



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.

# Deployment Guide

Genesys Widgets Current

# Table of Contents

<b>Genesys Widgets Deployment Guide</b>	<b>3</b>
Tested Browsers	13
Cookies	14
Required Versions of Genesys Components	19
Configuring Genesys Widgets	20
Localization	26
Styling the Widgets	31
Icons	42
Rich Messaging	45
Accessibility	69

# Genesys Widgets Deployment Guide

## Tip

The latest version of our documentation (titled “**Current**”) relates to release **9.0.x**.

This guide provides the steps required to instrument your website with Genesys Widgets.

## Important

As of release **9.0.017.01** Genesys Widgets is no longer available to download through the Genesys Downloads Center. Instead, it can be downloaded as a zip file from the **Content Delivery Network (CDN)**.

## Audience

This document is for website developers who are in charge of website code. You must have knowledge of HTML, JavaScript, and CSS.

## Cookies

Genesys Widgets uses cookies to store non-sensitive data in the browser. These cookies are used to restore a chat session, to track the state of the UI, to store a user's decision, and more. The end-user's browser must allow cookies for Genesys Widgets to operate properly.

## Important

No personally identifiable information (PII) is ever stored in cookies, local storage, or session storage by Genesys Widgets.

## Sub-domains

Normally, cookies cannot be transferred between sub-domains of a website unless they are configured to do so. Genesys Widgets automatically detects the domain of the host site and configures all cookies to be transferable between sub-domains. For example, you could start a chat

on **www.testsite.com** and restore that chat session on **store.testsite.com**, **support.testsite.com**, or **portal.testsite.com**.

### Important

For Genesys Widgets deployment recommendations, see [Chat API Deployment Notes](#).

## Cookie Support in Test Environments

Genesys Widgets uses special cookies that persist across sub-domains. This is a critical feature for plugins like WebChat that need to restore an active chat session while navigating around a website. The side effect of using this type of cookie is they won't work when using test environment domain names such as "localhost" or an IP address. You must use a fully-qualified domain name (FQDN) such as "localhost.com" or any other variant that can be identified as a domain name. Cookies will also fail to work if you run the test site as an HTML file path directly in the browser.

One workaround is to update your system's **hosts** file to create an FQDN alias for "localhost", your test environment's name, or an IP address.

### Example

```
127.0.0.1    localhost
127.0.0.1    localhost.com
```

### Introduced: 9.0.004.03

Starting **9.0.004.03** version, a fully-qualified domain name (FQDN) such as "localhost.com" or any other variant that can be identified as a domain name is not mandatory but recommended. These cookies will also work when using test environment domain names such as "localhost" or an IP address.

## How Can I Deploy Genesys Widgets?

We support two deployment methods:

- **Lazy loading**—Lazy loading breaks the JavaScript bundle apart into individual plugin files and loads them into the page only as you need them. This is the preferred method of deploying Genesys Widgets.
- **All-In-One**—The All-In-One method is the deprecated method of deploying Genesys Widgets. In this method, you have one JavaScript file and one CSS file that contain all plugins and resources.

## Deploying Genesys Widgets (Lazy loading)

### Files Used

- **widgets/cxbus.min.js**
- **widgets/plugins/widgets-core.min.js**
- **widgets/plugins/\*.\***

A good starting point is the following script:

```
<script src="widgets/cxbus.min.js"
onload="javascript:CXBus.configure({debug:true,pluginsPath:'build/
plugins/'});CXBus.loadFile('path/to/
widgets.config.js').done(function(){CXBus.loadPlugin('widgets-core')});"></script>
```

This script does the following:

1. Loads **cxbus.min.js**. This makes the global CXBus instance available.
2. Configures CXBus to turn on debug logging and set the path to the Widgets plugin folder.
3. Load your configuration file, **widgets.config.js**. (This is an imaginary file. You must create it).
4. Loads **widgets-core**, the core Genesys Widgets library.

Use this script as a starting point and customize it as needed.

Remember that your configuration can be defined inline on the page or loaded in as a separate file (as shown in this script).

### Important

Whichever method you choose, you must ensure your configuration is in the page before you load **widgets-core**. Otherwise, **widgets-core** cannot read the configuration.

### Important

Refer to this article for help [configuring Genesys Widgets](#).

## On-Demand Lazy loading

Genesys Widgets is designed to load plugins into the page on-demand as you use the product. For example, if you call the command **WebChat.open**, CXBus fetches the **webchat.min.js** plugin from the **plugins/** folder and loads it into the page. Any WebChat command triggers it to load. Likewise, WebChat calls WebChatService commands, thus CXBus loads **webchatservice.min.js** into the page

as well.

## Preloading Plugins

In some cases, you might not want to load plugins on-demand, or the demand is to load them at startup. A good example is SideBar. You probably want this plugin to appear on the screen immediately so the customer can use it. To make this possible, you can specify which plugins you want to preload at startup in your configuration.

```
_genesys.widgets.main.preload = [  
  "sidebar"  
];
```

You may specify as many plugins as you want in the preload list. The plugins load in order after you load Widgets Core.

All plugin names are lower-case. Please refer to the file names in the **plugins/** folder. For example, to preload **webchat.min.js**, specify **webchat**, the first part of the file name.

You may find other plugins or features of plugins that necessitate preloading.

## Deploying Genesys Widgets (All-in-One)

### Files Used

- **widgets/widgets.min.css**
- **widgets/widgets.min.js**

A good starting point is the following script:

```
<script src="widgets/widgets.config.js"></script>  
<script src="widgets/widgets.min.js"></script>  
<link id="genesys-widgets-styles" href="http://www.yourhost.com/path/to/widgets.min.css"  
type="text/css" rel="stylesheet"/>
```

First, you must define your configuration for Genesys Widgets. You can do this inline on the page by using a *script* tag, or you can store it in a separate file and load it in before **widgets.min.js**. In the script example above, we assume your configuration is stored in another file. You must create the **widgets.config.js** file for this script to function properly.

### Important

Whichever method you choose, you must ensure your configuration is in the page before you load **widgets.min.js**. Otherwise, **widgets.min.js** cannot read the configuration.

### Important

Refer to this article for help [configuring Genesys Widgets](#).

## Alternative Deployment Script

To simplify the deployment process while using tools like Google Tag Manager, you can use below script to embed Widgets.

```
<script>
  var widgetBaseUrl = 'https://apps.mypurecloud.ie/widgets/9.0/',
      widgetScriptElement = document.createElement('script');

  widgetScriptElement.setAttribute('src', widgetBaseUrl + 'cxbus.min.js');

  widgetScriptElement.addEventListener('load', function () {

    CXBus.configure({debug: false, pluginsPath: widgetBaseUrl + 'plugins/'});
    CXBus.loadPlugin('widgets-core');
  });

  document.head.appendChild(widgetScriptElement);
</script>
```

## Releases hosted on Content Delivery Network (CDN)

### Warning

As announced in version [9.0.010.02](#), customers must migrate to new CDN URLs, as the legacy URLs at [app.genesys.cloud](#) are now disabled, and will not be kept up to date with any later versions.

Genesys Widgets are now available over CDN, providing optimized load times and instant access to new releases.

<https://apps.mypurecloud.com/widgets/<version>/<filename>>

Note that <version> and <filename> are placeholders.

### <filename>

This value varies based on the deployment method you choose. Please refer to [deployment section](#) for more details.

### Tip

In the case where a CDN URL that you are trying to access is **not found**, it means that either the release or the file you are looking for, is not yet available.

### <version>

version can take one of the following 3 values.

- 9.0 - (Major) - A version that is company-wide or
- 9.0.xxx - (Major).(Minor) - Minor version is product specific and is tied to each Widget's iteration or
- 9.0.xxx.xx - (Major).(Minor).(Release candidate) - Specific release version

For all the available released versions, refer to the [Genesys Widgets Releases Notes](#).

### Choose Region

Widgets is available in a number of regions worldwide as shown below. Choose the nearest or appropriate region URL based on where you are located.

Region	URL
North America (East)	https://apps.mypurecloud.com/widgets/{version}/{path/to/file}
North America (West)	https://apps.usw2.pure.cloud/widgets/{version}/{path/to/file}
North America (Canada)	https://apps.cac1.pure.cloud/widgets/{version}/{path/to/file}
Australia or New Zealand	https://apps.mypurecloud.com.au/widgets/{version}/{path/to/file}
EU (Ireland)	https://apps.mypurecloud.ie/widgets/{version}/{path/to/file}
EU (Frankfurt)	https://apps.mypurecloud.de/widgets/{version}/{path/to/file}
UK (London)	https://apps.euw2.pure.cloud/widgets/{version}/{path/to/file}
Japan	https://apps.mypurecloud.jp/widgets/{version}/{path/to/file}
Mumbai	https://apps.aps1.pure.cloud/widgets/{version}/{path/to/file}
Seoul	https://apps.apne2.pure.cloud/widgets/{version}/{path/to/file}

## Using Genesys Widgets CDN with versions

Starting in the 9.0.006.02 release, all the released versions are accessible from the Genesys CDN URL. The sections below explain how to access the latest available released version or a specific released version using Genesys CDN.

To get the latest released version under the 9.0 family:

<https://apps.mypurecloud.com/widgets/9.0/widgets.min.js>

To get the last available released version under a specific (Major).(Minor) version (this also includes any hot fixes for that release):

<https://apps.mypurecloud.com/widgets/9.0.xxx/widgets.min.js>

Example: <https://apps.mypurecloud.com/widgets/9.0.006/widgets.min.js>

To get a specific release/hot-fix version:

<https://apps.mypurecloud.com/widgets/9.0.xxx.xx/widgets.min.js>

Example: <https://apps.mypurecloud.com/widgets/9.0.006.02/widgets.min.js>

### Important

Note that all older versions of Genesys Widgets may not be available in the CDN. All the released versions are available only starting with the 9.0.006.02 version.

## Versioning examples with scenarios

When a new release version comes out, it is available under all the 3 different CDN URLs below. In this example, if 9.0.006.01 is the first ever release announced, then it is available under the following CDN URLs.

- /9.0/
- /9.0.006/
- /9.0.006.01/

When 9.0.007.04 is released, it is available under /9.0/ but **not** under /9.0.006/ or /9.0.006.01/. Instead, /9.0.007/ and /9.0.007.04/ CDN URLs are created and this release is available under them:

- /9.0/
- /9.0.007/
- /9.0.007.04/

If a hot fix (such as 9.0.006.02) is released after 9.0.007.04 is released, then the hot fix is available under the following CDN URLs:

- /9.0.006/
- /9.0.006.02/

If a hot fix (such as 9.0.007.05) is released before announcing any new release, then it is available under the following CDN URLs:

- /9.0/
- /9.0.007/
- /9.0.007.05/

## Deployment Methods

### Lazy loading

**Recommended approach:** When using the lazy loading method, the base Genesys CDN URL must be prefixed in the [lazy loading deployment script](#). The <filenames> value is not needed in this scenario because they are auto-loaded from the base CDN configured. Here is what the deployment script looks like when using 9.0.006.02 release:

```
<script src="https://apps.mypurecloud.com/widgets/9.0.006.02/cxbus.min.js"
onload="javascript:CXBus.configure({debug:true,pluginsPath:'https://apps.mypurecloud.com/
widgets/9.0.006.02/plugins/'});CXBus.loadFile('path/to/
widgets.config.js').done(function(){CXBus.loadPlugin('widgets-core')});"></script>
```

### All-in-One

**Legacy approach (deprecated):** When using the all-in-one deployment method, the values are the files mentioned in the [All-In-One](#) section. For example, if you would like to use widgets.min.js and widgets.min.css under 9.0.006.02 release, CDN URLs will look like this:

```
https://apps.mypurecloud.com/widgets/9.0.006.02/widgets.min.js
https://apps.mypurecloud.com/widgets/9.0.006.02/widgets.min.css
```

### Important

Lazy loading is the recommended method for Widgets. The [\(All-In-One\)](#) method is deprecated.

