



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.

Developer's Guide

Localization

12/13/2025

Localization

Genesys Co-browse localization is split into two parts:

- [Localizing the Customer UI](#)
- [Localizing the Agent UI](#)

Co-browse is localized for English by default. To modify the default English localization or add localization for other languages, see the following sections on this page:

- [Localizing the Customer Co-browse UI](#)
- [Localizing the Co-browsing button](#)
- [Localizing the Agent UI](#)
- [Caching and updating Localization \(I10n\) files](#)
- [Built-in localization](#)

Localizing the Customer Co-browse UI

To localize the customer Co-browse UI, configure the `localization` option in the `cobrowse` subsection of the global configuration variable `_genesys`. To configure the localization option, do one of the following:

- Pass localization data to the configuration object in your website instrumentation. You can [provide localization directly as a JS object](#) or [provide localization using a function](#).
- Save the localization in [.json](#) files and serve them as JSONP. The Co-browse application uses these files on top of the default built-in values.

The localization files are plain JSON files, loaded through a JSONP request. This means they can be hosted on any domain as long as JSONP is supported by the hosting server. Co-browse uses a standard `callback` argument for the callback function name. See [Serving JSONP](#) for information on how to serve JSONP using Co-browse servers.

Provide Localization Directly as a JS Object

The simplest way to provide localization is to use a plain JavaScript object of key value pairs.

Example:

```
<script>
_genesys = {
  cobrowse: {
    localization: {
      exitBtnTitle: "Stop Co-browsing"
    }
  }
};
```

```
</script>
```

Provide Localization Using a Function

If a JavaScript object is not sufficient, you can use a function to provide localization. For example, you can use a function to figure out localization at runtime.

Example:

```
<script>
_genesys = {
  cobrowse: {
    localization: function() {
      if (window.currentLanguage = 'ru') {
        return { toolbarContent: "Ключ сессии: {sessionId}" }
      }
    }
  }
};
</script>
```

Asynchronous Function

Your function can also be asynchronous and you can use an asynchronous function to load localization how you prefer. To tell the configuration variable that your function is asynchronous, create your function with an argument. Customarily, the argument is done. We will pass a function that you have to call whenever the localization loads.

Example:

```
<script>
_genesys = {
  cobrowse: {
    localization: function(done) {
      jQuery.get('my/cobrowse/localization.json', done);
    }
  }
};
</script>
```

Warning

Functions you pass will be called after our scripts initialize, shortly after the document ready event. If you use an asynchronous function, initialization will be postponed until you call the done function.

Provide Localization Using an External JSON File

If you pass a string to the localization configuration option, the string will be treated as a URL to an external JSON file.

Example:

```
<script>
_genesys = {
  cobrowse: {
    localization: '//example.com/cobrowse-l10n/2014-10-08/cobrowse-fr.json'
  }
};
</script>
```

Important

Our scripts will attempt to load localization files using **JSONP**. You can use Co-browse server to serve files as JSONP for you. See [Serving JSONP](#).

If the files can not be loaded, initialization of our functionality will be blocked.

You must save localization JSON files with UTF-8 encoding.

Localizing the Co-browsing button

The "Co-browsing" default button image and localization changes will not affect the text on this button. Instead, you can localize the button using the following method:

Providing Custom HTML for buttons

You can pass a **function** that returns HTML elements to **buttons.cobrowse**. In this case, the output of the function is used to render the button instead of default image.

Note that in this case your custom button inherits the positioning of the default button.

Here's a simple example that makes use of the jQuery library to generate HTML elements:

```
function createCustomButton() {
  return jQuery('<div class="myButtonWrapper"><button class="myButton">Co-
browse</button></div>')[0];
}

var _genesys = {
  buttons: {
    cobrowse: createCustomButton
  }
};
```

Important

jQuery is NOT mandatory to use in order to provide a custom HTML element. The example above does return an HTML element out of a jQuery object by retrieving the first element from jQuery collection via [0].

