii. Keep this in mind when deciding on your building strategy of using Segments within Segments

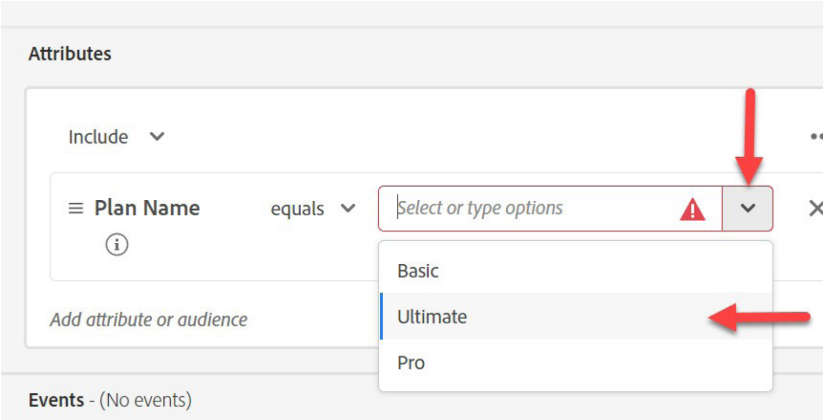
2. Recreate it the same way
3. Before saving, change the Evaluation Method to Edge, then click on the folder icon



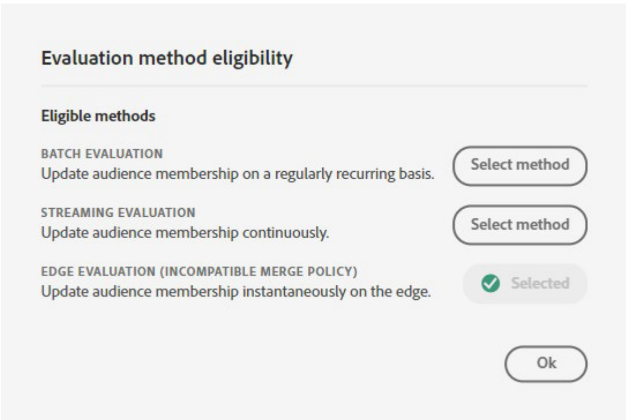
   a. You should see this:



4. While not obvious, the reason for this is we are using Product Name on a Lookup



5. Let's replace it
   a. Replace
      i. XDM Individual Profile Dxp Active Products Product ID properties Product Name
   b. With
      i. XDM Individual Profile Dxp Active Products Model
6. Before you save the Segment change the Evaluation Method to Edge and click the folder icon

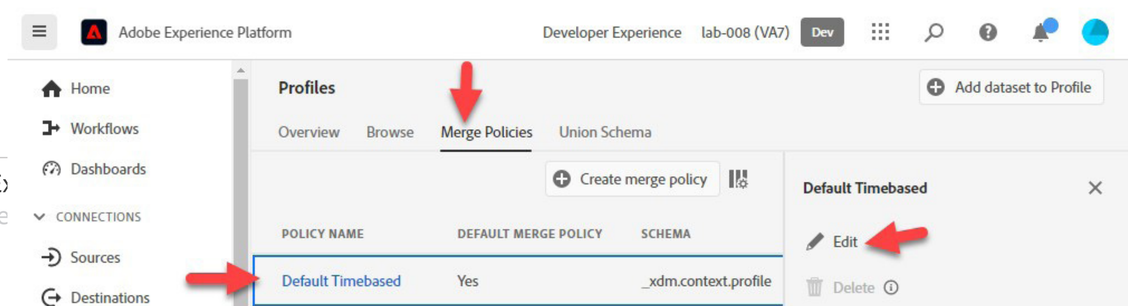


Note: if you are still getting an error

Go to your Merge Policies and Edit the Default Merge Policy for _xdm.context.profile

Note: If you are still getting an error

i. Go to your Merge Policies and Edit the Default Merge Policy for _xdm.context.profile



ii. Turn on the Active-On-Edge Merge Policy

iii. Save

iv. Go back to your Segment and save it (you may have to recreate it)

7. Copy the pseudocode
8. Save

Considerations why you may want to select Batch vs Streaming or Edge:

Guardrail	Limit	Limit Type	Description
Segments per sandbox	10,000	Soft	An organization can have more than 10,000 segments in total, as long as there are less than 10,000 segments in each individual sandbox. Attempting to create additional segments may affect system performance.
Streaming segments per sandbox	500	Soft	An organization can have more than 500 streaming segments in total, as long as there are less than 500 streaming segments in each individual sandbox. Attempt-ing to create additional streaming segments may affect system performance.
Batch seg-ments per sandbox	10,000	Soft	An organization can have more than 10,000 batch segments in total, as long as there are less than 10,000 batch segments in each individual sandbox. Attempting to cre-ate additional batch segments may affect system performance.

https://experienceleague.adobe.com/docs/experience-platform/profile/guardrails.html?lang=en

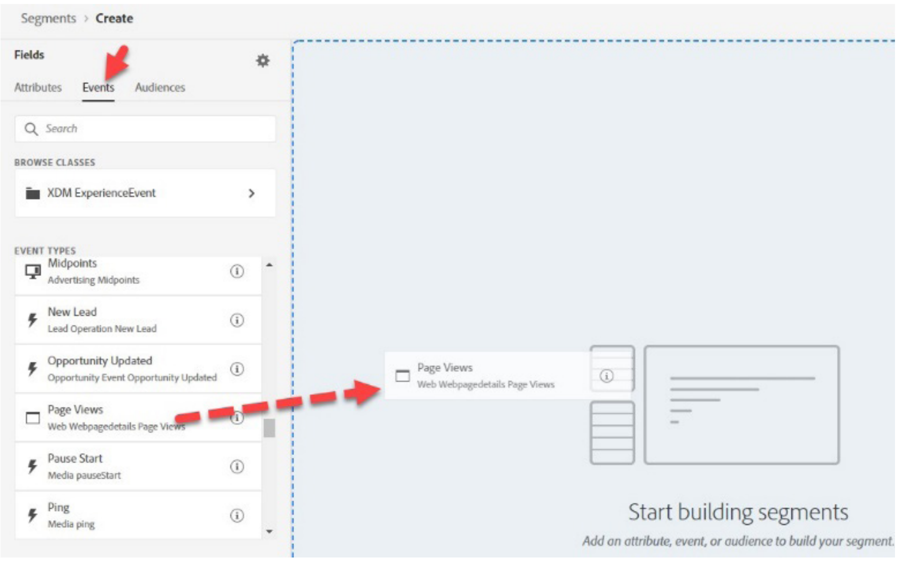
Note: We have discovered yet another area where Product is stored.

Segment #3 (Visited an iPhone 13 product page)

Remember: We are looking for a Page View, not a Product Purchase nor an eCommerce Event.

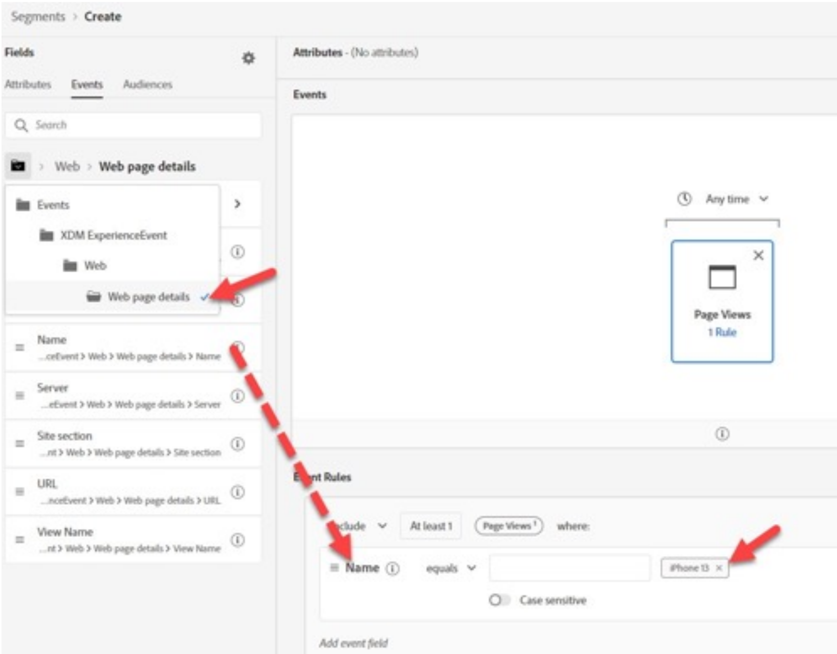
1. Create a new Segment
2. Find Page View event on the Event tab under Event Types in the left rail and add to the Segment

> - Why? Up until now we have been using the generic "Any Event". By using the Page View Event we ensure the Segment is only evaluating Page Name in the context of a Page View. It should be redundant since a Page Name only exists on a Page View, but provides two benefits:
>   - Provides high level visual documentation to the user when looking in the UI
>   - Provides filtering to ensure as new events are added that they are not included when that was not the intent
> - For this reason, we recommend each Event Schema you build should have a lot of thought put into the Event Types you use. They are fundamental to filtering and visual guides.



3. Save this Segment as "Visited Any Page"
4. Create another Segment
   a. Add the Page Views Event
5. Navigate to where Page Name is and add the Page Name field to the Event so we can filter on it.
   a. XDM ExperienceEvent > Web > Web page details > Name
   b. Add contains "iPhone 13"
   c. Note: for your site this could be a Contains or Equals depending on your naming

convention

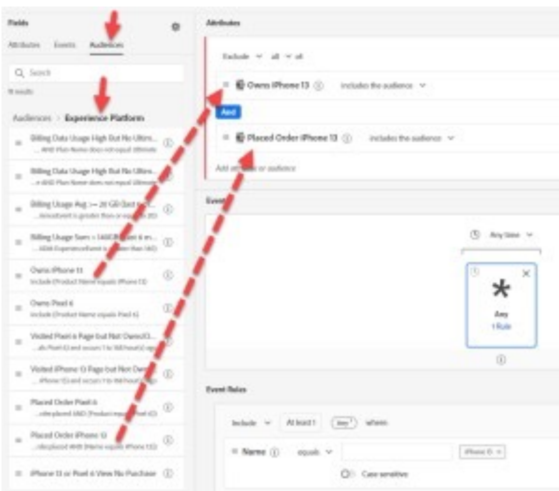


d. For those who fall in love with search, try searching on Page

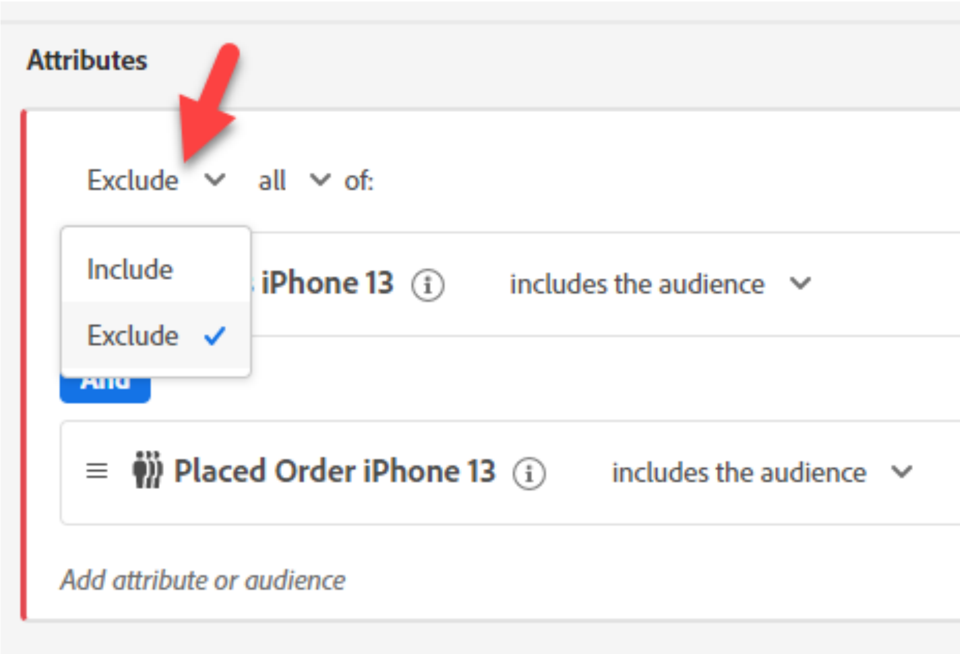
1. You will see Page Name does not come up

