



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

Co-browse Restrictions and Known Limitations

12/14/2025

Co-browse Restrictions and Known Limitations

Contents

- **1 Co-browse Restrictions and Known Limitations**
 - 1.1 Co-browse Server should be restarted after network settings are updated for Co-browse Server application
 - 1.2 Web Components browser feature support
 - 1.3 Support for Internet Explorer 11 is deprecated
 - 1.4 cookieFootprintReduce feature
 - 1.5 Co-browse Server needs to be restarted when Configuration Server does not see the disconnection with it
 - 1.6 Visibility of restricted DOM element frame
 - 1.7 Cookie based stickiness might not work properly
 - 1.8 Co-browse No Longer Supports Interaction Workspace 8.1
 - 1.9 Co-browse Must Be Deployed on the Same Second-level Domain as the Website
 - 1.10 Synchronization of Interactions with Browser Plugins is Not Supported
 - 1.11 Web Components are Not Supported
 - 1.12 Some Obsolete Web Techniques are Not Supported
 - 1.13 Some HTML5 Features are Not Supported
 - 1.14 Some Pseudo CSS Selectors are Not Supported
 - 1.15 Customer Representative Can Handle Only One Co-browse Session at a Time
 - 1.16 Conferencing, Consultation and Transfer are Not Supported for Co-browse Sessions
 - 1.17 Mouse synchronization
 - 1.18 Representation of Dynamically Shown List Items May be Partially Broken on Agent side Browsers Using IE11
 - 1.19 Co-browse Fails Silently on Internet Explorer 8, 9, 10
 - 1.20 Static resource synchronization behind authentication
 - 1.21 Static resources behind authentication prevents a DNS failover fallout
 - 1.22 Configuring Co-browse for multibyte character encodings
 - 1.23 Scaling effects are not supported
 - 1.24 Different co-browse sessions in the same browser instance are not supported for the user

- 1.25 Stricter policy with HTTPS-based localization
- 1.26 Co-browse handles the agent side CSS resources only in UTF-8

Co-browse Server should be restarted after network settings are updated for Co-browse Server application

If the network settings in the Co-browse Server application configuration is to be changed dynamically when the application is running, the Co-browse Server application should be restarted after the settings update.

Web Components browser feature support

- Web components with shadowDom can be synchronized with an agent only if shadowDom was created as {mode:open}. For components {mode:closed}, it is only possible to fill a component space with the same size mock.
- Input synchronization inside web components works only from the master page to an agent. Input will not be synchronized from an agent's component reflection to the master page.
- Though DOM-restrictions with jQuery syntax work for web components, it could be an issue that a component is restricted on master or agent side. A restricted image space might have a different background color or frame inside the web component on agent side.

Support for Internet Explorer 11 is deprecated

Starting from 9.0.014.xx, support for Internet Explorer 11 is deprecated and will be dropped in the future releases.

cookieFootprintReduce feature

cookieFootprintReduce feature might not work properly with some Content Security Policy settings because it uses hidden iframe loading to solve Cross-Domain localStorage restrictions. cookieFootprintReduce feature is not supported by Internet Explorer 11. Co-browse works in the Internet Explorer 11 without this feature.

Co-browse Server needs to be restarted when Configuration Server does not see the disconnection with it

In some rare cases when ADDP is configured for Co-browse Server when Co-browse Server is connected to Configuration Server, Configuration Server cannot observe the disconnection that happens between it and Co-browse Server. In this case, Co-browse Server sees the disconnection and constantly tries to restore the connection. Configuration Server is, however, unaware of the disconnection and can interpret requests to reconnect as attempts to make new connection for the second instance of Co-browse Server.

The only way to restore that connection is to restart the Co-browse Server nodes. Automatic server restart can be implemented on the customer side by checking /cobrowse/health or /cobrowse/health/all REST API from each Co-browse instance.

Visibility of restricted DOM element frame

Restricted DOM element frames may be visible in the Agent UI even when the base element is covered by another layer or element if the `display="none"` attribute is not set. Genesys recommends setting the `display="none"` attribute for these elements.

Cookie based stickiness might not work properly

- A Co-browse session cannot be established when all the following conditions are true:
 - The Co-browse URL domain differs from the client's website domain.
 - Third-party cookie-tracking privacy settings is suppressed in the client browser.
 - Headers `Access-Control-Expose-Headers` and `Access-Control-Allow-Headers` are not set in the website response.

