



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.

Widgets Developer's Guide

Accessibility

5/10/2025

Contents

- [1 Overview](#)
- [2 What is WCAG?](#)
- [3 Support](#)
 - [3.1 Web Content Accessibility Guidelines \(WCAG\) 2.1, Level AA Plugin Support by Platform](#)
- [4 Screen reader support](#)
- [5 Keyboard accessibility](#)
- [6 Focus trap](#)
- [7 Color contrast](#)
- [8 Browser zoom and text resizing](#)
- [9 Customization](#)
 - [9.1 Localization](#)
 - [9.2 Configuration options](#)
- [10 Resources and tools used](#)
 - [10.1 Online](#)
 - [10.2 Screen readers](#)

-
- Administrator
 - Developer

Learn how Widgets aligns with the Web Content Accessibility Guidelines (WCAG) 2.1, Level AA.

Related documentation:

-

Overview

Genesys provides a Voluntary Product Accessibility Template® - VPAT® report from ITI, to document conformance of Widgets to WCAG 2.1 specification. The VPAT® report is a standardized template for documenting conformance to various accessibility specifications. The VPAT® report provided by Genesys follows the W3C/WAI's WCAG 2.1 specification, as this is an international standard adopted and recognized by our customers worldwide. The Genesys VPAT® can be downloaded here: [Genesys Widgets WCAG 2.1 AA VPAT®](#).

What is WCAG?

Web Content Accessibility Guidelines (WCAG) 2.1 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content more accessible to a wider range of people with disabilities and will also often make Web content more usable to users in general. WCAG relies on four guiding principles for building accessible UIs:

1. **Perceivable:** Information and user interface components must be presentable to users in ways they can perceive.
2. **Operable:** User interface components and navigation must be operable.
3. **Understandable:** Information and the operation of user interface must be understandable.
4. **Robust:** Content must be robust enough that it can be interpreted by a wide variety of user agents, including assistive technologies.

Support

Widgets provides support for WCAG 2.1 Level AA for desktop and mobile web browsers. However, not all Widgets meet these guidelines. The table below lists the Widgets that address and meet the standard accessibility requirements to help assist users with vision, hearing, or mobility impairments in gaining greater access to the customer support.

Web Content Accessibility Guidelines (WCAG) 2.1, Level AA Plugin Support by Platform

| Plugin | Genesys Cloud CX |
|-----------------|------------------|
| WebChat | Level AA |
| CallUs | Level AA |
| ChannelSelector | Level AA |
| Engage | Level AA |
| SideBar | Level AA |
| Callback | Level AA |

Widgets supports all the WCAG 2.1 Level AA accessibility guidelines for both mobile and desktop. Some of the high-level features are listed below.

Screen reader support

Supported widgets are accessible via screen readers, which announce the following: all the textual and non-textual content on the Widgets window elements, new chat messages sent by the agent to the user, outgoing messages sent by the user to the agent, and error messages. To achieve a consistent reading behavior of live data across all the screen readers and the browsers, recommended ARIA live regions have been implemented in WebChat for reading new messages.

Genesys Widgets is built and maintained following WCAG A/AA accessibility standards. These standards are supported by popular screen readers, such as JAWS, VoiceOver (MacOS, iOS), TalkBack, and others.

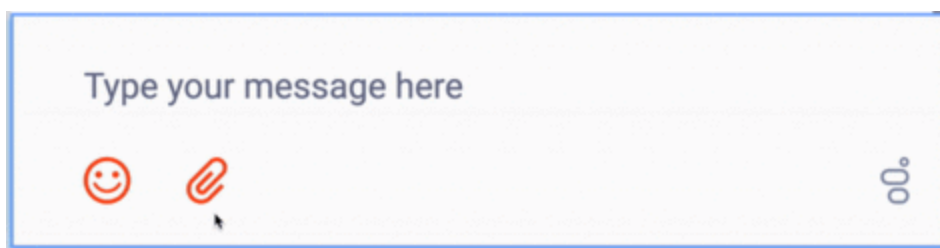
Genesys supports compatibility with most commonly used screen readers in the following cases:

- **PC Windows OS:** JAWS with Chrome and Internet Explorer 11 or Edge
- **Mac OS:** VoiceOver with Chrome and Safari
- **iOS:** VoiceOver with Safari
- **Android:** TalkBack with Chrome

Important

Not all screen readers may read all the textual and non-textual window functionality. There are known issues around Firefox and Internet Explorer with some screen readers. The content is read as long as the screen reader model is supported on that particular browser.