## Checking Widgets Version

At any point of time after deploying Widgets, the version used can be easily determined using one of the below scripts.

```
CXBus.command("App.info");
```

Prints out the debug header with version info.

```
window._genesys.widgets.common.data("version");
```

Returns the version number directly, as a string.

## Download a version

### Important

As of release 9.0.017.01, Genesys Widgets is no longer available to download through the Genesys Downloads Center. Instead, it can be downloaded as a zip file from the Content Delivery Network (CDN).

You can download the latest released version under the 9.0 family here:

<https://apps.mypurecloud.com/widgets/9.0/dist.zip>.

To download a specific version, simply change the version in the above URL to the desired version. The version here takes the same pattern as described above in the **Versioning examples with scenarios** section.

## Genesys Web Fonts

Google Fonts are now hosted in Genesys Infrastructure. Please Choose the nearest or appropriate region URL based on where you are located and configure it through the [googleFontUrl option](#).

### Important

By default, Genesys web fonts are loaded from the North America (East) region.

Region	URL
North America (East)	<a href="https://apps.mypurecloud.com/webfonts/roboto.css">https://apps.mypurecloud.com/webfonts/roboto.css</a>
North America (West)	<a href="https://apps.usw2.pure.cloud/webfonts/roboto.css">https://apps.usw2.pure.cloud/webfonts/roboto.css</a>
North America (Canada)	<a href="https://apps.cac1.pure.cloud/webfonts/roboto.css">https://apps.cac1.pure.cloud/webfonts/roboto.css</a>

Region	URL
Australia or New Zealand	<a href="https://apps.mypurecloud.com.au/webfonts/roboto.css">https://apps.mypurecloud.com.au/webfonts/roboto.css</a>
EU (Ireland)	<a href="https://apps.mypurecloud.ie/webfonts/roboto.css">https://apps.mypurecloud.ie/webfonts/roboto.css</a>
EU (Frankfurt)	<a href="https://apps.mypurecloud.de/webfonts/roboto.css">https://apps.mypurecloud.de/webfonts/roboto.css</a>
UK (London)	<a href="https://apps.euw2.pure.cloud/webfonts/roboto.css">https://apps.euw2.pure.cloud/webfonts/roboto.css</a>
Japan	<a href="https://apps.mypurecloud.jp/webfonts/roboto.css">https://apps.mypurecloud.jp/webfonts/roboto.css</a>
Mumbai	<a href="https://apps.aps1.pure.cloud/webfonts/roboto.css">https://apps.aps1.pure.cloud/webfonts/roboto.css</a>
Seoul	<a href="https://apps.apne2.pure.cloud/webfonts/roboto.css">https://apps.apne2.pure.cloud/webfonts/roboto.css</a>

# Tested Browsers

The following is a list of all Genesys-tested browsers for both **desktop** and **mobile**.

## Important

Support for the browsers listed below will only be available for as long as Genesys can properly reproduce the issue.  
Please let Genesys know of any issues you encounter with any of our tested browsers.

## Desktop Browsers

### Windows

- Google Chrome — Current release or one version previous
- Microsoft Edge — Current release or one version previous
- Mozilla Firefox — Current release or one version previous

### Mac OS

- Google Chrome — Current release or one version previous
- Microsoft Edge — Current release or one version previous
- Mozilla Firefox — Current release or one version previous
- Safari — Current release or one version previous

## Mobile Browsers

- Google Chrome — Current release or one version previous
- Safari — Current release or one version previous

---

# Cookies

## Purpose

Genesys Widgets uses cookies to store non-sensitive data in the browser. The end-user's browser must allow cookies for Genesys Widgets to operate properly. Each cookie is required, and without the ability to read and write these cookies, Genesys Widgets features will not function properly.

## Cookie creation

All cookies start with the prefix "\_genesys.widgets" to easily identify them. By default, Genesys Widgets cookies are created in a way that allows the cookies to be read across sub-domains by setting the "domain" attribute in the cookie options. We derive the proper domain value by parsing the host site's domain and extracting it.

### Important

Genesys Widgets never stores Personally Identifiable Information (PII) in its cookies.

## Duration

All cookies used by Genesys Widgets are created as session cookies and will be deleted when the user's browser is fully closed.

## Sub-domains

Normally, cookies cannot be transferred between sub-domains of a website unless they are configured to do so. Genesys Widgets automatically detects the domain of the host site and configures all cookies to be transferable between sub-domains. For example, you could start a chat on [www.testsite.com](http://www.testsite.com) and restore that chat session on [store.testsite.com](http://store.testsite.com), [support.testsite.com](http://support.testsite.com), or [portal.testsite.com](http://portal.testsite.com).

## Cookie support in test environments

Genesys Widgets uses special cookies that persist across sub-domains. This is a critical feature for

plugins like WebChat that need to restore an active chat session while navigating around a website. The side effect of using this type of cookie is they won't work when using test environment domain names such as "localhost" or an IP address. You must use a fully-qualified domain name (FQDN) such as "localhost.com" or any other variant that can be identified as a domain name. Cookies will also fail to work if you run the test site as an HTML file path directly in the browser.

One workaround is to update your system's hosts file to create an FQDN alias for "localhost", your test environment's name, or an IP address.

## Example

```
127.0.0.1      localhost
127.0.0.1      localhost.com
```

A fully-qualified domain name (FQDN) such as "localhost.com" or any other variant that can be identified as a domain name is not mandatory, but it is recommended. This way, the cookies will also work when using test environment domain names such as "localhost" or an IP address.

The following is a list of cookies used by Genesys Widgets.

## App

Cookie Name	Purpose
_genesys.widgets.app.autoLoadList	Contains a list of active plugin names that are updated based on the usage of Widgets during the <b>Lazy loading</b> deployment method. This is to ensure that a Widget is auto-loaded during a page refresh or page navigation when there is an active session associated with it.

## ClickToCall

Cookie Name	Purpose
_genesys.widgets.clicktocall.state.open	Contains the Widget open or close state based on the active session, if any.
_genesys.widgets.clicktocall.state.expire	A session cookie containing the expiring time of the current active session. It is automatically deleted when the session expires or the visitor closes his or her browser.
_genesys.widgets.clicktocall.state.phone_number	A session cookie containing the phone number generated for that active session. It is automatically deleted when the session expires or the visitor closes his or her browser.

Cookie Name	Purpose
_genesys.widgets.clicktocall.state.access_code	A session cookie containing the phone access code related to the active session. It is automatically deleted when the session expires or the visitor closes his or her browser.
_genesys.widgets.clicktocall.state.phone_tel_href	A session cookie containing the complete clickable phone number generated for that active session. It is automatically deleted when the session expires or the visitor closes his or her browser.

## Console

Cookie Name	Purpose
_genesys.widgets.console.session	A cookie containing the active Console plugin open/close state.
_genesys.widgets.console.commandPlugin	A cookie containing the selected plugin name from the Commands section.
_genesys.widgets.console.command	A cookie containing the selected command to run from the Commands section.
_genesys.widgets.console.eventPlugin	A cookie containing the selected plugin from the Events section to listen for any events.
_genesys.widgets.console.event	A cookie containing the selected event type to listen against, from the Events section.
_genesys.widgets.console.optionsArea	A cookie containing the command options to send when executing a command.
_genesys.widgets.console.activeSubscriptions	A cookie containing the list of all active event subscriptions listening via Console plugin.
_genesys.widgets.console.windowPosition	A cookie containing the position of the Console plugin on the screen.

## KnowledgeCenter

Cookie Name	Purpose
_genesys.widgets.knowledgecenter.state.session	A session cookie containing the session ID related to the current active knowledge search center.
_genesys.widgets.knowledgecenter.state.keys	A session cookie containing some encrypted session data related to the current active knowledge search center.

## WebChat

### Important

An active WebChat session cookie is automatically deleted whenever the chat session is ended/expired or the visitor closes his or her browser.

Cookie Name	Purpose
<code>_genesys.widgets.webchat.state.open</code>	A cookie containing the WebChat Widget open or close state for internal tracking purposes.
<code>_genesys.widgets.webchat.state.keys</code>	A session cookie that can contain some encrypted keys related to the current active chat session.
<code>_genesys.widgets.webchat.state.ping</code>	A session cookie containing the time at which the last successful request was made to the server.
<code>_genesys.widgets.webchat.metaData</code>	A session cookie containing all the <b>Metadata</b> details related to the current active chat session.
<code>_genesys.widgets.webchat.state.index</code>	A session cookie containing the last unique Message ID for internal tracking purposes.
<code>_genesys.widgets.webchat.state.filters</code>	A session cookie containing any pre-filters that were added using WebChatService plugin commands <b>addPrefilter</b> or <b>sendFilteredMessage</b> .
<code>_genesys.widgets.webchat.state.session</code>	A session cookie containing the unique Session ID related to the current active chat session. It is used to restore the active chat session during scenarios like page refresh or page navigation.
<code>_genesys.widgets.webchat.state.minimized</code>	A cookie containing the WebChat Widget minimized or maximized state for internal tracking purposes.
<code>_genesys.widgets.webchat.autoInvite.disabled</code>	Contains a value whether to disable or enable WebChat autoInvite feature. It is dynamically updated based on the user's response to the initial WebChat invite.
<code>_genesys.widgets.webchat.state.unreadMessages</code>	A session cookie that keeps track of the number of unread messages during an active chat session, when WebChat is minimized. It is cleared whenever the WebChat Widget is maximized by the user to read the new messages.
<code>_genesys.widgets.webchat.state.lastMessageCountRead</code>	A session cookie containing the number of messages that are read during an active chat session that calculates the number of unread messages when WebChat is minimized. It is automatically cleared whenever WebChat Widget is maximized or closed/ended.
<code>_genesys.widgets.webchat.state.asyncUnreadMessageCount</code>	A session cookie that keeps track of the number of unread messages related to an Async Chat, when WebChat is minimized. It is cleared whenever the WebChat Widget is maximized by the user to read

Cookie Name	Purpose
	the new messages.
<code>_genesys.widgets.webchat.state.pureengage-v3-rest.session</code>	A session cookie applicable only when using Genesys Engage V3 API. It contains the Session ID related to the current active chat session. It is used to restore the active chat session during scenarios like page refresh or page navigation.
<code>_genesys.widgets.webchat.state.pureengage-v3-rest.keys</code>	A session cookie applicable only when using Genesys Engage V3 API, containing the encrypted keys related to the current active chat session.
<code>_genesys.widgets.webchat.state.pureengage-v3-rest.index</code>	A session cookie applicable only when using Genesys Engage V3 API, containing the last unique message ID for internal tracking purposes.
<code>_genesys.widgets.webchat.state.pureengage-v3-rest.open</code>	A cookie applicable only when using Genesys Engage V3 API, containing the WebChat Widget open or close state for internal tracking purposes.
<code>_genesys.widgets.webchat.state.purecloud-v2-sockets.JWTtoken</code>	A session cookie applicable only when using Genesys Cloud V2 API, containing the JWT token related to the current active chat session.
<code>_genesys.widgets.webchat.state.purecloud-v2-sockets.ConversationID</code>	A session cookie applicable only when using Genesys Cloud V2 API, containing the active conversation ID related to the current chat session.
<code>_genesys.widgets.webchat.state.purecloud-v2-sockets.MemberID</code>	A session cookie applicable only when using Genesys Cloud V2 API, containing the user ID of the WebChat Widget related to the current active chat session.
<code>_genesys.widgets.webchat.state.purecloud-v2-sockets.WS_URL</code>	A session cookie applicable only when using Genesys Cloud V2 API, containing the WebSocket event stream URI for listening to new incoming messages.
<code>_genesys.widgets.webchat.state.purecloud-v2-sockets.LastMsgId</code>	A session cookie applicable only when using Genesys Cloud V2 API, containing the last unique ID of the message sent in the WebChat Widget.

