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

Genesys Co-browse 8.5.1

12/29/2021

Table of Contents

Genesys Co-browse 8.5 Deployment Guide

Welcome to the *Genesys Co-browse 8.5 Deployment Guide*. This document introduces you to the concepts, terminology, and procedures relevant to Genesys Co-browse. See the summary of chapters below.

About Genesys Co-browse

Find out about the core features of Genesys Co-browse.

Deploy Genesys Co-browse

Find procedures to set up Genesys Co-browse.

Genesys Co-browse Reporting Templates

Find templates for real-time and historical reporting.

What is Genesys Co-browse?

Overview

Genesys Co-browse provides the ability for an agent and the end customer to browse and navigate the same web page at the same time. In a Genesys Co-browse session, both the agent and the customer share the same instance of the screen, as opposed to a conventional screen sharing application, where one of the parties sees an image of the other party's browser instance.

Components

Genesys Co-browse is composed of the following components:

- **Genesys Co-browse Server** is a server-side component that is responsible for orchestrating the co-browsing activities between the end consumer and the agent.
- **Genesys Co-browse Plug-in for Workspace Desktop Edition** provides co-browsing functionality for Workspace Desktop Edition users.
- **Genesys Co-browse Sample Reporting Templates** provides configuration files and reporting templates for getting real-time and historical statistic data.
- **Genesys Co-browse JavaScript** includes Chat and Co-browse functionality. You should add this component to the pages on your website where you want to enable co-browsing. See Website Instrumentation.

Features

Genesys Co-browse includes the following features:

- Active participation—both the agent and the customer have the ability to take control.
- Browsing always happens on the customer side.
- Administrators are able to restrict what the agent can do and see on the web page. The customer can easily identify which fields are masked from the agent. Administrators can easily specify which DOM elements (buttons, check boxes, and so on) the agent must not be able to control.
- Pointer Mode and Write Mode—Co-browse sessions begin in Pointer Mode where the agent cannot enter information for the customer. The agent may send the customer a request to enter Write Mode where the agent can enter information for the customer. The customer must agree to enter Write Mode. You may also disable Write Mode and make all sessions Pointer Mode only. DOM Restrictions and Data Masking apply to both Pointer and Write Mode.
- Support for multiple browsers, cross-browser support, and same-browser support.
  Support for scenarios in which the agent and customer are using different browsers.

Support for scenarios in which the agent and customer are using different versions of the same browser.

- The customer can co-browse without downloading or installing any plug-ins.
- Co-browse keeps an agent's internal traffic contained within the internal network while still allowing the customer traffic to flow through the external network.

Browser Support

See Tested Browsers for a list of Genesys-tested browsers for web and mobile.

> **Warning**
>
> We strongly advise against IE Conditional Comments.

> **Important**
>
> Interaction Workspace uses *only* Internet Explorer as the embedded browser for working with Co-browse sessions.

Hardware Requirements

See Sizing Information for details.

Related Components

Genesys Co-browse interacts with the following Genesys Products:

- Workspace Desktop Edition — The Genesys Co-browse Plug-in for Workspace Desktop Edition is required to interface Genesys Workspace Desktop Edition with Genesys Co-browse. This plug-in enables the agent to join and terminate a co-browsing session with a customer.
- Genesys Widgets—a set of productized widgets that are optimized for use with desktop and mobile web clients, and which are based on the GMS APIs. Genesys Widgets provide for an easy integration with Co-browse, allowing you to proactively serve these widgets to your web-based customers.

For a full list of related components see Related Components.

> **Important**
>
> For supported operating systems and a list of other required/compatible non-Genesys components, see Genesys Co-browse in the *Genesys Supported Operating Environment Reference Guide*.

Genesys Co-browse 8.1.3+ and Previous Co-Browsing Solutions

The Genesys Co-browse 8.1.3+ solution should not be associated with the Web API Cobrowse Samples. These samples work with the old KANA-based Co-Browsing Server and do not work with this new Co-browse solution; however, you may use a chat interaction started from the Web API Chat Samples to initiate a new Co-browse session from an instrumented page with an agent.

> **Important**
>
> Genesys strongly recommends that you use the Co-browse chat widget to initiate a Co-browse session with an agent. This chat widget is designed to work with the new Co-browse solution in the most optimal way. Agents do not even need to paste the Co-browse session ID manually into their screen - the Co-browse page is opened automatically once the customer clicks the "Co-browsing" button during a live chat.

For more information about how to work with the Co-browse chat widget, see the following sections:

- Initiating a Co-browse session from an integrated chat
- Genesys Co-browse and Chat

For more information about how to work with external chats like the Web API Chat Samples, see:

- Initiating a co-browse session from a voice call or external chat without integration
- External Chat without Integration

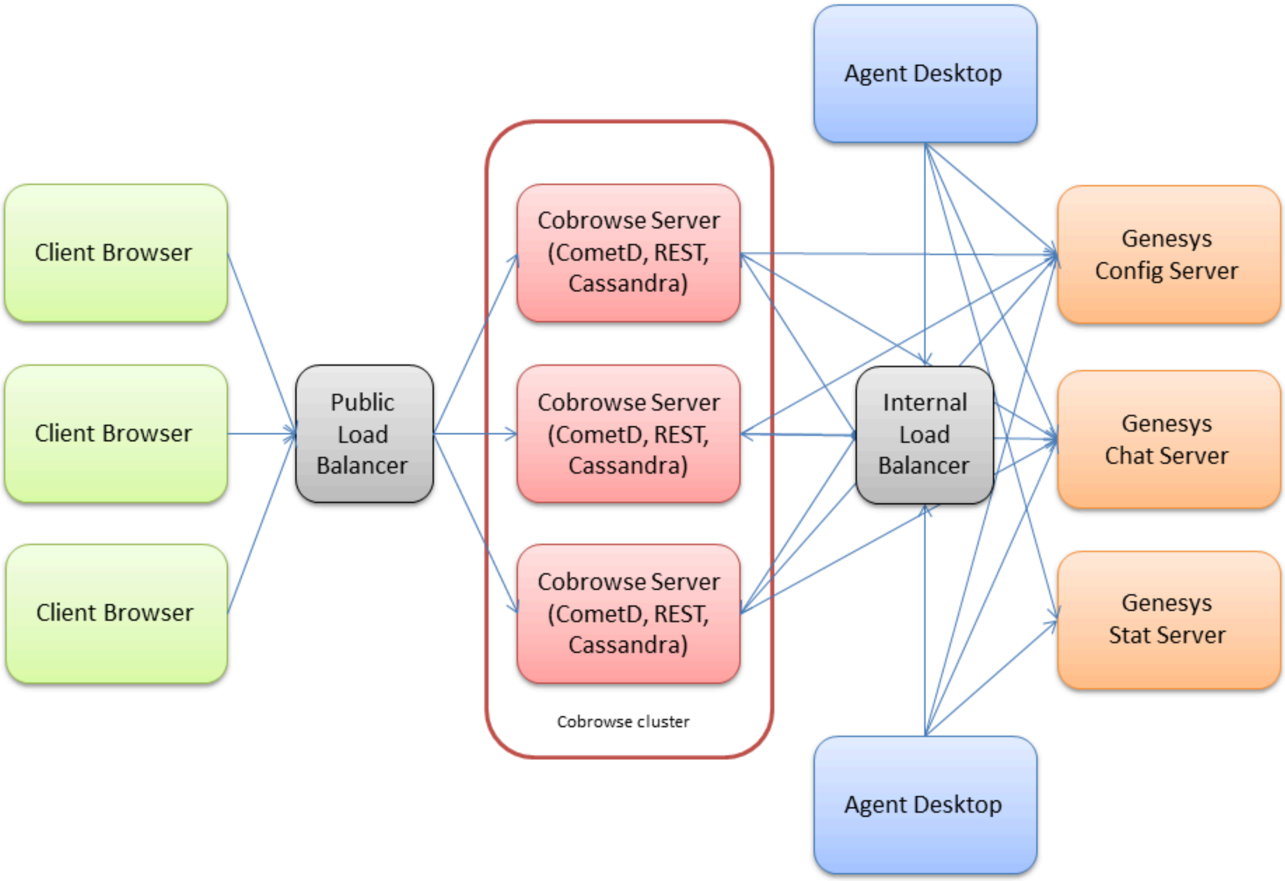
Restrictions and Known Limitations

See Co-browse Restrictions and Known Limitations.

Co-browse Architecture

Architecture Diagram

The following diagram shows an example of a three node cluster implementation of Co-browse:

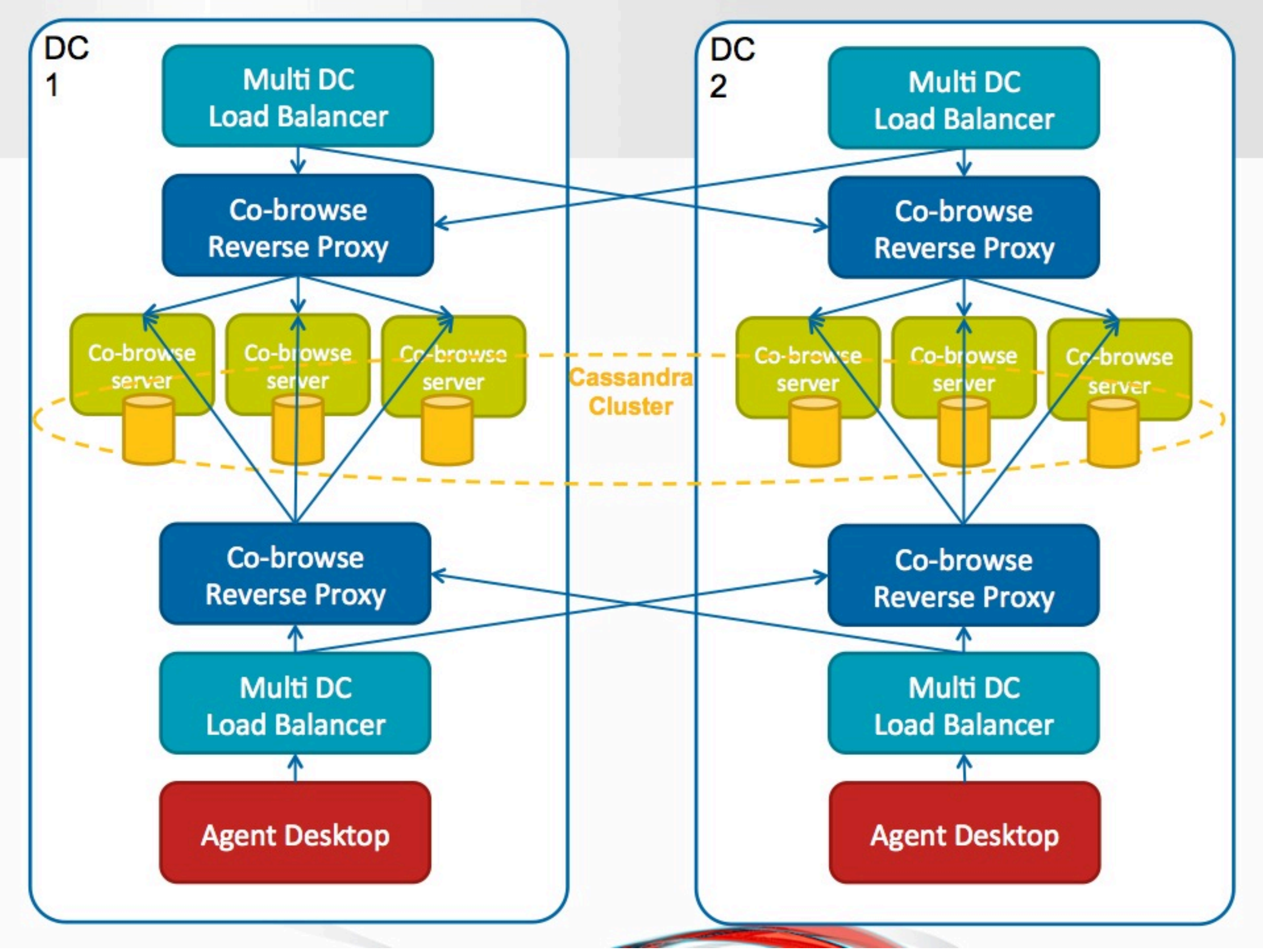


- Each Co-browse server has the same role in the cluster and must be identically configured.
- Each Co-browse server hosts the following:
  1. CometD server with Co-browse and Web Chat Services
  2. Live and Historical session REST APIs
  3. Embedded Cassandra Node
- A Co-browse cluster is formed through a load balance/reverse proxy. See Cluster Configuration.
- A Cassandra cluster is formed through appropriate cassandra.yaml configuration. See cbdb Options.

- Co-browse servers are usually deployed in the back end server environment and given access through a load balancer/reverse proxy.
- Internal Co-browse server resources are secured at the network level by not being exposed via the Public Load Balancer. Co-browse Server resources are exposed to internal applications via the Internal Load Balancer.
- Co-browse server web chat function acts as a gateway to Genesys Chat Server.
- Agent Desktops connect to the Co-browser server to receive web page representations from the Client Browser.
- The Co-browse plugin for Genesys Workspace Desktop Edition (Agent Desktop) reports Co-browse statistics via attached data on primary interactions.
- The Client Browser initiates a Co-browse session and transmits web page content to the Agent Desktop through the Co-browse Server.

Architecture with Multiple Data Centers

The following diagram shows an example of a two data center implementation of Co-browse:



A Co-browse solution can be deployed across multiple Data Centers to provide higher availability. Each Data Center deployment acts independently except for the following points:

- The Cassandra cluster is configured in multi-data center mode (http://www.datastax.com/dev/blog/deploying-cassandra-across-multiple-data-centers) to enable data sharing across multiple data centers.
- If a local Co-browse cluster does not respond, a multiple DC Load Balancers could be configured to forward requests to a remote Co-browse Reverse Proxy. Multi DC balancing logic can also be injected directly into a Co-browse Reverse Proxy.

Genesys Co-browse Sessions

> **Important**
>
> Co-browse sessions are not interactions like chat and voice interactions. Co-browse sessions take place *on top* of a primary interaction like chat or voice and attach user data for reporting. Co-browse sessions do not support operations that are standard for Genesys interactions like transfer and conference.

A session is initiated when a customer requests to co-browse. The session stays idle until the agent joins. Then the session is considered to be active. The session ends when one of the parties (the customer or the agent) exits. It is not possible to re-join a co-browse session. If one party exits accidentally, a new session must be initiated. Starting with Co-browse 8.5.003, an agent is by default limited to handling one co-browse session at a time.

Session Identifiers

Each live session has two identifiers that can be used to track the session:

- Session access token (Session ID)—A sequence of nine digits that is applicable only to live sessions.
- History session identifier (UUID)—A session identifier in the database.

Starting and Stopping a Session

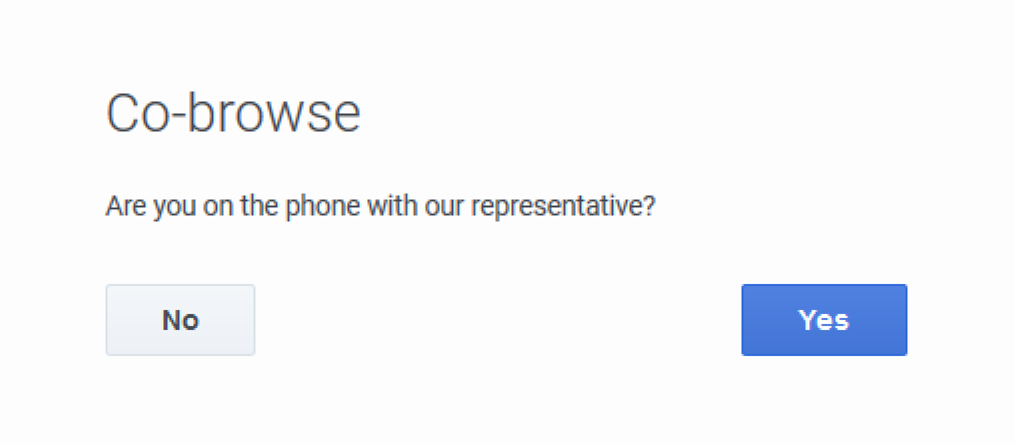
A co-browse session can only be initiated by a customer. An agent does not have the option or ability to send a co-browse request to a customer. This provides greater security to the customer. In order to initiate a co-browse session, the customer must already be engaged in an interaction with an agent, be it a voice call or a chat.