- This is because of how it is named:
    - i. XDM ExperienceEvent > Web > Web page details > Name
    - ii. So your folder will come up, but not the field itself
- When you are putting descriptions and your naming conventions together, consider this

6. Navigate to Audiences in the top left nav
7. Drill down to Experience Platform
8. Pull in the two other Segments we previously created



9. Change the Include to Exclude



10. Copy the "Pseudocode" version of the Segment Rule into the Description
11. Save Segment as "Visited iPhone 13 Page but Not Owns/Ordered it"

Go through the same steps for Pixel 6

Time Filter

> This has no time requirement, so if someone visited three years ago, they would qualify. Depending on our use case that may or may not work. It is worth asking. If we add a time filter, how far back can we go before an Edge Segment becomes Streaming or even Batch?

Ramifications of breaking this up:

We have split what is a simple requirement up into many Segments for a few reasons. The requirement is for a Streaming Segment, but these two requirements turn our Segment into a Batch Segment. More detail here on the rules around this.

https://experienceleague.adobe.com/docs/experience-platform/segmentation/ui/streaming-segmentation.html?lang=en

1. The exclude clauses
    a. "no order exists for iPhone 13/Pixel 6"
    b. "no active iPhone 13/Pixel 6"
2. Marketing wants to target iPhone & Pixel users as two Segments

Our Marketing team just gave us another requirement, so let's add this What does this mean to our data flow and using all this data?

To answer that we need to understand how Segment data is stored. Our Peeking Underneath the Hood of Segment blog talks a little about this below. It shows how the result of a Segment is stored on the Profile. This is important since as data streams in it is looking at the results of a Segment stored on the Profile, it is not rerunning the Segment at that point in time! A simple nuance but worth understanding. Most Profile attributes are updated periodically, so this approach makes sense. But we need to understand that when using a Segment within a Segment, AEP will attempt to sequence when it can. There are edge cases where this is not possible, e.g. If a segment of segments is used, profile disqualification will happen every 24 hours.

https://experienceleaguecommunities.adobe.com/t5/adobe-experience-platform-blogs/peeking-underneath-the-hood-of-segments-in-aep-adobe-experience/ba-p/453535

Thus, the way this Segment will behave is:

- Real-time qualification of these Segments as data streams in Ordered iPhone 13
    - Ordered Pixel 6
    - Owns iPhone 13
    - Owns Pixel 6
    - Visited iPhone 13 Page
- 24 hours latency disqualification of these Segments Owns
    - iPhone 13
    - Owns Pixel 6

Bottom line: we traded off faster entry into the Segment by breaking it into pieces with a 24 hour latency of them falling out of the Segment.

DESTINATION

We will cover this in another lab.

Break/Wait Here

Use Case #3 (Unlock New Opportunities: Upsell Ultimate Phone Plan)

Find all customers who have a total billing data usage in the last 6 months >140GB, a rolling 6 month avg. monthly data usage of >=20GB and they do not have an ultimate phone plan.

Activate into Facebook / Google and Direct Mail channels.

Direct Mail Personalization fields:

- First Name → used for greeting
- Mailing Address → used for mailing
- Plan Name → used for mailing statement (i.e. Eric, upgrade your <plan name> to an ultimate plan today!)”

Analysis Tasks

Analyze the above and write down:

1. What fields you think are needed to address this use case?
2. Does the segment need to be Streaming?
3. Why do we have two Billing Details?
4. What other information would you like to know?

> **Remember:** When we get requirements from the business stakeholders, they tend to be incomplete, use another terminology and make assumptions without knowing it. It is your job to bring as much of that to the surface and guide them to something that can be done.

Approach

For this use case we are going to evaluate two options:

1. Option 1 (Use Segment to Aggregate)
   a. This will have the Segment do the aggregation
      i. Billing data usage Sum >140GB (last 6 months)
      ii. Billing data usage Avg >20GB (last 6 months)
      iii. Billing Data Usage High But No Ultimate Plan
2. Option 2 (Use Pre-Aggregates)
   a. This will utilize aggregation that was done before putting the data on Profile
      i. Billing Data Usage High But No Ultimate Plan (Agg)

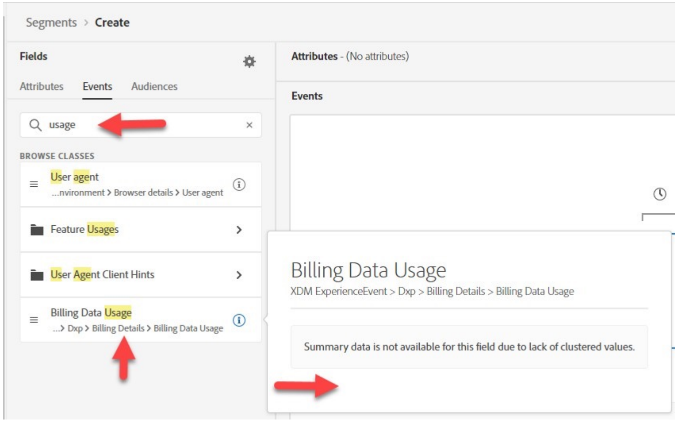
Build Tasks

This build will show two options in order to show some functionality and showcase why we took some design choices early on.

Pre-Work (breaking down the Segment and finding the fields)

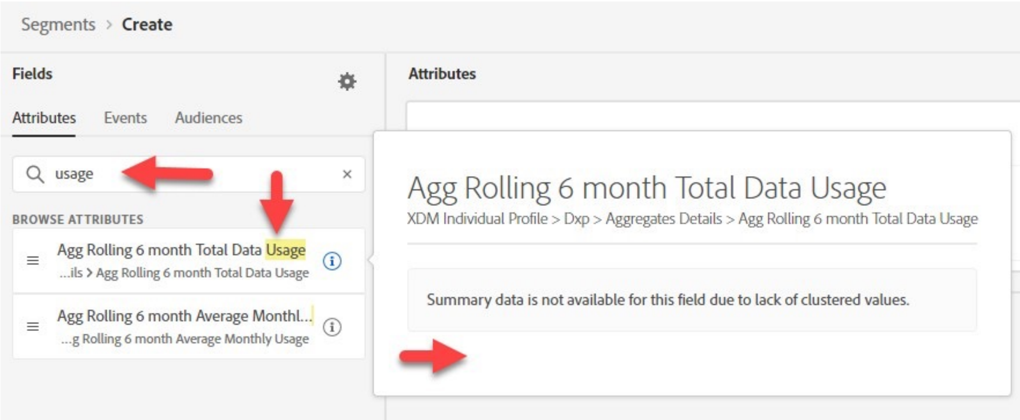
For this Segment there isn't much pre work to do. The usage data comes from one source. But does have some nuances. After looking at the data we realize it exists in two places:

1.Usage exists as a Billing **Event**



Note: there is no description

2. Data Usage exists as an **Attribute** on Profile



Note: there is no description

Notice:

1. Neither of these have any descriptions, so the Marketer will either make some assumptions. Descriptions are important.
   a. Without descriptions, how will the Marketer know
      i. Which to use?
      ii. Latency of data?
      iii. Recommended/preferred in specific use cases?
2. Searching for “Billing” only shows up as an Event, while “Usage” shows up in both places. Naming conventions exist for your Marketer too.
   a. aepending on what they search on or if they are looking/expecting this to be an Event or on Profile affects what they find and eventually use.

We will implement using both of these fields:

4. Option #1 Usage exists as a Billing Event
5. Option #2 Data Usage exists as an Attribute on Profile

Build Option #1 (Use Segment to Aggregate)

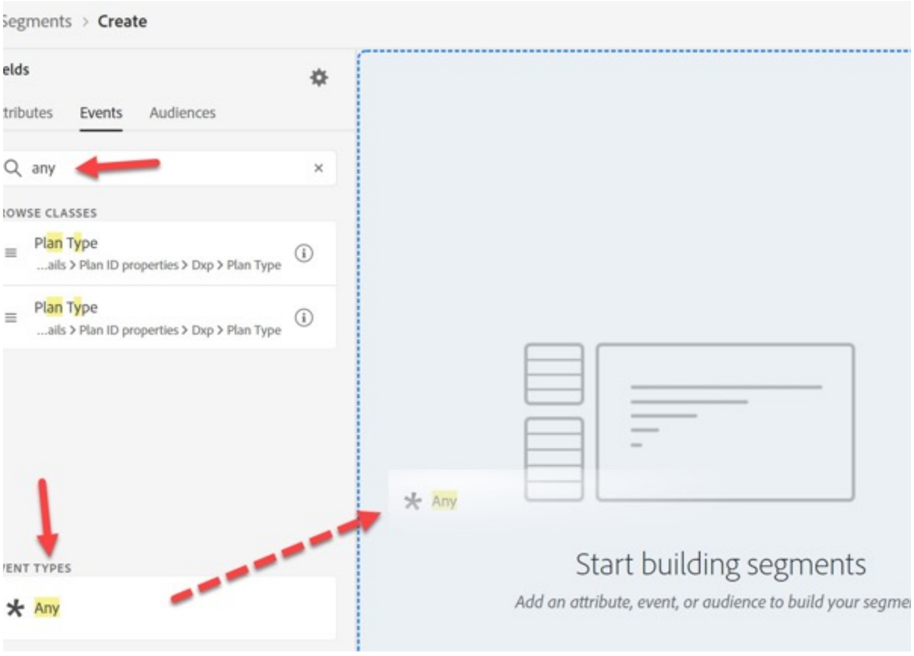
Aggregates in Segments allow us to aggregate Events in the Segment rule. But since we can only do one aggregate at a time, we need to split the two from our use case.

Aggregate Segment #1 (total billing data usage in the last 6 months >140GB)

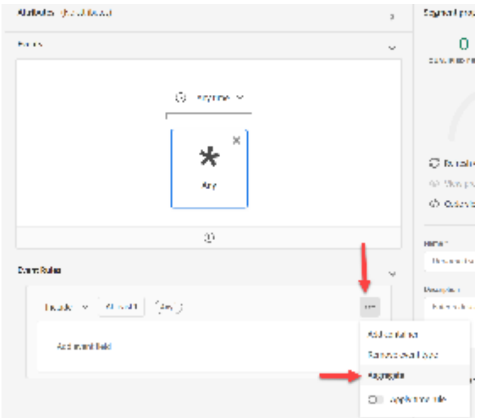
1. Create a new Segment
2. Since our exploration didn’t turn up any specific Bill Event Type, we will search for the “Any” Event

- This may not be obvious to the Marketer they can do this, nor what the Any Event is.
- A better approach is to make your Event Types standardized and well known across your Schemas. It also helps w/ misspellings and including Events that were unintended.