## Local Storage

Genesys Widgets uses local storage to store non-sensitive data in the the browser with no expiration date.

Key Name	Purpose
<code>_genesys.widgets.inFocus</code>	A globally unique identifier (GUID) to identify the current active chat session browser tab/window, when the WebChat Widget opens in multiple browser tabs/windows.

## Required Versions of Genesys Components

The following is a list of minimum versions required to achieve basic end-to-end functionality:

Component	Version
Chat Server	8.5.109.06
Genesys Co-browse Server	8.5.002.00
Genesys Mobile Services	8.5.106.14
Genesys Web Engagement Server	8.5.000.15
Genesys Knowledge Center Server	8.5.200.11

The following is a list of feature-dependent minimum versions:

Feature	Component	Version
<b>CometD</b>	Chat Server	8.5.109.06
	Genesys Mobile Services	8.5.109.05
<b>EWT v2</b>	Genesys Mobile Services	8.5.200.07
<b>Rich Messaging</b>	Bot Gateway Server	9.0.004.08
	Chat Server	8.5.109.06
	Genesys Mobile Services	8.5.201.04
<b>Typing Preview</b>	Workspace Desktop Edition	8.5.108.11

# Configuring Genesys Widgets

Genesys Widgets supports multiple widgets including Webchat, **Web Engagement**, and **Co-browse**. You can configure all widgets and services in the same configuration object. When you add new Genesys products and services, you must update your Genesys Widgets configuration to enable those widgets.

After you **deploy the Genesys Widgets** on your website, configure the CX Widget by defining the **global window.\_genesys** JavaScript object.

To include the JavaScript script, you can choose one of the following options:

- Place the script inline on your website; or
- Place it in a separate JavaScript file, and then include the file on your page.

The following example is a basic view of the global Genesys Widgets configuration object:

```
<script>
  if(!window._genesys)window._genesys = {};
  if(!window._gt)window._gt = [];

  window._genesys.widgets = {
    main: {},
    webchat: {},
    cobrowse: {},
    gwe: {}
  };
</script>

// include widgets.min.js after defining your configuration options
```

The following example is a populated Widget configuration that includes configuration options for Webchat, SendMessage, Web Engagement (GWE), and Co-browse (GCB):

## Important

Your configuration options must be defined on the page **before** widgets.min.js is loaded. Failing to do so may result in errors.

## Tip

Note that Web Engagement and Co-browse is optional.

```
<script>
  if(!window._genesys>window._genesys = {});
  if(!window._gt>window._gt = []);

  window._genesys.widgets = {
    main: {

      theme: "dark",
      lang: "en",
      i18n: "http://HOST:PORT/path/to/languages/file.json",
      customStylesheetID: "<ANY_ID_NAME>",
      mobileMode: 'auto',
      mobileModeBreakpoint: 600
    },
    webchat: {

      apikey: "0123456789", // Used for Apigee service only
      dataURL: "http://HOST:PORT/path/to/chat/service",
      userData: {},
      proactive: {

        enabled: true,
        idleTimer: 5,
        cancelTimer: 30
      },
      chatButton: {

        enabled: true,
        template: false,
        openDelay: 1000,
        effectDuration: 300,
        hideDuringInvite: true
      }
    },
    cobrowse: {

      src: "<COBROWSE_SERVER_URL>/cobrowse/js/gcb.min.js",
      url: "<COBROWSE_SERVER_URL>/cobrowse"
    },
    gwe: {
      httpEndpoint: "http://<GWE_SERVER_URL>:<PORT>",

```

```
        httpsEndpoint: "https://<GWE_SERVER_URL>:<PORT>",
        dslResource: "http://HOST:PORT/path/to/domain-model.xml"
    },
    sendmessage: {
        SendMessageButton : {
            enabled: true,
            template: false,
            openDelay: 1000,
            effectDuration: 300
        },
        apikey: "0123456789", // Used for Apigee service only
        dataURL: "http://HOST:PORT/path/to/sendmessage/service"
    },
};
</script>

// Lazy loading method of loading widgets
<script src="/path/to/cxbus.min.js" onload="javascript:CXBus.configure({debug:false, pluginsPath:'/path/to/plugins/'});
CXBus.loadPlugin('widgets-core');"></script>

// All-in-one method of loading widgets
<script id="genesys-cx-widget-script" src="/path/to/widgets.min.js"></script>
<link id="genesys-widgets-styles" href="/path/to/widgets.min.css" type="text/css" rel="stylesheet"/>
```

## Main Configuration

Genesys Widgets is a hub for multiple Genesys products and services. Some configuration options are set globally and therefore apply to all products and services running on the CX Widget platform. In the main application configuration you can configure options such as visual theme, language, and mobile support.

For detailed information on configuration options, see [App Configuration Options](#).

## Widget Configuration Options

- [WebChat Configuration](#)
- [WebChatService Configuration](#)
- [SendMessage Configuration](#)
- [SendMessageService Configuration](#)
- [CoBrowse Configuration](#)
- [GWE Configuration](#)
- [Channel Selector Configuration](#)

For a complete list of Widget configuration options, please see [Genesys Widgets Reference](#).

## Launcher

Launcher is a sample page which shows how Genesys Widgets are displayed on any host website. With this, you can:

- View Genesys Widgets with your own configuration.
- Copy the Configuration Script; for example, using the details you entered on the form, configuration script is generated in the **Need Configuration Script** section. You can copy this script and use it in your website to launch Widgets.
- Starting in the 9.0.008.03 version, use launcher tool to test and configure between the different APIs services available in Genesys, namely under Genesys Engage, Genesys Engage cloud and Genesys Cloud (formerly known as PureEngage, PureEnage Cloud, and PureCloud).

## How to use Launcher

### Sidebar

To enable Sidebar plugin, select this check box in launcher. By default, Sidebar will be shown right side of the screen on Widgets startup. You can change these [configurations](#) by using the options

---

---

shown under this section. Provide the sidebar channel configuration in the corresponding text area according to the [Sidebar documentation](#) or use the provided sample configuration links to prefill the sample data in the text area. Ensure that the channels defined here are enabled and configured as well.

### Enable Live Assist (EWT)

Select the check box next to **Enable Live Assist (EWT)** to enable the ChannelSelector plugin. Enter the Stats URL followed by the virtual queue names to fetch the Estimated Wait Time details. Refer to the [Channel Selector documentation](#) for more information. To show Chat, SendMessage and CallUs channels in this plugin, please make sure that you select these plugins in this Launcher page.

### WebChat

To enable it, select the check box next to the **WebChat**. Enter the mandatory field URL (marked with an asterisk). Other values are optional and self explanatory.

### Important

Starting in the 9.0.008.03 version, WebChat supports Genesys Engage v3 API via transports configuration section. The same can be tested In the launcher tool using the WebChat [with Transport only] section under Genesys Engage cloud tab.

### Co-browse

To enable it, select the check box next to the **Co-browse** heading. Enter required fields (marked with an asterisk).

### GWE

To enable it, select the check box next to the **GWE** heading. Enter required fields (marked with an asterisk).

### Send Message

To enable it, select the check box next to the **Send Message** heading. Enter required fields (marked with an asterisk).

### CallUs

Select the check box next to **Call Us** and provide the configuration data. Here, the **Edit/Use Sample Config** option is also provided and you can use this to prefill with the sample configuration and edit it for your own details. Ideally, call us is shown in the Live Assist widget or it can also be launched with bus command `CallUs.open`.

### Callback

Select the check box next to **Callback** to include Callback and Calendar plugins. Enter the callback service provider URL field and other details as required. Ensure **Enable Sidebar with Live Assist** is selected.

### Knowledge Center/Search

Select the check box next to **Knowledge Center/Search** to include the Knowledge Center plugin. Ensure that all the required details are entered and the sidebar plugin is enabled.

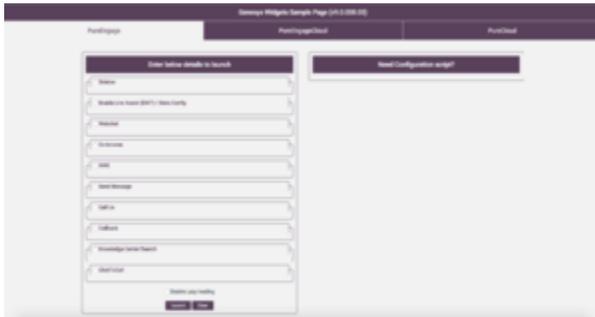
### ClickToCall

Select this checkbox to enable ClickToCall plugin and enter the mandatory field URL (marked with an asterisk). Other values are optional and self explanatory.

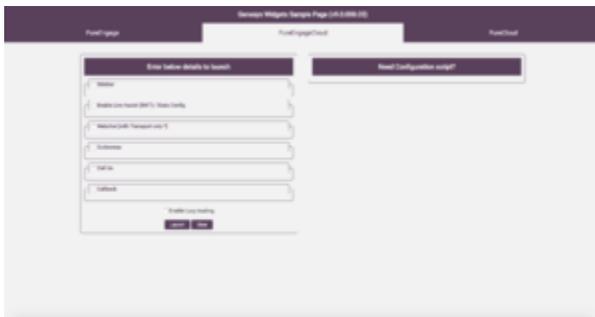
### Lazy Loading

Select the check box next to **Enable Lazy loading** to launch Widgets in the lazy load mode, otherwise they will be launched normally. At the minimum, Sidebar plugin show be enabled and configured with required channels to load it on Widgets startup.

Once all the necessary configuration details are entered, click on the **Launch** button to launch Widgets.



Launcher tool for Genesys Engage API (click to enlarge)



Launcher tool for Genesys Engage cloud API (click to enlarge)

---

# Localization

Genesys Widgets allow for localization of user messages and prompts. First, you must create and host a Language Pack that Genesys Widgets can access and use. The Language Pack is a file written in JSON format. Specify your Language Pack file by using Genesys Widgets configuration options, which you can configure in the **window\_genesys.widgets.main** section.

Example:

```
<script>

  if(!window._genesys)window._genesys = {};
  if(!window._gt)window._gt = [];

  window._genesys.widgets = {
    main: {
      theme: "dark",
      lang: "en",

      // Enter a URL that points to
      i18n: "http://HOST:PORT/path/to/lanaguages/file.json"

      // OR define the JSON object inline
      i18n: {
        "en": {
          "webchat": {
            "ChatStarted": "Chat Started",
            "ChatEnded": "Chat Ended",
            ...
          },
          "sendmessage": {
            "EmailFormFirstname": "First Name",
            "EmailFormLastname": "Last Name",
            ...
          }
        }
      }
    }
  };
</script>
```

## Master Localization File

[This URL](#) provides the latest i18n localization content containing all the language codes and strings of all Widgets. This acts as a centralized master file that you can use as a reference to create your own modified localization file and host it. In this way, you can use this to override the language content.

## Important

The English language pack file provided in the above URL is just for reference. Do not load this file into Widgets because it is already built into Widgets by default.

## Multiple Translated Language Packs

Beginning in version **9.0.014.05**, multiple i18n language pack files are available as individual JSON files in the `/i18n` folder. You can select the desired language pack file and then set the `i18n` and `lang` properties in the **`window._genesys.widgets.main`** configuration section. Each language pack file is named using the language code to identify easily. The same language code is also used inside the language pack file to construct the i18n JSON. This language code must be specified in the **`main.lang`** configuration option.