When the session is established, the agent's browser displays a view of the customer's browser. The view the agent sees is loaded from Genesys Co-browse Server, so only certain images might be loaded from the original website. The agent is not a client of the website. All actions taken by the agent are passed onto and "replayed" on the customer's side.

Initiating a co-browse session from a voice call or external chat without integration

If a customer and agent are engaged in a voice call or external chat without integration, a co-browse session can be initiated by the customer if the need arises. For example, the agent might be trying to walk the customer through how to submit a specific form, but the customer is having issues understanding where the agent is directing him or her to go on the page. In this scenario, the agent might suggest they engage in a co-browse session. While the agent can verbally suggest a co-browse session, the customer is the one who must *initiate* the session.

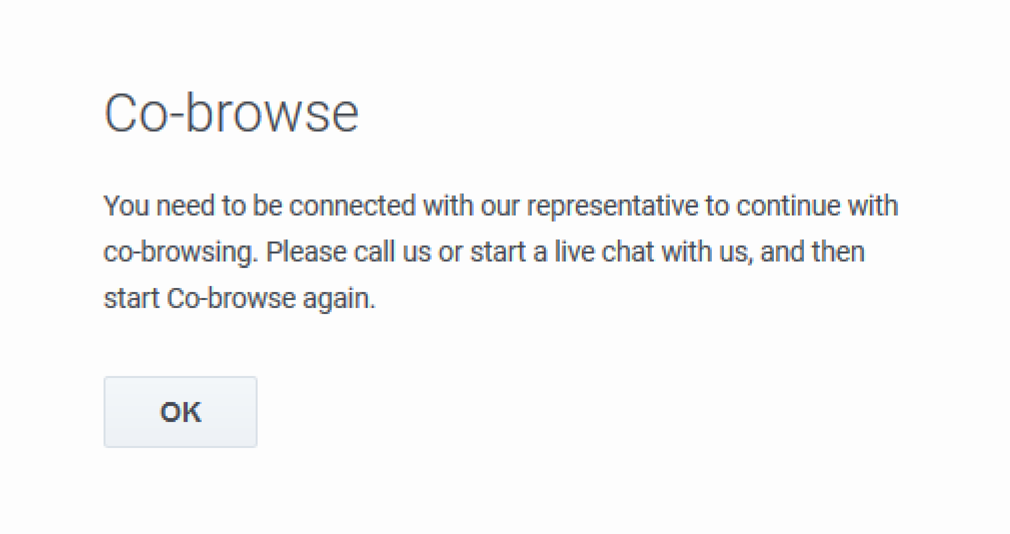
By default, there is a "Co-browsing" button on the left side of every web page that supports co-browsing. Note that the location on the page can vary, depending on configuration. When the customer clicks this button, they are presented with a message window asking them to confirm that they are engaged in a voice call with a representative.


Co-browse

Are you on the phone with our representative?

Co-browse message

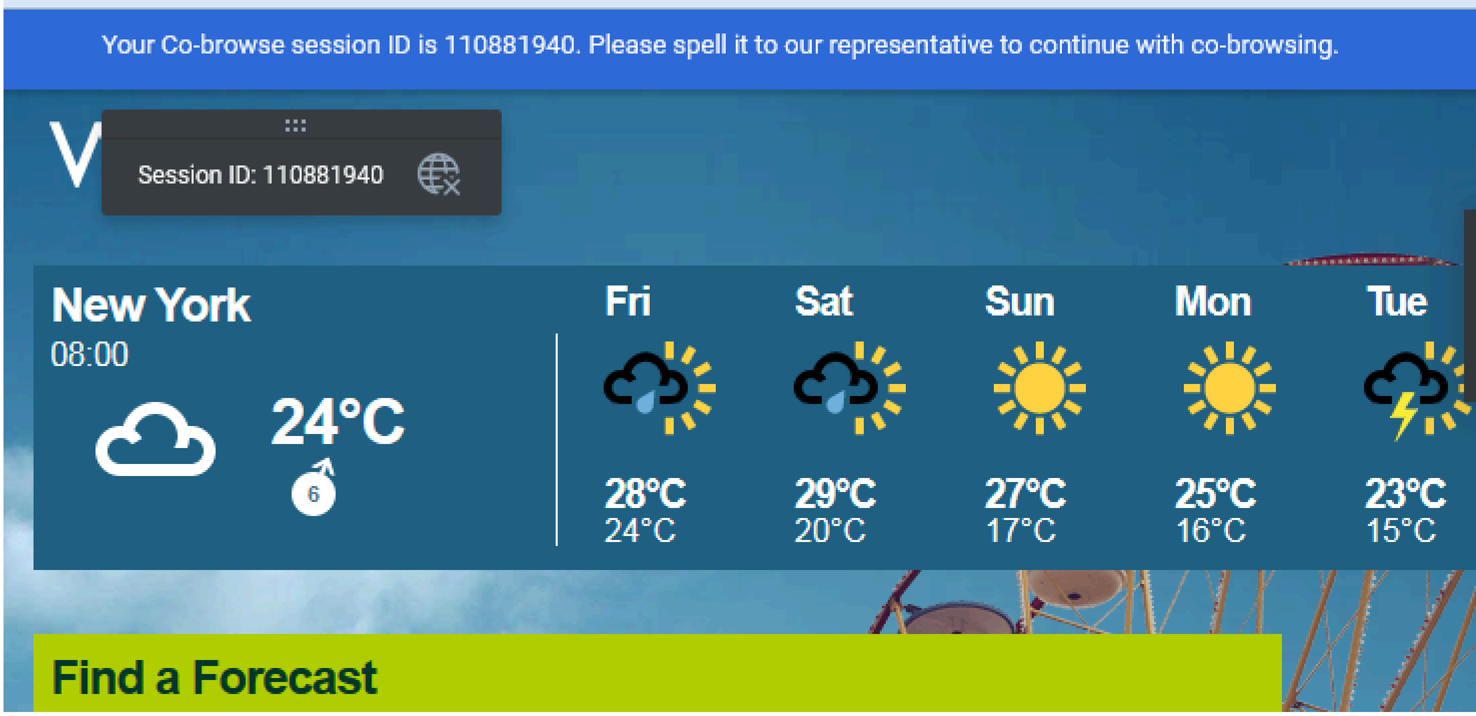
If the customer selects "No", a new message advises them to either initiate a voice call or a chat in order to co-browse.


Co-browse

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.

The customer must initiate a voice call or chat

If the customer selects "Yes", a numeric session identifier appears on the customer's screen.

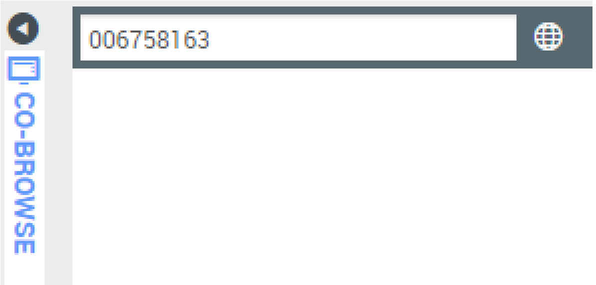


Your Co-browse session ID is 110881940. Please spell it to our representative to continue with co-browsing.

Session ID: 110881940

Session identifier

This identifier can then be read to the agent over the phone or the customer might have to send the session ID through their external chat window. The agent enters the session identifier in the appropriate field in Interaction Workspace, and then the customer's browser is displayed in the agent's view. There is no need to navigate to the web page the customer is viewing; the session identifier ensures the exact page is embedded in Interaction Workspace for the agent. The customer is notified on his or her screen that the session has been established.



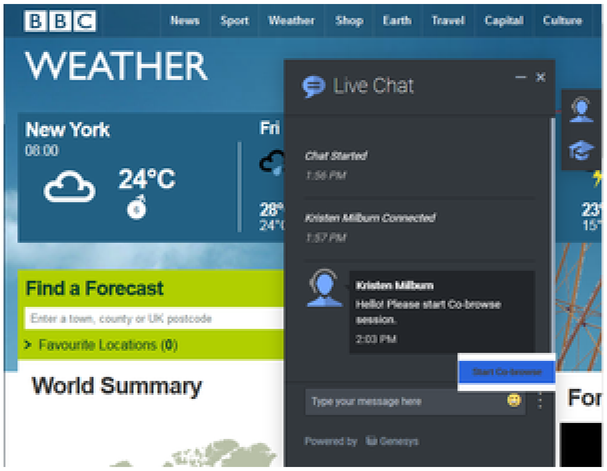
006758163

Session identifier, Agent view

Initiating a co-browse session from Genesys Widgets

A customer can also initiate a co-browse session through a Genesys Widget integrated into the website. You can enable Genesys Co-browse in several Genesys Widgets, for example:

Initiating a co-browse session from the Web Chat Widget

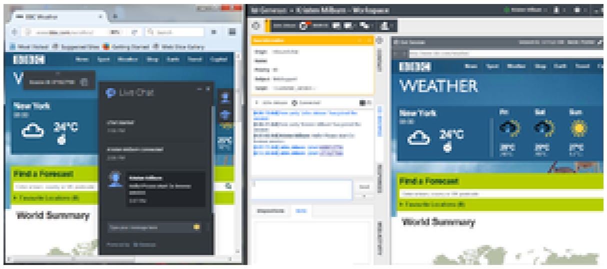


Chat Started
1:56 PM

Kristen Milburn Connected
1:57 PM

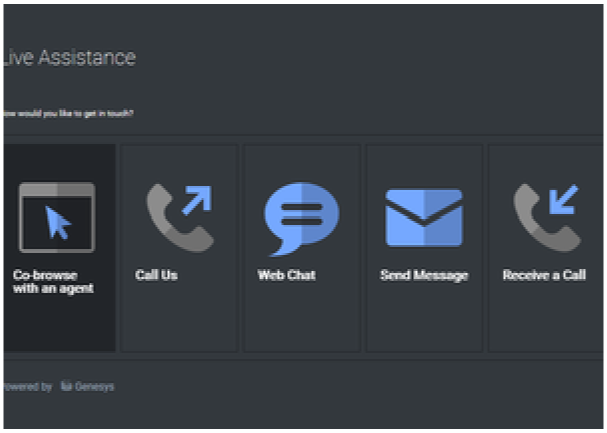
Kristen Milburn
Hello! Please start Co-browse session.
2:03 PM

Starting a co-browse session from the Web Chat Widget.



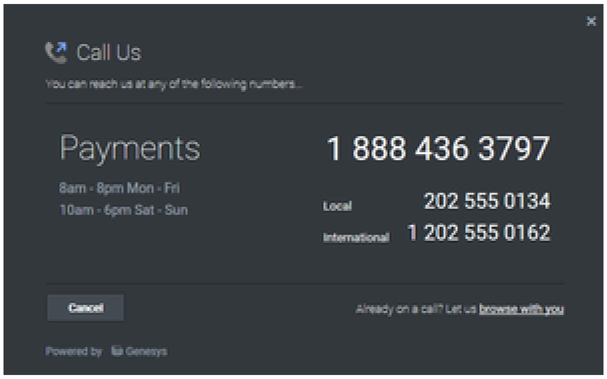
Customer and agent view of a co-browse session started from the Web Chat Widget.

Initiating a co-browse session from the ChannelSelector Widget



Starting a co-browse session from the ChannelSelector Widget.

Initiating a co-browse session from the CallUs Widget


Call Us

You can reach us at any of the following numbers...

Payments

8am - 8pm Mon - Fri
10am - 6pm Sat - Sun

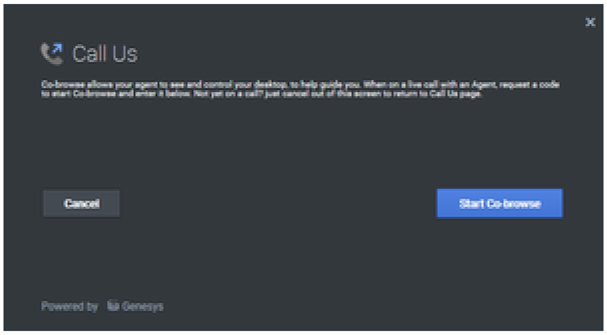
1 888 436 3797

Local 202 555 0134

International 1 202 555 0162

Already on a call? Let us browse with you

CallUs Widget with co-browse option.

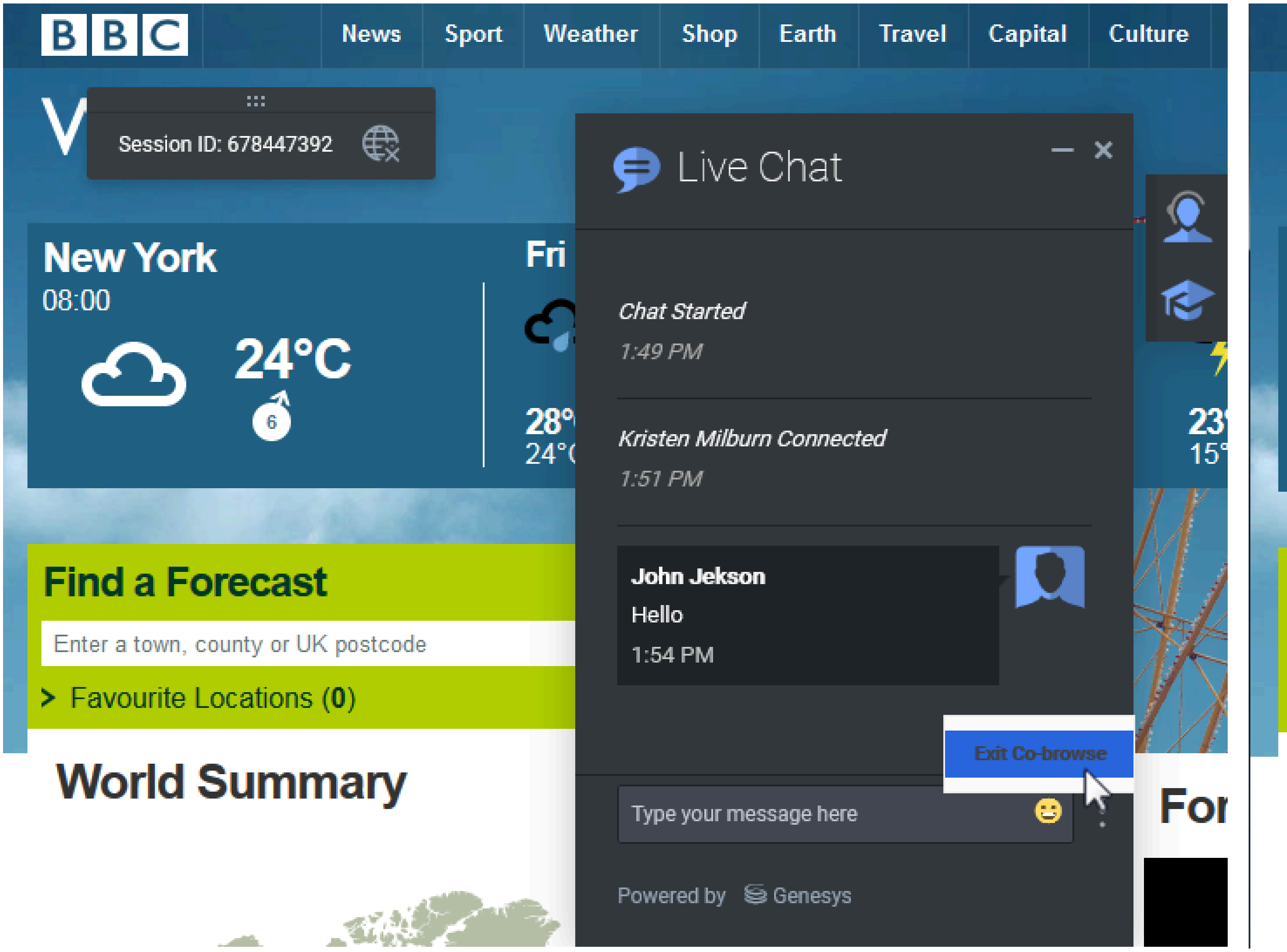

Call Us

Co-browse allows your agent to see and control your desktop, to help guide you. When on a live call with an Agent, request a code to start Co-browse and enter it below. Not yet on a call? just cancel out of this screen to return to Call Us page.