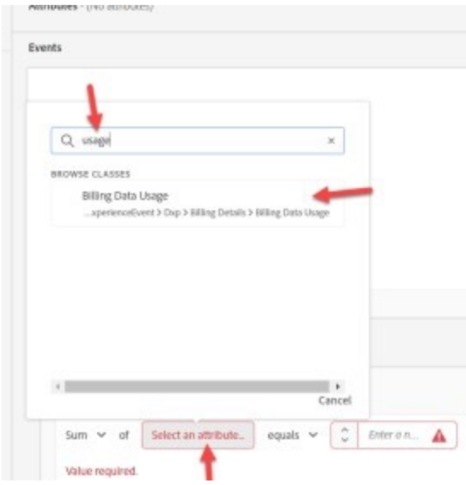
3. Drag and Drop the Any Event Type onto the canvas



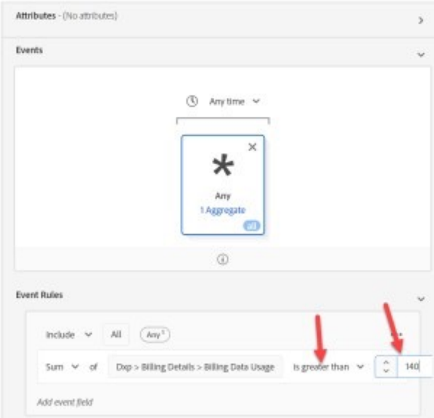
4. Click on the Any Event Card
5. Click on the Ellipse in the bottom right rules and choose Aggregate



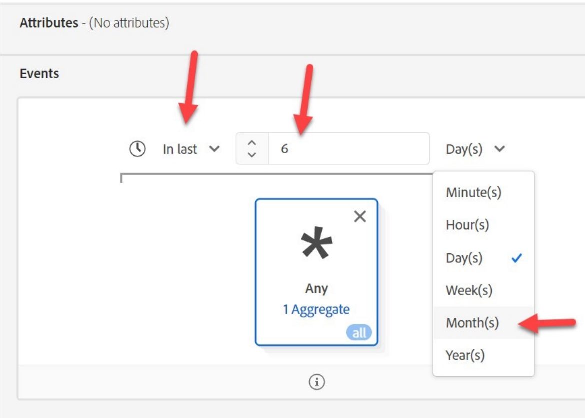
6. Click on Select an Attribute
7. Type Usage
8. Select the Billing Data Usage field



9. Change the Equals to Greater than and the value of 140



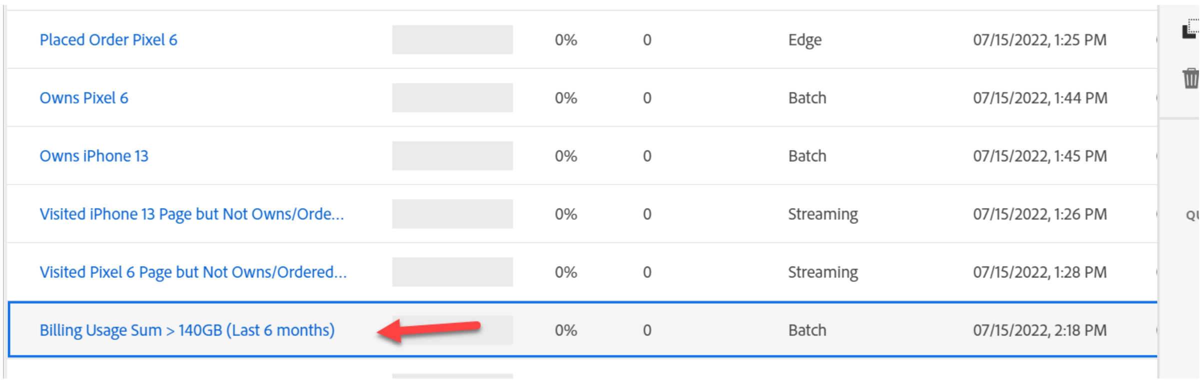
10. Change the time above the Any Event card from Any Time to In Last and the value to 6 and the days to months



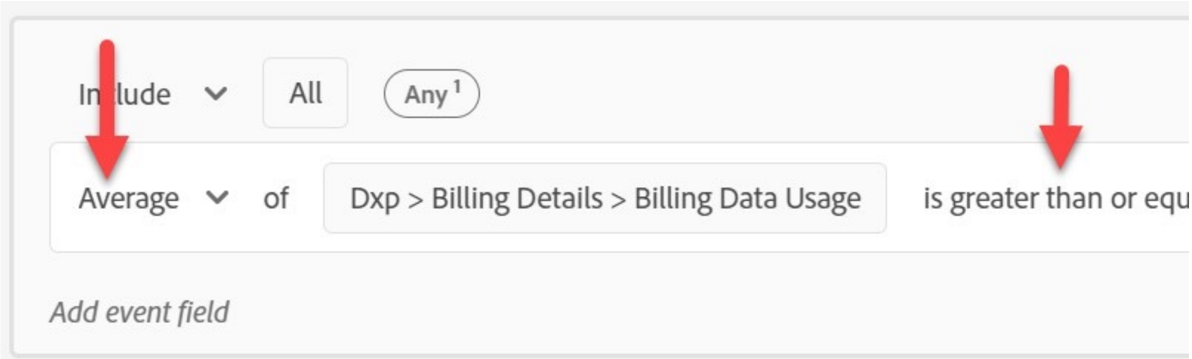
11. Copy the pseudo code into the description
12. Give the Segment the name “Billing Usage Sum > 140 GB (last 6 months)”

Aggregate Segment #2, (Rolling 6 month avg. monthly data usage of >=20GB)

1. Do not click on the hyperlink, but select the row in the Segment List UI so it is highlighting the segment we just created
2. Once it is highlighted, click copy



3. Click the copy and Edit it
4. Click on the Any Event card
5. Change the Sum to Average
6. Change the greater than to greater than or equal to
7. Change the value to 20



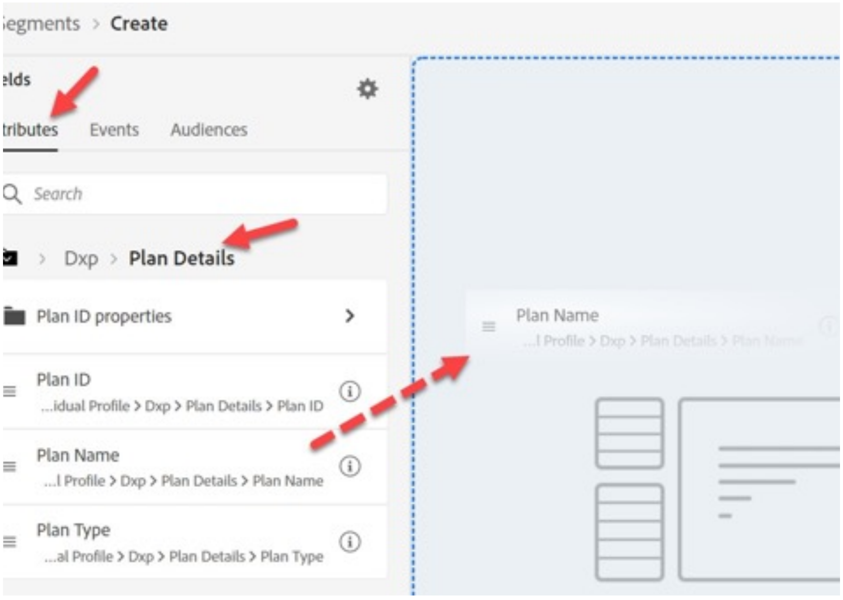
8. Copy the pseudo code into the description
9. Give the Segment the name "Billing Usage Avg > 20 GB (last 6 months)"

Segment #3, (do not have an ultimate phone plan)

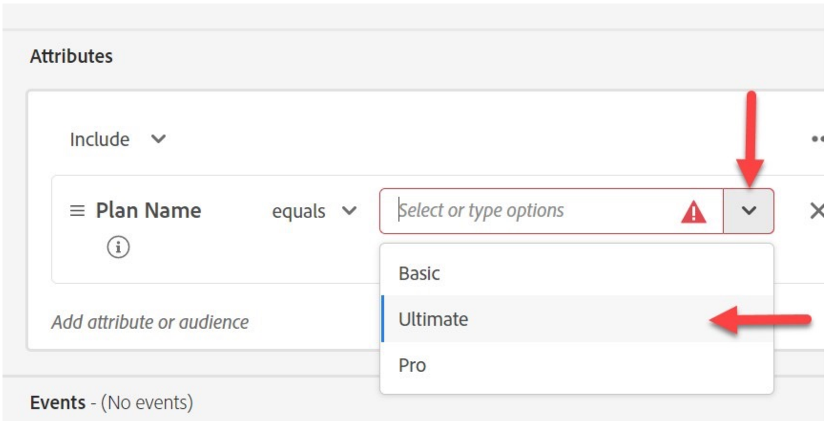
1. Create a new Segment
2. In Attributes, search for Plan

> **Why two Plan Name's?**
>
> i. XDM Individual Profile > Dxp > Plan > Plan ID properties > Plan Name
> ii. XDM Individual Profile > Dxp > Plan > Plan Name
> iii. The descriptions don't help. How do we know which to use?
> - Let's use the one that is using the lookup dataset for now since it should have the most up to date names
>   - i.e. in the case a Plan Name changed, most likely the lookup dataset would be updated vs every person on that Plan

- XDM Individual Profile > Dxp > Plan > Plan ID properties > Plan Name

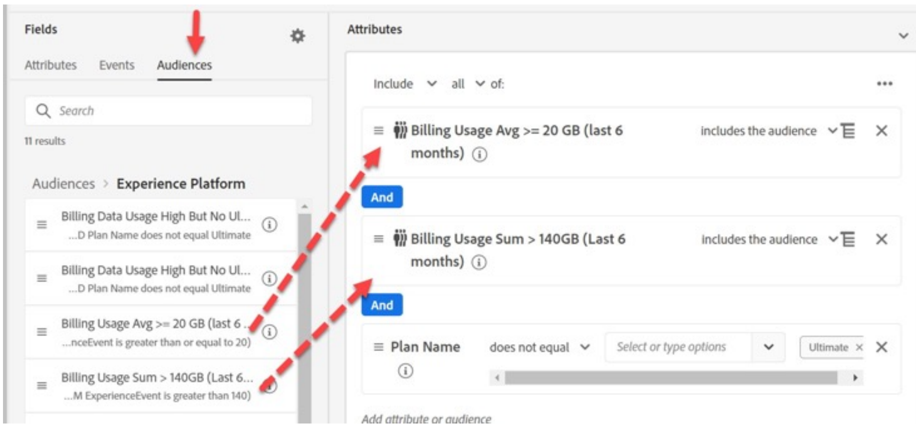


3. Select Ultimate



- Change to Does Not Equal

4. Click on Audiences
5. Click on Experience Platform
6. Drag Billing Usage Sum > 140 GB and Billing Usage Avg Avg >= 20 GB next to Plan Name



7. Copy the pseudo code into the description
8. Check this can be Streaming
    a. It can't. Let's make some changes
    b. Lets use
        i. XDM Individual Profile > Dxp > Plan > Plan Name

> Remember Plan Name is in two places
>
> i. If you remember, this was a design decision we made during the Denormalize step in SID and this is why. It allows for removing the join to our lookup data and thus our Segment is no longer Batch
> ii. This is a common thing to do in a JSON structure, but if the data point isn't in the source, can be quite problematic
>   - E.g. An order may only contain a Plan SKU and not Plan SKU and Plan Name

9. Check this can be Streaming
    a. It can, let's select it, or Edge if you want
10. Save Segment as "Billing Data Usage High But No Ultimate Plan"

> This approach will work, but we now have an Edge segment using Batch segments that will run once every 24 hours. If this works for our use cases and data loads, then this is a good choice (e.g. maybe our billing data is loaded daily or monthly which is highly likely but not all use cases will be like this). If not, a common approach is to aggregate the data before sending to AEP.

