Journey JavaScript SDK

autotrackOfferStateChangesInAdobeAnalytics

5/1/2026

Contents

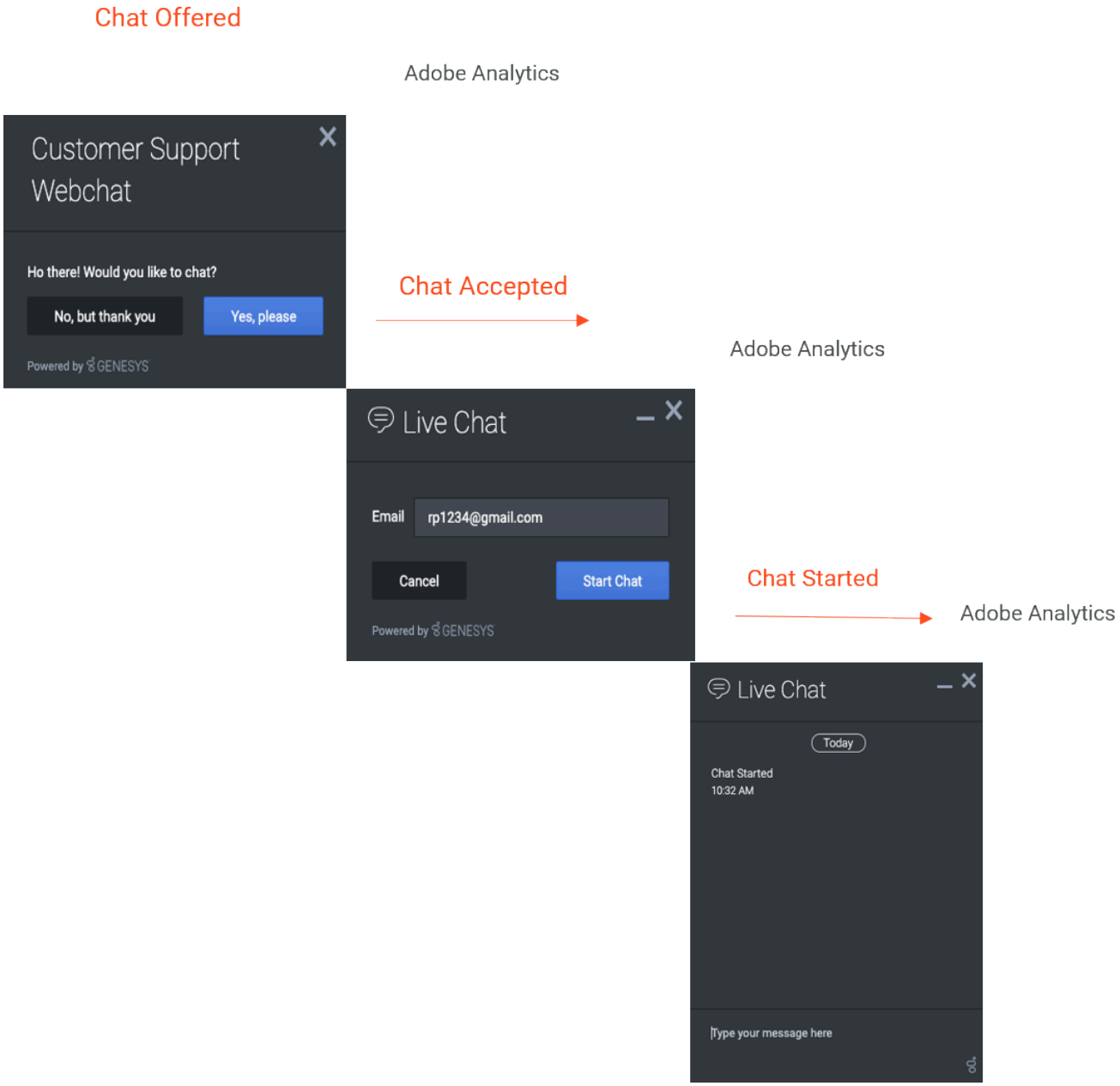
Learn how to track web chat and content offer state changes on a webpage and send them to Adobe Analytics.

