



Documentation

Getting Started

What is Segment?
[How Segment Works](#)
Getting Started Guide
A Basic Segment Installation
Planning a Full Installation
A Full Segment Installation
Sending Data to Destinations
Testing and Debugging
What's Next
Use Cases

Guides

Connections

Unify

Engage

Privacy

Protocols

Segment App

API

Partners

Glossary

Config API

Help

✓ Mobile

✓ Mobile

○ Server

✓ Server

Once you enable Adobe Analytics (formerly known as Omniture or Sitecatalyst) in Segment, you can start sending data from any of the Segment [libraries](#) to an Adobe report suite. When you send events from Segment's mobile SDKs or Cloud-mode libraries, Segment translates that data using a mapping that you configure, and then passes it to the Adobe Analytics [Data Insertion API](#).

The following documentation provides detailed explanation of how both destination the Device-mode and Cloud-mode components work. For FAQs about Device- vs Cloud-mode tracking, unique users, identifiers, and more, see the [Best Practices](#) page!

• [Mapping Segment for Adobe Analytics](#)

• [Identity Resolution](#)

• [Adobe Analytics for Mobile](#)

• [Adobe Analytics for Heartbeat](#)

Planning for Adobe Analytics

Adobe Analytics uses a slightly different approach to tracking than Segment, and it's important to understand the difference so you can effectively set up your account. Segment uses a user-action data model, which uses different types of calls to track different activities of a user on a website or app. Adobe Analytics uses page views as the basic unit of activity, and variables like custom traffic variables (also called 'props'), eVars, list variables, and hierarchy variables to add details that allow more nuanced analysis.

For example, a `Welcome Dialog Dismissed` event in Segment (where the action is "dismissed") with properties that contain the user ID (`user123`) and the dialog name `welcome-dialog`, could be modeled in Adobe Analytics as a `pageView` with variables that represent the dialog name, `visitorID`, and the event name mapping the user action ("dismissed") to an eVar.

Both Segment and Adobe Analytics have recommended standard data for tracking events. Segment has [the Spec](#), and Adobe uses predefined events. Segment automatically maps incoming event data and some product level properties to Adobe's predefined events, when the event data is in the correct Segment Ecommerce Spec format. Video calls using the format described in this document are also automatically mapped. If you're using the Mobile SDKs, mobile lifecycle events are also automatically mapped. If you will be creating Page and Track events that are outside the scope of the Ecommerce spec, you'll need to map those to your Segment destinations settings UI.

We strongly recommend that you create a tracking plan for both your Segment and Adobe Analytics events before you send any events or properties to Adobe. This will help you map your Segment events to Adobe `events`, and Segment properties to Adobe variables. If you decide to set up Adobe Analytics for mobile, you'll have to do this mapping in both the Segment settings, and the Adobe Mobile Services dashboard - so it's good to keep your options open!

Choosing between Device-mode and Cloud-mode

If you're using device-mode JavaScript, by default Segment "bundles" (mobile) or "wraps" (when using Analytics.js) the Adobe libraries. In this configuration, Segment sends Events directly from the client using the Adobe Analytics `AppMeasurement.js` [library](#). Adobe's client-side libraries can provide services to other Adobe suites and products, however they can also increase the size of your page.


If you prefer, you can enable [Cloud-mode](#), and send data through the Segment servers where it is then mapped and sent on to Adobe Analytics. By default, mobile and server sources will use Adobe Analytics in Cloud-mode. You can enable Cloud-mode for Javascript sources from the Adobe Analytics source settings in the Segment app.

Our Cloud-mode Adobe Analytics destination also provides support for **Adobe Mobile Services** "states", "actions", and lifecycle events, metrics, and dimensions.

Note: Segment only supports Adobe Heartbeat through device-mode implementations, using Segment's JavaScript source or mobile SDKs bundled.

Connecting Segment to Adobe Analytics

To set up Adobe Analytics as a destination for your Segment data, Segment needs some information on how to connect to Adobe.

