



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

Serving JSONP

Contents

- [1 Serving JSONP](#)
 - [1.1 JSONP Examples](#)

Serving JSONP

External Co-browse resources, such as [localization files](#) or [custom chat templates](#), must be served via [JSONP](#). Genesys Co-browse provides a simple way to serve resources using JSONP.

Important

This functionality is supported in Co-browse **8.5.001.05+**

To serve a resource, put the resource into the "static" Jetty webapp directory (**server/webapps/static** in Co-browse deployment) of every Co-browse node in your cluster. You can then reference the resource as `http://<COBROWSE_SERVER_URL>/static/your-resource.extension`.

Supported extensions are ***.json**, ***.html**, ***.xml**.

JSONP Examples

Example 1: Serving Localization (JSON)

Suppose you wanted to override one key in the [localization files](#) so that the title of all UI dialogs would be "My Company" instead of "Co-browse". You could accomplish this by doing the following:

1. Create a file with the following content:

```
{
  "modalTitle": "My Company"
}
```

2. Save the file with the **.json** extension. For example, **my-localization.json**.
3. Copy the file into the **server/webapps/static** folder of every Co-browse server in your cluster.
4. In your instrumentation, tell Co-browse to use this file for localization using the [Configuration API](#):

```
<script>
var _genesys = {
  cobrowse: {
    localization: 'http:<COBROWSE_URL>/static/my-localization.json'
  }
};
</script>
```

Example 2: Serving Custom Chat Templates (HTML)

Suppose you want to use a custom [chat template](#). You could do so by doing the following:

1. Download the default templates from Web Engagement or the Co-browse server. See [Template-based Customization](#).
2. Modify and save the file. For example, **myChatTemplates.html**. See examples in [ChatWidgetAPI#Customizing the User Interface](#).
3. Copy the file into the **server/webapps/static** folder of every Co-browse server in your cluster.
4. In your instrumentation, configure chat to use this template file:

```
<script>
var _genesys = {
  chat: {
    templates: 'http:<COBROWSE_URL>/static/myChatTemplates.html'
  }
};
</script>
```

Important

You must put the resource(s) on all nodes in the Co-browse cluster.

Tip

Web Engagement servers support similar functionality, see [Web Engagement Architecture#Hosting Static Resources](#).