Prerequisites

- Required products for integration:
  - Genesys Predictive Engagement with any Genesys platform chat widget
  - Adobe Analytics with latest version of AppMeasurement (to ensure capturing of the Experience Cloud Identifier)
  - Adobe Experience Cloud Identity Service (formerly Marketing Cloud)

> **Important**
>
> This article only applies to customers using web chat. If you are a Genesys Cloud CX customer, we encourage you to use the new web messaging feature to replace web chat.

Description

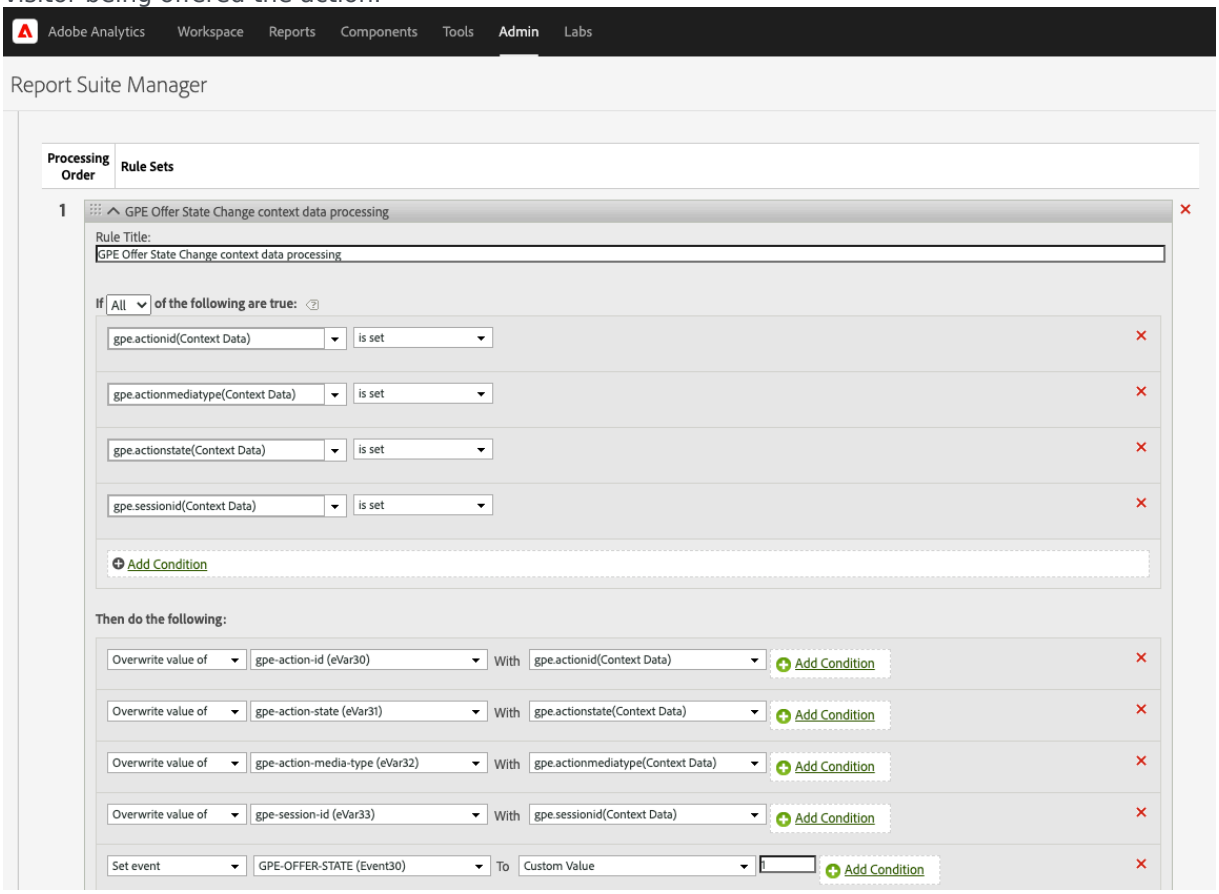


The `autotrackOfferStateChangesInAdobeAnalytics` module tracks web chat and content offer states and sends them in real time to Adobe Analytics for reporting purposes. You can use the data to create reports in Adobe Analytics. This data helps you to evaluate where these events occur in the customer journey and understand how they are influencing your conversion rate.