Starting a co-browse session from the CallUs Widget.

Stopping a co-browse session

Once a co-browse session has been established, both parties have the ability to terminate the session. At any time, either party may click the "Exit Co-browse session" button (again, the name and location of this button can vary).



Chat Started
1:49 PM

Kristen Milburn Connected
1:51 PM

John Jekson
Hello
1:54 PM

Exiting a co-browse session

The other party will be notified that the session has ended, and the agent's browser will no longer display a view of the customer's browser. Also, if the primary interaction (chat or voice call) is terminated, the co-browse session will be terminated automatically. Sessions can also terminate due to inactivity, after a pre-configured timeout expires. Likewise, if the agent closes their browser, or navigates to a third-party website, the session will terminate if the agent does not return back to the session page within the pre-configured timeout.

Once a session has been terminated, it cannot be reactivated. If the session was deactivated accidentally, a new session has to be initiated, with a new session identifier.

Participating in a Co-browse Session

Once a co-browse session begins, the agent can use his or her mouse pointer to guide the customer through the web site. Agents start co-browse sessions in *Pointer Mode*. In Pointer Mode, the customer and the agent can see each other's mouse pointer but the agent can not enter any information into the web page, click buttons, or navigate the customer's browser. If the agent needs to enter information into the web page or to navigate the browser, he or she can send the customer a request to switch the co-browse session to *Write Mode*. For more information on Pointer Mode and Write Mode, see Pointer Mode and Write Mode.

All actions (mouse clicks, key presses, and so on) are actually performed on the customer side. Any actions taken by the agent are sent to the customer's browser. This ensures a secure approach, as all browsing is done on one side—the customer's side. This approach also provides for greater performance and a more seamless customer experience. Each participant can see the other participant's mouse movements as well. This enables an agent to point to specific sections on the web page to help direct the customer through their task.

Administrators can limit which fields are visible to and editable by the agent. Some fields might be grayed out entirely, and some might have the data masked. For example, administrators might choose to hide the customer's password, Social Security Information, and so on from the agent. The customer can easily identify which information is hidden from the agent. By default, all `Submit` buttons are deactivated for the agent. If he or she clicks on a `Submit` button, nothing happens. The customer always has permission to submit any web forms, just as they would while browsing normally.

One-Session Agent Limitation

By default, agents are prohibited from handling more than one co-browsing session at the same time. Starting with Genesys Co-browse release 8.5.003.04, you can disable one-session limitations and configure the number of simultaneous co-browsing sessions an agent can participate in with the `agentSessionsLimit` option in the `cobrowse` section of the Workspace Desktop Edition application.

When an agent is busy with a co-browsing session and a session limitation is enabled, other customers can still start additional Co-browse sessions from their browsers but the sessions immediately end with a configurable notification explaining the agent is currently busy with another Co-browse session. See Localization to configure this message.

Note that in Workspace Desktop Edition you can override any option on the agent group and agent levels, WDE Configuration Options and Annexes.

Stickiness

Genesys Co-browse sessions are *sticky*. This means that all requests from the customer and agent sides have to be routed (*stick*) to the same Co-browse node within a given session.

Although the stickiness principles are mostly important for load balancing, the Co-browse application adheres to them even in a single-node setup (for example, in a demo or test environment). Moreover, if you use URL-based stickiness for the agent side (for example, when using Co-browse with Workspace Web Edition), the proper configuration is required even for the single node.

Generally, stickiness in Co-browse works like this:

1. The customer initiates a session on any server, which is routed by Load Balancer using the round-robin method or any other load-balancing technique. For more on load balancing, see Configure the Load Balancer.
2. After the session is created, the customer *sticks* to that server. All further requests must go to the server that owns the session.
3. After it has been given the session token, the agent side must figure out on which server the session was established. This is done via special request to any Co-browse node.
4. After that, all requests from the agent side must be routed to that same server.

There are two ways to achieve this stickiness:

- cookie-based
- URL-based

> **Important**
>
> Customer-side stickiness is always cookie-based.

Cookie-Based Stickiness

In a cookie-based scenario, the Co-browse application sets the **gcbSessionServer** cookie in every one of the following situations:

- When a Co-browse session is created.
- When a chat session is created.
- When an agent joins an existing session.

After the cookie is set, Load Balancer must use it to route requests to the specified node.

URL-Based Stickiness

In the URL-based stickiness, the agent side receives a public URL for the Co-browse node that owns the current session. The public URL is configured via the **serverUrl** configuration option. After receiving the URL, the agent application routes all further requests to that URL. If **serverUrl** is configured for URL-stickiness, the agent side always uses the URL instead of cookies for stickiness.

Important

To avoid making co-browse nodes publicly accessible, you can hide them behind the Load Balancer and differentiate them, for example, by a query parameter.

For example, the first server might have the URL `http://<load-balancer>?co-browse-node-id=1`, and the second might have the URL `http://<load-balancer>?co-browse-node-id=2`. The Load Balancer then would route the requests to the corresponding nodes.

Warning

Genesys Workspace Web Edition only supports URL-based stickiness for Co-browse. If you use it for co-browsing, you must configure the **serverUrl** option.

CSS Synchronization

This article gives an overview of how Genesys Co-browse synchronizes CSS between the customer and agent browsers.

Why does Co-browse need to synchronize CSS?

When a customer and agent are in a Co-browsing session, Co-browse tracks DOM-based and CSS-based changes in each browser and replicates changes from one browser to the other. All DOM-based changes pass from one browser to the other through the Co-browse server. Examples of DOM-based changes include creating new elements in the web page and adding or removing attributes from an element. To replicate CSS-based changes in the browser, Co-browse must make sure both browsers use the same CSS rules even when the customer and agent use different browsers. CSS-based changes can include drop-down menus and other hover events.

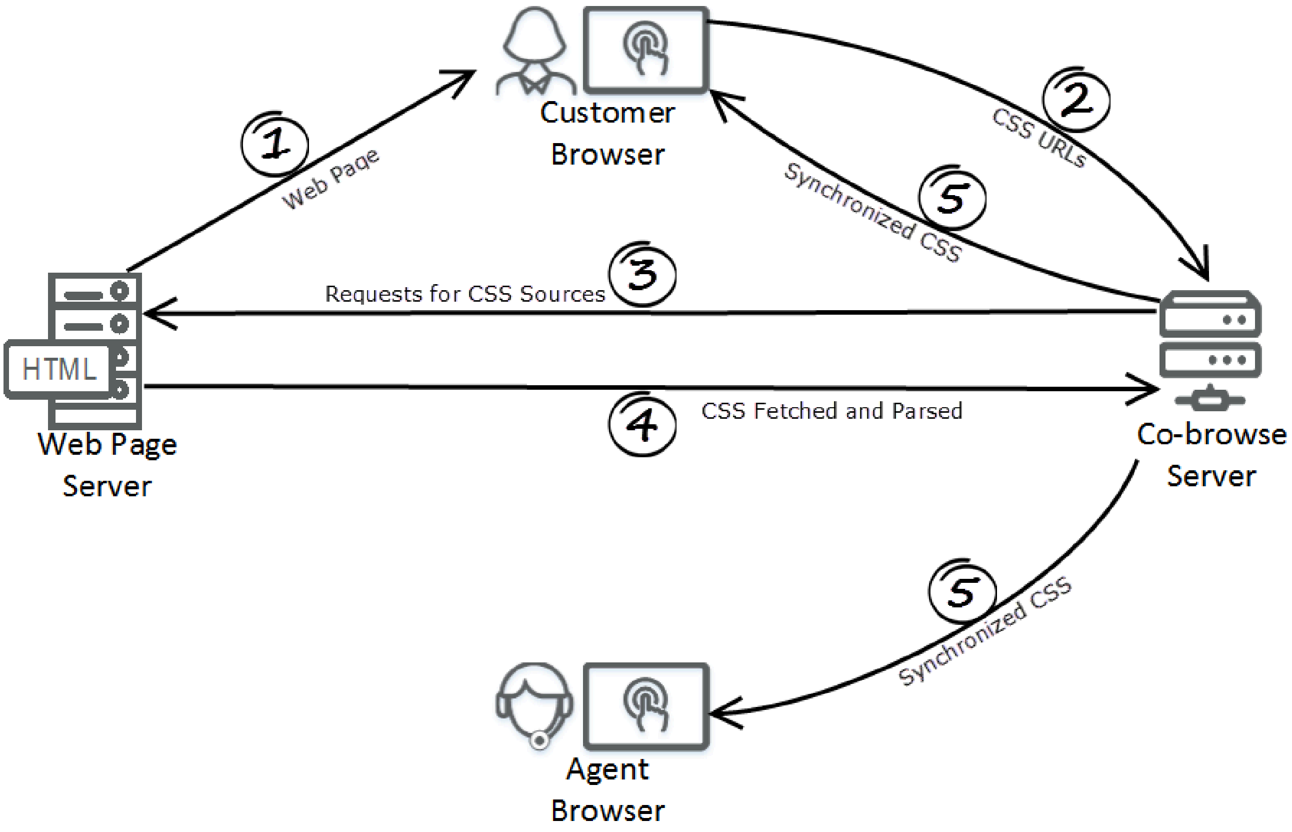
Synchronizing Browser Events

Some CSS-based changes depend on browser events that the server can not push from one browser to the other. By synchronizing CSS between the customer and agent, Co-browse server can replicate browser events it can not push. For example, the customer and agent browsers each have their own `mouseover` event that fires when the mouse pointer hovers over a web page element. Without CSS synchronization, CSS-based changes that fire based on the mouse pointer will show in one browser but not the other. To synchronize hover events, Co-browse server parses the web page CSS and adds a DOM-based pseudo-hover that fires the hover event on both browsers.

CSS Synchronization Architecture

The following diagram describes how Co-browse synchronizes CSS between the customer browser and the agent browser:

1. The customer opens a website and starts a Co-browsing session with an agent.
2. Co-browse server reads the CSS sources from the customer's view of the web page.
3. Co-browse server fetches all the required CSS stylesheets.
4. Co-browse parses the CSS stylesheets and adds additonal Co-browse specific CSS.
5. Co-browse sends the synchornized CSS stylesheet to the agent and customer browsers.



Configuring CSS Synchronization

The **css** option of the JavaScript Configuration API manages CSS synchronization. Genesys recommends using the `server` strategy and the **css** option is set to `server` by default. In some edge cases, changing the **css** option may produce better CSS synchronization results. For more about improving CSS synchronization, see the CSS synchronization section of the troubleshooting page.

Synchronizing CSS Through a Secure Zone

If you deploy Co-browse into a secure zone like a DMZ or local intranet, you must make sure the Co-browse server can still access your public web page by configuring a Forward Proxy. Otherwise, Co-browse will not be able to synchronize CSS and the agent side may not properly render.

Troubleshooting CSS Synchronization

If some or all of the content of your website is not properly rendered on the agent side, it is most likely a CSS synchronization problem. See the CSS synchronization section of the troubleshooting page.

Pointer Mode and Write Mode

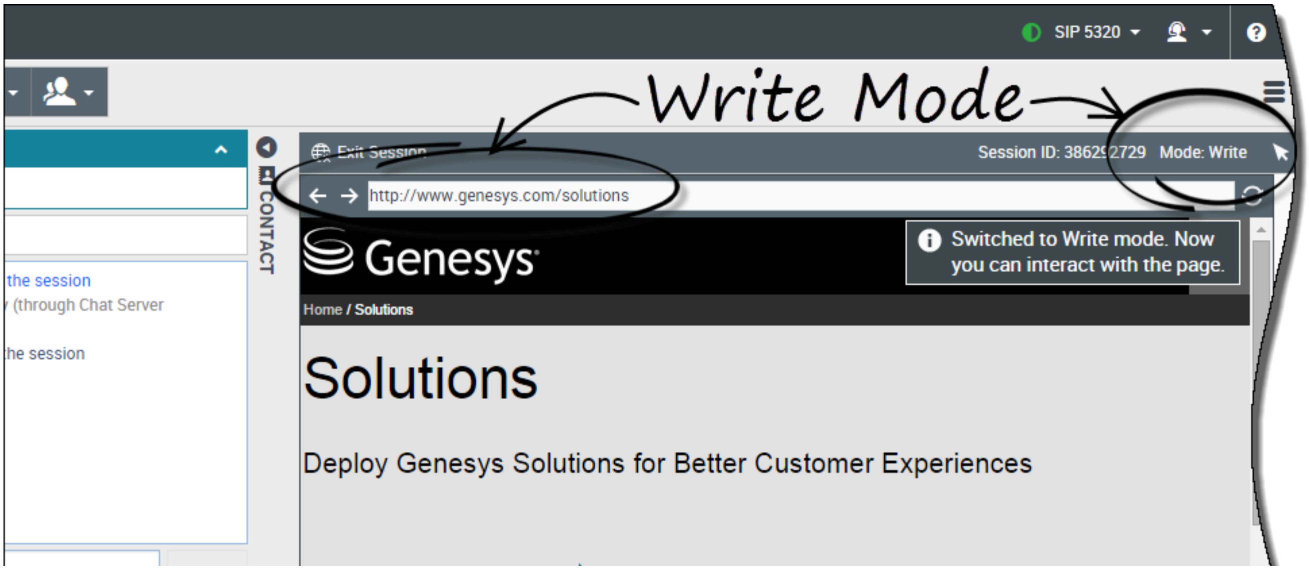
Co-browsing sessions can be in either **Pointer Mode** or **Write Mode**. Co-browse sessions begin in Pointer Mode where the agent can guide the customer using his or her mouse pointer. In Pointer Mode, the agent can not enter information into the webpage or navigate the customer's browser. If the agent needs to enter information into the web page and navigate the customer's browser, he or she must send the customer a request to enter Write Mode. By having two different Co-browse modes, the customer controls how much an agent can interact with his or her browser.

Pointer Mode

While in Pointer Mode, the agent can see what the customer sees but the agent can not perform any actions in the customer's browser. The agent can not navigate, input information, or submit forms. The agent and the customer can see each other's mouse movements at all times and the agent's mouse clicks will create a red circle effect around their mouse pointer. The agent can use the red circle effect to point to specific sections on the web page and to direct the customer.

Agents *always* join a Co-browse session in Pointer Mode.