 If you're using Device-mode data collection with Analytics.js, or using a server-side library, you need your Adobe Report Suite ID, and your Tracking Server URL. You'll add this information in the Destination settings in the Segment app UI so that Segment can send information to Adobe. An example tracking server is `jimsbrims.sc.omtrdc.net`. You do not need to include the hypertext transfer protocol (ie. `http://`). For more information on how to identify your analytics tracking server and report suites see Adobe's [documentation](#)

here.

[Settings](#) [Filters](#) [Event Tester](#) [Event Delivery](#)

General

Other

Mappings

Identity Resolution

Timestamps

Adobe Analytics Settings ☒ Enabled

Docs

Copy Settings

Connection Mode

Cloud Mode

Tracking Server URL

This is the URL of your Adobe Analytics server.

jimsbrims.sc.omtrdc.net

Report Suite ID(s) *

You can find your Report Suite ID in your Adobe Analytics Settings page. Multiple report suite ids can be separated by commas: suite1,suite2,suite3.

testreportsuite

Save

Delete Destination

If you're collecting data from mobile devices, you can download the `ADBMobileConfig.json` file instead of specifying these settings in the UI which contains that information. Follow the instructions in Adobe's documentation, [here for iOS](#), and [here for Android](#).

Once you've done that, you can either use the default Ecommerce Spec and the mappings Segment provides, or write custom Page and Track events, and then configure how they map to your Adobe Analytics fields and settings. For more information on how to get started with implementation see the [Mapping Segment to Adobe Analytics](#) guide.

When Will I See Data?

If you just enabled Adobe Analytics for an app already deployed with the Segment library, Adobe can require up to 24 hours to process and display new data. There is also a known [reporting delay](#) on the Adobe side due to [processing times](#).

It can also take up to an hour for all of the mobile users' Segment settings caches to refresh. Once they refresh, the mobile devices learn about the new service and begin sending data to Adobe Analytics.

Adobe Analytics has a real-time reporting feature which displays web page traffic and ranks page views in real time. Configuring and enabling these reports are restricted to Adobe Admin users. To learn more see Adobe's [overview on Real-time reporting](#)

Device-mode - Analytics.js

Device-mode web data is sent using Analytics.js, with Analytics.js either serving as a wrapper/bundle around the Adobe Analytics code, or sending directly to Segment servers where the data is then sent on to the Adobe destination.

Our Adobe Analytics device-mode destination code is open sourced on GitHub. Feel free to check it out: [iOS](#), [Android](#) and [JS](#).

Initialization

When you use Adobe Analytics with Analytics.js, Segment first checks to see if you have any global properties (such as `window.s_account`) or any properties on the `window.s` object, and uses them. However, if it doesn't find anything, it uses the **Report Suite ID**, **Tracking Server URL**, and **Tracking Server Secure URL** (optional) you defined in the destination settings in the Segment app.

Once these required properties are set, Segment loads `appmeasurement.js` version 2.20.0.

Marketing Cloud Visitor ID Service

Segment's Analytics.js destination loads the Adobe `visitorAPI.js` library, but does not initialize it unless you provide your Marketing Cloud Organization ID. When you do, Segment sets `window.s.visitor` with the return value from `window.Visitor.getInstance(<Your Marketing Cloud Org Id>)`. See [the Adobe Marketing Cloud documentation](#) for more information.

Note: Segment loads `visitorAPI.js` in the same script as `appmeasurement.js` because Adobe Analytics requires synchronous execution of the two scripts. Using the visitor API is **optional** but if you do, Segment makes it available on the page.

To use Adobe's Marketing Cloud Visitor ID Service, enter your **Marketing Cloud Organization ID** in the **Identity Resolution** settings tab in the Segment app.

SettingsFiltersEvent TesterEvent Delivery

GeneralOtherMappingsIdentity ResolutionTimestamps

Adobe Analytics SettingsDocs

Marketing Cloud Organization Id

If you would like to use the Marketing Cloud Id Service and use visitorAPI.js, please enter your Marketing Cloud Organization ID. If you do not know your organization ID, you can find it on the Marketing Cloud administration page. It should look something like '1234567ABC@AdobeOrg'.

1234567ABC@AdobeOrg

Drop Visitor ID

This will disable Visitor ID from being passed to Adobe.