Example:

The French language pack file is available as **`widgets-fr.i18n.json`**. To use this language pack file, follow the example below.

```
window._genesys.widgets = {
  main: {
    lang: "fr",
    i18n: "/relative/path/to/i18n/widgets-fr.i18n.json"
    // OR using the CDN URL
    i18n: "https://apps.mypurecloud.com/widgets/<version>/i18n/widgets-fr.i18n.json"
  }
};
```

### Language code mapping

Language	Code
Brazilian Portuguese	pt-BR
Chinese Simplified	zh-CN
Chinese Traditional	zh-TW
Danish	da
Dutch	nl
English	en
Finnish	fi
French	fr
German	de
Italian	it
Japanese	ja

---

Language	Code
Korean	ko
Norwegian	no
Polish	pl
Spanish	es
Swedish	sv
Thai	th
Turkish	tr

## Configuration Options

### **main.lang**

**Type:** string

**Default:** "en"

**Requirement:** Optional

**Description:** A language code to specify which language to display in the Widgets. Language codes are set by the customer.

### **main.i18n (external file)**

**Type:** string

**Default:** built-in English words and phrases

**Requirement:** Required when using main.lang option.

**Description:** A URL that the Widgets use to fetch the Language Pack file upon startup. Can be partial or complete. Unspecified strings will use default values.

### **main.i18n (inline object)**

**Type:** object

**Default:** built-in English words and phrases

**Requirement:** Required when using main.lang option.

**Description:** An inline JSON object. Can be partial or complete. Unspecified strings will use default values.

## Language Pack JSON Format

The Language Pack is written in JSON format.

```
// Root
{
  // Language Code
  "en": {
    // Widget name
    "webchat": {
      // Localized strings
      "ChatStarted": "Chat Started",
      "ChatEnded": "Chat Ended",
```

```

    "ChatFailed": "There was a problem starting the chat session. Please Retry.",
    // Customer Defined Strings - Match & Replace messages received from chat server
    "SYS0001": "An Agent will be with you shortly"
  },
  "sendmessage": {
    // Localized strings
    "SendMessageButton": "Send Message",
    "EmailFormFirstname": "First Name",
    "EmailFormLastname": "Last Name",

    //Errors
    "ErrorServerNotAvailable": "Unable to reach server. Please try again.",
    "ErrorAttachfileSizeMax": "Total size of attachments exceeds limit: "
  }
}

```

## Localization Namespaces

Plugin	Namespace
Calendar	calendar
CallBack	callback
CallUs	callus
ChannelSelector	channelselector
ClickToCall	clicktocall
KnowledgeCenter	knowledgecenter
Offers	offers
SendMessage	sendmessage
SideBar	sidebar
WebChat	webchat

## Language Codes

To allow flexibility in the way that your website currently handles multiple languages and language codes, there are no rules for language codes other than that they must be strings. You can use any language code system. The language code that you set in **window.\_genesys.widgets.main.lang** must correlate to a language code in the Language Pack File.

### Important

Beginning in version **9.0.014.05**, when using one of the available pre-translated language packs, ensure the language code maps with the one included in the

---

language pack file.

## Plugin Localization Options

- [WebChat Localization](#)
- [WebChatService Localization](#)
- [SendMessage Localization](#)
- [SendMessageService Localization](#)
- [CoBrowse Localization](#)
- [GWE Localization](#)
- [Callback Localization](#)
- [CallUS Localization](#)
- [ChatDeflection Localization](#)
- [Search Localization](#)
- [SideBar Localization](#)
- [ClickToCall Localization](#)
- [Offers Localization](#)

# Styling the Widgets

## Themes

You can change the appearance of Genesys Widgets using *themes*. Themes allow you to change colors and fonts for all widgets.

Genesys Widgets includes two built-in themes, "dark" and "light". The "dark" theme is active by default.

### Dark Theme

The image displays two screenshots of Genesys Widgets forms in the dark theme. The left screenshot shows a 'Live Chat' form with fields for First Name, Last Name, Email, and Subject, and buttons for 'Cancel' and 'Start Chat'. The right screenshot shows a 'Send Message' form with fields for First Name, Last Name, Email, and Subject, a message input field, and buttons for 'Attach files' and 'Send'.

### Light Theme

## How do I set the active theme?

There are two methods for setting the active theme:

### Configuration

```
window._genesys.widgets.main.theme = "light"; // or "dark"
```

### Widget Bus Command

```
window._genesys.widgets.bus.command("App.setTheme", {theme: "light"}); // or "dark"
```

## How do I create my own themes?

### Theme Templates

Genesys Widgets uses special LESS files called "Theme Templates" to define themes. Using this Theme Template, you can modify the color palette and add new styles. Everything is laid out clearly in the template file.

LESS syntax is used because we can define local variables that allow us to create a clear color

palette. The LESS file color palette consists of no less than 28 separate color variables. These are grouped by their usage:

- Background Colors
- Text Colors
- Icon Colors
- Border Colors
- Outline Colors

At a bare minimum, you can create a new style by simply changing the color values in the color palette. You may add or remove colors from this palette as you see fit.

### Color Palette Example

```
/* Color Palette */
@bg_color_1: #33383D; // Main Background Color
@bg_color_2: #444A52; // Form Inputs
@bg_color_3: #222529; // Button default
@bg_color_4: #5081E1; // Button primary gradient 1
@bg_color_5: #4375D6; // Button primary gradient 2
@bg_color_6: #CCCCCC; // Button disabled
@bg_color_7: #212529; // Native scrollbar track color
@bg_color_8: #A3A8AE; // Scrollbar color

@txt_color_1: #FDFDFD; // Main text color
@txt_color_2: #98A7B8; // footer text
@txt_color_3: #FDFDFD; // Button default & primary / autocomplete text hover
color
@txt_color_4: #FDFDFD; // Hyperlink color
@txt_color_5: #C5CCD6; // Placeholder color
@txt_color_6: #F53131; // Alert/error color

@icon_color_1: #FDFDFD; // Base icon color
@icon_color_2: #8C8C8C; // Secondary icon color (multitone only)
@icon_color_3: #000000; // Icon shadow color (multitone only)
@icon_color_4: #000000; // Icon secondary shadow color (multitone only)
@icon_color_5: #98A7B8; // Window control icon color
@icon_color_6: #98A7B8; // Form input icon overlay color (e.g. "clear" icon)
@icon_color_7: #5081E1; // Interactive icon color 1 (attach files, delete
file, etc)
@icon_color_8: #4AC764; // Positive Color (confirmation, availability,
usually green)
@icon_color_9: #F53131; // Negative Color (error, exception, usually red)
@icon_color_10: #F8A740; // Warning Color (warning, pending, offline, usually yellow
or orange)
@icon_color_11: #FDFDFD; // Icon color for primary buttons

@border_color_1: #222529; // Main border color
@border_color_2: #2E69DB; // Button primary
@border_color_3: transparent; // Button default
@border_color_4: transparent; // Button disabled
@border_color_5: #F53131; // Alert/error color
@border_color_6: #758384; // Form controls default state

@outline_color_1: #75A8FF; // Form input focus outline / autocomplete hover
background color
@outline_color_2: #DAE6FC; // Outline color for primary buttons
```

## Example Theme Template Files

**Note:** Clicking the example template files automatically downloads them to your computer.

[theme-template-dark.less](#)  
[theme-template-light.less](#)

Theme Templates are LESS files, which must be converted to CSS before being used on a website. Use a website or tool to convert them when you're ready to test and implement them on your site.

By default, the Theme Template will override styles for all Genesys Widgets generally, but you may add more specific changes that affect only a specific widget. More information on this is provided later on this page.

## Naming Your Theme

In the "dark" theme template file, the first class selector is defined as:

### **.cx-widget.cx-theme-dark**

**.cx-widget** is the base class for all Genesys Widget UI. The outermost container of every widget or standalone UI element has this class and is used to identify UI elements that belong to Genesys Widgets.

**.cx-theme-dark** is the class name created for the "dark" theme. Themes are applied by searching for all elements with the **.cx-widget** class and appending the theme's classname to it. Thus, the combined class selector indicates styles that will be applied only when your custom theme is active in the configuration object.

You may name your theme classname anything you wish. There are no restrictions or limitations.

In a later step, you will register this theme classname in your configuration.

## Customization Guidelines

When creating your own themes, they are restricted to the following CSS properties:

- color
- background
- font-family
- font-style
- border-color
- border-style
- and other non-structural properties

### Warning

Widgets primarily relies on class names for CSS selectors, rather than fixed node path selectors. Using class names allows for the HTML structure to be changed without breaking selectors. For example, the selector ".cx-webchat .cx-message" is all that's needed to target message bubbles inside WebChat. Using a fixed node path equivalent, like "div.cx-webchat > div.cx-body > div.cx-transcript > div.cx-message-group > div.cx-message" creates a dependency on the HTML node type and structure. If any changes are made to WebChat's HTML structure, this CSS selector will break. Use the smallest necessary specificity in your selectors and try to use class names only.

### Warning

Exercise caution when you make changes to the CSS to make sure structure and functionality are retained.

### Warning

Avoid setting CSS properties that change height, width, thickness, size, visibility, or other properties that change the structure of widgets. These properties are not supported and changing them could break widget stability and usability.

### Important

By default, the Widgets CSS refers to the Roboto font, available at <https://fonts.google.com/>

## How do I register my themes with Genesys Widgets?

You can register themes in the Genesys Widgets configuration.

```
window._genesys.widgets.main.themes = {  
  "blue": "cx-theme-blue"  
};
```

The name:value pair used here consists of a key ("blue") and the theme's CSS classname ("cx-theme-blue").

You can add as many themes to this list as you need and select the active theme using one of the key values.

```
window._genesys.widgets.main.theme = "blue";  
// OR  
window._genesys.widgets.bus.command("App.setTheme", {theme: "blue"});
```

## How do I change styles for a specific widget?

It is possible to specify specific widgets and even specific elements within a widget by appending the widget's CSS classname to the theme classname.

In the following example, we extend the "cx-theme-blue" class with a widget-specific entry that makes the WebChat widget's background color a darker shade.

```
.cx-widget.cx-theme-blue, .cx-widget .cx-container{  
    color: #FDFDFD;  
    background: #1e5799;  
}  
.cx-widget.cx-theme-blue *{  
    border-color: #7DB9E8;  
}  
.cx-widget.cx-theme-blue.cx-webchat, .cx-widget.cx-theme-blue .cx-webchat{  
    background: #225897;  
}
```

### Important

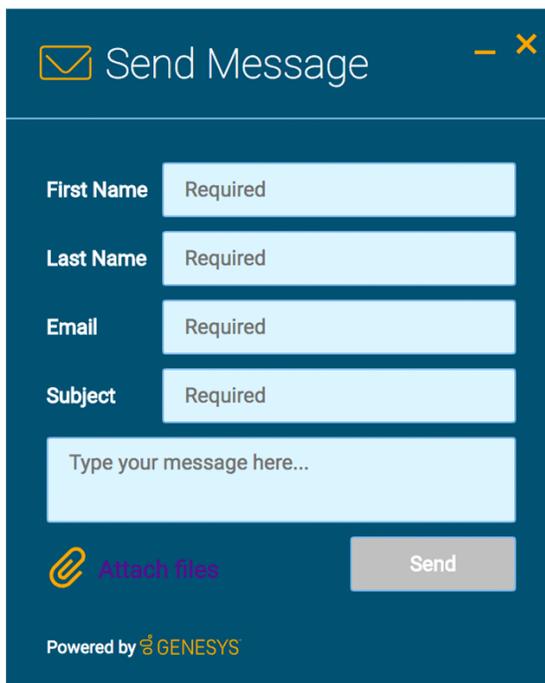
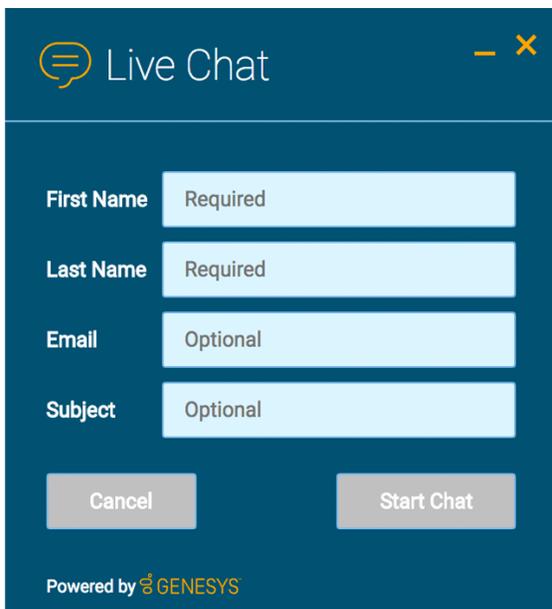
Notice the dual CSS selector used when specifying a widget. This is required to make sure your styles always apply properly.

