This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Gplus Adapter for Siebel CRM Deployment Guide

After Call Work Timer

1/11/2026

This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Journey JavaScript S

autotrackOfferStateChangesInAdobeAnal

Contents

- 1 ACW Timer description
- 2 Configuring the ACW Timer
- 3 Change the ACW Thresholds and colors

For the OpenUI mode only, the After Call Work (ACW) timer enables agents to see their remaining after call work time.

ACW Timer description

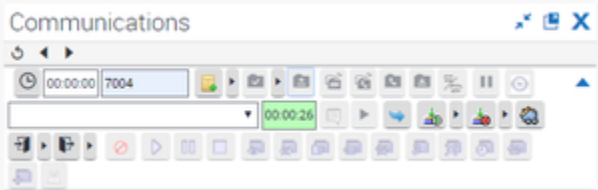
The standard Siebel solution provides only one timer in the CTI toolbar, which shows a work item's (a call, for example) duration. When using the OpenUI mode, this additional ACW timer enables agents to see the remaining after call work time that the agent has to complete after call work. The ACW timer appears in CTI toolbar near the Work Item List's drop-down menu.

Important

Be aware of the following:

- The remaining time shown in the timer is approximate. It's usually a little greater than the actual time remaining, but it depends on the overall performance of the Adapter and Siebel.
- If you are not implementing the ACW Timer, this page and the procedure do not apply. However, if you are not implementing it, you will need to remove the following event handlers from your CTI configuration: `UntimedWrapupMode`, `StartWrapupMode`, `StopWrapupMode`, `StopWrapupMode2`, and `StopWrapupMode3`.

When you, as the administrator, configure this ACW Timer, when an agent switches to the corresponding After Call Work, the Timer appears with a light green background in the CTI toolbar to the right of the Work Item List's drop-down menu.



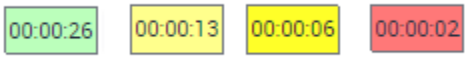
ACW Timer (CTI Toolbar)

The ACW Timer counts down the remaining after call work time, as you configured in your after call work timeout. When the After Call Work mode ends, the timer disappears.

You can configure the Timer to change the background color and the countdown thresholds that display to the agent as the agent approaches the end of the after call work period.

By default, there are four thresholds configured: 0%, 50%, 70% and 90%. When the spent time is greater than the threshold, the background color changes

progressively in this order: light green, light yellow, yellow, red.



Default ACW Timer Threshold Colors

> **Tip**
> If the after call work time extends until the agent makes ready himself/herself), the ACW Timer counts the time in seconds, starting from zero (0), independently of the configured ACW timeout. It also displays with a white background (the color for the CTI toolbar timers), as defined in the Siebel theme.

Configuring the ACW Timer

> **Important**
> This new procedure is associated with the following section in the 8.0 Deployment Guide: Chapter 6, Deploying the Voice Component > Configuring Siebel > Configuring Siebel Using the Siebel Web Client., that starts on page 220, per the PDF.

To implement this ACW Timer:

1. Copy all of the script files to the Siebel Server Open UI customization folder (**siebsrvr\WEBMASTER\siebel_build\scripts\siebel\custom**, for example).
   - For 8.1.1.11/8.2.2.4 (IP2013) or IP 2014, the source directory is:
     * **//SCRIPTS/OUI**, where is 8.1_8.2_OUI or IP 2014.
   - For IP 2017, the source directory is:
     * **//SCRIPTS/**, where is IP 2017.
2. To register the custom scripts, do the following:
   1. Log into Siebel Server as a Siebel administrator.
   2. Navigate to Site Map > Administration - Application > Manifest Files.
   3. Add the following new record into the Files applet: `siebel/custom/GplusCommToolbarUpdate2.js`
   4. Navigate to Site Map > Administration - Application > Manifest Administration.
   5. Add a new record into the UI Objects and subordinate applets, as shown in the following tables:

UI Objects applet row

Parameter name	Value
Inactive Flag	N

Parameter name	Value
Type	Application
Usage Type	Common
Name	PLATFORM INDEPENDENT

Object Expression applet row

Parameter name	Value
Inactive Flag	N
Group Name	
Expression	Desktop
Level	1
Operator	
Web Template Name	

Files applet row

Parameter name	Value
Inactive Flag	N
Name	siebel/custom/GplusCommToolbarUpdate2.js

Change the ACW Thresholds and colors

You can change the countdown percentage and the background colors associated with the thresholds in the the `GplusCommToolbarUpdate2.js` file. The default values have the following data structure:

```
// Wrap-up timer background colors map
var gplusWrapupTimerBgColors = [
   { "percent":0,  "color":"#bbffbb" },
   { "percent":50, "color":"#ffff8c" },
   { "percent":70, "color":"#ffff28" },
   { "percent":90, "color":"#ff7878" },
   ];
```

You can modify these rows, delete them, or add new rows.