Write Mode



Session ID: 38629?2729 Mode: Write

http://www.genesys.com/solutions

Switched to Write mode. Now you can interact with the page.

the session
(through Chat Server
he session

Solutions

Deploy Genesys Solutions for Better Customer Experiences

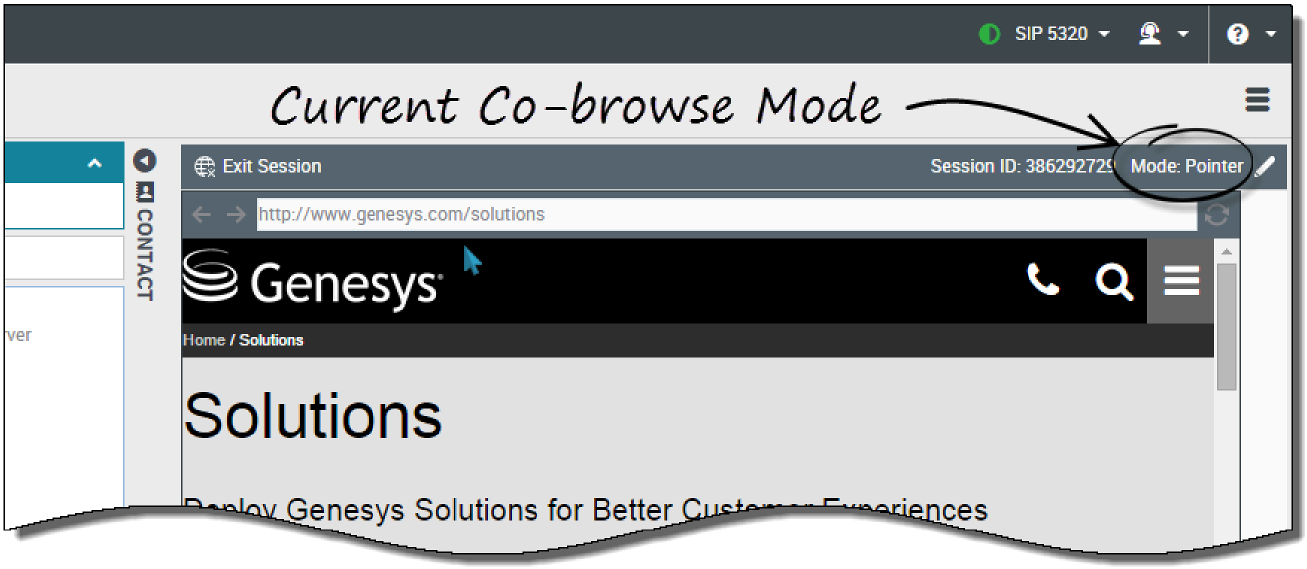
In Write Mode, both the agent and the customer can perform conventional user actions. Both can enter text and click buttons. The agent can navigate by clicking links in the web page or by using the following navigation options in Agent Dektop:

- Back and forward arrows
- URL bar
- Refresh button

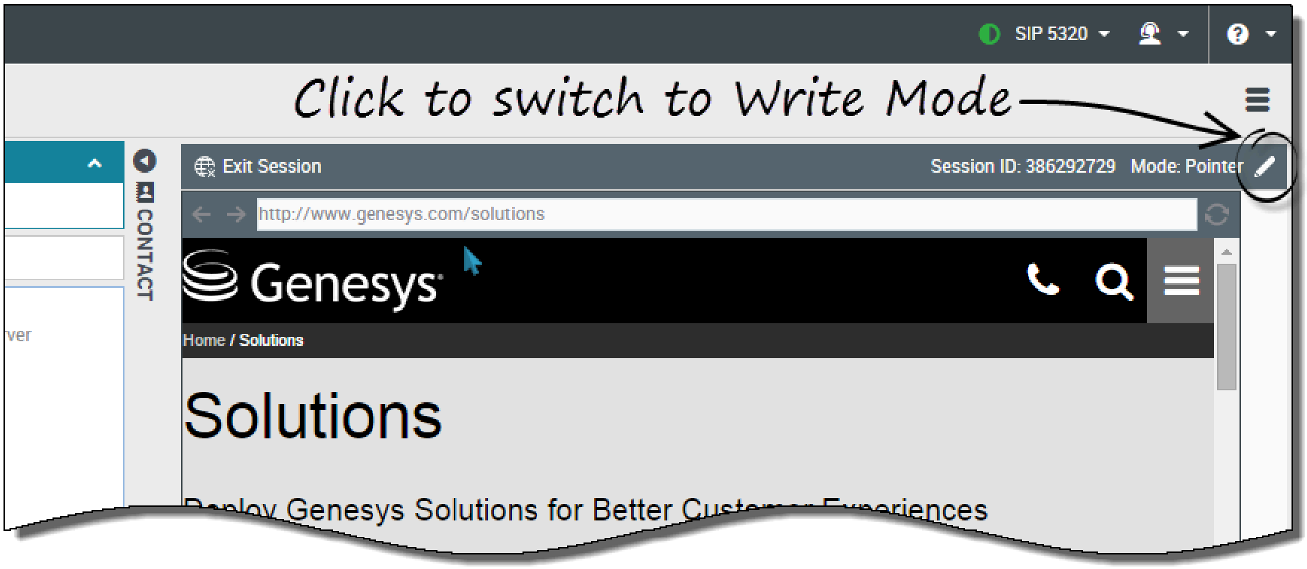
Administrators can limit which interactive elements are enabled for an agent in Write Mode. For example, administrators may choose to disable certain links. By default, all **Submit** buttons are deactivated for agents and nothing will happen when an agent clicks one. Customers can always submit forms as if they were browsing normally. For more information about restrictive interacive elements, see DOM Restrictions

> **Important**
>
> Navigation is limited to the website domain instrumented with Co-browse. If the agent tries to navigate the customer to an external page, Co-browsing will cease until the customer returns to the instrumented website. For example, if the agent enters http://example.com in the URL bar while Co-browsing http://www.genesys.com will halt the Co-browse session until the customer returns to http://www.genesys.com.

Switching to Write Mode



The top right corner of the **Co-browse** area in Agent Dektop shows the agent the current Co-browse Mode.


Click to switch to Write Mode
SIP 5320
Exit Session
Session ID: 386292729 Mode: Pointer
http://www.genesys.com/solutions
Genesys
Home / Solutions
Solutions
Deploy Genesys Solutions for Better Customer Experiences
CONTACT

To switch to Write Mode, the agent clicks the pencil icon at the top right corner of the Co-browse area.

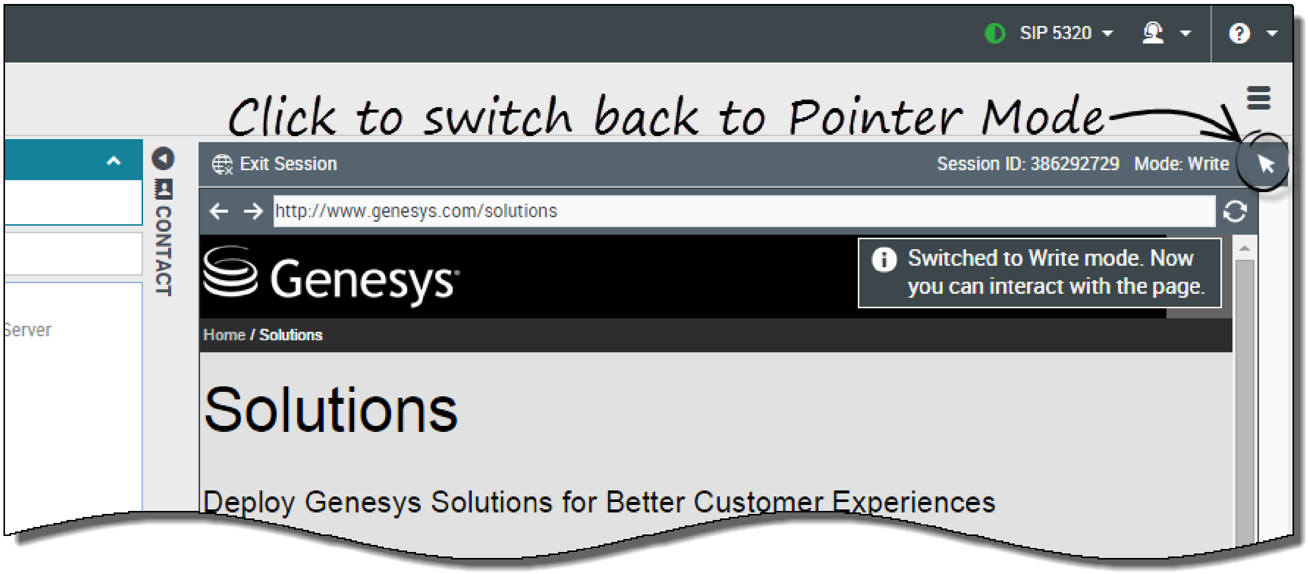
The customer will be asked to approve the switch to Write Mode. Write Mode will be enabled only if the customer approves. The agent will receive a notification about the customer's response.

If the customer approves the switch to Write Mode, the pencil icon turns into a pointer icon.

Tip

Write Mode can be completely disabled using the writeModeAllowed option. If Write Mode is disabled by administrators, agents will not see the pencil icon.

Switching Back to Pointer Mode


Click to switch back to Pointer Mode

Switched to Write mode. Now you can interact with the page.

SIP 5320
Exit Session
Session ID: 386292729 Mode: Write
http://www.genesys.com/solutions
Genesys
Home / Solutions
Solutions
Deploy Genesys Solutions for Better Customer Experiences
CONTACT
Server

To switch back to Pointer Mode, agents click the pointer icon at the top right corner of the Co-browse area.

The customer may also switch back to Pointer Mode at any time.

Configuring Write Mode

By default, Write Mode is allowed and an agent can send the customer a request to enter Write Mode. Write Mode can be disabled completely using the writeModeAllowed option.

Co-browsing in Iframes

Genesys Co-browse supports co-browsing in iframes. Behavior depends on whether the iframe is from the same domain or a different domain.

Iframes from the Same Domain

By default, agents can co-browse within iframes from the same domain or sub-domain as long as the session meets these requirements:

- The web page in the iframe contains your website instrumentation.
- The **setDocumentDomain** option is set to `true` (default value) in your website's instrumentation.

If the web page in the iframe is in your domain but not instrumented for Co-browse, the iframe does not load and the agent sees a blank page in the iframe.

Iframes from a Different Domain

By default, an iframe pointing to a webpage from a different domain loads on the customer side but not on the agent side. You can enable iframes from specific domains using the allowedThirdPartyDomains option. Once you add a third-party domain to this option, iframes pointing to that domain load for agents and they can see its content.

Customer Q&A

This page answers some of the most common questions we receive about Genesys Co-browse.

Q: Which emerging technologies and industry standards related to the Co-browse product are supported and will be evolved?

A: The key technologies targeted are HTML5, JavaScript and CSS.

Q: What is the Co-browse solution's architecture in relation to hardware and software?

A: See Co-browse Architecture.

Q: Can you describe any high availability and redundancy solutions?

A: In the current release, common resources not pertaining to a certain live co-browse session are served by any node in the cluster. Each co-browse session is hosted on a single server in the cluster. Future releases will support server failover functionality for Co-browse sessions where live sessions will be almost transparently transferred to another server in the cluster. Web chat sessions are already transparently migrated to another Co-browse server. Co-browse historical data redundancy is achieved through the Cassandra cluster.

Q: What is the highest amount of simultaneous users successfully handled?

A: Thanks to horizontal solution scalability, the highest amount of simultaneous users is limited only by the server hardware involved.

Q: Has the site traffic been verified by any third-party?

A: Not applicable. HTTP communication with Co-browse server is tested with ZAP proxy, https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.

Q: Can you describe Alert, Monitoring and control options and functionality (Administrative control and notifications for server and edge site errors)?

A: Co-browse server monitoring is performed through standard Genesys platform tools (messages are displayed in Genesys Administrator and the Solution Control Interface). The same goes for alerts.

Q: Can you describe the technologies that the application is written in?

A: Java (Jetty 9.x as a container), JavaScript, HTML5 (including Mutation Observers and Web Sockets), CSS, and CometD.

Q: Is the company’s core technology developed by internal engineering staff, or is it outsourced to partner developers?

A: Internal engineering develops the core technology.

Q: How is quality control ensured?

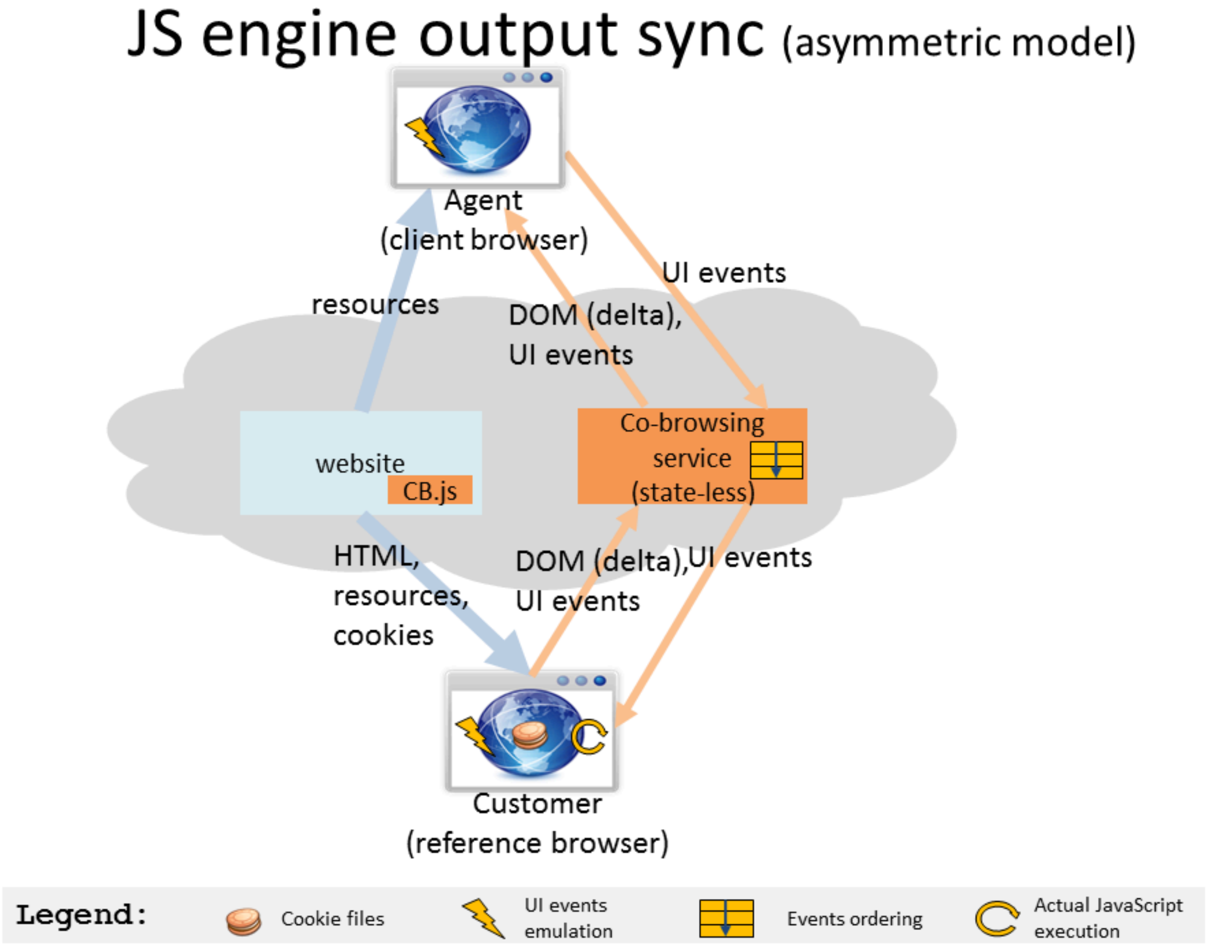
A: Daily builds are verified by Quality Assurance. The main use cases are automated.

Q: Has the technology been recognized by third-party endorsements?