Let's look at another option if we need a more real time approach.

Build Option #2 (Use Pre-Aggregates)

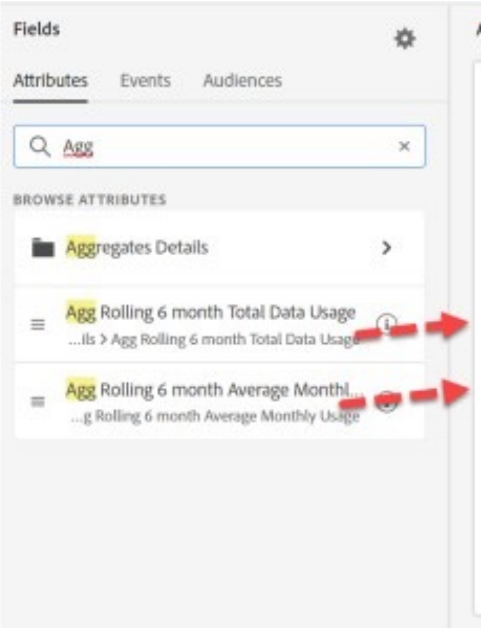
The challenge with Aggregates in our Segment is that our Segment (while Edge), is built on aggregations done inside Segments which are Batch Segments.

Since Marketing has determined a more real time approach is needed, we have done three things to work this into the design:

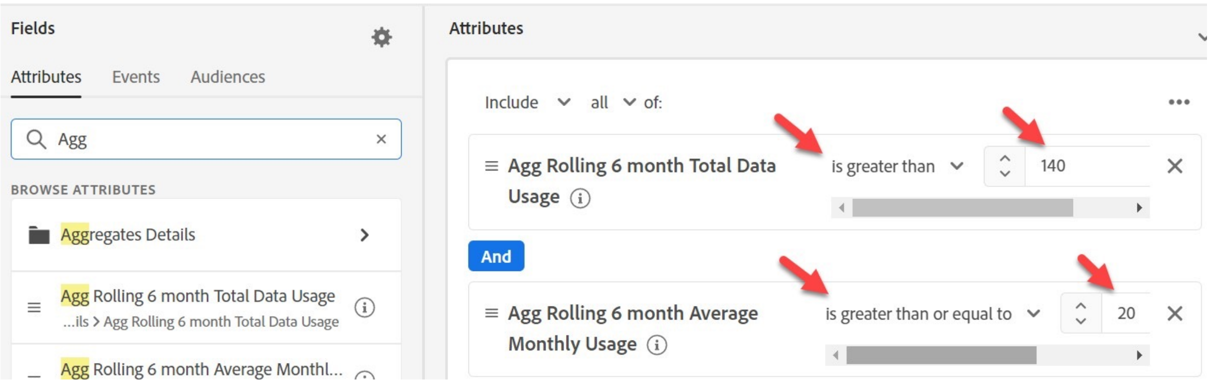
1. Calculate the Aggregates before streaming the data in
    a. This is quite uncommon as most streamed data is designed around a single event vs an aggregate
2. Use the denormalized the Plan Name
3. Stream the data in

Build our new Segments:

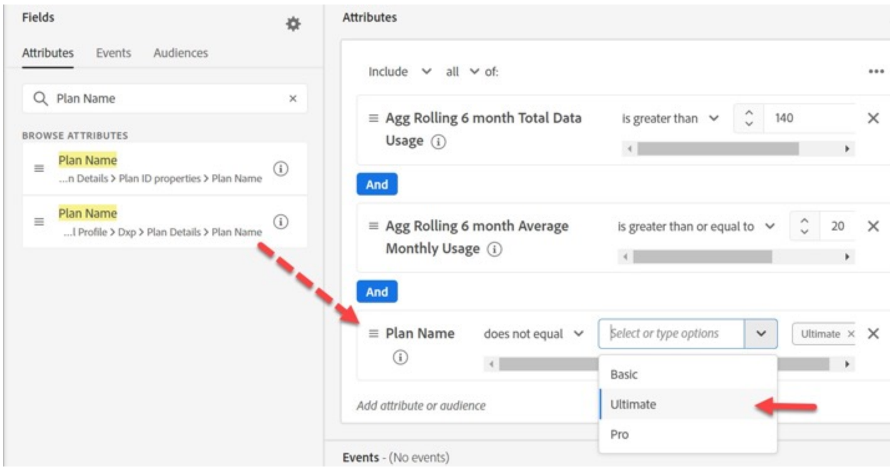
1. Create a new Segment
2. Search for "Agg" on the **Attributes** not Event tab and drag the two Aggregates onto the canvas



3. Set the appropriate operators and values for each



4. Search for Plan Name and add it



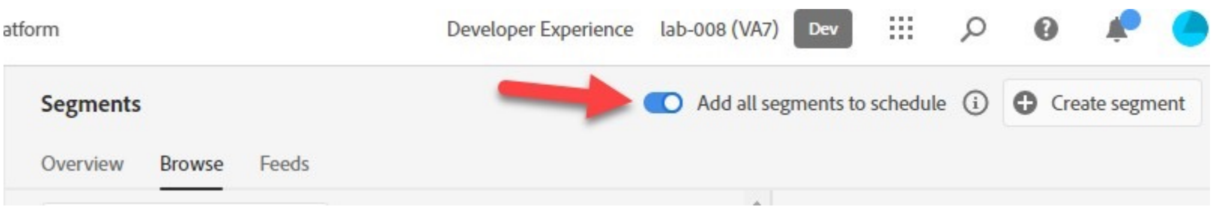
a. This time we use the denormalized Plan Name
    - XDM Individual Profile > Dxp > Plan > Plan Name

b. Does not Equal "Ultimate"

5. Copy the Description
6. Save the Segment as "Billing Data Usage High But No Ultimate Plan (Agg)"
    a. Validate your Segment is Edge

> - Remember, we shifted the aggregate logic into our upstream Streaming ETL layer
> - This choice is a trade off between have a Batch Segment where the marketer controls the logic vs a Streaming Segment by pushing that to the ETL layer where Engineering controls the logic

7. Toggle the Segment Schedule on



Destinations

We will cover this in another lab.

Use Case #4 (Order Placed with Order Cancel)

Our previous Use cases were really successful, but a trend in both orders placed and orders cancelled has been seen. Management now wants to target people who are purchasing items and cancelling them within a week with an email to find out why.

Analysis Tasks

Analyze the above and write down:

1. What fields you think are needed to address this use case?
2. Does the segment need to be Streaming?
3. Do we need aggregation or any ETL?
4. How do we ensure they are cancelling the same item?
5. What other information would you like to know?

Remember: When we get requirements from the business stakeholders, they tend to be incomplete, use another terminology and make assumptions without knowing it. It is your job to bring as much of that to the surface and guide them to something that can be done.

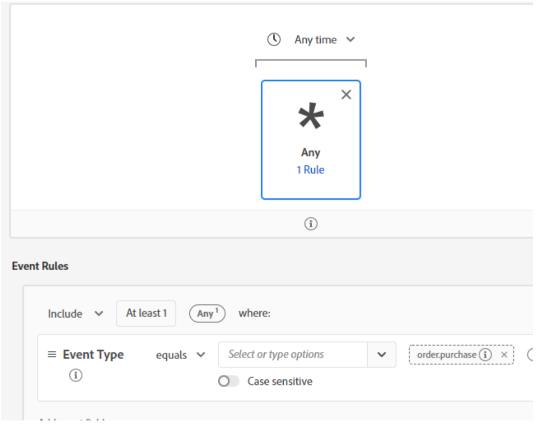
Approach

For this use case we are going to use two features of the Segmentation Engine:

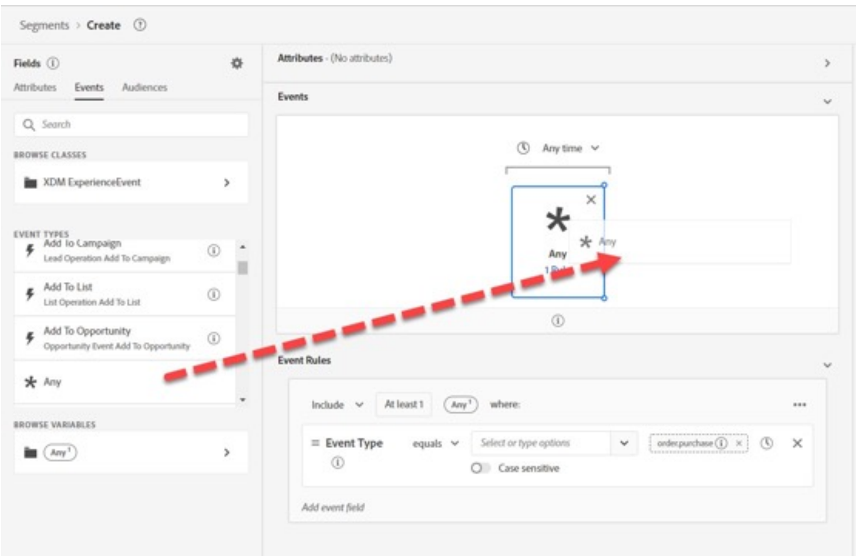
1. Sequential Segmentation
2. Dynamic Segmentation

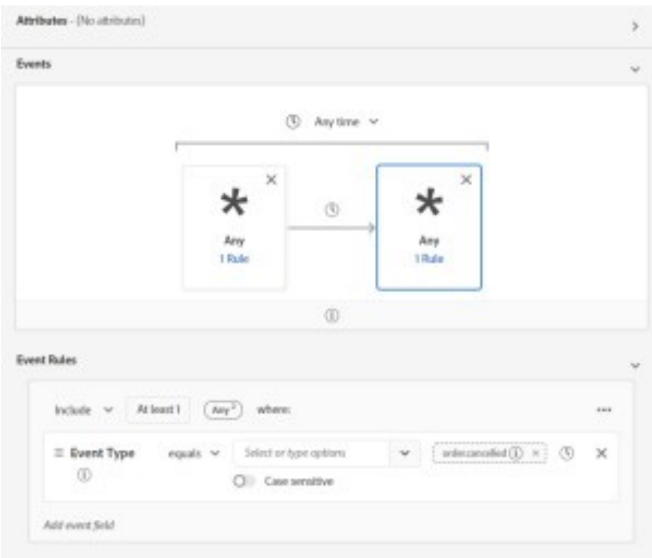
Build Tasks

1. Add the Order Purchase Event or the Any Event filtered by Event Type=order.purchase

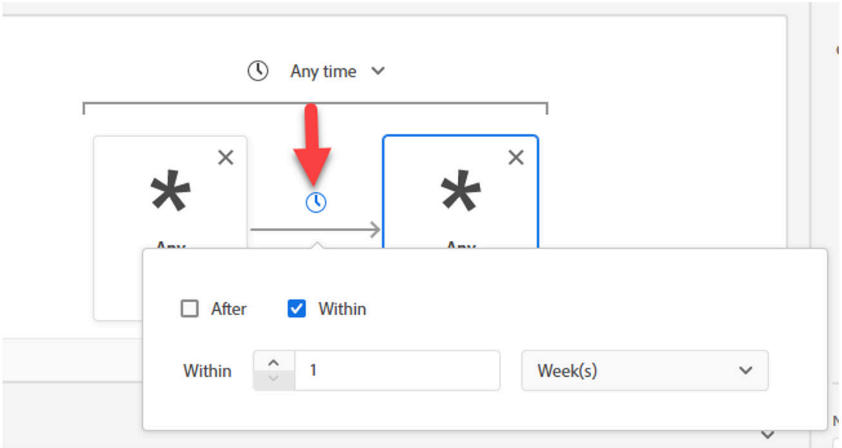


2. Add the Order Cancelled Event or the Any Event filtered by Event Type=order.cancelled to the right of the Order Purchase Event