Contents

- 1 Description
- 2 High-level workflow
- 3 States tracked and event information sent
  - 3.1 States tracked

- 3.2 Event information sent
- 4 Reporting examples
- 5 Signature
- 6 Example
- 7 Config (optional)
  - 7.1 Example
- 8 Callback (optional)

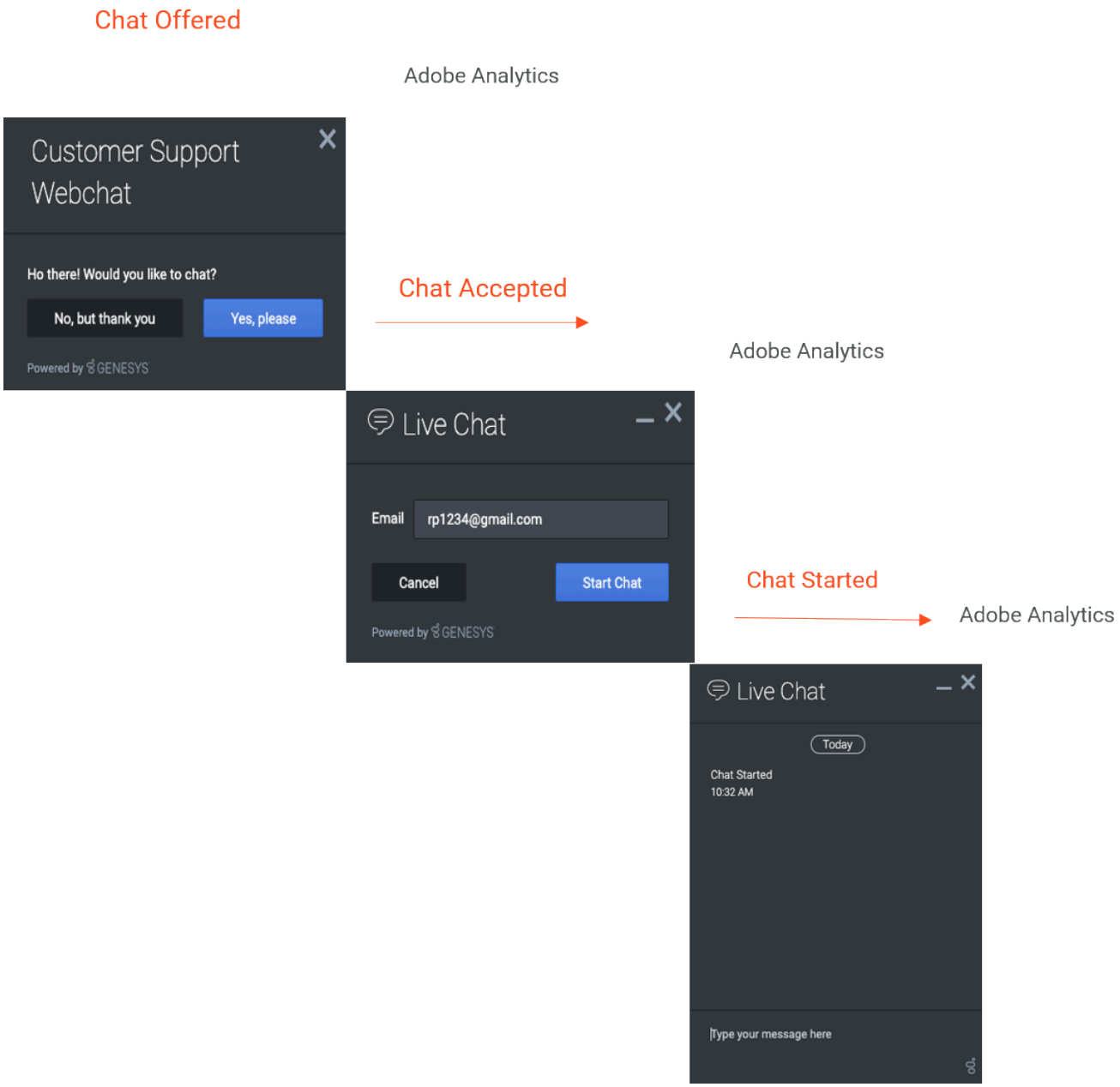
Learn how to track web chat and content offer state changes on a webpage and send them to Adobe Analytics.