Configuring Buttons

The **_genesys.buttons** section allows some basic configuration of the "Co-browsing" button. It has two optional properties:

- **position**: Can be either "left" (default) or "right"
- **cobrowse**: Defaults to true

Note that you can override only the properties that you want to be changed. Other properties are used with their default values. For example this configuration:

```
var _genesys = {  
  buttons: {  
    cobrowse: false  
  }  
};
```

actually means this:

```
var _genesys = {  
  buttons: {  
    cobrowse: false,  
    position: 'left' // inherited default  
  }  
};
```

Disabling Buttons

As seen in the snippet above, you can pass false to disable the "Co-browsing" button. This might be useful if you want to start co-browsing from your own custom button (or from any other element or event), using the [Co-browse API](#).

Overriding Buttons

Override how the button looks using CSS. See [Customizing the CSS](#) for details.

Localizing the Agent UI

The agent side UI is localized by updating the localization [configuration option](#) to the URL of the JSON localization file in the [slave](#) section of the Co-browse Cluster application. Unlike the customer side UI, the value is changed in the Genesys Configuration server and not via the instrumentation script. The agent side UI can not be configured by passing a JavaScript object or function.

Important

The localization file is loaded using [JSONP](#). You can use Co-browse server to serve files as JSONP for you or configure your own infrastructure for JSONP support. See [Serving JSONP](#).

Important

Starting in release 9.0.005.33, Genesys Co-browse Plug-in for Workspace Desktop Edition (WDE) now has a stricter policy for working with origins against the agent's localization. To allow working with the localization resource via HTTPS, you must place the resource in the same origin that the Co-browse Plug-in for WDE uses to work with Co-browse.

If load balancing is used for the Co-browse Plug-in for WDE to access Co-browse, place the JSON localization file in the static folder of the Co-browse nodes, and add the following snippet in your NGINX configuration file.

```
location /static {
    proxy_pass https://<cobrowsecluster>$uri?$args;
}
```

Caching and updating Localization (l10n) files

Requests to the l10n files are made on every page that is instrumented with Co-browse, before the Co-browse UI is displayed. Genesys recommends that you implement a caching mechanism for these files if you host them on your servers.

For best performance, add far-future expiration headers (for example, Expires, Cache-Control or both) to your l10n files. This prevents the browser from requesting these files on each page. Instead, it will take them from the cache. This reduces the start-up time for the Co-browse UI and cuts down traffic for the end user. If you modify a localization file with a far-future expiration header, the browser must request the new version of the file from the server instead of taking it from the cache. To force the browser to do this you must change the URL of the file. You can do this by updating the corresponding localization parameter in Co-browse instrumentation after either putting the modified file in a new directory or updating the file's name.

For example, consider the case where you have set up your own server to host the Co-browse localization files for the customer side UI:

```
<script>
var _genesys = {
  cobrowse: {
    localization: '//example.com/cobrowse-l10n/2014-09-07/cobrowse-fr.json'
  }
};
</script>
<COBROWSE_INSTRUMENTATION_SCRIPT>
```

Next, you modify some of your localization files and want them to be refreshed for all users, so you create a new directory and update your Co-browse instrumentation:

```
<script>
var _genesys = {
  cobrowse: {
    localization: '//example.com/cobrowse-l10n/2014-10-08/cobrowse-fr.json'
  }
};
</script>
```

```
    }  
};  
</script>  
<COBROWSE_INSTRUMENTATION_SCRIPT>
```

Now, any browsers that had cached the files will be reload the files and re-cache them.

Built-in localization

This section lists the default localization values and keys.

You can use the code snippets in this section to create your own localization files. To do so, copy and save the code snippet as a .json file.

Important

You must save your localization JSON files using UTF-8 encoding.

Tip

It is not necessary to list **all** the keys in your localization JSON file. In your JSON file, you can specify only the keys you wish to override. Any key not specified will use the default localization value.

