



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

Localize widgets and services

7/12/2025

---

## Contents

- [1 Master localization file](#)
- [2 Multiple translated language packs](#)
  - [2.1 Example](#)
- [3 Configuration options](#)
- [4 Language pack JSON format](#)
- [5 Localization namespaces](#)
- [6 Language codes](#)
- [7 Plugin localization options](#)

- 
- Developer

Localize your Genesys Widgets user messages and prompts by creating and hosting a Language Pack that Genesys Widgets can access.

### Related documentation:

- 

The Language Pack is a special file written in JSON format.

You also have to specify your Language Pack file in the **window\_genesys.widgets.main** section of your Genesys Widgets configuration options, as shown in this example:

## Master localization file

The widgets-en.i18n.json file 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

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 option. 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 this example:

```
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//i18n/widgets-fr.i18n.json"  
}  
};
```

### Language code mapping examples:

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

#### Important

You may use any language code you wish. The above table is for reference only.

## 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 |
| Offers          | offers          |
| WebChat         | webchat         |

## Language codes

To allow flexibility in the way your website handles multiple languages and language codes, there are no rules for language codes other than that they must be strings. This means that you can use any language code system.

However, the language code that you set in **window.genesys.widgets.main.lang** must correspond to a language code in the Language Pack File.

### Important

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

- ChannelSelector
- CallUs
- Calendar
- Callback
- SideBar
- WebChat