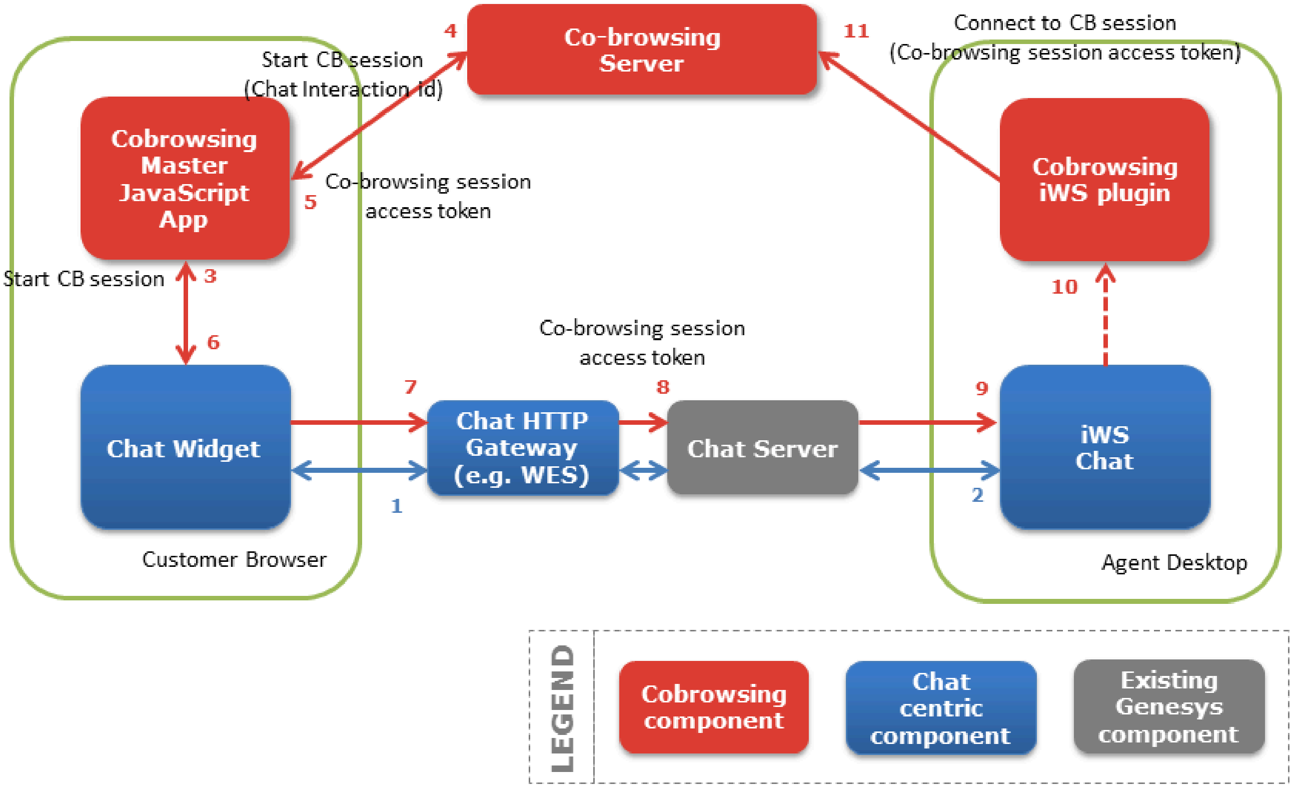
A: Similar principles are implemented by competitors.

Q: What are the basic principles of the system's inner workings?

A: Conceptual diagram:



The diagram below shows chat and Co-browse integration. Co-browse server incorporates Web Chat Gateway function as well.



Q: Can the web page be easily branded with company colors and logos?

A: Yes.

Q: Explain the process of embedding links on BAC web pages as well as any other user interfaces. How much integration is required?

A: Each cobrowsable website page must include a script. For more information, see Website Instrumentation. The following is a basic example of the required script:

```
<script>(function(d, s, id, o) {
  var fs = d.getElementsByTagName(s)[0], e;
  if (d.getElementById(id)) return;
  e = d.createElement(s); e.id = id; e.src = o.src;
  e.setAttribute('data-gcb-url', o.cbUrl);
  fs.parentNode.insertBefore(e, fs);
})(document, 'script', 'genesys-js', {
  src: "<COBROWSE_SERVER_URL>/gcb.min.js",
  cbUrl: "<COBROWSE_SERVER_URL>/cobrowse"
});</script>
```

Default functionality can be customized through JavaScript based configuration or the Co-browse JavaScript API. Co-browse website functionality can be roughly split into three parts:

- Co-browse session initiation UI
- The Co-browse session itself

- Chat widget

The UI for each can be replaced or customized using CSS, the JavaScript API, or Localization. For more information, see Customize the Genesys Co-browse UI

Q: Is the application flexible? Is it easy to add, customize and track new fields?

A: Yes.

Q: What are the desktop requirements for the customer and servicing agents using Genesys Co-browse?

A: Genesys Co-browse requires Workspace Desktop Edition starting with release 8.5.000.30.

Q: What is the minimum bandwidth required for a co-browse session?

A: The bandwidth will depend on a co-browsed web site's content and the techniques used to present it. Unlike screen sharing solutions, Genesys Co-browse syncs initial HTML page/resources, DOM deltas, and actions so it does not require high bandwidth. With Web Socket (which is supported by modern mobile devices and where ever bandwidth is normally concerned) support, there is not even over-head associated with HTTP requests or responses to Co-browse server. Web Sockets must be supported by the reverse proxy/load balancer infrastructure.

Q: Do you support secure WebSockets (wss://)?

A: Genesys Co-browse recommends using WebSockets and supports secure WebSockets. Co-browse uses secure WebSockets (wss://) when the website instrumentation **cbUrl** uses `https://`. If the **cbUrl** uses `http://` then Co-browse uses `ws://` WebSockets. When using `https://` in the **cbUrl**, you do not need any additional configuration to use secure WebSockets.

Q: Can you co-browse within iframes?

A: Co-browse supports co-browsing in iframes from the same domain. Agents can view but not co-browse iframes pointing to a different domain. For more information, see co-browsing in iframes.

Cluster URL Configuration Option

Starting with Co-browse Server 8.5.002.00, only the agent browser uses the cluster `url` option. The consumer's browser always uses the URL in the JS instrumentation for css-proxy and url-proxy. With this update, you can configure Co-browse to keep internal traffic from agents within the network while allowing customer traffic to flow through an external network. The following questions relate to this update:

Q: Will this address Agent Side driven CSS overwrite for actions like mouse hover?

A: All existing features continue working as before. Hover sync works in both directions.

Q: Are there any known limitations/impacts related to this approach?

A: Currently, no known limitations/impacts relate to this functionality. In this implementation, Instrumentation Scripts from the Customer Browser should point to an externally facing load balancer. The cluster `url` option in Co-browse Server is only used by agent desktops and can point to an internal load balancer.

Q: Is this change backwards compatible?

A: Yes, backwards compatibility is maintained.

Q: What do I need to do next?

A: Upgrade all Co-browse Servers in the cluster when the next release is available. If applicable, update the cluster url to an internally facing load balancer. Ensure the instrumentation script contains the correct externally facing URL for the Co-browse Server cluster.

Installing and Deploying Genesys Co-browse

Important

Genesys recommends that you first install Co-browse in a test environment. This will allow you to customize and test Co-browse before moving it to your production environment.

Objective	Related procedures and actions
1. Prepare your deployment.	Review Related Components and Sizing Information.
2. Install Genesys Co-browse Server.	See Install Genesys Co-browse Server for details.
3. Install the related plug-in for Workspace Desktop Edition.	Install the Genesys Co-browse Plug-in for Workspace Desktop Edition. See Install the Genesys Co-browse Plug-in for Workspace Desktop Edition. **Optional**: Configure token-based agent authentication.
4. If you are using Genesys Workspace Web Edition, configure it to work with Co-browse.	See Configure Genesys Workspace Web Edition to Work with Co-browse for configuration details.
5. Load the certificate and private keys into the Java and Jetty keystores.	See Loading Certificate for SSL for details.
6. Configure **allowedOrigins** and **allowedExternalDomains**.	As a security best practice, configure the **allowedOrigins** and **allowedExternalDomains** options to control which websites can access your Co-browse server and which external resources Co-browse server may proxy. Additionally, consider configuring the **allowedThirdPartyDomains** option to control which third-party iframes agents can view.
7. Add the Co-browse JavaScript snippet to your website.	See Website Instrumentation for details.
8. Configure a cluster of Co-browse servers.	See Configure a Cluster of Co-browse Servers for details.
9. Start and stop Genesys Co-browse Server.	See Start and Stop Genesys Co-browse Server for details.
10. Import the reporting templates.	You can use the provided Genesys Co-browse Sample Reporting Templates for real-time and historical reporting. See Genesys Co-browse Reporting Templates for details.

Objective	Related procedures and actions
11. Test and troubleshoot.	Complete the procedures on the Testing and Troubleshooting the Co-browse Solution page to ensure that your Co-browse solution is properly configured. This page also provides solutions to common problems that you might encounter while testing the Co-browse solution.

Related Components

The following components are mandatory for Genesys Co-browse:

Server Name	Compliant Versions (and later)
Configuration Server	8.1.100.14+
Workspace Desktop Edition	8.5.100.05+
Workspace Web Edition	8.5.200.80+

See also, Genesys Co-browse in the *Genesys Supported Operating Environment Reference Guide*.

Sizing Information

Before deploying the Genesys Co-browse solution to your production site, you should estimate the solution size needed to handle your expected user load. Genesys recommends using the Co-browse Sizing Calculator, an Excel workbook that helps you calculate the number of Co-browse Server nodes required for your production deployment.

Download Co-browse Sizing Calculator-85.xlsx

Estimating Load

For Co-browse load capacity planning, use the following input parameters in the Co-browse Sizing Calculator:

- Expected maximum parallel Co-browse sessions
- Website complexity. In the Sizing Calculator, you can select from two boundary options, average (genesys.com) and high (amazon.com). Choose high if your website is highly dynamic, and interactive. For example, websites including a large single-page application, a lot of multimedia content, and/or dynamic page options should select high website complexity.
- WebSocket connection availability
- CPU cores per node, 8 or 4

WebSocket Support

To achieve the best performance, we highly recommend WebSocket support. Genesys Co-browse enables WebSockets by default. WebSocket-based Co-browse sessions appear smoother to users and consume significantly less traffic by avoiding HTTP overhead. The Co-browse server also consumes less hardware resources when using WebSockets and you may require fewer nodes for your Co-browse cluster.

If you use WebSockets, make sure your load balancers, proxies, and firewalls allow WebSocket connections through. Co-browse uses either WebSockets (ws://) or secure WebSockets (wss://) depending on your website instrumentation.

Planning Scalability

Since Co-browse server nodes do not share many resources besides the Cassandra cluster, you have nearly linear scalability from your Co-browse cluster. Each node adds the same amount of capacity to the cluster.

For high-availability purposes, we recommend **at least one additional server** to handle the load in case of server failure. The Co-browse Sizing Calculator includes this recommendation. The Sizing

Calculator recommends no fewer than two nodes, even if the single server capacity can handle estimated server load.

Cassandra Cluster Deployment

Most Cassandra clusters should sufficiently support a Co-browse cluster because the Co-browse solution does not produce large amounts of data to store. Confirm that your Cassandra cluster provides enough availability and capacity.

> ### Important
>
> Starting in 8.5.0, Embedded Cassandra mode is deprecated in Genesys Co-browse; support for this mode will be discontinued in 9.0.

Tested Browsers

The following is a list of all Genesys-tested browsers for both web and mobile.

Important

If you do not see your device/OS/browser combination listed below, please contact Genesys support. Help will be decided on a per-case basis.

Support for the device/OS/browser combinations listed below will only be available for as long as Genesys labs can properly reproduce the issue.

Please let Genesys know of any issues you encounter with any of our tested browsers.

Tip

For a list of all supported devices for Genesys Widgets, see Genesys Widgets - Tested Browsers.

Web Browsers

- Microsoft Internet Explorer 11
- Google Chrome 47+
- Firefox 43+
- Safari 8+
- Microsoft Edge

Mobile Browsers

OS Family	Device	Operating System	Browser	Release Version	Known Limitations
Android	Galaxy S7	Android 6	Chrome 58	Co-browse Server 8.5.101.02	n/a
	Galaxy Tab 4 10.1	Android 4.4	Chrome 58	Co-browse Server	n/a

OS Family	Device	Operating System	Browser	Release Version	Known Limitations
				8.5.101.02	
	Nexus 7 Tab	Android 6	Chrome 58	Co-browse Server 8.5.101.02	n/a
	Nexus 9 Tab	Android 5.1	Chrome 58	Co-browse Server 8.5.101.02	n/a
iOS	iPhone 6S Plus	iOS 9	Safari	Co-browse Server 8.5.101.02	n/a
	iPad Air 2	iOS 8	Safari	Co-browse Server 8.5.101.02	n/a
	iPad Pro	iOS 9.3	Safari	Co-browse Server 8.5.101.02	n/a
	iPad Pro	iOS 10.3	Safari	Co-browse Server 8.5.101.02	n/a

Install Genesys Co-browse Server

> **Important**
>
> - Starting in 8.5.0, Embedded Cassandra mode is deprecated in Genesys Co-browse; support for this mode will be discontinued in 9.0.
> - Genesys strongly recommends no more than one Cassandra instance on the same machine due to performance. If you intend to use a Cassandra instance embedded into Co-browse Server, you should not have a `CASSANDRA_HOME` environment variable to avoid conflict between embedded and external `CASSANDRA_HOME` variables.

Creating the Co-browse Server Application Object in Genesys Administrator

> **Important**
>
> Make sure to select the appropriate version tab below. The Application objects and templates are different for versions **8.5.000** and **8.5.001+**.

8.5.001+

Overview

Co-browse 8.5.001+ introduces a new cluster configuration, which includes cluster and node Application objects. In the steps below, you create multiple Co-browse Server Application nodes, each with unique configuration, and unite them under a common configuration in the Cluster Application object:

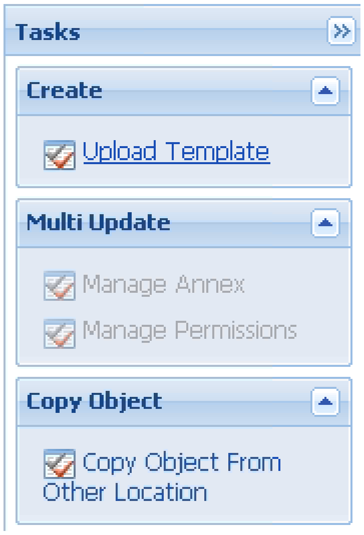
- Importing the Application Templates
- Creating the Co-browse Cluster Application
- Creating the Co-browse Node Application

Importing the Application Templates

To support this new configuration, you must first import two application templates, one for the node and one for the cluster.

Start

1. Open Genesys Administrator and navigate to **PROVISIONING > Environment > Application Templates**.
2. In the **Create** menu of the **Tasks** panel, click the **Upload Template** link.



Upload Template link in the Tasks panel

3. When the dialog box appears, Click **Add** to choose the application template (APD) file to import.
4. Choose the **Co-browse_Cluster_850.apd** file in the **templates** directory of your installation CD. The **New Application Template** panel opens.
5. Click **Save & Close**.
6. Complete the same import steps for the *Node Application Template* by importing the **Co-browse_Node_850.apd** file in the **templates** directory of your installation CD.

End

Creating the Co-browse Cluster Application

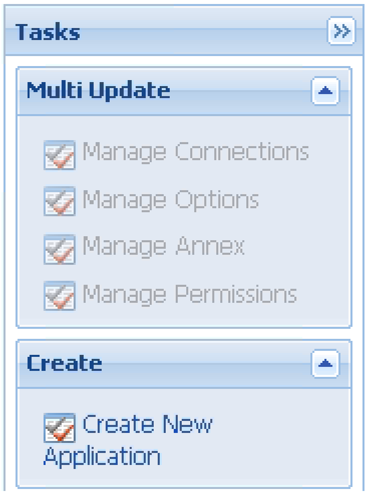
Prerequisites

- You completed Importing the Application Templates.

Start

1. Open Genesys Administrator and navigate to **PROVISIONING > Environment > Applications**.

2. In the **Create** menu of the **Tasks** panel, click the **Create New Application** link.



Create New Application link.

3. In the **Select Application Template** panel, click **Browse for Template** and select the Co-browse Cluster template you previously imported. Click **OK**.
4. The template appears in the **Select Application Template** panel. Click **Next**.
5. In the **Select Metadata File** panel, click **Browse** and select the **Co-browse_Cluster_850.xml** file. Click **Open**.
6. The metadata file appears in the **Select Metadata File** panel. Click **Next**.
7. In **Specify Application parameters**:
   - Enter a name for your application—for example, `Co-browse_Cluster`.
   - Enable the State.
   - Select the host of the Co-browse Server. Co-browse does not actually use this value for the Cluster Application, so you can specify any host.
   - Click **Create**.
8. The **Results** panel opens. Enable **Open the Application details form after clicking Finish** and click **Finish**. The Co-browse Cluster Application form opens to the **Configuration** tab.
9. If you use an external Cassandra cluster, add connections to the external Cassandra Resource Access Points (RAPs). For each RAP connection, set the **ID** to `default`.
10. Create any necessary connections to other Genesys servers. For example:
    - Primary Configuration Server
    - Primary Message Server
    - Chat Server(s) or Chat Application Cluster

Important

> The connection to Chat Server needs to be done through the **webapi** port.