When the module loads, specify the identifier for the Adobe Analytics library on your webpage.

High-level workflow

1. Enable the following on your website:
   - Genesys Predictive Engagement chat
   - Integration with Adobe Analytics using this `autotrackOfferStateChangesInAdobeAnalytics` module
   - Adobe Analytics tracking including the Experience Cloud Identity Service
2. Genesys Predictive Engagement begins sending data tracking requests for each web chat and content offer state change event in real time to Adobe Analytics.
3. Configure processing rules in Adobe Analytics:
   - Identify which eVars to use and which action states to map. The eVars that you use vary based on the eVars in use in your report suite currently. We recommend that you create a new eVar for each piece of context data that the SDK module sends.
   - Map contextData variables to eVars and, if necessary, map contextData elements to props. The following image is an example of what the processing rules might look like in Adobe Analytics. Once the data is flowing correctly, eVar30 will contain a unique identifier for the proactive chat or content offer. eVar 31 will contain the state associated to the event. eVar32 will identify whether it was a web chat or content offer. eVar33 will contain the web session identifier associated to the visitor being offered the action.



- Increment an event every time the state change tracking call is made.

4. Set up your Adobe Workspace dashboard with custom reporting (see reporting examples).

States tracked and event information sent

States tracked

Following are the available web chat and content offer states tracked for Adobe Analytics.

States	Web Chats	Content Offers
Offered	X	X
Accepted	X	X
Started	X	
Engaged	X	
Rejected	X	X
Ignored	X	X
Errored	X	X
Timedout	X	
Abandoned	X	
Completed	X	

Event information sent

Following are the event information that Genesys Predictive Engagement sends to Adobe Analytics as contextData variables for each state tracked.

Event Information	Web Chats	Content Offers
gpe.actionId	X	X
gpe.actionState	X	X
gpe.actionMediaType	X	X
gpe.sessionId	X	X
gpe.conversationId (available for Genesys Cloud CX customers only)	X	

Reporting examples

Following are some reporting examples that this integration provides:

- Number or percent of Adobe Analytics Reporting segment visitors who accepted a chat and number that proceeded to place an order post-chat
    - Further breakdown by webpage and Adobe segment to reveal patterns of behavior
- Predictive Engagement as an influencing touchpoint on buying behavior
- Quantify orders or sales post-chat with attribution modeling within 30 days of the chat interaction
- Online revenue (transaction value) generated post-chat
- Conversion rate as a percent of overall visitors and as a percent of total chats engaged
- Funnel reporting to show chat impact on conversions
- Page reporting to show the number of pages with the highest number of offered and accepted chats
- Next and previous page flow reporting of web chat engagements
- Break down of web chat engagements by Adobe segments

Signature

```
ac('load', 'autotrackOfferStateChangesInAdobeAnalytics', config, [callback]);
```

Example

```
ac('load', 'autotrackOfferStateChangesInAdobeAnalytics', {
  adobeAnalyticsObjectName: 'customNameForAnalyticsLibrary'
}, function () {
    console.log('"autotrackOfferStateChangesInAdobeAnalytics" has been loaded');
});
```

Config (optional)

Description: Only use this property when the Adobe Analytics tracker library is accessible globally on your webpage but its name is something other than "s." For more information, see Make the Analytics Object Globally Accessible.

Type: Object

Properties: adobeAnalyticsObjectName

Example

The value that you provide is the name of the Adobe Analytics library scoped globally under window. So, if your tracker library is accessible under "window.customNameForAnalyticsLibrary," the config object looks like the

following:

```
{
  adobeAnalyticsObjectName: 'customNameForAnalyticsLibrary'
}
```

Callback (optional)

When a module loads, callback executes. No arguments pass to the callback.