



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

Widgets Developer's Guide

Genesys Widgets deployment guide

5/26/2022

Contents

- 1 Audience
- 2 How can I deploy Genesys Widgets?
- 3 Deploying Genesys Widgets (lazy loading)
 - 3.1 Files used
 - 3.2 On-demand lazy loading
 - 3.3 Preloading plugins
- 4 Deploying Genesys Widgets (all-in-one)
 - 4.1 Files Used
- 5 Alternative deployment script
- 6 Releases hosted on Content Delivery Network (CDN)
 - 6.1
 - 6.2
 - 6.3 Choose region
 - 6.4 Using Genesys Widgets CDN with versions
 - 6.5 Versioning examples with scenarios
 - 6.6 Deployment methods
- 7 Checking Widgets version

-
- Developer

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

Related documentation:

-

Audience

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

How can I deploy Genesys Widgets?

There are two deployment methods available:

- Lazy loading— The recommended method, 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 (deprecated) —The all-in-one method is the "classic" 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)

Important

Lazy loading is the recommended method for Widgets. The (all-in-one) method is deprecated.

Files used

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

A good starting point is the following 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. Loads 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 Configuring Genesys Widgets for help.

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)

Important

Lazy loading is the recommended method for Widgets. The (all-in-one) method is deprecated.

Files Used

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

A good starting point is the following script:

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 Configuring Genesys Widgets for help.

Alternative deployment script

To simplify the deployment process while using tools like Google Tag Manager, you can use the below

script to embed Widgets.

Releases hosted on Content Delivery Network (CDN)

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

`https://apps.mypurecloud.com/widgets//`

Note that `{}` and `{}` are placeholders.

This value varies based on the deployment method you choose.

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 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 Widgets Release Note.

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)	<code>https://apps.mypurecloud.com/widgets/{version}/{path/to/file}</code>
North America (West)	<code>https://apps.usw2.pure.cloud/widgets/{version}/{path/to/file}</code>
North America (Canada)	<code>https://apps.cac1.pure.cloud/widgets/{version}/{path/to/file}</code>
Australia or New Zealand	<code>https://apps.mypurecloud.com.au/widgets/{version}/{path/to/file}</code>
EU (Ireland)	<code>https://apps.mypurecloud.ie/widgets/{version}/{path/to/file}</code>
EU (Frankfurt)	<code>https://apps.mypurecloud.de/widgets/{version}/{path/to/file}</code>

Region	URL
UK (London)	<code>https://apps.euw2.pure.cloud/widgets/{version}/{path/to/file}</code>
Japan	<code>https://apps.mypurecloud.jp/widgets/{version}/{path/to/file}</code>
Seoul	<code>https://apps.apne2.pure.cloud/widgets/{version}/{path/to/file}</code>

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 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:

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
```

Checking Widgets version

CXBus.command("App.info");

Prints out the debug header with version information.

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

Returns the version number directly, as a string.