Keyboard accessibility



Supported Widgets are accessible via the keyboard. Users may navigate to and within any widget using the **tab** key or **shift+tab** key combo. For dropdowns and the date picker, the user can highlight a selection using the **arrow** keys. The **enter** or **space** key can then be used to make a selection, send a message, or activate a button.

- **tab** - step forward to the next element
- **shift+tab** - step backward to the previous element
- **arrow keys** - move between options within a dropdown or date picker
- **enter** - make a selection or submit
- **space** - make a selection or activate a button

Important

In macOS, Safari Browser's accessibility settings must be enabled to allow for proper keyboard navigation in Widgets.

Focus trap

In desktop browsers, when the Engage Offer Widget is rendered in an overlay modal dialog with the background disabled, the focus is trapped within the content until it is closed. In mobile devices, all the widget layouts are expanded to full screen modal dialog. These mobile layouts contain the aria-modal property as recommended in the W3C ARIA Dialog modal best practices.

Important

Widgets does not add the "aria-hidden" attribute on the customer page html elements. Due to this limitation, when using screen reader gestures in some Android devices, the focus may not be trapped within the widget. To trap the focus, a custom event handling script needs to be added subscribing to the widget opened event. Also, add the "aria-hidden" attribute on the host page html elements, and remove them from subscribing to the corresponding closed/minimized events.

Color contrast

Text and background colors and buttons now meet WCAG 2.1 Level AA accessibility contrast guidelines. This allows text to be read clearly. There are changes in the default Widgets themes to increase color contrast in our Dark and Light themes. Changes include border, button, link, text, and background color adjustments to meet the contrast requirements while maintaining the same look and feel. In addition, there is an outline to indicate which element or section of each widget is in focus.

The following table details some examples of the changes included as part of WCAG implementation. The changes apply to both the light and dark themes, and the light theme is used in the table examples.

| Description | Before | After |
|---|---|---|
| As per the “1.4.11 Non-text contrast” success criterion, icon color has been modified to meet the contrast requirement of at least 3:1 ratio against the adjacent/background color. |  |  |
| As per the “1.4.3 Contrast (minimum)” success criterion, background color of the primary button has been modified to ensure that the contrast ratio of at least 4.5:1 exists between text and background. |  |  |
| As per the “1.4.3 Contrast (minimum)” success criterion, placeholder text color has been modified to ensure that the contrast ratio of at least 4.5:1 exists between text and background. |  |  |
| As per the “2.4.7 Focus visible & 1.4.11 Non-text contrast” success criterion, borders with 3:1 contrast ratio have been added to highlight the focused state of the menu items. |  |  |

Browser zoom and text resizing

Genesys Widgets supports zooming in and out, or resizing text using the browser's built-in controls. This makes it easier for some viewers to read text on the screen.

Important

The SideBar Widget can only support the zoom feature properly if it contains six or fewer rows.

Customization

Localization

Aria labels are used throughout Genesys Widgets to supply callouts and context for screen readers. These labels have been added to the standard localization language pack definition, allowing you to customize these labels yourself. All aria label strings are prefixed with "aria" to make them easy to identify. Review each widget's localization reference page to find these new aria labels. Example: WebChat Widget Localization reference.

Configuration options

| Widget | Option name | Description |
|---------|----------------------------|---|
| WebChat | ariaIdleAlertIntervals | An array containing the intervals as a percentage at which the screen reader will announce the remaining idle time. By default, it is enabled with the following time intervals, and it is customizable according to the user's needs. Configuring a value of <code>false</code> will let the screen reader call out idle time for every change. |
| WebChat | ariaCharRemainingIntervals | An array containing the intervals as a percentage at which the screen reader will announce the remaining characters when the user inputs text into the message area. By default, it is enabled with the following intervals, and it is customizable according to user needs. Configuring a value of <code>false</code> will let the screen reader call out remaining characters for every change. |
| WebChat | emojiList | <code>emojiList</code> must be configured with display names to support the screen reader calling out the emoji name. These emoji names are applied as aria-label attributes on the non-text emoji markup. |

Resources and tools used

Online

- webaim.org
- deque.com
- contrastchecker.com

Screen readers

- JAWS
- NVDA
- VoiceOver