11. In the **Server Info** section, add the **Default Listening Port**. Co-browse does not use this value for the Cluster Application, so you can specify any port.
12. Make sure the **Working Directory** and **Command Line** fields are set to `.` (period).
13. In the **Options** tab, set the following configuration options:

> Important
>
> Application Node configuration has priority over Application Cluster Configuration. If you configure an option in both the Application Node and Application Cluster objects with different values, Co-browse uses the Application Node value.

- In the cassandraEmbedded section: If you use an external Cassandra cluster, set **enabled** to `false`. Genesys recommends using an external Cassandra cluster. Support for embedded Cassandra will be discontinued in 9.0.
- In the cassandraKeyspace section: Set **replicationStrategyParams** using the following format:

  ```
  '<Data Center name>':<Co-browse Nodes amount>
  ```

  For example: `'OperationalDC':3`
- In the cluster section: Set the **url** option to the HTTP(S) Co-browse load balancer. For example:

  ```
  https://<LB_host>:<LB_port>/cobrowse
  ```

  > Important
  >
  > The **url** option is mandatory and you must configure this option to properly set up a Co-browse Server cluster. Be careful not to confuse the **url** option with the **serverUrl** option which is not mandatory.

14. Set additional options according to your needs. See Configuration Options for details.
15. Click **Save & Close**. The **Confirm** dialog displays the following message:

    The host and/or port(s) of the application will be changed. Do you want to continue?

    Click **Yes**.

End

Creating the Co-browse Node Application

Prerequisites

- You completed Importing the Application Templates.

- You completed Creating the Co-browse Cluster Application.

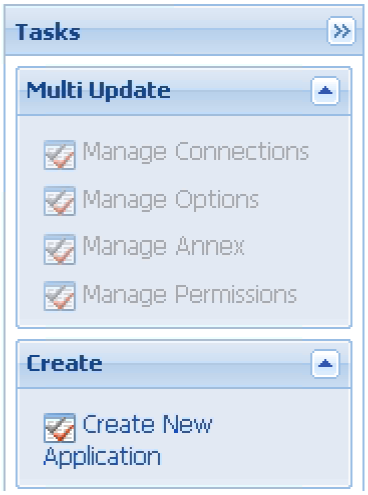
Complete the following steps for *each* Co-browse Node Application in your cluster.

> ### Important
> If your Cassandra cluster is built from Cassandra nodes that are embedded in your Co-browse nodes, the minimum number of Co-browse nodes is three. If you use an external Cassandra cluster, the minimum number of Co-browse nodes is two.

Start

1. Open Genesys Administrator and navigate to **PROVISIONING > Environment > Hosts**.
2. Click **New...** and create the **Host** object where the Co-browse Node will run.
3. Navigate to **PROVISIONING > Environment > Applications**.

   In the **Create** menu of the **Tasks** panel, click **Create New Application**.

   Create New Application link.

4. In the **Select Application Template** panel, click **Browse for Template** and select the Co-browse *Node* template you previously imported. Click **OK**.
5. The template appears in the **Select Application Template** panel. Click **Next**.
6. In the **Select Metadata file** panel, click **Browse** and select the **Co-browse_Node_850.xml** file. Click **Open**.
7. The metadata file appears in the **Select Metadata File** panel. Click **Next**.
8. In **Specify Application Parameters**:
   - Enter a name for your application—for instance, `Co-browse_Node`.
   - Enable the **State**.

- Select the **Host** you created in Step 2. This is the host where the Co-browse Server will reside.
- Click **Create**.

9. The Results panel opens
   Enable **Open the Application details form after clicking 'Finish'** and click **Finish**. The Co-browse Node Application form opens to the **Configuration** tab.
10. Create a connection to the Co-browse Cluster Application you previously created and set the **ID** to `default`.
11. In the **Server Info** section, add the **Default Listening Port**. Set **Connection Protocol** to `http` and **Listening Mode** to `unsecured`.
12. If you intend to use https, add the **https port**. Set **Connection Protocol** to `https` and **Listening Mode** to `secured`.
13. Make sure the **Working Directory** and **Command Line** fields are set to `.` (period). Co-browse automatically populates these fields during installation.
14. In the **Options** tab, set the following configuration options:

> Important
>
> Application Node configuration has priority over Application Cluster Configuration. If you configure an option in both the Application Node and Application Cluster objects with different values, Co-browse uses the Application Node value.

   - In the cassandraEmbedded section: If you use an external Cassandra cluster, leave all values at their defaults. Otherwise, specify the necessary values for your configuration.

15. Set any other options according to your needs. See Configuration Options for details.
16. Click **Save & Close**. The **Confirm** dialog displays the following message:
    The host and/or port(s) of the application will be changed. Do you want to continue?
    Click **Yes**.

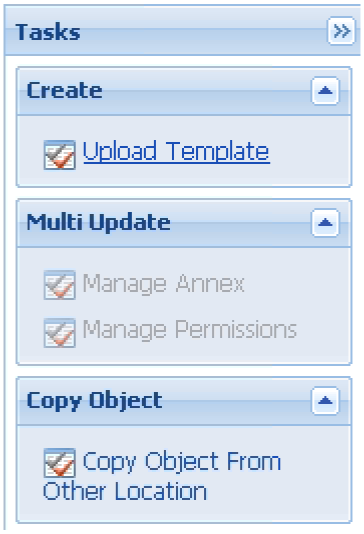
End

8.5.000

Importing the Application Template for the Co-browse Server

Start

1. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Application Templates`.
2. In the `Create` menu of the `Tasks` panel, click the `Upload Template` link.



Upload Template link in the Tasks panel

3. Click Add in the `Click 'Add' and choose application template (APD) file to import` dialog box.
4. Browse to the `Co-browse_Server_850.apd` file, available in the `templates` directory of your installation CD. The `New Application Template` panel opens.
5. Click `Save & Close`.

End

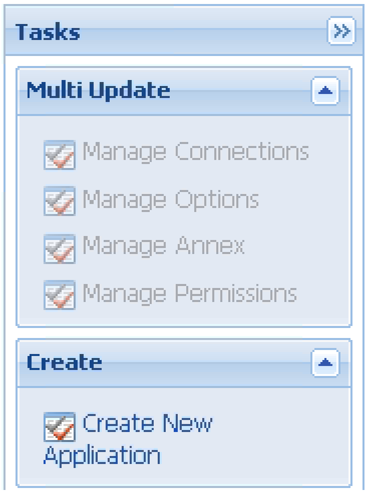
Creating the Co-browse Server Application

Prerequisites:

- You must have a `webapi` listening port configured on your Chat Server Application. For details, refer to the eServices 8.1 Deployment Guide.
- You completed Importing the Application Template for the Co-browse Server.

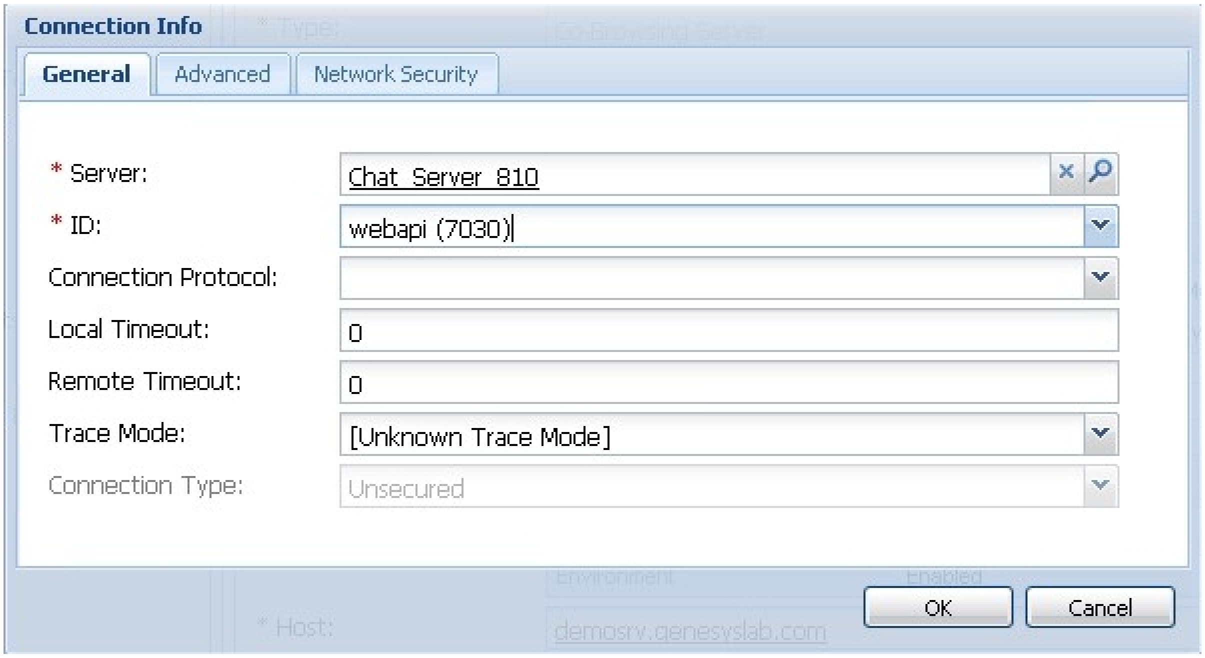
Start

1. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`.
2. In the `Create` menu of the `Tasks` panel, click the `Create New Application` link.



Create New Application link.

3. In the Select Application Template panel, click Browse for Template and select the Co-browse Server template that you imported in Importing the Application Template for the Co-browse Server. Click OK.
4. The template is added to the Select Application Template panel. Click Next.
5. In the Select Metadata file panel, click Browse and select the Co-browse_Server_850.xml file. Click Open.
6. The metadata file is added to the Select Metadata file panel. Click Next.
7. In Specify Application parameters:
   - Enter a Name for your application—for instance, Co-browse_Server.
   - Enable the State.
   - Select the Host on which Co-browse Server will reside.
   - Click Create.
8. The Results panel opens.
   - Enable Opens the Application details form after clicking 'Finish' and click Finish. The Co-browse Server application form opens to the Configuration tab.
9. Add a connection to Chat Server.
   - Click Add in the Connections section.
   - Select the Chat Server and click OK.
   - Edit the connection and set ID to the webapi listening port on your Chat Server.



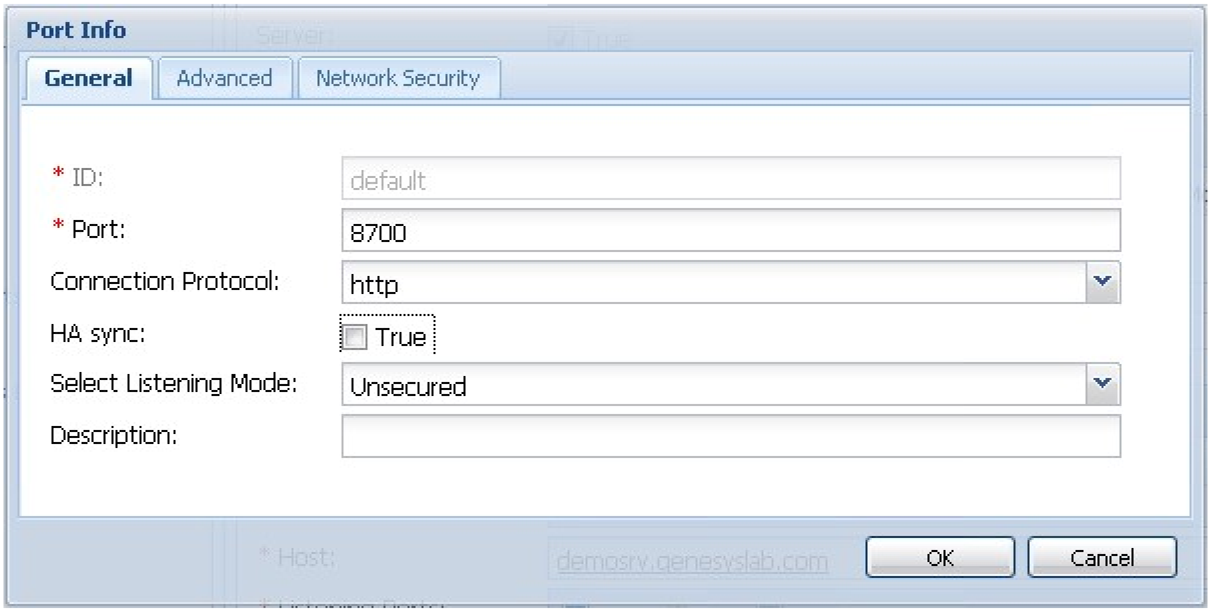
Connection Info

* Server:	Chat_Server_810
* ID:	webapi (7030)
Connection Protocol:	
Local Timeout:	0
Remote Timeout:	0
Trace Mode:	[Unknown Trace Mode]
Connection Type:	Unsecured

Chat Server connection

- Click OK.

10. If your Host is not defined, click the lookup icon to browse to the hostname of your application.
11. Add the default port.
    - Click Add in the `Listening Ports` section. The `Port Info` dialog box opens.
    - Enter the application's `Port`. For instance, 8700. This must be a Jetty port configured in the Jetty configuration file.
    - Mandatory: Enter `http` for the `Connection Protocol` field. This connection protocol setting will be used by IWS to connect to the Co-browse Server.
    - Optional: Enter a description.



* ID: default

* Port: 8700

Connection Protocol: http

HA sync: □ True

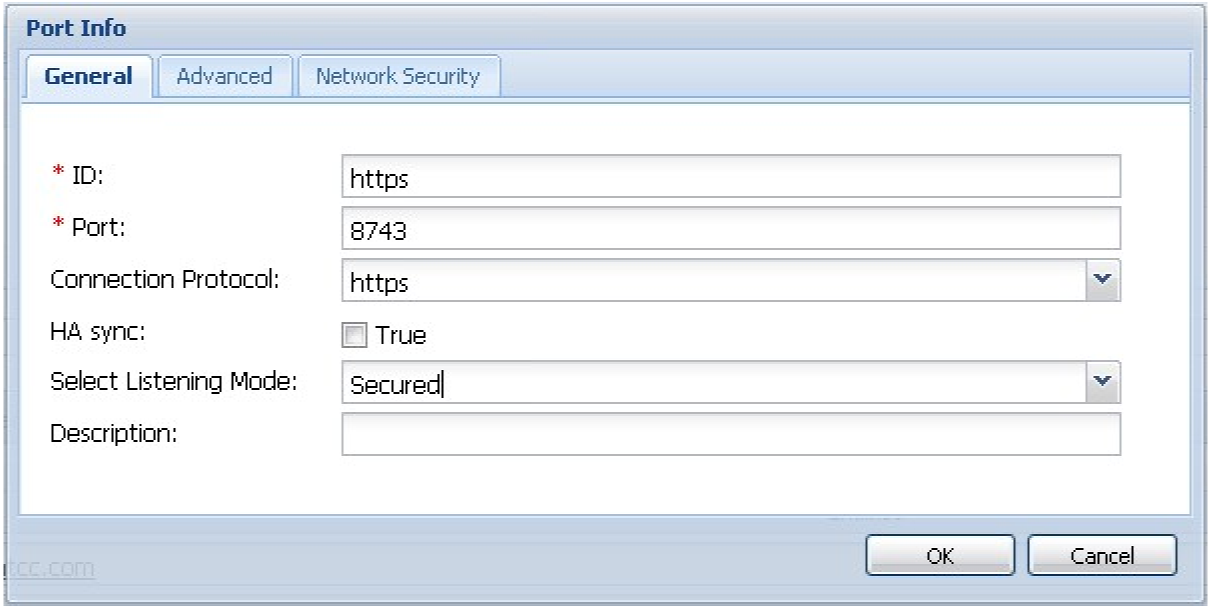
Select Listening Mode: Unsecured

Description:

The default http listening port.