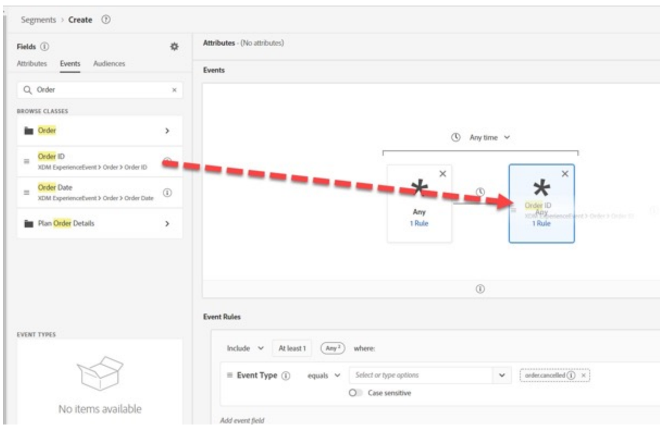


3. Change the time to within a week



> **Note:** The Segmentation Engine uses only the Timestamp to interpret the order of Events. Thus, if you have multiple datetime fields on the Event, keep in mind the Timestamp field is the one used.

4. Clear any search
5. Search for Order ID and drag the field onto the Order Cancelled Event

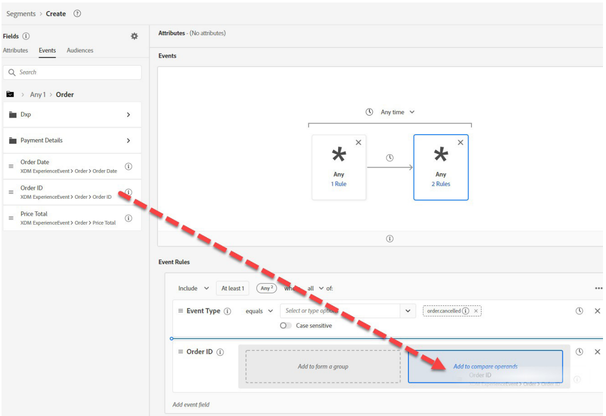


- We are adding a filter for Order ID to ensure the Order Placed is the same Order Cancelled

6. Clear the Search

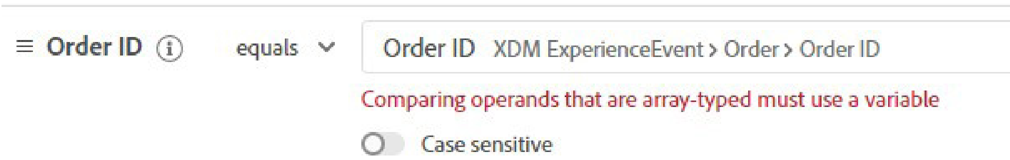
7. Click into Order Placed (or Any1 if you used the Any Event and filtered on Event Type) under the Browse Variables

   a. Drill down to Order ID
   b. Drag over to add a compare operand

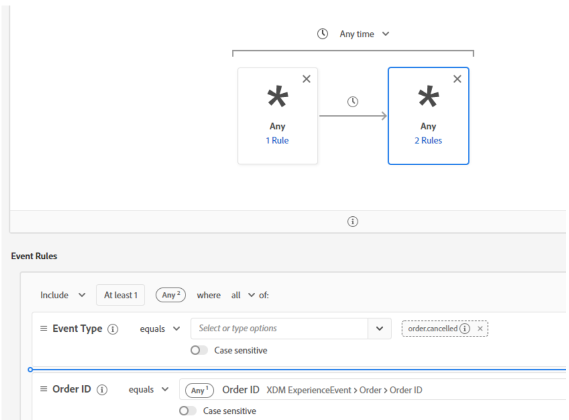


Note: Do not use search within a Variable, it will not keep the context of the variable

i. If you do you will get this error



8. You should have this



   a. This is using the variable container to ensure the Order Cancelled is the same Order that was Placed

Containers

Earlier we used a Container to isolate an element in an Array. Here we are using Containers to reference a specific Event in a filter criteria inside another Event.

- The Order Cancelled Event is filtering its own Order ID by comparing to the Order Placed Order ID
- How else could we use this?
  - Comparing a Product SKU for a Page View is the Product SKU Purchased
  - Comparing an Ship to City is different than the Bill to City
  - Comparing any two fields of the same data type should be possible even though the Events can come from different Schemas

https://experienceleaguecommunities.adobe.com/t5/adobe-experience-platform-blogs/how-exactly-do-containers-work-in-aep- segmentation-a-deeper-look/ba-p/458780

9. Copy your Pseudocode
10. Check your Evaluation method
11. Save your Segment as Order Placed and Order Cancelled within a Week

Edge Network & Edge Segmentation

Link & Lecture

https://experienceleague.adobe.com/docs/blueprints-learn/architecture/data-ingestion/deployment/websdk.html

Segment APIs

Streaming Segment Baseline

Keep in mind that Streaming Segments must run a baseline once they are created. It is created with the next scheduled Segmentation job. This means any downstream Destinations will not get any Profiles until this is run. To address this we are going to use the API to run a Segment Job manually. Normally, you will just set everything to run on a schedule daily and let the scheduler pick it up.

Run your Segment Job:

1. Copy paste all your Segment IDs into Postman in this call
2. Run it

```
POST: https://platform.adobe.io/data/core/ups/segment/jobs [
    {
    "segmentId": "[yourSegmentID]"
    },
    {
    "segmentId": "[yourSegmentID]"
    }, ...
]
```

In general Segment **building** is handled by Marketing and usually through the UI. That doesn't mean we can't use the API. For this lab we are not going to go through the API, but here is the documentation on doing this.

Things to know about building a Segment using the API:

1. You will need to provide or write the Segment rules in PQL

> - You can see this by using the UI and clicking on "Code View"
>
> a. e.g. for our Segment "Billing Usage Sum > 140GB (Last 6 months)"
> CHAIN(xEvent, timestamp, [C0: WHAT(true) WHEN(<= 6 months before now)
> COUNT(ALL) WHERE(C0.sum(_devbc.billing.dataUsage) > 140.0)])
>
> - This PQL must be provided in the API call
>   - POST: https://platform.adobe.io/data/core/ups/segment/definitions
>   - https://developer.adobe.com/experience-platform-apis/references/segmentation/#operation/createSegmentDefinition

2. Segments created using the API cannot be modified in the UI

Break/Wait Here

Activation & Destinations

So far we have done a lot of behind the scenes work:

1. Built our schema
2. Ingested data
3. Defined how we want Profiles to merge
4. Created Segments

Now is when we get to see AEP in action.

Activation of a Profile takes many forms. We are going to talk about the Destinations Framework, a subset of Activation.

Destinations

We have many Destinations in AEP, but they roughly break down to two types on the Hub:

1. Streaming
2. Batch

Use Case #2 (Social Media Activation: Nurturing Warm Audiences) Streaming Approach:

- Create a Streaming Destination for the Segments we created above.
- For this Use Case we are sending a Segment to a Social Media Destination.
- We are going to use a HTTP API Streaming Destination as a Proxy so we don't send anything to Facebook.

Segment: Visited iPhone 13 Page but Not Owns/Ordered it

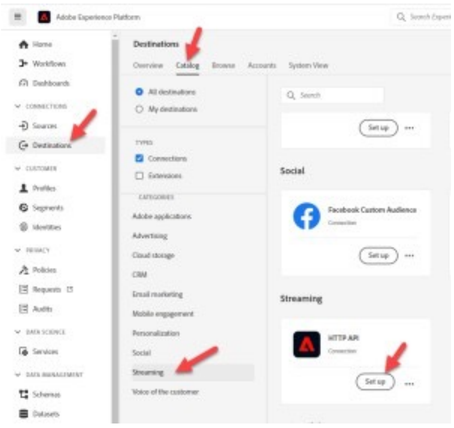
Pre-work (configure Webhook)

1. Go to https://developer.adobe.com/dep/webhook/
2. Enter your sandbox name and hit connect

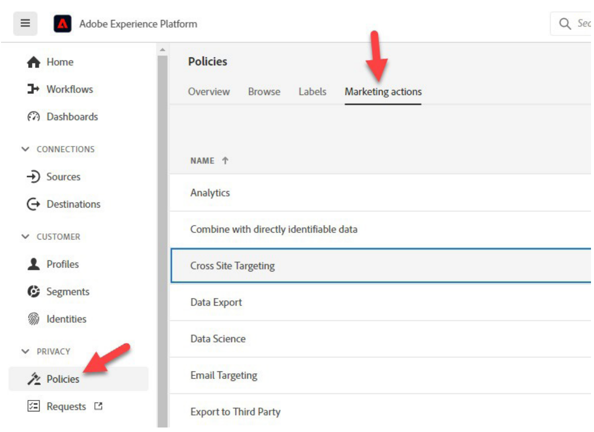
- Copy the webhook URL (not the page URL)

Build Activity (Destination)

1. Create a Destination



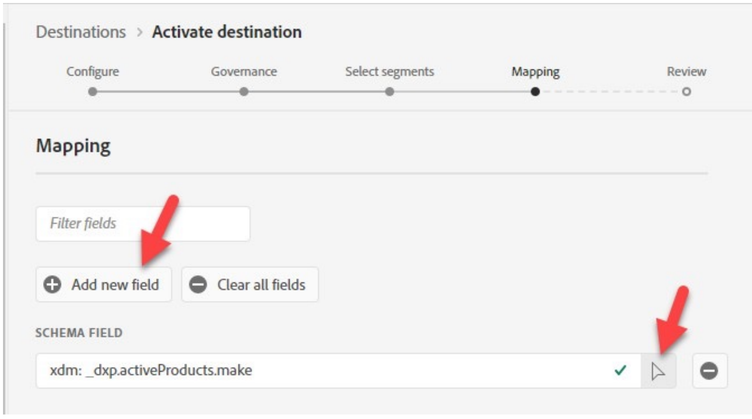
2. Connect
    a. Here you would select the Connection type and add any authentication
    b. For us we are just going to Connect
3. Add Details
    a. Note: the endpoint, header and query params cannot be changed in the UI once saved
        - If you need to change them in the UI, you will need to delete the Destination and the Account
    b. Name: Streaming Webhook DEP Destination
    c. Description: [your webhook endpoint you copied above]
        - I do this since you can't see the endpoint in the UI once saved
    d. HTTP Endpoint: https://27200-depwebhook-stage.adobeio-static.net/api/v1/web/dx-excshell-1/webhook/[sandbox]
        - Your sandbox should be something like lab-001
    e. Select Include Segment Names
4. Next
5. Select Cross Site Targeting
    a. A list of what is available is in the Policies section of AEP