No Fallbacks for Visitor ID: Server-Side Only

Note: This setting is for Server-Side only, and only applies when the Drop Visitor ID setting is disabled and you send a marketingCloudId in the Adobe Analytics integration object. Segment's default behavior is to set the Adobe Analytics visitorID based on the destination specific setting for visitorId, falling back to userId then anonymousId. This setting removes the fallbacks.

Save

Cloud-mode - Server-side

"Cloud-mode" data is data sent *without* bundling the Segment-Adobe-Analytics SDK. It can be sent using mobile libraries, Analytics.js, and other server-based sources. Cloud mode data is sent to Adobe using Adobe's data insertion API in XML format. To learn more about Adobe's data insertion API see the [documentation here](#).

For more information on mobile native integrations using Segment's iOS and Android Adobe Analytics SDKs, see the [section "Setting up Adobe Analytics for Mobile"](#).

Important: For data sent from server-side libraries, you must define your events and custom properties BEFORE

you send events to Adobe Analytics server-side destination. However, *for data sent from mobile devices*, Segment sends *every* event along automatically, and you can use the Adobe Analytics [processing rules](#) UI to map actions, lifecycle dimensions, and custom properties from `contextData` to events, props and eVars.

Segment's server-side integration is not open-source. Let's explore what happens when Segment generates the XML to send to Adobe:

1. your **Timestamp Option** is **Timestamp Enabled** or **Timestamp Optional**, Segment sets the `<timestamp>`.

2. or `.track()` events, Segment sets `<channel>` as `properties.channel` or fallback to `properties.category`. If neither property is provided, the XML tag is not set.

For `.page()` events, this XML tag is set as the category of the page call, which is sent by providing both category and name (ie. `.page('Some Category', 'Some Name');`)

3. or `.track()` events, Segment sets `<pageName>` as `properties.pageName`, `properties.page`, `context.page.title`, or `context.screen.name` (in order of precedence). If none of these fields are available and **Page Name Fallback to Screen** is enabled, Segment sets `properties.screen` to `<pageName>`. Otherwise, Segment falls back to 'None'. For `.page()` calls, Segment sets the tag as the name. This can be sent by providing the first parameter:
`.page('Some Name');`

4. Since Adobe Analytics does not support sending timestamped hits a `<visitorID>`, if you have set your Report to **Timestamp Disabled**, Segment sets `<visitorID>` as these values in order of precedence:

- `visitorId` passed manually from the client. Adobe Analytics sets a cookie with their own `visitorId` on the client. You can read from this cookie and pass it to your servers manually and then send it to Segment. This is generally unnecessary, but may be worth considering if you're concerned about unique user identification.

(In Node.js)

```
analytics.track({
  userId: '019mr8mf4r',
  event: 'Gotta catch em all',
  properties: {
    caught: 1738
  },
  integrations: {
    'Adobe Analytics': {
      visitorId: '12345'
    }
  }
});
```

- `userId`

- `anonymousId`



Adobe Analytics timestamp options need to match in the Segment destination settings and Adobe setting. Otherwise, events might be silently dropped by Adobe despite sending a 200 success response to Segment

5. Segment maps a number of other supported XML tags. For example, it sets `<ipAddress>` with the `ip` of the call.

Note: For server side libraries, the `ip` is by default be the `ip` address of your company servers, NOT the customers' own. This means that for server side events, it is best practice to record the customer's `ip` from their requests, and manually send that to Segment as `context.ip`.

Segment also set your `context.locale` (which is automatically collected if using a mobile library) to `<language>`.

Since mobile libraries also send your traits from previous `.identify()` calls inside the `context.traits`, Segment tries to send `<state>` and `<zip>` by looking up `context.traits.address.state` and `context.traits.postalCode` respectively, as noted in the [Identify spec](#). If these lookups fail, Segment uses the default `properties.state` and `properties.zip`.

For mobile libraries, since Segment can detect whether the event occurred while the user had a wifi

connection, it also sends the `<connectionType>` as `lan/wifi`. All other events are treated as `Mobile Carrier` inside Adobe's Mobile Web Reports.