- Click OK. The HTTP port with the default identifier appears in the list of Listening Ports.

12. Add the secure port.
    - Click Add in the Listening Ports section. The Port Info dialog box opens.
    - Enter https for the ID.
    - Enter the application's secure Port. For instance, 8743. This must be a Jetty port configured in the Jetty configuration file.
    - Mandatory: Enter https for the Connection Protocol field.
    - Mandatory: Select the Secured for the Select Listening Mode field.
    - Optional: Enter a description.



- * ID: https
- * Port: 8743
- Connection Protocol: https
- HA sync: □ True
- Select Listening Mode: Secured
- Description:

The https listening port.

- Click OK. The HTTPS port with the https identifier appears in the list of `Listening Ports`.

13. Ensure the `Working Directory` and `Command Line` fields contain "." (period). They will be automatically populated when the Co-browse Server is installed.
14. Click `Save & Close`. The `Confirm` dialog displays the following message: `The host and/or port(s) of the application will be changed. Do you want to continue?` Click Yes.

End

Next Steps

- You can now install the Co-browse Server as described in Installing the Co-browse Server.

Installing the Co-browse Server

With basic Configuration Server details in place, you can now complete the installation process.

> ### Important
> Genesys does not recommend installation of its components using a Microsoft Remote Desktop connection. You should perform the installation locally.

Prerequisites: You completed Creating the Co-browse Server Application Object in Genesys Administrator.

Start

1. In your installation package, locate and run the setup application for your platform as specified below:
   - Linux: **install.sh**
   - Windows: **setup.exe**

   The Install Shield opens to a welcome screen.
2. Click **Next**. The **Connection Parameters to the Configuration Server** screen appears.
3. Under **Host**, specify the host name and port number of your Configuration Server. This value should be the same as the main listening port in the **Server Info** tab of your Configuration Server.
4. Under **User**, enter the user name and password to log in to Configuration Server.
5. Click **Next**. The **Select Application** screen appears.
6. Select the Co-browse Server Application you are installing. The **Application Properties** area shows the **Type**, **Host**, **Working Directory**, **Command Line executable**, and **Command Line Arguments** information previously entered in the **Server Info** and **Start Info** tabs of the selected Application object.
7. Click **Next**. The **Choose Destination Location** screen appears.
8. Under **Destination Folder**, keep the default value or click browse to set the installation location.
9. Click **Next**. The **Backup Configuration Server Parameters** screen appears.
10. Under **Host**, specify the host name and the port number where the Backup Configuration Server is running.
11. Click **Next**. The **Ready to Install** screen appears.
12. Click **Install**. When The Genesys Installation Wizard finishes installing Co-browse Server, the **Installation Complete** screen appears.
13. Click **Finish** to complete your installation of Co-browse Server.

End

Next Steps

- Complete the configuration of the Co-browse Server Application, as described in Configuring the Co-browse Server.

Configuring the Co-browse Server

Complete the steps below to configure the Co-browse Server application in Genesys Administrator. This procedure only covers a few of the mandatory options. Most options can be left at their default values.

Prerequisites: Creating the Co-browse Server Application.

Start

1. Open Genesys Administrator and navigate to **PROVISIONING > Environment > Applications**.
2. If you are installing version **8.5.000**, select the Co-browse Server Application that you previously created. If you are installing **8.5.001+**, select the Co-browse *Cluster* Application you previously created.
3. In the **Options** tab, locate the **session** section and update the following options:
    - **domRetrictionsURL**—a URL that points to the XML file that contains DOM restrictions. By default, all **Submit** buttons are disabled for the agent. For information about customizing this XML file, see DOM Restrictions
4. If you deploy Co-browse Server to an environment where the Internet is accessed using a forward proxy (for example, DMZ or local intranet), configure the options in the **forward-proxy** section.
5. If your configuration uses Genesys Chat, update the following option in the **chat** section:
    - **queueKey**—specifies the endpoint configured in Chat Server. Use the format `tenantid:endpointname`.
6. If you are installing **8.5.001+**, **Save & Close** your Co-browse Cluster Application and open the Co-browse Node Application that you previously created.
7. Configure the options in the **log** section. These options are standard Genesys log options. For details, refer to the Management Framework 8.1 Configuration Options Reference Manual.
8. If you use Genesys Workspace Web Edition, configure the **serverUrl** option in the **cluster** secion.
9. Click **Save & Close**.

End

Next Steps

- Install the Genesys Co-browse Plug-in for Workspace Desktop Edition
- Configure Genesys Workspace Web Edition to Work with Co-browse

Install the Genesys Co-browse Plug-in for Workspace Desktop Edition

> **Important**
>
> Genesys does not recommend installation of its components via a Microsoft Remote Desktop connection. The installation should be performed locally.

> **Important**
>
> For compliant versions of each component, see Related Components.

Installing the Genesys Co-browse Plug-in for Workspace Desktop Edition in Application Mode

Prerequisites

- You have installed Workspace Desktop Edition in Application mode.
- You have installed Internet Explorer 11.

Installing the Genesys Co-browse Plug-in

Start of procedure

1. In your installation package, locate and double-click the `setup.exe` file. Click **Next**. The **Select Installed Application** screen appears.
2. Select your Workspace Desktop Edition application.
3. Click **Next**. The **Ready to Install** screen appears.
4. Click **Install**. The Genesys Installation Wizard indicates it is performing the requested operation for the Genesys Co-browse Plug-in for Workspace Desktop Edition. When done, the **Installation Complete** screen appears.
5. Click **Finish** to complete your installation of the Genesys Co-browse Plug-in for Workspace Desktop Edition.

End of procedure

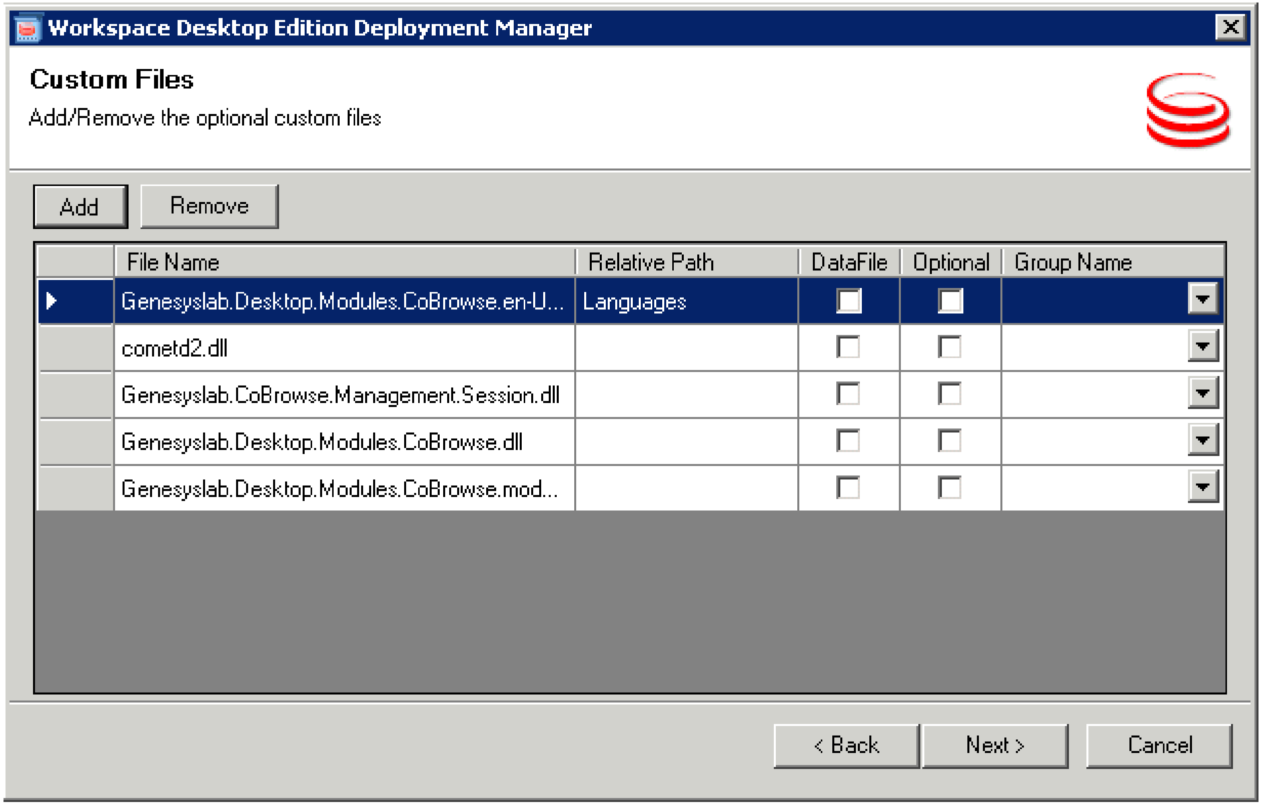
Installing the Genesys Co-browse Plug-in for ClickOnce/ Developers Toolkit Workspace Desktop Edition

Prerequisites

- You have Installed Workspace Desktop Edition in ClickOnce or Developers Toolkit mode.
- You have installed Internet Explorer 11.

Start of procedure

1. Install the Co-browse WDE Plug-in in your WDE installation as described in Installing the Genesys Co-browse Plug-in.
2. From the `Start` menu, open **Workspace Desktop Edition- Deployment Manager**.
3. Click **Next**. On the next screen, check the topmost check box and click **Next** again.
4. Check the **Add custom files** check box. Note and remember the `Base URL*` value. This value will be used as the agent's login. Click **Next**
5. Use the **Add** button to add to the Custom Files list all plug-in files placed in the WDE installation installation folder by setup. Leave all check boxes unchecked. Click **Next**.



Workspace Desktop Edition Deployment Manager

Custom Files

Add/Remove the optional custom files

	File Name	Relative Path	DataFile	Optional	Group Name
▶	Genesyslab.Desktop.Modules.CoBrowse.en-U...	Languages	□	□	
	cometd2.dll		□	□	
	Genesyslab.CoBrowse.Management.Session.dll		□	□	
	Genesyslab.Desktop.Modules.CoBrowse.dll		□	□	
	Genesyslab.Desktop.Modules.CoBrowse.mod...		□	□	

6. Enter the **Config Server host**, **Config Server port**, and WDE **application name** in the Client Configuration page. Take care to enter this information correctly. Check both check boxes below the page. **Click Next**.

7. Click **Next** in the next two screens.
8. At the next screen, leave all check boxes unchecked. Click **Finish**.
9. When you see the Application Installation Wizard, click **Install**.
10. Log in an agent. You should see the WDE main window.

End of procedure

You have set up a WDE application and deployed the Co-browse WDE plug-in. You can now log in any agent using URL `<Base URL*>publish.htm` from any other host.

Configuring Workspace Desktop Edition to allow the Plug-in to work with co-browsing

Prerequisites

- You have installed Workspace Desktop Edition.
- You have installed the Genesys Co-browse Plug-in for Workspace Desktop Edition
- You have prepared a Co-browse Server cluster or a stand-alone Co-browse Server.

To configure Workspace Desktop Edition to work with Co-browse:

1. Open Genesys Administrator and navigate to **PROVISIONING > Environment > Applications**.
2. Select the Workspace Desktop Edition application.
3. In the application's **Options** section, create a `cobrowse` section and specify the `url` option in this section, see the `url` option for details.

> Important
>
> If you use a stand-alone Co-browse Server, you can also configure co-browsing using a Connection to the Co-browse Server/Node application. However, this functionality is deprecated since **8.5.0** and will be removed in the **8.5.101** release. If you configure a connection to a Co-browse Node application, make sure Workspace Desktop Edition has read access to all applications it connects to, including the Co-browse Node.

Next Steps

- Loading Certificate for SSL
- Configure Token-based Agent Authentication (Optional)

Configuring Role-Based Access Control for Co-browse

To configure Co-browse privileges for an agent you should:

- Import the Co-browse Plug-in for Workspace Desktop Edition template.
- Configure Role-Based Access Control.

> **Important**
>
> Starting in version 8.5.001.09, the plug-in is shipped with the Application Template and XML metadata that provide the plug-in options to configure in the `cobrowse` section. The Application Template also includes the `Agent - Can Monitor Co-browse` privilege, a dedicated privilege that the Co-browse plug-in supports starting in this release.

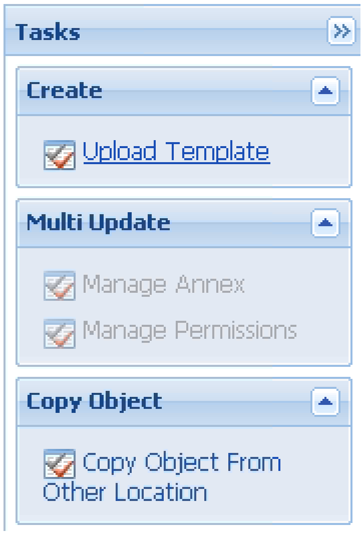
Importing the Co-browse Plug-in for Workspace Desktop Edition Template

Prerequisites

- You have configured Workspace Desktop Edition to allow the plug-in to work with Co-browsing.

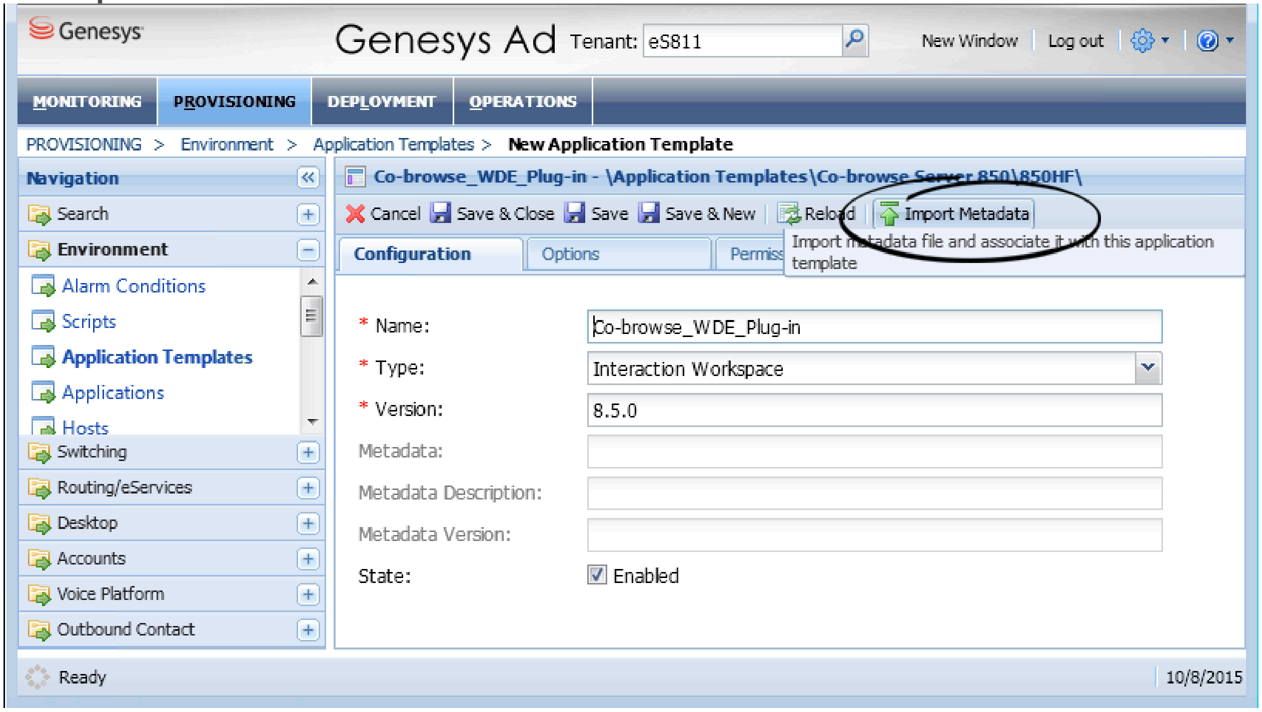
Start

1. Open Genesys Administrator and navigate to **PROVISIONING > Environment > Application Templates**.
2. In the **Create** menu of the **Tasks** panel, click the **Upload Template** link.