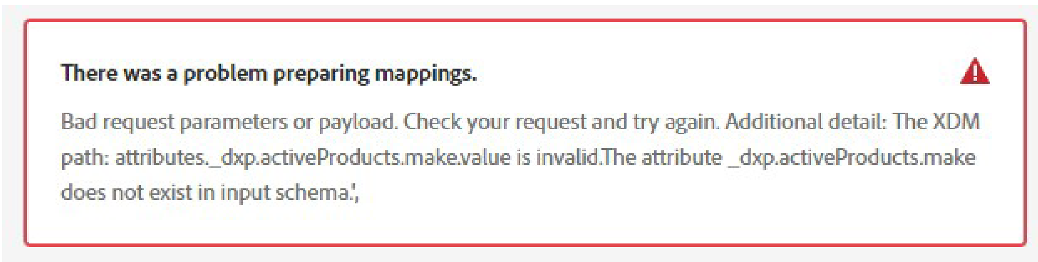
Note: there are no descriptions here, go to Experience League to find them

- https://experienceleague.adobe.com/docs/experience-platform/data-governance/policies/overview.html?lang=en#core-actions

6. Next
7. Select our Segments:
    a. Visited iPhone 13 Page but Not Owns/Ordered it
    b. Visited Any Page
8. Next
9. Select the fields you would like to be sent along with this
    a. For our purposes we are going to include Make



    b. We should get this error



    c. Why?
        i. The error is really hard to understand, but the reason is activeProducts is an Array
        ii. For now, let's use Plan Name instead
            1. xdm: _devbc.plan.name
10. Next and Save

The Destination is Configured and waiting for:

1. Data to be ingested
    a. Every 5 minutes we are streaming data in, so hopefully in a few minutes you will start seeing data in the Webhook
2. Profile to be updated
3. Segment qualifications to happen
4. Destination to send to the webhook

Things to consider

While the design of the overall solution is fine, we may want to go back to the business and see about adjusting our Segment definitions.

Why?

1. As people are viewing the site, they will qualify for theses Segments because I have viewed an iPhone page but not ordered. Seems pushy.
    a. Should we put in a wait x hours/days?
2. As new phones are introduced, we have to go in and change our Segment for the new Version.
    a. Should we only look for the latest version?
    b. Should we only care about Brand?
3. As new phones are introduced, does age of someone's current phone play a role in getting a new one?
    a. Should we look at the age of someone's phone?
    b. Should we keep track of how often someone buys a new phone and use that in our Segments?

Use Case #3 (Unlock New Opportunities: Upsell Ultimate Phone Plan) Batch

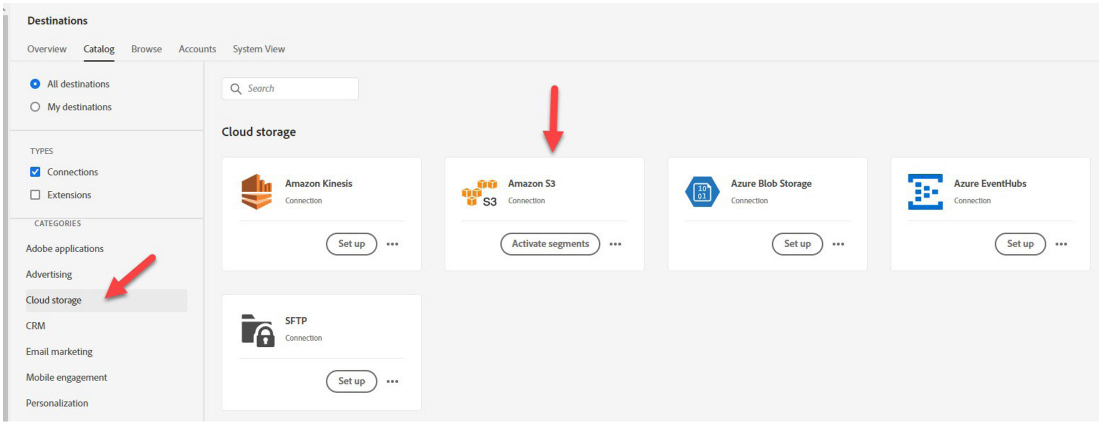
Approach:

- Create a Batch & Streaming Destination for our Segments created above.
- For this Use Case we are sending a Segment to an External Email provider.
- We are going to use an S3 bucket as a Proxy. This will be a File based batch.
- To see the data flow we are going to also send this to a Streaming Destination.

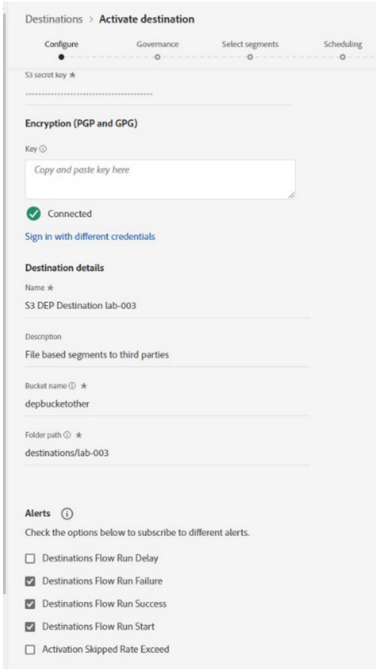
Segment: Billing Data Usage High But No Ultimate Plan

https://experienceleague.adobe.com/docs/experience-platform/destinations/catalog/cloud-storage/amazon-s3.html

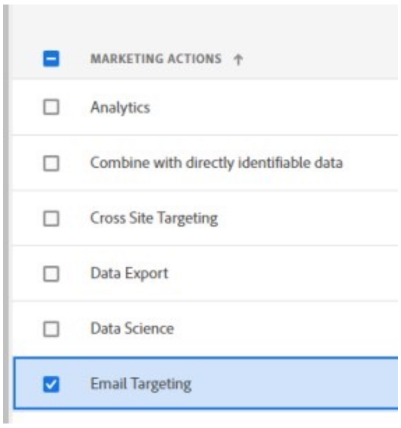
1.Create a Destination



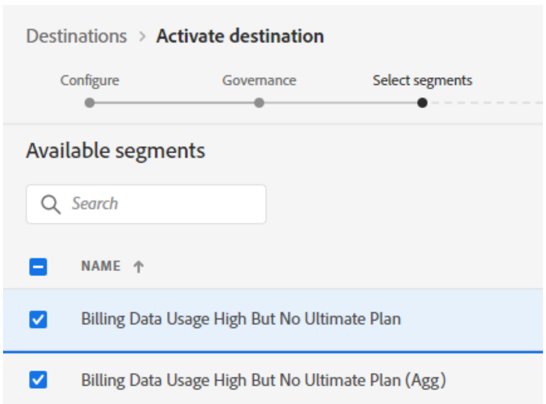
2. Add in the details
   a. Note: you can only create one S3 account
   b. Access Key: AKIAV2G3TC3RHK3UM5GN
   c. Secret Key: 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
   d. Name: S3 DEP Destination lab-003
   e. Description: File based segments to third parties. [bucket]:[folder]
      i. We put [bucket] [folder] here because you can't see that information in the UI once it is saved.
   f. Bucket: depbucketother
   g. Folder: destinations/[sandbox]
   h. Alerts: Start, Success, Failure



3. Next
4. Marketing Action - Email Targeting

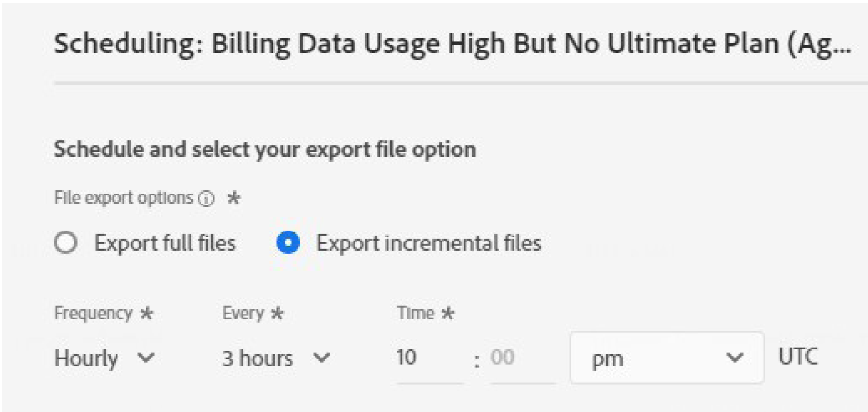


5. Next
6. Select our Segments

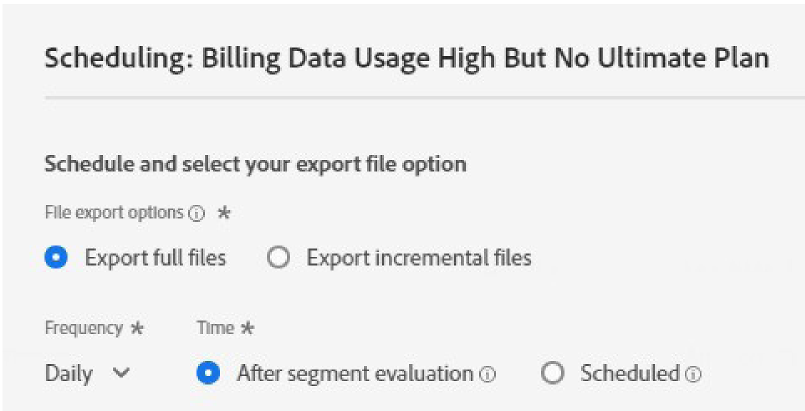


7. Next

8. Schedule
    a. There are a lot of nuances on schedules beyond this lab you can read about here
        i. https://experienceleague.adobe.com/docs/experience-platform/destinations/ui/activate/activate-batch-profile-destinations.html?ang=en#scheduling
    b. For us let's select two different schedules
        i. Incremental every 3 hours



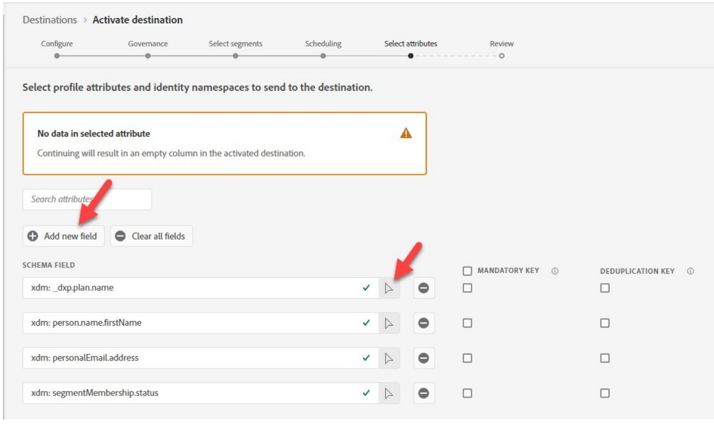
        ii. Incremental daily after segmentation



        iii. set both for one week
    c. Hover over the i to learn a little about full vs incremental