To fix this issue, set the headers `Access-Control-Expose-Headers` and `Access-Control-Allow-Headers` with the same domain as the Co-browse URL domain.

Co-browse No Longer Supports Interaction Workspace 8.1

Starting with release **8.5.0**, Interaction Workspace 8.1.x is no longer supported by Genesys Co-browse.

Co-browse Must Be Deployed on the Same Second-level Domain as the Website

Due to some browsers' strict cookie policies, Genesys highly recommends that you host the Load Balancer on the same domain as the website or on one of its sub-domains. Otherwise, chat and Co-browse stickiness cookies may be rejected as third-party and the solution will not work. Users will not be able to start chat nor begin co-browsing.

Synchronization of Interactions with Browser Plugins is Not Supported

By design, synchronization of interactions with browser plugins is not supported. HTML markup managed by browser plugins (Flash, Java, Silverlight, ActiveX, etc.) is synchronized as is and may be displayed if both browsers support the plugin.

Web Components are Not Supported

Custom Elements and Shadow DOM are not supported.

Some Obsolete Web Techniques are Not Supported

- Quirks Mode, Almost Standards Mode — Co-browse always uses Full Standards Mode when rendering the customer's website on the agent side and requires the valid document type definition to be set on the customer's pages. Technically, it means that `doctype` is always set to `<!DOCTYPE html>` when rendering anything on the agent side. Pages in Quirks Mode or Almost Standards Mode are not supported.

- Framesets — Obsolete technology is not supported.

Some HTML5 Features are Not Supported

The following HTML5 features are not supported:

- Canvas
- WebGL
- HTML5 audio and video—HTML markup is synchronized. Synchronization of playing, pausing, etc. is not supported.

SVG

Genesys Co-browse **8.5.1** adds support for SVG in co-browse sessions.

Some Pseudo CSS Selectors are Not Supported

The following pseudo selectors are not supported:

- `:visited`
- `:target`
- `:active`
- `:focus`
- `:fullscreen`
- `:scope`
- CSS3 form selectors such as `:valid` and `:required`

For other pseudo selectors (For example, `:dir()`, `:read-only`, and `:nth-last-of-type()`), synchronization depends on the browsers. The pseudo-selector will be synchronized only if it is supported by both browsers.

Important

The `:hover` selector is supported. For more information, see [JavaScript Configuration API—css](#).

Customer Representative Can Handle Only One Co-browse Session at a Time

Starting with Co-browse **8.5.003**, an agent is by default **limited to handling one co-browse session at a time**. You can allow an agent to handle more than one co-browse session at a time by configuring co-browse session limitations in Workspace Desktop Edition.

Conferencing, Consultation and Transfer are Not Supported for Co-browse

Sessions

As Co-browse sessions are not interactions like chat or voice, these standard operations for interactions are not currently supported for Co-browse.