### Widget-Specific and Element-Specific

In the next example, we further extend the "cx-theme-blue" class with a widget and element-specific entry that changes the background color of the input fields within the WebChat widget to a light shade of blue.

```
.cx-widget.cx-theme-blue, .cx-widget .cx-container{
  color: #FDFDFD;
  background: #1e5799;
}
.cx-widget.cx-theme-blue *{
  border-color: #7DB9E8;
}
.cx-widget.cx-theme-blue.cx-webchat, .cx-widget.cx-theme-blue .cx-webchat{
  background: #225897; // Darker Shade
}
```

```
.cx-widget.cx-theme-blue.cx-webchat .form input, .cx-widget.cx-theme-blue .cx-webchat .form
input{
    background: #DCF5FF; // Lighter Shade
}
```



## How do I change the layout and structure of widgets?

Genesys Widgets only support customizing a limited set of styles through themes. If you wish to use an alternate layout of your own design, you can disable the widget you want to replace and utilize the provided service plugins to build your own.

### Choosing Which Plugins to Load

Refer to the 'plugins' configuration option here: [App Configuration](#)

### Service Plugins

Service plugins provide a high-level API for quickly integrating a UI with back-end services. Each widget is matched with a corresponding service plugin. This separation allows for advanced integrations.

- [WebChatService](#)
- [SendMessageService](#)
- [CallbackService](#)

- [KnowledgeCenterService](#)

### Warning

Changing the layout of official Genesys Widgets is not supported. We cannot guarantee that your changes will remain when upgrading to a newer version of Genesys Widgets.

## How do I change fonts?

By default, Genesys Widgets downloads and uses the Google font 'Roboto'. You can change the font used in Genesys Widgets by using the following CSS:

```
.cx-widget{ font-family: name-of-font-here; }
```

Choose whichever font you wish to use and it will then apply throughout Genesys Widgets.

## Disabling Google Font Download

If you would like to prevent the Roboto font file from being downloaded at startup, you can disable the download by changing the configuration option `main.downloadGoogleFont` to `false`:

```
_genesys.widgets.main.downloadGoogleFont = false;
```

If set to "**false**", the Google font 'Roboto' will not be downloaded. If set to "**true**", the Google font 'Roboto' will be downloaded. The default value is "**true**".

### Important

Use this configuration option if you have security concerns regarding including fonts from 3rd party sources, to optimize your page load time, or if you already include Roboto on your website.

---

# Icons

Genesys Widgets utilize icons in SVG format. Using SVG allows icons to be themed with different color fills and other SVG CSS properties. Using SVG icons also provides the highest rendering quality on all devices, no matter the zoom level, resolution, or DPI of the screen. Our icons scale to fit in any container, can be used inline or as blocks, and can be animated in orientation, colors, and other styles.

Our icons are available to our customers to use in their own custom extensions. This allows custom extensions to retain the look and feel of our prebuilt widgets. Some examples might be:

- Creating a custom launcher button for chat, using the chat icon
- Creating a custom widget with your choice of icon in the title bar
- Using our icons inline, inside text copy to refer to widgets by icon

Widgets includes two different sets of icons. The Multi-tone icon set utilizes multiple layers and colors per icon. The Outline icon set offers a minimalist approach to both design and color. You may use any of the icons for your own purpose.

Currently, icons cannot be customized or replaced.

## How to use icons

### Automatic HTML Injection

By applying the CSS class "cx-icon" and the attribute "data-icon" to an element, you can specify which icons you want and where.

**Example:**

```
<div id="your-element" class="cx-icon" data-icon="chat">  
  ...SVG icon will be inserted here  
</div>
```

You then pass your element as a jquery wrapped set or HTML string into the `CXCommon.populateAllPlaceholders( $("#your-element") )` function, Widgets will insert the appropriate SVG icons into each specified element and return the HTML to you.

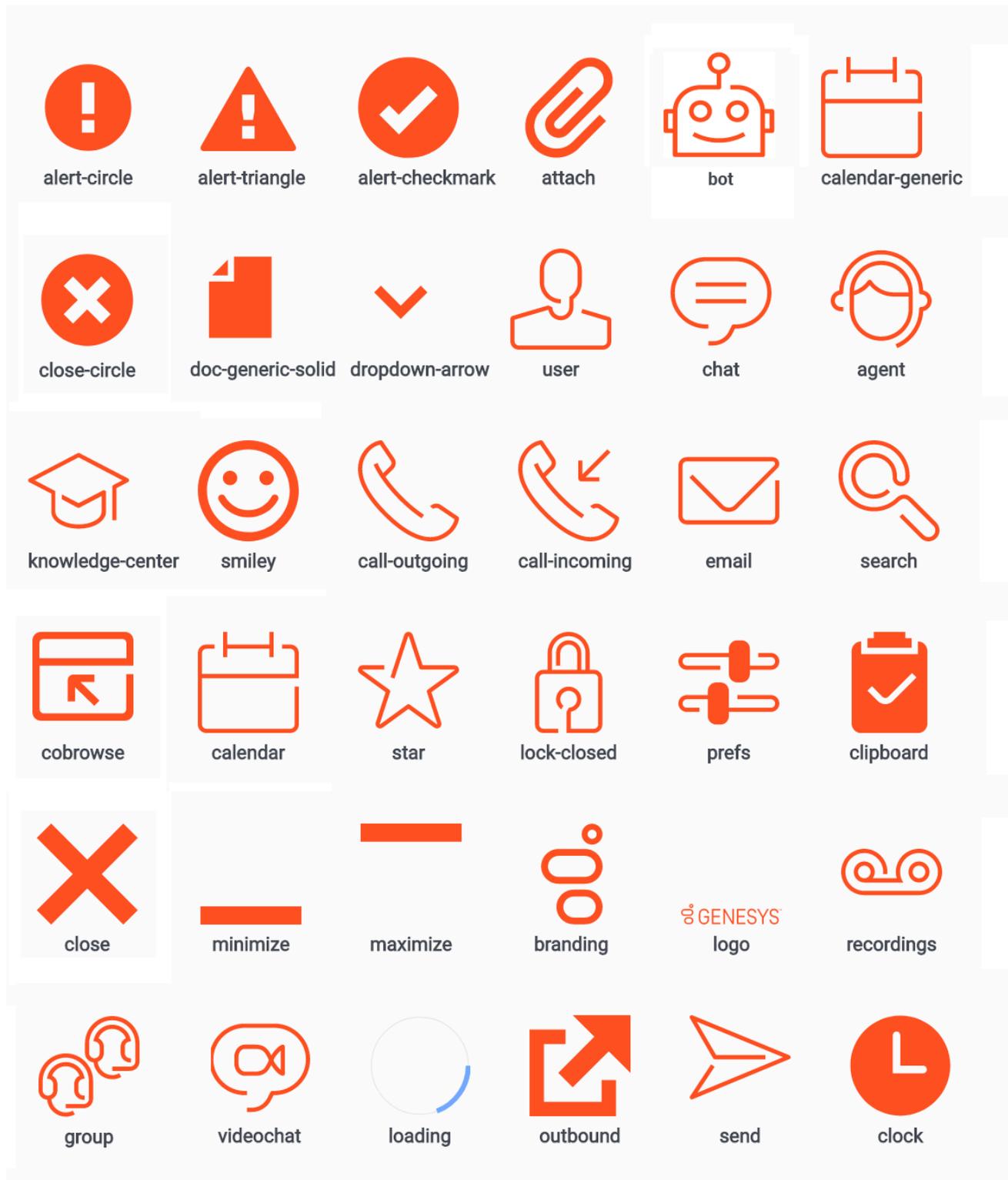
### Fetching SVG Icon Markup

You can also fetch the markup for each SVG icon manually.

**Example:**

```
$("#your-element").append( CXCommon.Generate.Icon("chat" ) );
```

## Included Icons



# Rich Messaging

The Chat Widget provides the ability to display rich messages which enables a more interactive digital experience with your end users. For the Genesys Engage on-premises Chat solution, please refer to [Rich Messaging Support](#) in the *Chat Server Administration Guide* for more information.

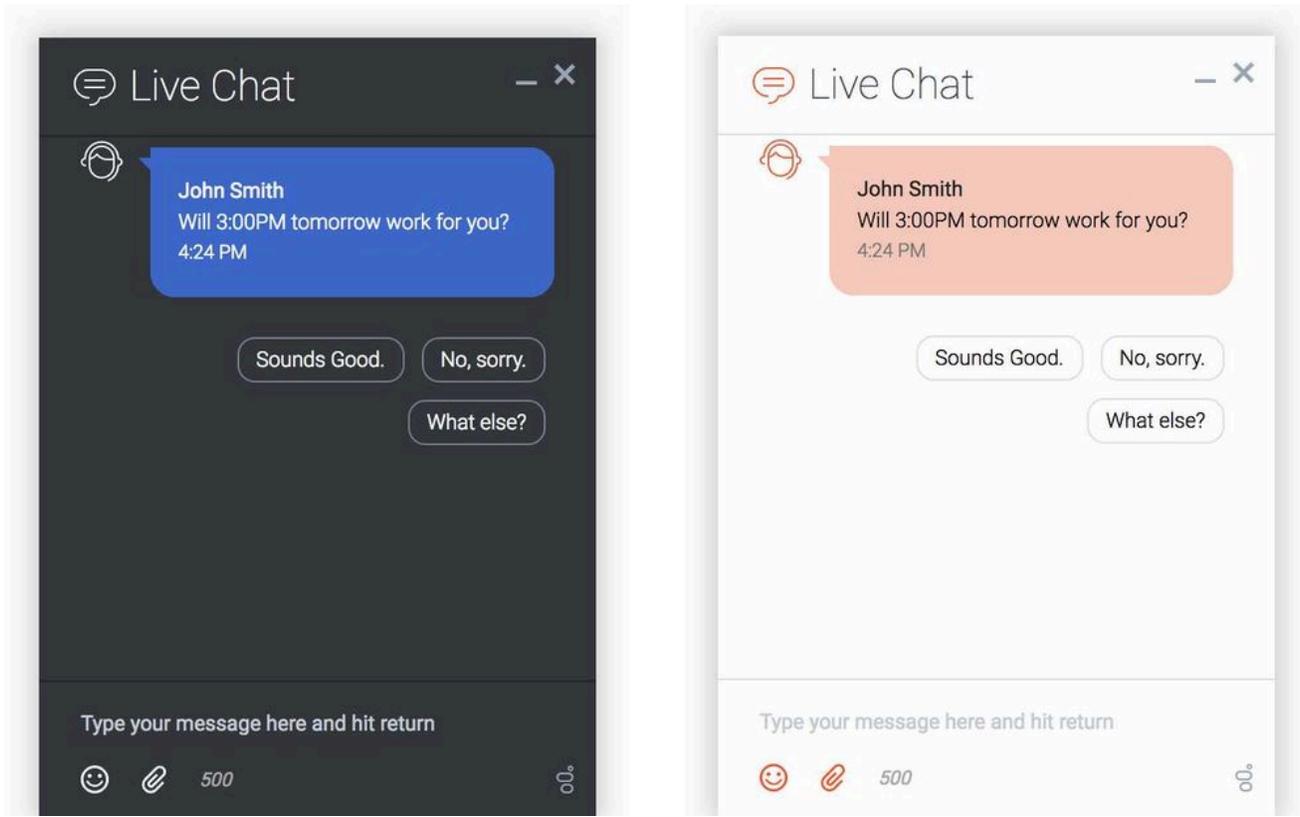
The following is a list of supported Rich Messaging templates:

- [Quick Replies](#)
- [Generic Template](#)
  - Images
  - Videos
  - Buttons
  - Markdown
- [List Template](#)
  - Simple List
  - Selectable List
- [Carousel](#)

## Quick Replies

Quick Replies offer the customer a choice of responses to the last agent or last chat bot message in the transcript. Tapping or clicking one of these Quick Replies posts that reply back to the agent as a text message. This saves the customer from having to type a response manually.