9. Attributes
    a. We need to add:
        i. First Name & Plan Name so we can personalize the email or direct mail
        ii. Remember we denormalized this field. This is another reason why

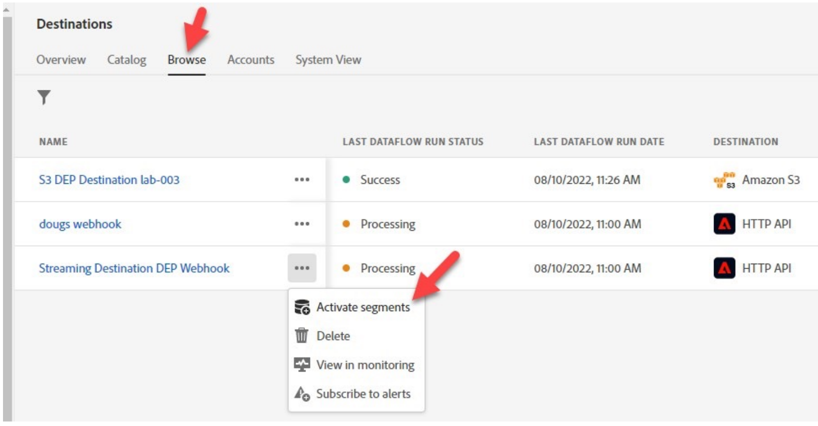
b. We don't need last name, so remove it



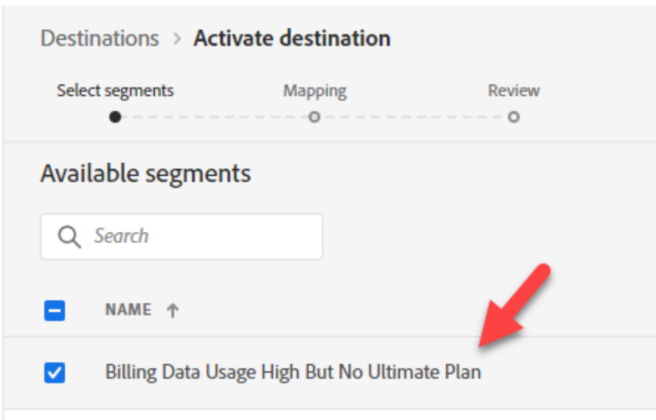
c. Keep in mind, you cannot add Events to a Destination

i. However, we have aggregated the billing data usage so we could include it if we wanted to use that in the content, but probably not needed for this

10. Next
11. Save
12. Go back to the Browse Destinations, find the Streaming Webhook DEP Destination and click Activate Segments



13. Add the same Segment we are sending to S3



14. Map the same fields you did above when you created the S3 Destination

15. Next
16. Save
    a. Now we have one segment going to two destinations
        i. One S3
        ii. One Streaming

Note: Each Destination has its own set of Documentation and what it supports.

Things to be careful of with Destinations

- The API is very powerful
- In the UI you can't:
    1. Create more than one Account to one type of Destination (e.g. only one S3 Account)
    2. Change the schedule
    3. Change the method (e.g. Incremental to Full)
    4. See the schedule or method once saved
- In the UI, you can:
    1. Remove or add/remove a Segment sent to a Destination
    2. Delete a Destination
    3. Disable/enable an Account

Break/Wait Here

Appendix

Alerting

As part of this lab we turned on three Alerts on our Destinations, but there are many more:

https://experienceleague.adobe.com/docs/experience-platform/observability/alerts/rules.html?lang=en

To work with alerts in the Platform UI, you must have the following access control permissions enabled through Adobe Admin Console:

Permission	Description
View Alerts	Allows you to view received alert messages.
View Alerts History*	Allows you to view a history of received alerts via the Alerts tab.
Manage Alerts*	Allows you to enable and disable alert rules via the Alerts tab.
Resolve Alerts*	Allows you to resolve triggered alerts via the Alerts tab.

AEP has Alerting across all of these areas:

- Data Ingestion
- Identity Service
- Real-time Customer Profile Segmentation
- Destinations

Each Alert can also be Subscribed to in Adobe I/O allowing workflow automation or integration into other applications:

https://experienceleague.adobe.com/docs/experience-platform/observability/alerts/subscribe.html?lang=en

Monitoring & Troubleshooting

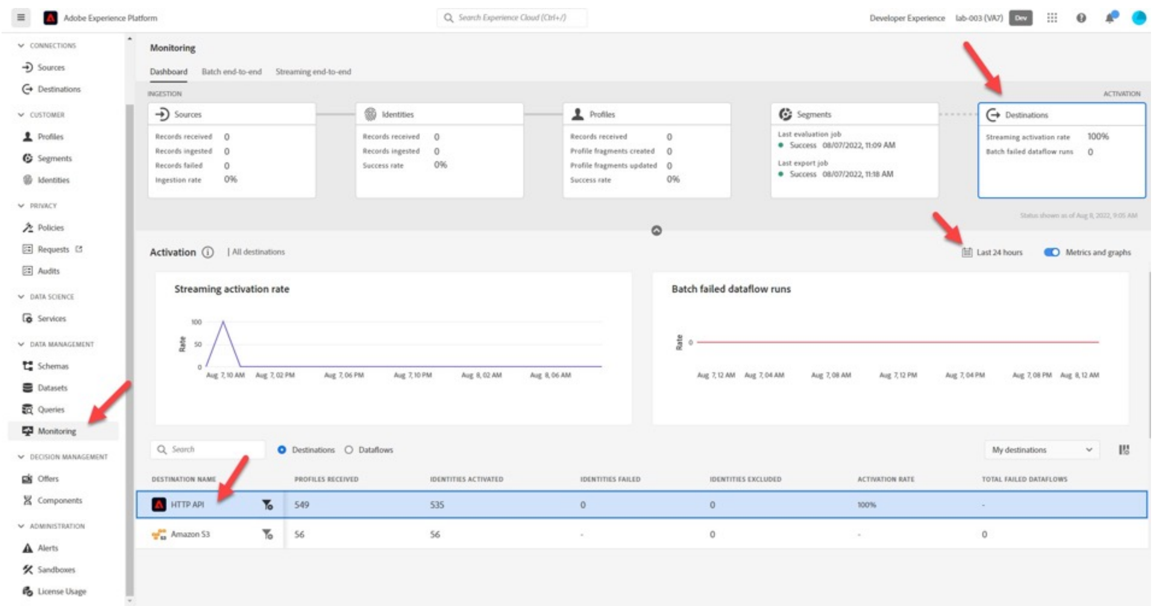
High level (from external Destination back to Data Source)

1. Did the Segment arrive at our Destination inside the system as expected?
2. Did it arrive at the endpoint/file location?
3. Did it get sent by Destinations?
    a. Did the schedule/time pass (daily/hourly)?
    b. If the Destination Incremental, did the Profile enter or exit within the timeframe?
4. Did the Segment Job/Stream run?
5. Did the Profile qualify for the Segment?
    a. Status = entered
    b. Did the Profile exit the Segment?
6. Should the Profile qualify for the Segment?

7. Did the data land on the Profile?
8. Did the data add to the Identity Graph?
9. Did AEP Sources pick up the data?
10. Did the data arrive at the AEP endpoint/file location?
11. Did the data get sent from the source to AEP?

Low level (from external Destination back to Data Source)

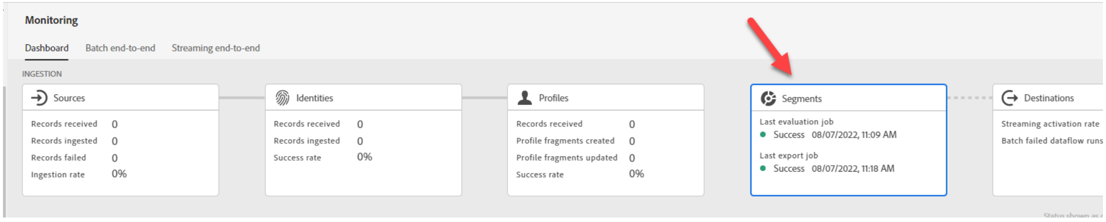
1. Did the Segment arrive at our Destination inside the system as expected?
    a. We need credentials to see this in the downstream system
    b. This will have its own set of:
        i. Ingestion workflows, did they kick off?
        ii. Profile updates, did they update or create a new one?
        iii. Segmentation updates, did they qualify?
        iv. Campaign runs, did they kick off?
        v. etc.
2. Did it arrive at the endpoint/file location?
    a. We need credentials to see this
3. Did it get sent by Destinations?
    a. Did the schedule/time pass (daily/hourly)?
        i. Validate schedule and time has passed or is this a streaming destination
    b. If the Destination Incremental, did the Profile enter or exit within the timeframe?
        i. Validate type
    c. Open the Monitoring Dashboard and see if the Destination Ran



        ii. Drill down into the Destination of Interest by clicking the Filter icon
        iii. Look at the Last Run Date
        iv. Drill down to see the Dataflow Runs
        v. Drill down to see a specific Dataflow Run metrics, timestamps and any Errors

or Identities Excluded

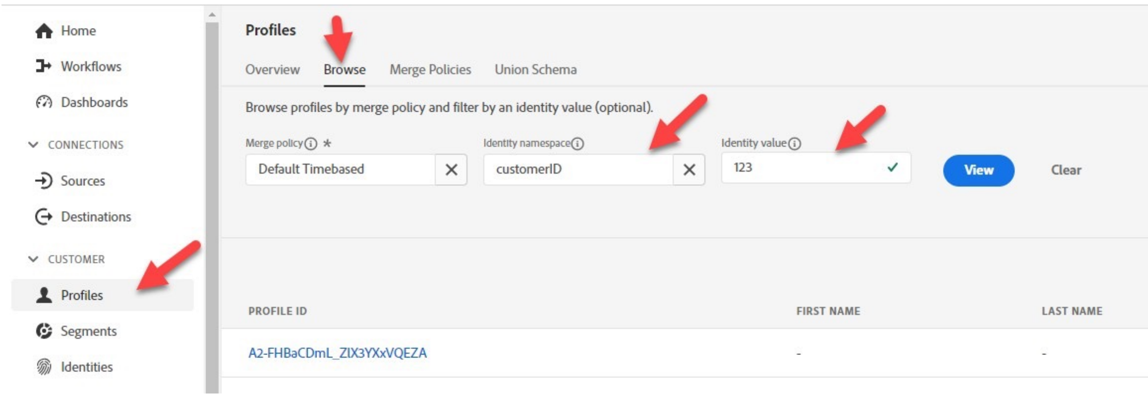
4. Did the Segment Job/Stream run?
    a. Open the Monitoring Dashboard and click on Segments



    b. Drill down into a Segment
    c. Look at the Type, timestamps, status, counts and Destination

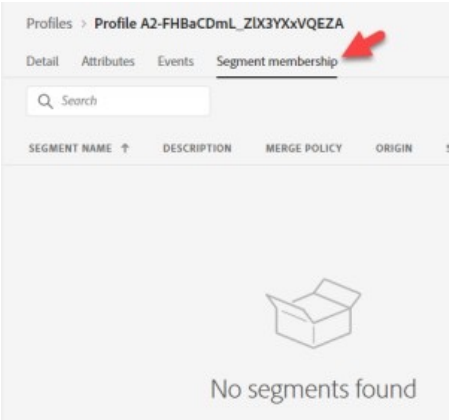
JOB START	TYPE	JOB COMPLETE	PROCESSING TIME	JOB STATUS	PROFILE
08/07/2022, 11:18 AM	Activation	08/07/2022, 11:19 AM	0 minutes	Success	-
08/07/2022, 11:03 AM	Evaluation	08/07/2022, 11:09 AM	7 minutes	Success	56