Segment also calculates the timezone offset from UTC/GMT as required by Adobe, and sends a `<timezone>` based on your `context.timezone` and the `timestamp` fields.

Since many out of the box reports from Mobile Web services rely on the `<userAgent>` tag, Segment also maps this to your `context.userAgent`.

Important: The Android library can collect the `userAgent` automatically - however, the iOS library cannot do so. However, since Segment does collect other contextual metadata about a device, it can render a valid iOS `userAgent` string to populate your Mobile Web Reports.

If you are using the Marketing Cloud ID Service, you can pass the **Marketing Cloud Visitor ID** as a destination specific setting and which Segment sets as `<marketingCloudVisitorId>`.

(In Node.js)

```
analytics.track({
  userId: '019mr8mf4r',
  event: 'Gotta catch em all',
  properties: {
    caught: 1738
  },
  integrations: {
    'Adobe Analytics': {
      marketingCloudVisitorId: '12345'
    }
  }
});
```

For `.track()` events only, set some custom link report parameters such as:

- `<linkType>` to `'o'` (stands for `'other'`)
- `<linkURL>` to `context.page.url` with a default fallback to `'No linkURL provided'`
- `<linkName>` to `'Link Name - <whatever was set as the linkURL>'`

On the server, Segment sends *all* property values as `contextData.$propertyKey` by default, so you can further map them with Adobe Processing Rules. You can also choose to add a prefix for properties in the destination's advanced settings page. Properties with a prefix are sent as `contextData.<prefix>.$propertyKey`.

Segment automatically translates [native mobile spec](#) events and forwards them to Adobe Analytics as [Mobile Services Lifecycle Metrics](#).

Specifically, Segment maps the following events:

| Segment Event Name | Adobe Analytics Event Name |
|-----------------------|----------------------------|
| Application Opened | a.LaunchEvent |
| Application Installed | a.InstallEvent |
| Application Updated | a.UpgradeEvent |

The following metrics and dimensions are supported:

- a.AppID
- a.HourOfDay
- a.DayOfWeek
- a.OSVersion
- a.DeviceName
- a.CarrierName

Set `<userAgent>` with `context.userAgent` (which is automatically populated by the Segment libraries). Note this is omitted for mobile events and superseded by `DeviceName` and `OSVersion`.

For any ecommerce events, Segment tries to set `<products>` if possible. The product description has the same logic as the ecommerce event processing done on the client side destination.

Segment follows the same logic as the client side: it looks up any mappings for custom properties and generates the correct `<eVar>`, `<prop>`, and `<hVar>` XML tags.

Finally, Segment sends the event POST request to your **Tracking Server URL**!

Engage

You can send computed traits and audiences generated using [Engage](#) to this destination as a **user property**. To learn more about Engage, schedule a [demo](#).

For user-property destinations, an [identify](#) call is sent to the destination for each user being added and removed. The property name is the snake_cased version of the audience name, with a true/false value to indicate membership. For example, when a user first completes an order in the last 30 days, Engage sends an Identify call with the property `order_completed_last_30days: true`. When the user no longer satisfies this condition (for example, it's been more than 30 days since their last order), Engage sets that value to `false`.

When you first create an audience, Engage sends an Identify call for every user in that audience. Later audience syncs only send updates for users whose membership has changed since the last sync.



Real-time to batch destination sync frequency

Real-time audience syncs to Adobe Analytics may take six or more hours for the initial sync to complete. Upon completion, a sync frequency of two to three hours is expected.

Settings

Segment lets you change these destination settings from the Segment app without having to touch any code.