Quick Replies are very flexible. A chat bot or agent can provide context-sensitive quick replies that aid in making a selection, polite responses (such as "OK", "Thank you", "Yes, please" in the customer's native language), numeric responses, or choose from a set of preset time slots.



## JSON Examples

Quick Replies are attached to standard chat message objects by adding a few additional properties:

- **type** - The type of message object received. For quick replies, the value is "Message" or "Structured".
- **contentType** - The type of rich message content attached to the message. For quick replies, the value is "quick-replies".
- **content** - An array of rich message content and other attachments. For quick replies, this contains quick reply objects.

Top level of message object

### Notes:

- This example is only showing properties related to quick replies. All other properties normally included inside a message object have been omitted.
- This example assumes the message text property name is "text", but this may differ from one chat API to another. Please use the appropriate message text property for your chat API.
- Unlike other rich message types, Quick Replies does not use the text property as fallback text for clients that don't support Rich Messages. Rather, it always shows the text as a standard message, with the quick replies as attachments to that message.

```
{
  "type": "Structured",
  "text": "Will 3:00PM tomorrow work for you?",
  "contentType": "quick-replies",
  "content": [
    {"id": "1", "type": "quick-reply", "action": "message", "text": "Sounds Good."},
    {"id": "2", "type": "quick-reply", "action": "message", "text": "No, sorry."},
    {"id": "3", "type": "quick-reply", "action": "message", "text": "What else?"}
  ]
}
```

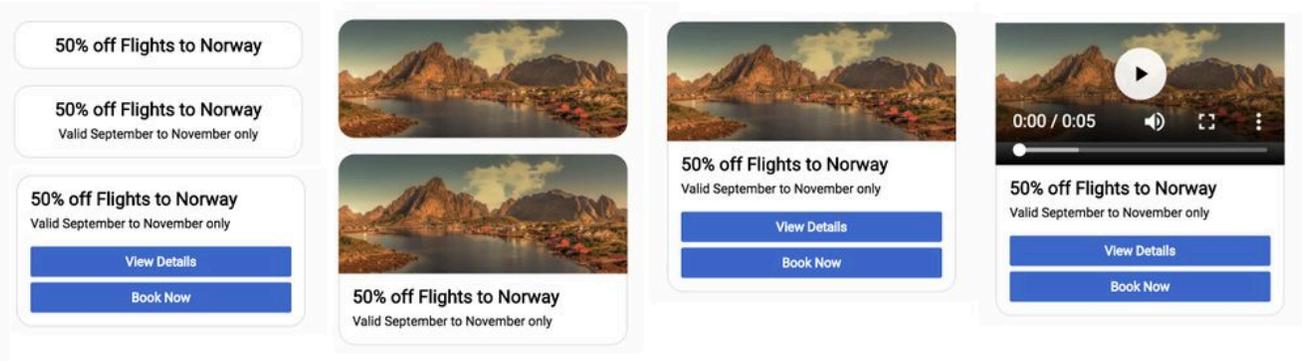
Top-Level of Quick Reply object

Option	Type	Description
id	string	An ID assigned to the quick reply. Each object inside the content array has a unique ID.
type	string	Type identifier for object. For Quick Replies, this value is always "quick-reply".
action	string	Specifies the type of action that is triggered upon clicking the quick reply. Currently, the only supported action is "message" which sends a message using the quick reply text.
text	string	Text to show inside the quick reply. This is also used as the response text after clicking on the quick reply.
image	image URI or base64 data	Path to an image resource or base64 image data to show inside the quick reply. Images appear on the left side of each quick reply. Images are not scaled to fit. Choose an image with the desired dimensions.
imageText	string	Alternative text for image.

## Generic Template

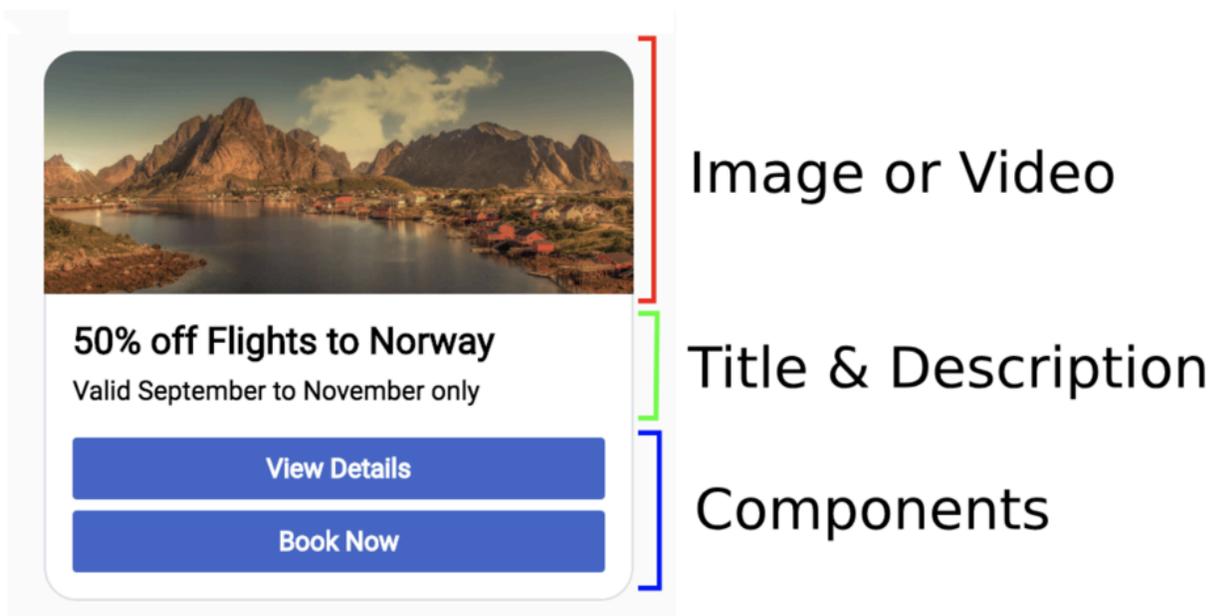
The Generic Templates allow you to craft custom cards with an array of different components like buttons, links, and text.

You can craft your generic templates in many different ways by adding or omitting any property or component. This affords you great flexibility to build your generic templates the way you want.



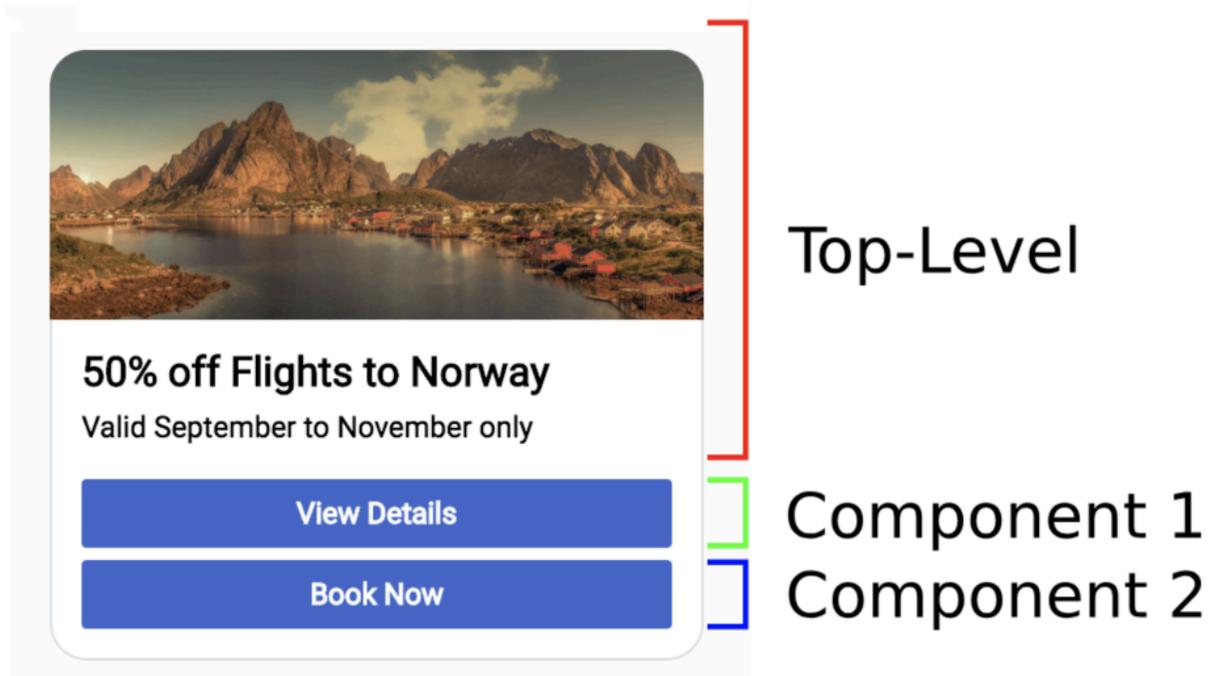
### Template Structure

The structure of the generic template is a simple stack. If you omit any of the content, the stack collapses that space. This gives you the option to use the generic template for showing single elements. This could be as simple as an image or video by itself, a single button or group of buttons, or just text.



### Clickable Areas

The generic template is divided into clickable areas, each having its own actions defined. The main top-level of the structure allows you to click on the image, title, or description to trigger its action.



## JSON Examples

Generic Templates are attached to standard chat message objects by adding a few additional properties:

- **type** - The type of message object received. For generic templates, the value is "Structured".
- **contentType** - The type of rich message content attached to the message. For generic templates, the value is "generic".
- **content** - An array of rich message content and other attachments. For generic templates, this contains generic template objects.

### Top level of message object

#### Notes:

- This example is only showing properties related to generic templates. All other properties normally included inside a message object have been omitted.
- The "text" property shown in this example is used to show a fallback message for clients that don't support Rich Messages. The example below assumes the property name is "text", but this may differ from one chat API to another. Please use the appropriate message text property for your chat API.

```
{
```

```
"type": "Structured",
"text": "fallback text for clients that don't support rich messages.",
"contentType": "generic",
"content": [
  {
    "id": "987a6c84-ada0-468f-86e1-e9ea715b8c32",
    "title": "50% off Flights to Norway",
    "desc": "Valid September to November only",

    "image": "http://www.samplesite.com/flights/img/norway_promo.jpg",
    "imageText": "Norway Promo",
    "video": "",
    "track": {
      "src": "captions_en.vtt",
      "kind": "captions",
      "srcLang": "en",
      "label": "english_captions"
    },
    "actions": {
      "url": "http://www.samplesite.com/flights/norway",
      "urlTarget": "__BLANK"
    },
    "components": [
      {
        "id": "0",
        "type": "button",
        "text": "View Details",
        "title": "View Details",
        "ariaLabel": "View Details about Norway flights",
        "actions": {
          "url": "http://www.samplesite.com/flights/norway",
          "urlTarget": "__BLANK"
        }
      },
      {
        "id": "1",
        "type": "button",
        "text": "Book Now",
        "title": "Book Now",
        "actions": {
          "url": "http://www.samplesite.com/flights/norway/book",
          "urlTarget": "__BLANK"
        }
      }
    ]
  }
]
```

## JSON Reference

### Top-Level of content object

Option	Type	Description
id	string	An ID assigned to this rich message instance. Each instance inside the content array has a unique ID.
image	image file URI	Path or URI to an image file. The image scales to fit the width of the rich message while constraining its aspect ratio. Also used as the video placeholder if both image and video are set.
imageText	string	Alternative text for image.
video	video file URI	Path or URI to a video file. Support is determined by browser. MP4 files are generally supported by all browsers. The video scales to fit the width of the rich message while constraining its aspect ratio. If the image property is also set, that image is used as the placeholder or 'poster' for the video.
track	object	An object containing properties that specify text tracks for video. Note: Added for accessibility support.
track.src	track file URI	Path or URI to track file.
track.kind	string	Kind of text track. Supported options are captions, chapters, descriptions, metadata and subtitles. Default value is set to 'captions'.
track.srclang	string	The language of the track text data. This is required if 'track.kind' is set to subtitles. Default value is set to 'en'.
track.label	string	The title of the text track. Default value is set to 'english_captions'.
title	string	Text to show in the title row.
desc	string	Text to show in the description row. This is immediately below the title.
actions	object	An object containing properties that define the actions that is executed when you click on this

Option	Type	Description
		area or element.
components	array	An array of component objects. Components are stacked in the order defined by this array. Each component can have unique properties. Refer to each component's JSON structure.