5. Did the Profile qualify for the Segment?
    a. Status = entered
    b. Did the Profile exit the Segment?
    c. A good way to validate this step is to pick a few Profiles who should qualify for the Segment and look at them in depth
    d. Open the Profile Viewer, put in the namespace and ID you want to lookup, then open the Profile



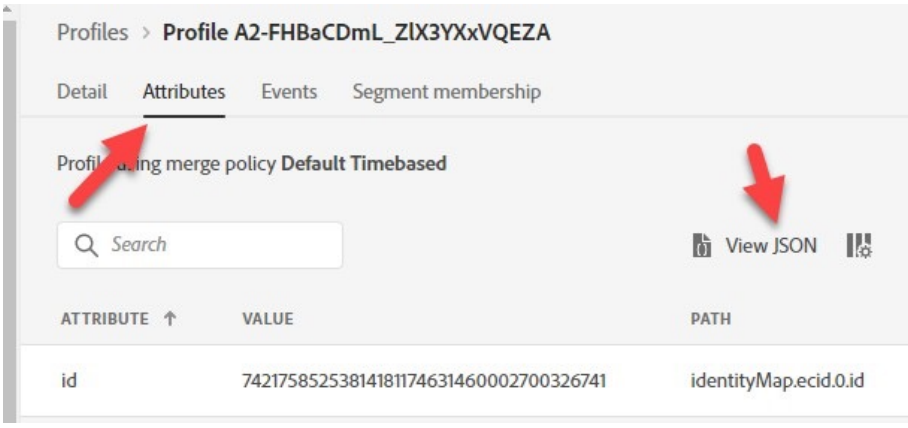
        i. If your Segment using a non-Default Merge Policy, make sure to select it here

e. Click on the Segment Membership tab and see if they are in the Segment



Note: this only displays if they are currently in the segment, not if they used to be and have exited.

f. To see exited status, go to the Attributes tab and click on View JSON



Search for the Segment ID in question. We are looking for something like this:

```
"segmentMembership": {
  "ups": {
    "439ded5b-bf81-4547-89c6-10f5e5bf1307":  {
      "lastQualificationTime": "2022-08-06T18:08:29Z",
      "status": "exited"
    }
  }
},
```

1. This tells us the Profile exited this Segment and what time.
2. It is possible a Profile enters and exits a Segment before a Destination runs in some cases.

g. Validate the timestamp entry into the Segment also.

i. It is possible the entry happened after the Destination ran.

6. Does the Profile reflect what we expect?

a. A good way to validate this step is to pick a few Profiles and ensure their attributes and/or Events have the values that would qualify them for the Segment and look at them in depth

b. e.g.

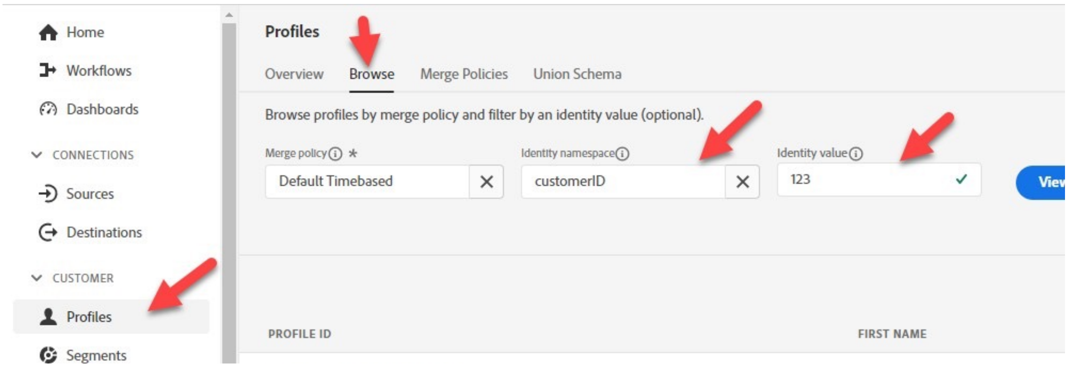
- My Segment filters on City=Lehi. I sent in City=Lehi. Does Profile ABC have City=Lehi?
- My Segment filters on PageView=Homepage. I sent in PageView=Homepage. Does Profile ABC have PageView=Homepage

c. Open the Profile Viewer, put in the namespace and ID you want to lookup, then open the Profile

i. You could also use the API

1. https://developer.adobe.com/experience-platform-apis/references/profile/#operation/retrieveEntity

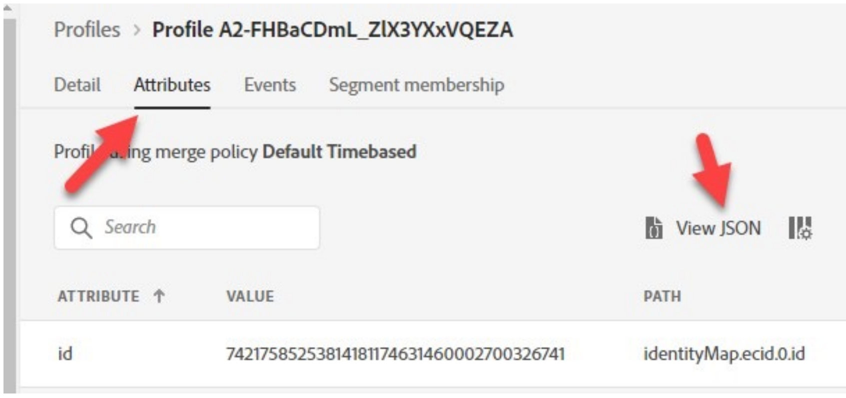
ii. UI



iii. If your Segment using a non-Default Merge Policy, make sure to select it here

iv. Click over to the Attributes tab and look for the attribute you are filtering on in your Segment

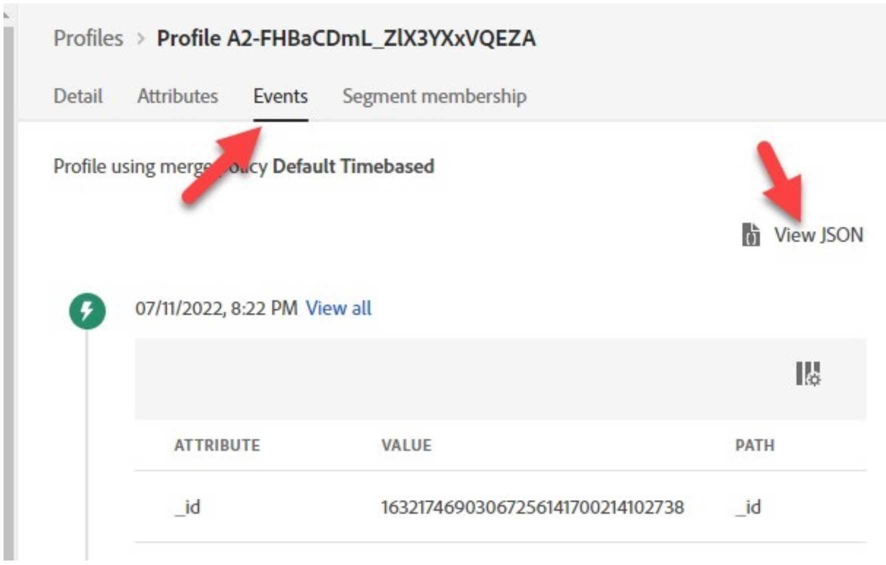
1. I recommend using the View JSON button



2. Does the Profile show the update that was sent in?
3. Did the timestamp update?
4. Spelling?
5. Path to field is the same

v. Click over to the Events tab and look for the Event you are filtering on in your Segment

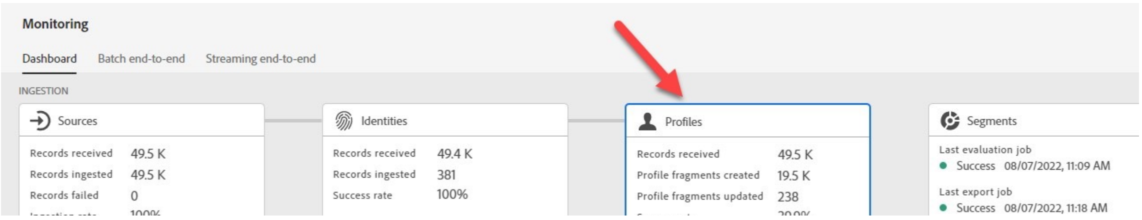
1. I recommend using the View JSON button



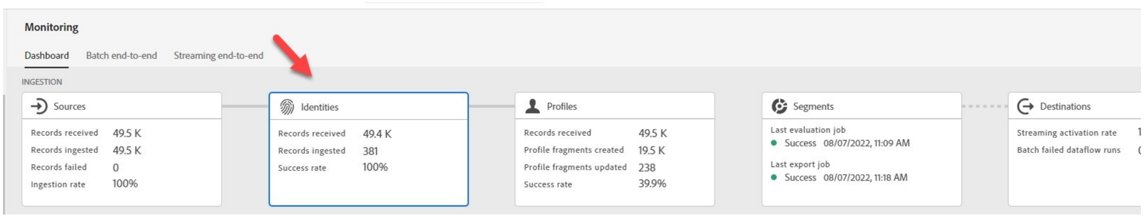
2. Note: the UI has a cap on the number of filters it brings back (~1k confirm number)
   a. If you have a high, use the API
      i. https://developer.adobe.com/experience-platform-apis/references/profile/#operation/retrieveEntity

        b. API also has the ability to filter on the Events, helping you focus on the Event in question
            i. e.g. property=property=web.webPageDetails.isHomepage=false&property=placeContext.localTime<="2020- 07-20"
            ii. This helps if your Segment is filtering on specific criteria
        c. We have saved an Example in your Postman collect
    3. Validate the event exists and meets all your Segment criteria
        a. Timestamp
        b. Spelling
        c. Path to field is the same

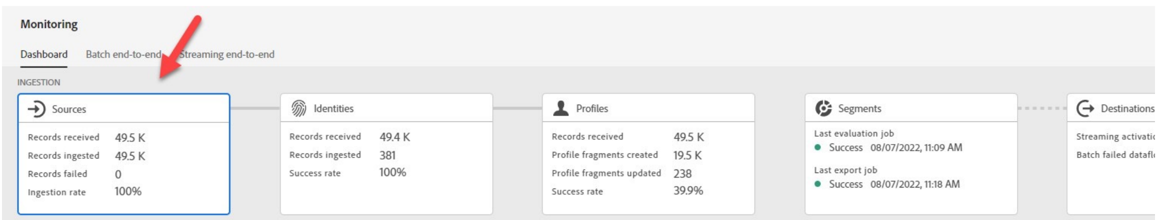
7. Did the data land on the Profile?



8. Did the data add to the Identity Graph?



9. Did AEP Sources pick up the data?



10. Did the data arrive at the AEP endpoint/file location?
    a. In theory, this is the same as Sources, but validate if you are sending a file/payload but Sources is not picking it up
11. Did the data get sent from the source to AEP?
    a. This is external to AEP and without credentials we have no visibility to this