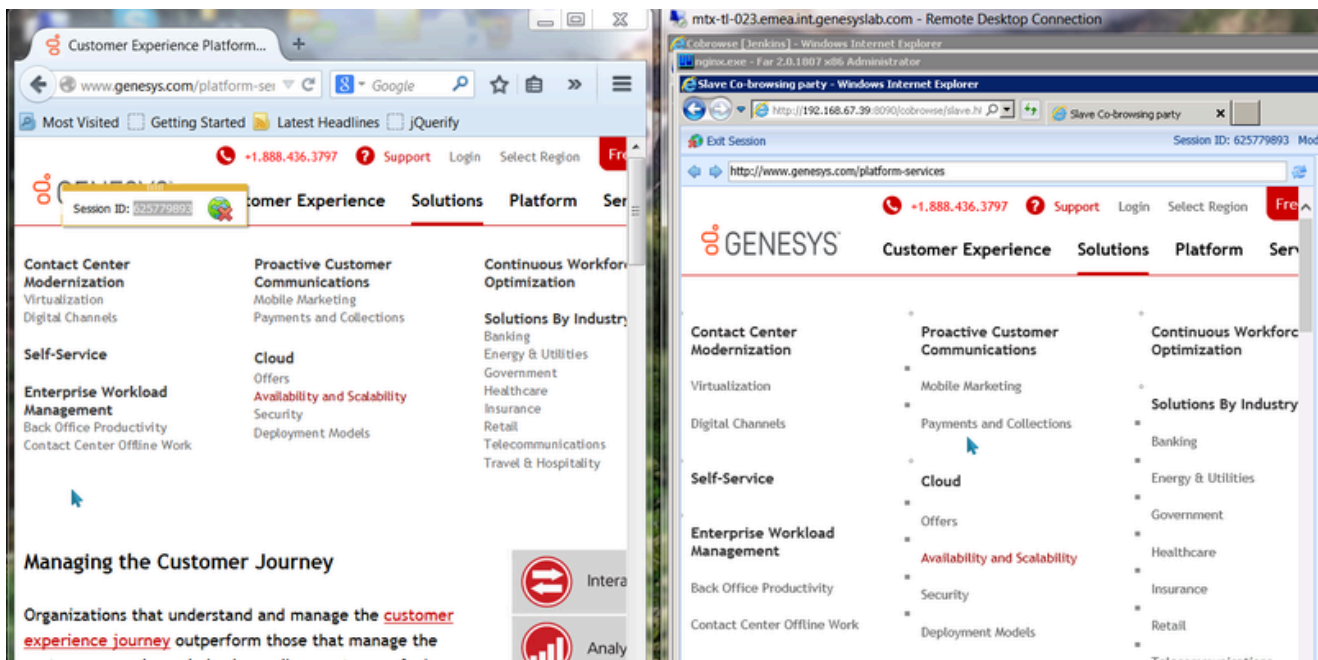
Mouse synchronization

Mouse positions may differ slightly on the Customer side and Agent side if websites render differently in different browsers.

Representation of Dynamically Shown List Items May be Partially Broken on Agent side Browsers Using IE11

Issues may arise when IE11 is used as a Agent side browser on a website with dynamically shown/hidden sub-menus.

Example:



Workaround

1. Create a CSS file with a rule that sets the fixed display property of submenu list items.
Example:

```
.my-submenu li {  
    display: block !important;  
}
```

2. Host this file somewhere that is accessible via HTTP

3. Specify the URL of the file in the `slave.cssPatchUrl` option in Config Server.

Co-browse Fails Silently on Internet Explorer 8, 9, 10

Co-browse will always fail silently when run on Internet Explorer 8, 9, or 10, meaning that the error message *Your browser is not supported* will not generate.

Static resource synchronization behind authentication

Static resource synchronization is supported starting in release 9.0.005.15. For information about known issues, see the [Co-browser Server 9.0.x Release Note](#).

Static resources behind authentication prevents a DNS failover fallout

With DNS Failover, Co-browse is able to properly keep active sessions until they finish naturally. However, all new sessions cannot be established successfully because the Co-browse Server web-proxy that passes resources to the agent side requires DNS to download the resources from the customer site.

Web-proxy that is affected by DNS Failover can be replaced with Static Resources Behind Authentication, which provides a server-based caching service that does not require DNS because all resources are sent from the customer client browser directly to the Co-browse Server, and then downloaded by the agent side via hash-based direct links from the Co-browse Server.

For information about known issues with Static Resources Behind Authentication (such as the hover effects), see the [Co-browser Server 9.0.x Release Note](#).

Configuring Co-browse for multibyte character encodings

In rare cases, when the co-browsed web page uses multibyte encoding like Shift JIS and some other factors combined, Co-browse may not work in Internet Explorer on certain versions of Windows. To mitigate this, configure your Load Balancer to serve the Co-browse JavaScript file (the `gcb.min.js`) in UTF-8. For Nginx, this can be done by adding the `charset UTF-8;` line to any `http`, `server`, or `location` directive. For example, your Nginx config (see [Configuring a Load Balancer for Co-browse Cluster](#)) may look like this:

```
...
location /cobrowse {
    charset UTF-8;
    ...
}
```

Scaling effects are not supported

If an end-user or an agent are scaling their browser, the scaling effect will not be transferred to the other side, however, the size of shared page area remains the same for both sides.

Different co-browse sessions in the same browser instance are not supported for the user

Co-browse does not support two different sessions in the same browser instance at the same time for

the user.

Stricter policy with HTTPS-based localization

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;
}
```

Co-browse handles the agent side CSS resources only in UTF-8

Use of multi-byte encoding like `shift_jis` on the webpage may encounter parsing issues. As per the [W3C](#) recommendation, use UTF-8 encoding.