| SETTING | DESCRIPTION |
|---|--|
| Add application build to Adobe's App ID | <code>boolean</code> , defaults to <code>FALSE</code> . If this setting is enabled, we will add <code>context.app.build</code> , if present, to Adobe's AppID, as <code>''''('')</code> . |
| Collect High-Entropy Client Hints | <code>boolean</code> , defaults to <code>FALSE</code> . If you enable this option, Adobe's library will request high-entropy hints. High-entropy hints contain more detailed information about a user's device. See Adobe documentation for more information. Note: This setting is for web device-mode only. |
| Context Data Variables | <code>text-map</code> , defaults to <code>{}</code> . Map values you pass into the context object to Context Data Variables in Adobe Analytics. Then you can use processing rules to map you Context Data Variables in Adobe to other variables with Adobe Analytics Processing Rules. In the box on the left, put your Segment context key. The box on the right is what Context Data Variable you'd like it to be mapper to in Adobe. If you have a nested object, separate the name with a <code>.</code> . For example if you wanted to map the page referrer, you would put: <code>page.referrer</code> . NOTE: By default Segment send alls your properties as Context Data Variables so you do not need to map them again here. |
| Context Data Property Prefix | <code>string</code> . If you would like to prefix your Segment properties before sending them as <code>contextData</code> , enter a prefix here. |

| SETTING | DESCRIPTION |
|---|---|
| List Variable and Prop Custom Delimiter: Server-Side Only | <p><code>map</code>, defaults to <code>{}</code>.</p> <p>Add a custom delimiter to concatenate Adobe Analytics IVars or props sent as an array. Note, if you do not specify a custom delimiter, arrays will be concatenated with a comma. Please add the Adobe Analytics IVar or prop on the left and select a custom delimiter on the right. IVars must be in format 'list1', 'list2', etc. and props in format 'prop1', 'prop2', etc. Must be all lowercase with no whitespace.</p> |
| Drop Visitor ID | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>This will disable Visitor ID from being passed to Adobe.</p> |
| Enable pageName for Track Events | <p><code>boolean</code>, defaults to <code>TRUE</code>.</p> <p>If you do not want to attach <code>pageName</code> for your <code>.track()</code> calls, you can disable this option.</p> |
| eVars | <p><code>map</code>, defaults to <code>{}</code>.</p> <p>Map your Adobe Analytics eVar names to the property names you're using in your Segment events. Enter a Segment property name on the left and an Adobe Analytics eVar number on the right. You can view your Segment events and properties in your Schema.</p> |
| Events | <p><code>mixed</code>, defaults to <code>.</code></p> <p>Map your Segment events to custom Adobe events.</p> |
| Events V2 (Bundled Mobile Only) | <p><code>text-map</code>, defaults to <code>{}</code>.</p> <p>Note: If you are bundling our Adobe Analytics SDK on your mobile device, you should map your events here. If you are sending via server side you should use the Events option instead. Map your Adobe Analytics property names on the left to Adobe event names on the right. Your events MUST be set up in Adobe and mapped here in order to be forwarded to the destination. This setting only applies to Mobile integrations with Adobe.</p> |
| Heartbeat Tracking Server URL | <p><code>string</code>. This is the URL of your Adobe Heartbeat server. Please contact Adobe to obtain this URL.</p> |
| Hierarchy Variables | <p><code>map</code>, defaults to <code>{}</code>.</p> <p>Map your Adobe Analytics hVars to the property names you're using in your Segment page calls. Enter a Segment property name on the left and an Adobe Analytics hVar number on the right. You can view your Segment page calls and properties in your Schema.</p> |
| List Variables | <p><code>map</code>, defaults to <code>{}</code>.</p> <p>Map your Adobe Analytics list variables names to the property names you're using in your Segment events. Enter a Segment property name on the left and an Adobe Analytics list variable number on the right. You can view your Segment events and properties in your Schema.</p> |
| Marketing Cloud Organization Id | <p><code>string</code>. If you would like to use the Marketing Cloud Id Service and use visitorAPI.js, please enter your Marketing Cloud Organization ID. If you do not know your organization ID, you can find it on the Marketing Cloud administration page. It should look something like '1234567ABC@AdobeOrg'.</p> |
| Merchandising Events | <p><code>mixed</code>, defaults to <code>.</code></p> <p>Configure merchandising event, such as purchase or currency events. This is currently in Beta Testing and not generally available.</p> |
| OAuth Client ID | <p><code>string</code>. Client ID for your OAuth Server-to-Server Credential</p> |
| OAuth Client Secret | <p><code>password</code>. Client Secret for your OAuth Server-to-Server Credential</p> |
| Page Name Fallback to Screen | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>If "Enable pageName for Track Events" is enabled but page name is not populated by default mapping, then page name will be populated by <code>screen</code> property if this is enabled.</p> |
| Prefer VisitorID for Hybrid Timestamp Reporting | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>If you enable this option and you also have a <i>Timestamp Optional</i> reporting suite, you can opt to send your visitorID instead of the timestamp since Adobe does not allow you to send both. If you care more about your user attribution, you should enable this if you're using a hybrid timestamp reporting suite.</p> |