Customer UI

```
{  
  "agentJoined": "Representative has joined the session",  
  "youLeft": "You have left the session. Co-browse is now terminated.",  
  "sessionTimedOut": "Session timed out. Co-browse is now terminated.",  
  "sessionInactiveTimedOut": "Session timed out. Co-browse is now terminated.",  
  "agentLeft": "Representative has left the session. Co-browse is now terminated.",  
  "sessionError": "Unexpected error occurred. Co-browse is now terminated.",  
  "sessionsOverLimit": "Representative is currently busy with another Co-browse session. Co-browse is now terminated.",  
  "serverUnavailable": "Could not reach Co-browse server. Co-browse is now terminated.",  
  "sessionStarted": "Your co-browse session ID is {sessionId}. Please spell it to our representative to continue with co-browsing.",  
  "navRefresh": "Representative has refreshed the page. Reloading.",  
  "navBack": "Representative has pressed the \"Back\" button. Reloading page.",  
  "navForward": "Representative has pressed the \"Forward\" button. Reloading page.",  
  "navUrl": "Representative has requested navigation. Reloading page.",  
  "navFailed": "Navigation request by representative has failed.",  
  "toolbarContent": "Session ID: {sessionId}",  
  "contentMasked": "Content is hidden from representative",  
  "contentMaskedPartially": "Some content is hidden from representative",  
  "exitBtnTitle": "Exit Co-browse session",  
  "areYouOnPhone": "Are you on the phone with our representative?",  
  "areYouOnPhoneOrChat": "Are you on the phone or chat with our representative?",  
}
```

```
"connectBeforeCobrowse": "You need to be connected with our representative to continue with co-browsing. Please call us or start a live chat with us, and then start Co-browse again.",
"sessionStartedAutoConnect": "Co-browse session started. Waiting for representative to connect to the session...",
"browserUnsupported": "Unfortunately, your browser is not currently supported.<br><br>Supported browsers are: <ul><li><a target='_blank' href='http://www.google.com/chrome'>Google Chrome</a></li><li><a target='_blank' href='http://www.firefox.com/'>Mozilla Firefox</a></li><li><a target='_blank' href='http://microsoft.com/ie'>Internet Explorer 9 and above</a></li><li><a target='_blank' href='https://www.apple.com/safari'>Safari 6 and above</a></li></ul>",
"modalTitle": "Co-browse",
"modalYes": "Yes",
"modalNo": "No",
"writeModeInProgress": "Agent has control over the page.",
"downgradeMode": "Revoke control",
"modeUpgraded": "Co-browse session was upgraded. Agent has control over the page.",
"modeDowngraded": "Co-browse session was downgraded. Agent has no control",
"modeUpgradeRequested": "Agent requests upgrading Co-browse session to \"write\" mode. In \"write\" mode agent will have control over the page."
}
```

Important

The modalTitle and serverUnavailable keys were added to Genesys Co-browse in release 8.5.001.xx.

Agent UI

```
{
  "invalidSessionID": "Session ID is invalid or has expired.",
  "navRefresh": "Refresh is pressed. Reloading page.",
  "navBack": "Back is pressed. Reloading page.",
  "navForward": "Forward is pressed. Reloading page.",
  "youLeft": "You have left the session. Co-browse is now terminated.",
  "customerLeft": "Customer has left the session. Co-browse is now terminated.",
  "exitBtnText": "Exit Session",
  "sessionIdText": "Session ID: {sessionId}",
  "enterSessionIdText": "Session ID:",
  "modeWrite": "Mode: Write",
  "modePointer": "Mode: Pointer",
  "writeModeTip": "In Write mode you are able to interact with the website on customer's behalf.",
  "pointerModeTip": "In Pointer mode customer can only see your mouse cursor and clicks. No interactivity.",
  "requestUpgradeToWrite": "Request upgrade to Write mode.",
  "downgradeToPointer": "Downgrade to pointer mode",
  "navigationDenied": "Navigation request has failed.",
  "maskedNodeTitle": "This content is visible only to customer",
  "unsupportedNodeTitle": "Some data is missing due to Co-browse limitations",
  "navigationRestricted": "Navigation is restricted in Pointer mode.",
  "modeUpgraded": "Switched to Write mode. Now you can interact with the page.",
  "modeDowngraded": "Switched to Pointer mode. Customer can only see your cursor and clicks.",
  "writeModeRequested": "Requested Write mode. Waiting for customer approval.",
  "modeUpgradeDenied": "Customer declined upgrading to Write mode."
}
```