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

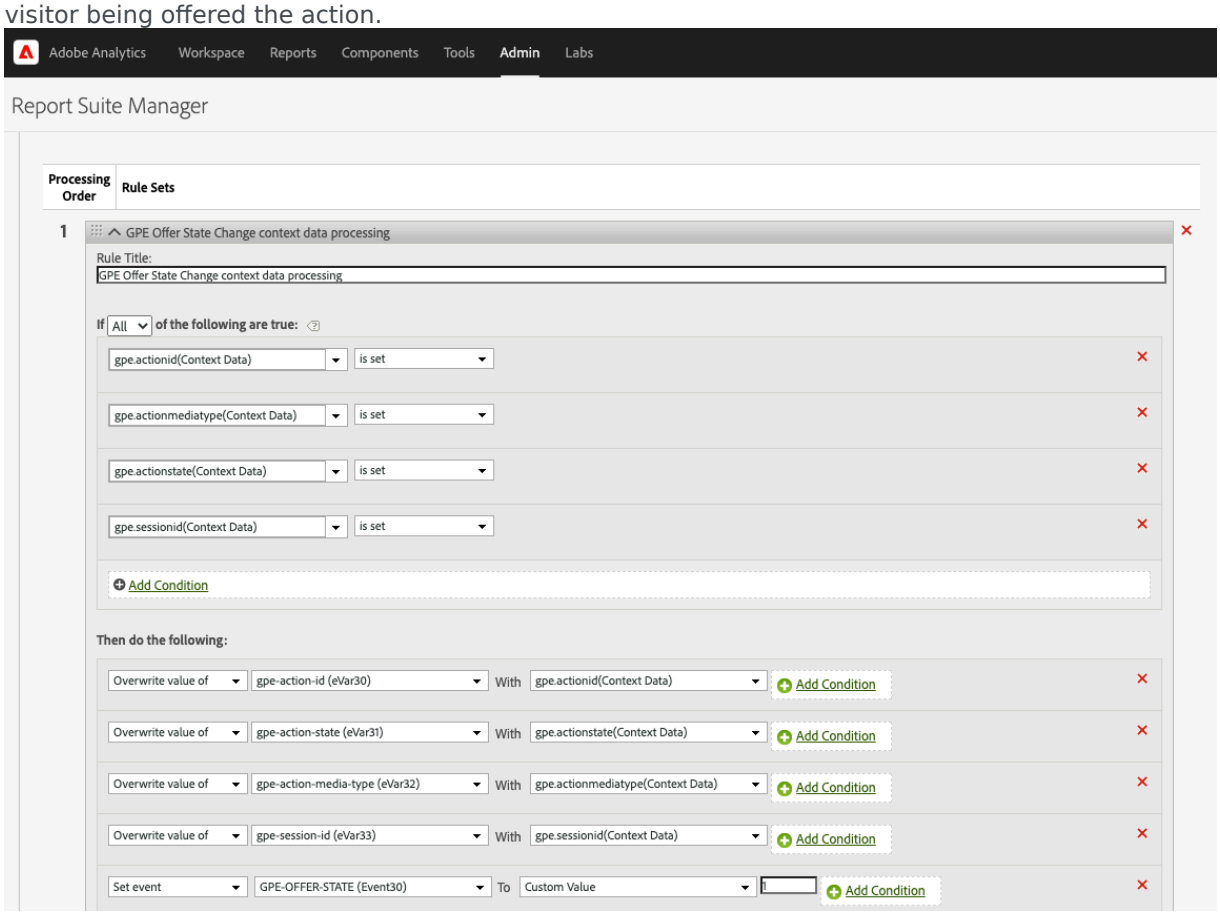


The `autotrackOfferStateChangesInAdobeAnalytics` module tracks web chat and content offer states and sends them in real time to Adobe Analytics for reporting purposes. You can use the data to create reports in Adobe Analytics. This data helps you to evaluate where these events occur in the customer journey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