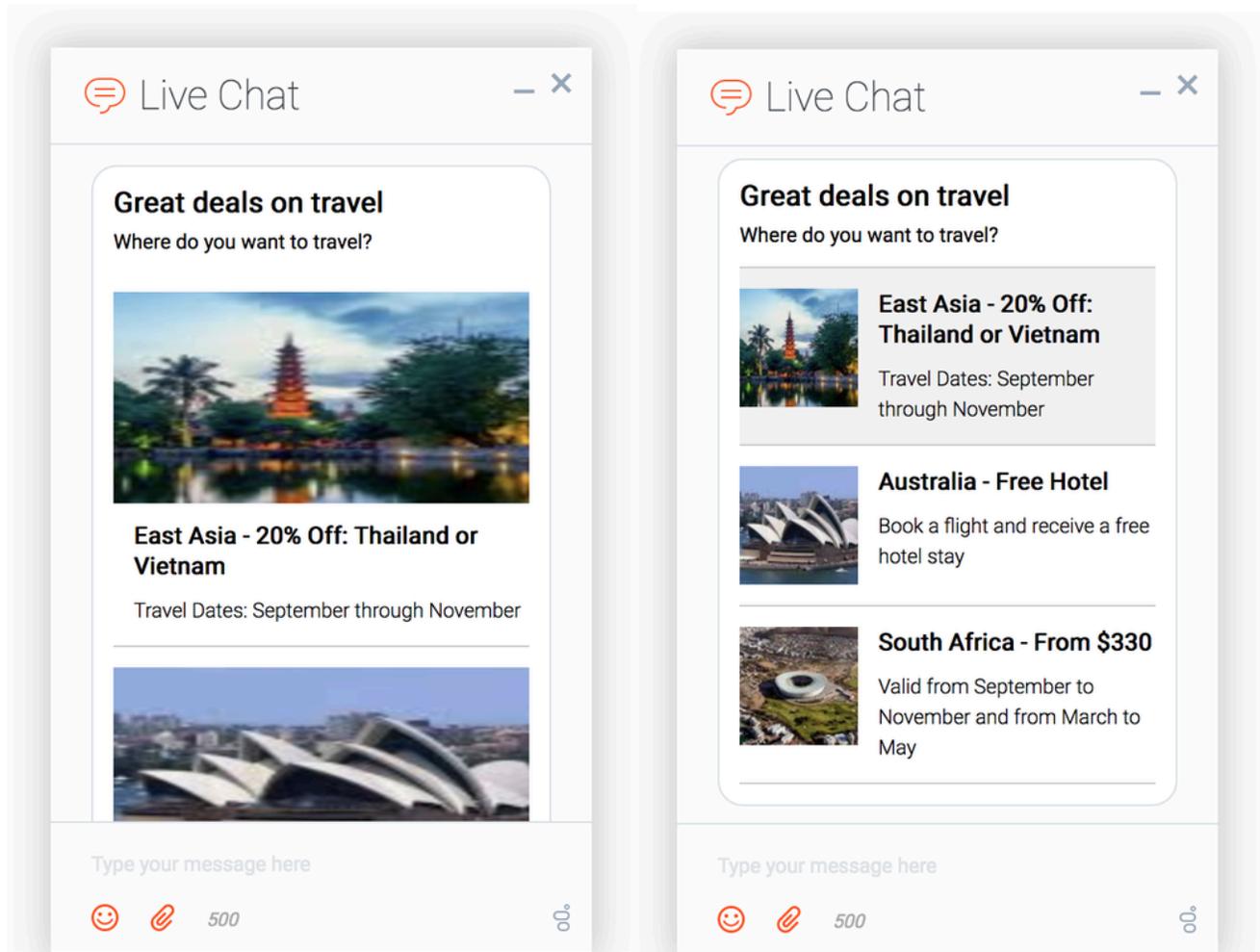
## Button Component

Option	Type	Description
id	string	An ID assigned to this component. Component IDs are for reference only and can be assigned any value.
type	string	The type of component to render. To render a button, the value must be "button".
text	string	Text to show inside the button.
title	string	Tooltip text to show when hovering over the button.
ariaLabel	string	Descriptive text for screen reader to read on focus. Note: Added for accessibility support.
actions	object	An object containing properties that define the actions that are executed when you click on this area or element.

## List Template

List Templates allow you to craft custom cards with an array of list items that can include an image and text.

You can choose between two different list item modes for displaying image and text together: side-by-side or stacked.



### Template Structure

The structure of the list template is main title and description, followed by a stack of list items beneath it.

### Single Selection List

The single selection list provides an option to choose a single list item from the stack of list items. When the list item is selected, the "Submit" button is visible and enabled. By clicking the button, the textback attached to the selected list item is sent back to the agent.

 Live Chat - X

**Great deals on travel**

Where do you want to travel?

---

 **East Asia - 20% Off: Thailand or Vietnam**  
Travel Dates: September through November

---

 **Australia - Free Hotel**  
Book a flight and receive a free hotel stay

---

 **50% off Flights to Norway**  
Valid September to November only

---

Type your message here

  500 00

 Live Chat - X

Where do you want to travel?

---

 **East Asia - 20% Off: Thailand or Vietnam**  
Travel Dates: September through November

---

 **Australia - Free Hotel**  
Book a flight and receive a free hotel stay

---

 **50% off Flights to Norway**  
Valid September to November only

---

[Send Response](#)

Type your message here

  500 00°

 Live Chat - X

### Great deals on travel

Where do you want to travel?



**East Asia - 20% Off: Thailand or Vietnam**

Travel Dates: September through November

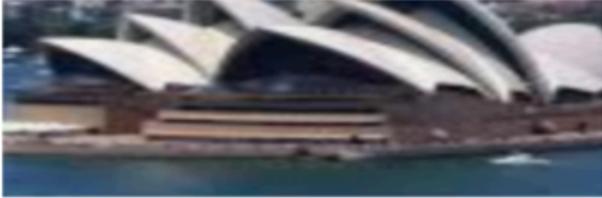


Type your message here

  500



Live Chat



**Australia - Free Hotel**  
Book a flight and receive a free hotel stay



**50% off Flights to Norway**  
Valid September to November only

**Send Response**

Type your message here

😊 📎 500

---

## Clickable Areas

List items are the only clickable areas in a normal list template. Clicking on the title or description does not trigger an action. Each list item can be configured with its own actions that are executed when you click anywhere inside that list item.

## JSON Examples

- **type** - The type of message object received. For list templates, the value is "Structured".
- **contentType** - The type of rich message content attached to the message. For normal list templates, the value is "list-vertical" and for single selection list templates, the value is "single-selection-list".
- **content** - An array of list item objects.

### Top level of message object

Notes:

- This example is only showing properties related to list templates. All other properties normally included inside a message object have been omitted.
- The "text" property shown in this example is used to show a fallback message for clients that don't support Rich Messages. The example below assumes the property name is "text", but this may differ from one chat API to another. Please use the appropriate message text property for your chat API.

```
// Normal List Template
{
  "type": "Structured",
  "text": "Fallback text for clients that don't support rich messages",
  "contentType": "list-vertical",
  "content": [
    {
      "id": "987a6c84-ada0-468f-86e1-e9ea715b8c32",
      "title": "Great deals on travel",
      "desc": "Where do YOU want to travel?",
      "components": [
        {
          "id": "0",
          "rmid": "987a6c84-ada0-468f-86e1-e9ea715b8c32",
          "type": "list-item",
          "image": "http://www.samplesite.com/flights/img/southeastasia.jpg",
          "imageText": "East Asia Flights",
          "title": "East Asia - 20% Off: Thailand or Vietnam",
          "desc": "Travel Dates: September through November",
          "actions": {
            "url": "http://www.samplesite.com/flights/southeastasia",
            "urlTarget": "__BLANK"
          }
        },
        {

```

```

        "id": "1",
        "rmid": "987a6c84-ada0-468f-86e1-e9ea715b8c32",
        "type": "list-item",
        "image": "http://www.samplesite.com/flights/img/australia.jpg",
        "title": "Australia - Free Hotel",
        "desc": "Book a flight and receive a free hotel stay",
        "actions": {
            "url": "http://www.samplesite.com/flights/australia",
            "urlTarget": "__BLANK"
        }
    },
    {
        "id": "2",
        "rmid": "987a6c84-ada0-468f-86e1-e9ea715b8c32",
        "type": "list-item",
        "image": "http://www.samplesite.com/flights/img/southafrica.jpg",
        "title": "South Africa - From $330",
        "desc": "Valid from September to November and from March to May",
        "actions": {
            "url": "http://www.samplesite.com/flights/southafrica",
            "urlTarget": "__BLANK"
        }
    }
]
}
]
}

// Single Selection List template
{
    "type": "Structured",
    "text": "Fallback text for clients that don't support rich messages",
    "contentType": "single-selection-list",
    "content": [
        {
            "id": "987a6c84-ada0-468f-86e1-e9ea715b8d43",
            "title": "Great deals on travel",
            "desc": "Where do YOU want to travel?",
            "submitLabel": "Send Response",
            "ariaSubmitLabel": "Send your selection",
            "actions": {
                "textback": "Textback: Cliked on the SubmitButton",
                "commandName": "SendMessage.open",
                "commandOptions": {"form": {"firstname": "test"}}
            },
            "components": [
                {
                    "id": "0",
                    "rmid": "987a6c84-ada0-468f-86e1-e9ea715b8d43",
                    "type": "list-item",
                    "image": "http://www.samplesite.com/flights/img/southeastasia.jpg",
                    "title": "East Asia - 20% Off: Thailand or Vietnam",
                    "desc": "Travel Dates: September through November",
                    "actions": {
                        "textback": "Trip to Thailand",
                        "url": "http://www.samplesite.com/flights/southeastasia",
                        "urlTarget": "__BLANK"
                    }
                }
            ]
        }
    ]
}

```

```

    },
    {
      "id": "1",
      "rmid": "987a6c84-ada0-468f-86e1-e9ea715b8d43",
      "type": "list-item",
      "image": "http://www.samplesite.com/flights/img/australia.jpg",
      "imageText": "Australia Flights",
      "title": "Australia - Free Hotel",
      "desc": "Book a flight and receive a free hotel stay",
      "actions": {
        "textback": "Trip to Australia",
        "url": "http://www.samplesite.com/flights/australia",
        "urlTarget": "__BLANK"
      }
    },
    {
      "id": "2",
      "rmid": "987a6c84-ada0-468f-86e1-e9ea715b8d43",
      "type": "list-item",
      "image": "http://www.samplesite.com/flights/img/southafrica.jpg",
      "title": "South Africa - From $330",
      "desc": "Valid from September to November and from March to May",
      "actions": {
        "textback": "Trip to South Africa",
        "url": "http://www.samplesite.com/flights/southafrica",
        "urlTarget": "__BLANK"
      }
    }
  ]
}

```

### JSON Reference

Top-Level of content object

Option	Type	Description
id	string	An ID assigned to this rich message instance. Each instance inside the content array has a unique ID.
title	string	Text to show in the title row.
desc	string	Text to show in the description row. This is immediately below the title.
submitLabel	string	Label for Submit button. Note: This is applicable only for "single-selection-list" template.
ariaSubmitLabel	string	Descriptive text of Submit button for screen reader to read on focus. Note: Added for accessibility support.

