Deployment Guide

Localization

5/10/2025

Contents

- 1 Localization
  - 1.1 Master Localization File
  - 1.2 Multiple Translated Language Packs
  - 1.3 Configuration Options
  - 1.4 Language Pack JSON Format
  - 1.5 Localization Namespaces
  - 1.6 Language Codes
  - 1.7 Plugin Localization Options

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