| SETTING | DESCRIPTION |
|---|--|
| Product Identifier | <p><code>select</code>, defaults to name.</p> <p>Adobe Analytics only accepts a single product identifier. Use this option to choose whether we send product name, id, or sku.</p> |
| Props | <p><code>map</code>, defaults to {}.</p> <p>Map your Adobe Analytics property names to the property names you're using in your Segment events. Enter a Segment property name on the left and an Adobe Analytics property number on the right. You can view your Segment events and properties in your Schema.</p> |
| No Fallbacks for Visitor ID: Server-Side Only | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>Note: This setting is for Server-Side only, and only applies when the Drop Visitor ID setting is disabled and you send a marketingCloudId in the Adobe Analytics integration object. Segment's default behavior is to set the Adobe Analytics visitorID based on the destination specific setting for visitorId, falling back to userId then anonymouId. This setting removes the fallbacks.</p> |
| Report Suite ID(s) (required) | <p><code>string</code>. You can find your Report Suite ID in your Adobe Analytics Settings page. Multiple report suite ids can be separated by commas: <code>suite1,suite2,suite3</code>.</p> |
| Send Both Timestamp and VisitorID for Timestamp Optional Reporting Suites | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>If you have a <i>Timestamp Optional</i> Reporting Suite, you can opt to send <i>both</i> the timestamp and the visitorID in your XML when sending events server side. However, note that this is <i>NOT</i> recommended by Adobe as it may lead to out of order data. This setting will only work for reporting suites that have optional timestamp setting enabled.</p> |
| Send False values | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>By default, we don't send properties with a <code>false</code> as value on cloud mode. Enabling this setting will send any boolean property where value is <code>false</code>.</p> |
| SSL | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>Check this box if you would like your Adobe Heartbeat calls to be made over HTTPS.</p> |
| Timestamp Option | <p><code>select</code>, defaults to enabled.</p> <p>Adobe Analytics can have Report Suites that will accept timestamped, non-timestamped or hybrid data. Note that we can only play historical data for timestamped or hybrid Report Suites.</p> |
| Tracking Server Secure URL | <p><code>string</code>. This is the secure URL of your Adobe Analytics server. Please input your URL without <code>https://</code> prepended.</p> |
| Tracking Server URL | <p><code>string</code>. This is the URL of your Adobe Analytics server. Please input your URL without <code>http://</code> prepended.</p> |
| Use Legacy LinkName | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>Before sending LinkName to Adobe Analytics, prepend the URL with <code>Link Name -</code>. This setting enables a legacy behavior for backwards compatibility. You probably want to keep this setting turned off.</p> |
| Use Secure URL for Server-side | <p><code>boolean</code>, defaults to <code>FALSE</code>.</p> <p>Enable this option if you want to use the 'Tracking Server Secure URL' endpoint instead of the normal URL for server-side and Cloud Mode calls.</p> |
| Use UTF-8 Charset | <p><code>boolean</code>, defaults to <code>TRUE</code>.</p> <p>Only applicable on server-side or cloud-mode. If this setting is enabled, we will send the payload to Adobe Analytics with UTF-8 charset. Useful when your events contains accents or other special characters.</p> |

This page was last modified: 27 Feb 2024

Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

Visit our Support page

Help improve these docs!

Edit this page

+ Request docs change

Was this page helpful?

Yes

No

Get started with Segment

Segment is the easiest way to integrate your websites & mobile apps data to over 300 analytics and growth tools.

Your work e-mail

Request Demo

or

Create free account

© 2025 Segment.io, Inc.

Privacy

Terms

Website Data Collection Preferences