Option	Type	Description
actions	object	An object containing properties that define the actions that are executed when you click on submit button. Note: This is applicable only for "single-selection-list" template.

### List Item Component

Option	Type	Description
id	string	An ID assigned to this component. Component IDs are for reference only and can be assigned any value.
rmid	string	An ID of the rich message instance.
type	string	The type of component to render. Either "list-item" for side-by-side mode, or "list-item-big" for large image, stacked mode.
image	image file URI	Path or URI to an image file. An image to show inside the list item. Note: Images are optional; leave blank if no image is preferred.
imageText	string	Alternative text for image.
title	string	The main headline of the list item.
desc	string	Text to show in the description row. This is immediately below the title.
actions	object	An object containing properties that define the actions that are executed when you click on this list item.

## Carousel

If multiple rich messages are defined inside the "contents" array of a single JSON response, those rich messages are shown together in a horizontal carousel. The carousel view shows the first rich message by default, and also provides left and right navigation arrows to move between rich messages inside the carousel.

The carousel view has no configuration options or special features. It is created automatically when multiple rich messages of the same type are sent in a single message. There is no other way to generate a carousel view.

[Link to video](#)

## Important

- Only multiple rich messages of the same type can be shown inside a carousel view (for example, a carousel of Generic Templates, or a carousel of List Templates).
- Quick Replies are not supported by the carousel view.

## JSON Schema

```
{
  "id": "",
  "type": "Structured",
  "contentType": "",
  "content": [
    {
      .. // Rich Message object 1
    },
    {
      .. // Rich Message object 2
    },
    ..
  ]
}
```

## JSON Example

This example contains multiple rich messages inside the contents array forming a carousel view of Generic Templates.

// Carousel with Generic Template

```
{
  "id": "30dd879c-ee2f-11db-8314-0800200c9a66",
  "type": "Structured",
  "contentType": "generic",
  "content": [
    {
      "id": "987a6c84-ada0-468f-86e1-e9ea715b8c33",
      "title": "50% off Flights to Norway",
      "desc": "Valid September to November only",
      "image": "https://www.samplesite.com/flights/img/norway.jpg",
      "actions": {
        "url": "http://www.samplesite.com/flights/norway",
        "urlTarget": "__BLANK",
        "textback": "User Clicked on Link"
      }
    },
  ]
}
```

```

"styles": {
  "align": "center"
},
"components": [
  {
    "id": "2",
    "type": "text",
    "text": "If you're interested, this is some plain text."
  },
  {
    "id": "",
    "type": "link",
    "text": "Click Me!",
    "cxbus": {
      "command": "WebChat.open",
      "options": {}
    },
    "styles": {
      "color": "blue"
    }
  }
]
},
{
  "id": "987a6c84-ada0-468f-86e1-e9ea715b8c38",
  "title": "50% off Flights to Norway",
  "desc": "Valid September to November only",
  "image": "https://www.samplesite.com/flights/img/norway.jpg",
  "actions": {
    "url": "http://www.samplesite.com/flights/norway",
    "urlTarget": "__BLANK",
    "textback": "User Clicked on Link"
  },
  "components": [
    {
      "id": "0",
      "type": "button",
      "text": "Yes, I'm interested!",
      "title": "View Details",
      "actions": {
        "textback": "User clicked on 'Yes, I'm interested'"
      }
    },
    {
      "id": "1",
      "type": "button",
      "text": "Not at this time",
      "title": "Book Now",
    }
  ]
}

```

```

        "actions": {
            "textback": "User clicked on 'Not at this time'"
        }
    }
],
},
{
    "id": "987a6c84-ada0-468f-86e1-e9ea715b8c37",
    "title": "50% off Flights to Norway",
    "desc": "Valid September to November only",
    "video": "http://techslides.com/demos/sample-videos/small.mp4",
    "components": [
        {
            "id": "0",
            "type": "button",
            "text": "Yes, I'm interested!",
            "title": "View Details"
        },
        {
            "id": "1",
            "type": "button",
            "text": "Not at this time",
            "title": "Book Now"
        }
    ]
},
{
    "id": "987a6c84-ada0-468f-86e1-e9ea715b8c32",
    "title": "50% off Flights to Norway",
    "desc": "Valid September to November only",
    "actions": {
        "url": "http://www.samplesite.com/flights/norway",
        "urlTarget": "__BLANK",
        "textback": "User Clicked on Link"
    },
    "styles": {
        "align": "center"
    }
}
]
}

```

## Actions

Rich Messages allow you to configure actions to be performed when certain elements are clicked. The following actions are supported:

- Open a URL and set the target window
- Send a text response back through the transcript

- Execute a CXBus command and pass options to that command

```
"actions": {
  "url": "http://www.samplesite.com/flights/norway",
  "urlTarget": "__BLANK",
  "textback": "User Clicked on Link",
  "commandName": "SendMessage.open",
  "commandOptions": {"form": {"firstname": "john", "lastname": "smith", "email":
"john@smith.com"}}
}
```

All actions are optional and all actions are executed simultaneously.

You can define actions for the top-level of a rich message (in other words, the main image, title, and description), and you can define unique actions for each component inside your components array. This allows you to create a stack of buttons that each perform different actions when clicked.

## Open URLs

One of the most common and simple actions is to open a link in a new window.

```
"actions": {
  "url": "http://www.samplesite.com/flights/norway",
  "urlTarget": "__BLANK"
}
```

You can control the target window or tab by specifying the target in the `urlTarget`. This is applied to the link element's "target" attribute. The default target is "\_\_BLANK", which opens a new window or tab.

## Textback

Another common action is to send a message back to the agent through the transcript.

```
"actions": {
  "textback": "I clicked the Norway link"
}
```

This posts a message as if the user had typed it. This functions much like quick replies.

## Commands

You can trigger client-side commands to help the user open other widgets or run custom commands you've created on your website.

```
"actions": {
  "commandName": "SendMessage.open",
  "commandOptions": {"form": {"firstname": "john", "lastname": "smith", "email":
"john@smith.com"}}
}
```

Commands are executed by name and you can pass in an object with properties known as "options". These options control the command in the same way arguments control a function call.

An example scenario might go like this:

1. A user starts a chat and while he is waiting for an agent, a chatbot queries the user
2. If no agents are available, or if it is after hours, the chatbot can send a Generic Template with a button offering the user to send an email instead
3. If the user clicks the button, the SendMessage widget opens next to the WebChat widget.
4. This has assisted the customer in using Genesys Widgets based on situation the chatbot identified.

## Markdown

Starting in Genesys Widgets version 9.0.009.04, Markdown text formatting is supported in both Generic and List templates. Our implementation follows the [CommonMark spec](#). Please review syntax rules and available formatting [here](#).

Markdown support is currently limited to the "desc" properties of **Generic** templates and **List Item** components. Markdown will not be rendered from any other property in the JSON payload.



### **This is a title!**

This is some *emphasized text* and some **strongly emphasized text**.

This is a bulleted list:

- item 1
- item 2
- item 3

This is a numbered list:

1. item 1
2. item 2
3. item 3

[This is a link](#)

This is an image:



## Example

This is some *\*emphasized text\** and some **\*\*strongly emphasized text\*\***

This is a bulleted list:

- \* item 1
- \* item 2
- \* item 3

This is a numbered list:

1. item 1
2. item 2
3. item 3

```
[This is a link](http://www.genesys.com)
```

This is an image:

```
![foo](http://www.genesys.com/images/genesys-logo.png)
```

## Markdown Links

To create a link, use the following syntax:

```
[genesys](http://www.genesys.com/images/genesys-logo.png)
```

As of 9.0.016.07, you can set the link's target attribute using this syntax.

```
[genesys](http://www.genesys.com/images/genesys-logo.png{{target="_self"}})
```

This new syntax component is appended to the URL directly, inside the parentheses, is case-sensitive, and must use double-quotes. Here is the new component by itself.

```
{{target="_self"}}
```

The value between the double-quotes can be any value needed to target the window or tab needed.

## Rich Messaging in Chat Server

Chat Server provides chat session participants (either an agent or chat bot) with the ability to use rich messaging in a chat session. For more information, please see, [Using rich messaging](#).

# Accessibility

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. 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.

Plugin	WCAG 2.1 Support
<a href="#">Web Chat</a>	Level AA
<a href="#">Send Message</a>	Level AA
<a href="#">Callback</a>	Level AA
<a href="#">Calendar</a>	Level AA
<a href="#">Call Us</a>	Level AA
<a href="#">Channel Selector</a>	Level AA
<a href="#">Click To Call</a>	Level AA
<a href="#">Engage</a>	Level AA

---

Plugin	WCAG 2.1 Support
Search	Level AA
Side Bar	Level AA
Co-browse	Not Available
Chat Deflection	Not Available

### Important

Co-browse and Chat Deflection are not included in our WCAG support program.

Widgets supports 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. They announce all the textual and non-textual content on the Widgets window elements, announce new chat messages sent by the agent to the user as well as the outgoing messages sent by the user to the agent. They also read error messages and live Estimated Wait Times automatically. 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 the following:

- WebChat for reading new messages
- Callback, ClickToCall, and ChannelSelector to read live Estimated Wait Time

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

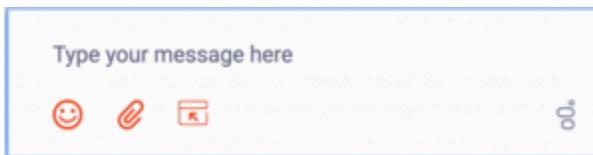
### Important

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](#).

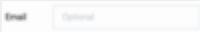
### Important

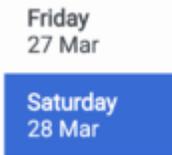
Widgets does not add the "aria-hidden" attribute on the customer page html elements. Due to this limitation, when using screen reader gestures in some Android devices, the focus may not be trapped within the widget. To trap the focus, a custom event handling script needs to be added subscribing to the 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.		

Description	Before	After
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.		
As per the “1.4.11 Non-text contrast” success criterion, Estimated Wait Time status icon color upon hovering over the channels has been modified to ensure that it has a contrast ratio of at least 3:1 against background or adjacent colors.		
As per the “1.4.3 Contrast (minimum)” success criterion, text color has been modified to ensure that the contrast ratio of at least 4.5:1 exists between text and background.		

## 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 screenreaders. 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	<code>ariaIdleAlertIntervals</code>	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	<code>ariaCharRemainingIntervals</code>	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	<code>emojiList</code>	<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 <b>aria-label</b> attributes on the non-text emoji markup.
ClickToCall	<code>ariaNumberExpirationIntervals</code>	An array containing the intervals in a percentage at which the screen reader will announce the remaining expiry time for the phone number. By default, it is enabled with the following time intervals, and is customizable according to user needs. Configuring a value of <code>false</code> will let the screen reader read the phone number expiry time for every change.

## Resources and Tools Used

### Online

- [webaim.org](http://webaim.org)

- [deque.com](https://deque.com)
- [contrastchecker.com](https://contrastchecker.com)

## Screen Readers

- JAWS
- NVDA
- VoiceOver