3. Click **Add** in the **Click ′Add′ and choose application template (APD) file to import** dialog box.
4. Browse the `Co-browse_WDE_Plug-in_850.apd` file, available in the `Templates` directory of the plug-in installation. Click **Open**. The **New Application Template** panel opens.

5. Click **Import Metadata**.



Import metadata file and associate it with this application template

* Name: Co-browse_WDE_Plug-in
* Type: Interaction Workspace
* Version: 8.5.0
Metadata:
Metadata Description:
Metadata Version:
State: ☑ Enabled

6. Browse to the `Co-browse_WDE_Plug-in_850.xml` file, available in the `Templates` directory of the plug-in installation. Click **Open**. The metadata fields in the **Configuration** tab are now filled.
7. Click **Save & Close**.

End
Now, when you are provisioning the Privileges assigned to a Role, the list of Privileges available for `Interaction Workspace` includes the privileges defined in the Co-browse Plug-in template.

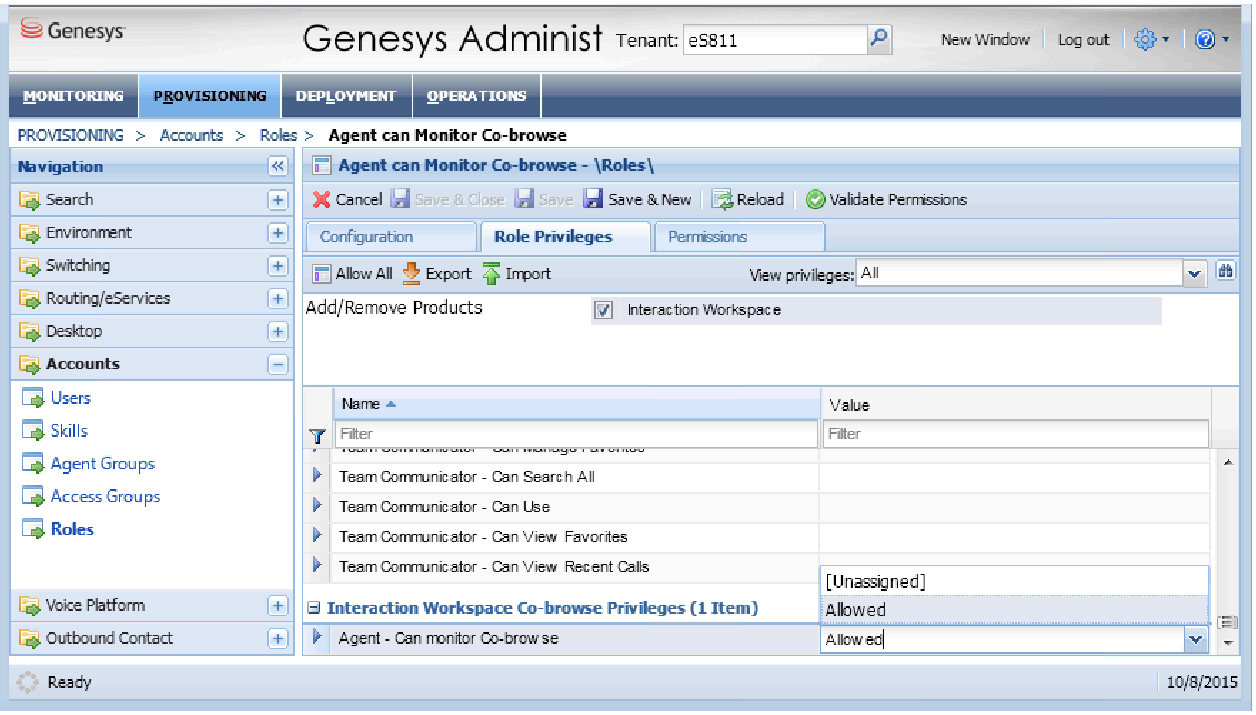
Configuring Agent Privileges to Work with Co-browse

You must complete this procedure to allow specific users or groups to manage Co-browse in Workspace Desktop Edition.
Prerequisites

- You have imported the Co-browse Plug-in for Workspace Desktop Edition Template.

Start

1. In Genesys Administrator, navigate to **Provisioning > Accounts > Roles**.
2. Edit or create a Role responsible for managing Co-browse in Workspace Desktop Edition. For instance, click **New** to create the `Agent - Can Monitor Co-browse` role.
3. Select the **Role Privileges** tab.
4. In the **Add/Remove Products** top panel, enable **Interaction Workspace** and expand the **Interaction Workspace Co-browse Privileges** section.
5. Set the `Agent - Can Monitor Co-browse` option to `Allowed`.



Tenant: eS811

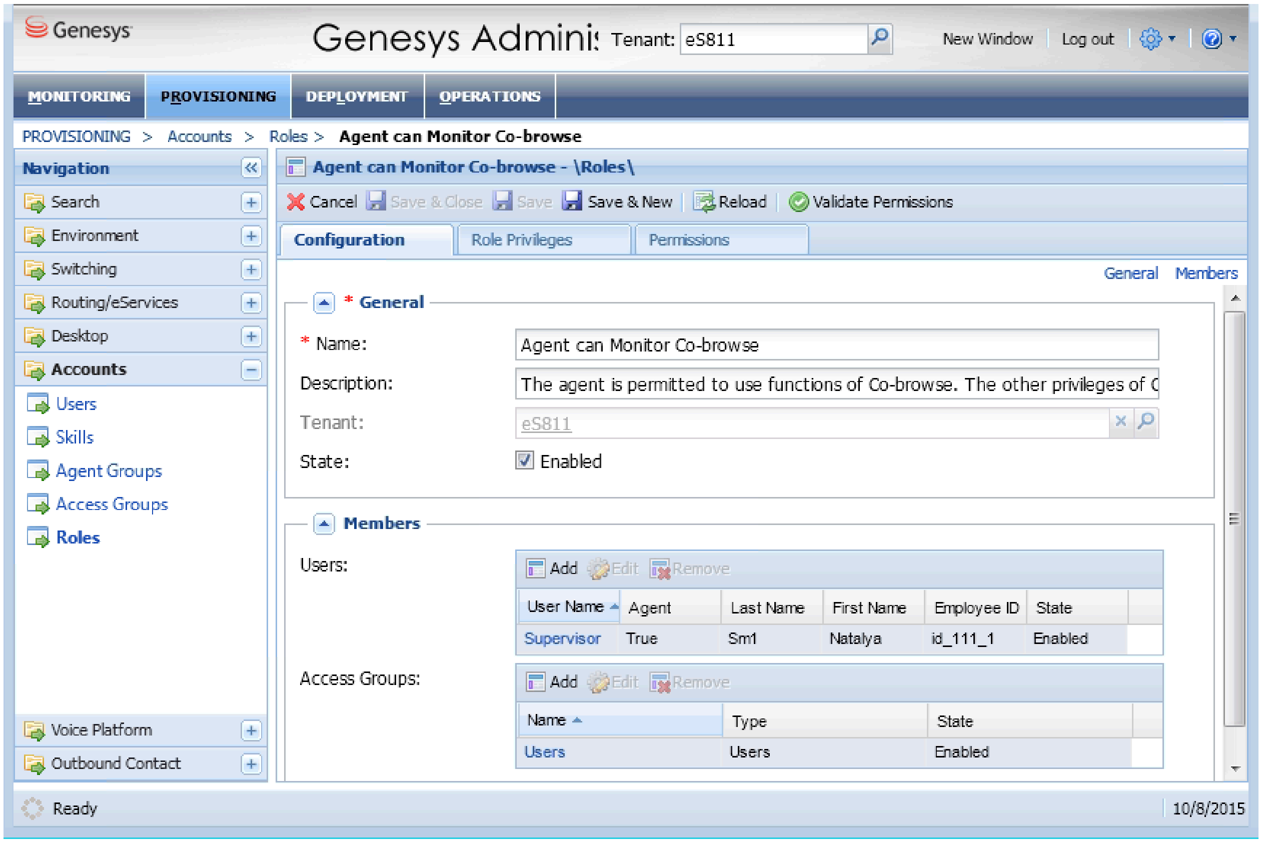
View privileges: All

Add/Remove Products ☑ Interaction Workspace

Name	Value
Team Communicator - Can Search All	
Team Communicator - Can Use	
Team Communicator - Can View Favorites	
Team Communicator - Can View Recent Calls	
Interaction Workspace Co-browse Privileges (1 Item)	
Agent - Can monitor Co-browse	Allowed

[Unassigned]
Allowed

6. In the **Members** section of the **Configuration** tab, add the users or groups who should get this role.



General

* Name: Agent can Monitor Co-browse

Description: The agent is permitted to use functions of Co-browse. The other privileges of C

Tenant: eS811

State: ☑ Enabled

Members

Users:

User Name	Agent	Last Name	First Name	Employee ID	State
Supervisor	True	Sm1	Natalya	id_111_1	Enabled

Access Groups:

Name	Type	State
Users	Users	Enabled

7. Click **Save & Close**.

End

Configure Genesys Workspace Web Edition to Work with Co-browse

> **Important**
> For compliant versions of each component, see Related Components.

To use Co-browse with Workspace Web Edition (WWE), do the following:

1. Configure the Co-browse options in the WWE Application object. For a list of the options and the appropriate settings, see the Co-browse topic in the *Workspace Web Edition Configuration Guide*.
2. Configure the **serverUrl** option for each Co-browse node in your cluster.
3. Configure the **wweOrigins** option for your Co-browse Cluster application.

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.

Start and Stop Genesys Co-browse Server

Start the Co-browse Server

Select a tab below to start Co-browse Server on either Windows or Unix:

Windows

Start the Co-browse Server on Windows

> Important
>
> You can start the Genesys Co-browse Server on Windows from:
>
> - Windows Services
> - the startserver.bat script
> - the cobrowse.bat script
> - Genesys Administrator

Start

- You can start the server from Windows Services.
  1. Open Windows Services
  2. Select and start the Co-browse Server service.
- You can use the provided `startserver.bat` script.
  1. Navigate to the Co-browse installation `server` directory and launch the Windows command console (cmd.exe).
  2. Type and execute `startserver.bat`, without any parameters.
- You can use the provided `cobrowse.bat` script.
  1. Navigate to the Co-browse installation `server` directory and launch the Windows command console (cmd.exe).
  2. Type and execute `cobrowse.bat`, along with the '-host', '-port', and '-app' parameters. For example, `cobrowse.bat -host demosrv.genesyslab.com -port 2020 -app Co-browse_Server`
     You can find your parameters in the `Server Info` section of your Co-browse application in

Genesys Administrator.

- You can start the server from Genesys Administrator.
    1. Navigate to `PROVISIONING > Environment > Applications`.
    2. Select the Co-browse Server.
    3. Click `Start applications` in the Runtime panel.

End
The Genesys Co-browse Server is shown in `Started` status in Genesys Administrator.

Unix

Start the Co-browse Server on Unix

> **Important**
> You can start the Genesys Co-browse Server on Unix from:
>
> - the run.sh script
> - the cobrowse.sh script
> - Genesys Administrator

Start

- You can use the provided `run.sh` script.
    1. Navigate to the Co-browse installation root directory in the Unix command console.
    2. Type and execute `run.sh`, without any parameters.
- You can use the provided `cobrowse.sh` script.
    1. Navigate to the Co-browse installation `server` directory in the Unix command console.
    2. Type and execute `cobrowse.sh`, along with the '-host', '-port', and '-app' parameters. For example, `cobrowse.sh -host demosrv.genesyslab.com -port 2020 -app Co-browse_Server`. **Note:** You can start the application as a daemon by adding `-d` to the command.
       You can find your parameters in the `Server Info` section of your Co-browse application in Genesys Administrator.
- You can start the server from Genesys Administrator
    1. Navigate to `PROVISIONING > Environment > Applications`.
    2. Select the Co-browse Server.
    3. Click `Start applications` in the Runtime panel.

End
The Genesys Co-browse Server is shown in `Started` status in Genesys Administrator.

Stop the Co-browse Server

Select a tab below to stop Co-browse Server on either Windows or Unix:

Stop the Co-browse Server on Windows

Stop the Co-browse Server on Windows

> **Important**
> You can stop the Genesys Co-browse Server on Windows from:
>
> - Windows Services
> - Genesys Administrator
> - A console window

Start

- You can stop the server from Windows Services.
  1. Open Windows Services
  2. Select and stop the Co-browse Server service.
- You can stop the server from Genesys Administrator.
  1. Navigate to `PROVISIONING > Environment > Applications`.
  2. Select the Co-browse Server.
  3. Click `Stop applications` in the Runtime panel.
- If you previously started Co-browse Server in a console window, you can stop the server by closing the window.

End
The Genesys Co-browse Server is shown in `Stopped` status in Genesys Administrator.

Stop the Co-browse Server on Unix

Stop the Co-browse Server on Unix

> **Important**
>
> You can stop the Genesys Co-browse Server on UNIX from either **Genesys Administrator** or a **console window**.

Start

- You can stop the server from Genesys Administrator.
  1. Navigate to `PROVISIONING > Environment > Applications`.
  2. Select the Co-browse Server.
  3. Click `Stop applications` in the Runtime panel.
- Or you can stop the server from the console window where it was started.
  1. Press `Ctrl+C` while the window is active.
  2. Type Y and press `Enter`.

End
The Genesys Co-browse Server is shown in `Stopped` status in Genesys Administrator.

Configure a Cluster of Chat Servers

> ## Important
>
> **Genesys Co-browse no longer requires Chat Server.**
>
> This functionality is now available through a single set of consumer-facing digital channel APIs that are part of Genesys Mobile Services (GMS), and through Genesys Widgets, a set of productized widgets that are optimized for use with desktop and mobile web clients, and which are based on the GMS APIs.
>
> Genesys Widgets provide for an easy integration with Co-browse, allowing you to proactively serve these widgets to your web-based customers.

Genesys Co-browse 8.5 can support either a standalone Chat Server or a cluster of Chat servers.

This chapter describes how to configure the necessary Genesys components to allow Co-browse Server to work with a cluster of Chat Server application objects.

Prerequisites

Chat Server Applications

You must have two or more configured Chat servers in order to organize them in a cluster. Each Chat Server must:

- Work in the same tenant.
- Have the same Interaction Server and Universal Contact Server in the Connections list.
- Have the same Chat inbound queue configured in its `endpoints:<tenant_id>` section.
- Have a `webapi` port. This port **must** be configured in the `Listening Ports` section.

Configure the Chat Server Cluster

To organize the Chat Server applications in a cluster, do **one** of the following:

- Add all of your Chat servers in the Connections list for the Co-browse Cluster application.

OR

- Add your Chat servers to an Application Cluster application object and then add this application object in the Connections list for Co-browse Cluster.

Application Cluster Object

An Application Cluster is a set of application objects united in a cluster (such as Chat servers, Email servers, and so on). This configuration object has a template of type "Application Cluster", and you can find it in the Templates directory for any Web API Server version.

Configure Co-browse Server to work with the Chat Server Cluster

Start of procedure

1. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`. Select the application defined for the Genesys Co-browse Cluster and click `Edit...`.
2. In the `Connections` section of the `Configuration` tab, click the `Add` button. The `Browse for applications` panel opens.
   - Select the Genesys application defined for the Solution Control Server, then click `OK`. Solution Control Server is added to the Connections list.
   - Select the Genesys application defined for the Chat Server, then click `OK`. Chat Server is added to the Connections list.
     Note: Add all Chat Server applications in the same way.
3. In the `Options` tab, locate the `chat` section and update the following options:
   1. Set `useChat` to `true`
   2. Set `queueKey` to `<tenant_id>:<your_Chat_inbound_queue>`
4. Click `Save & Close`.
5. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`. Select the application defined for the Genesys Co-browse Node and click `Edit...`.
6. In the Tenants section, click the `Add` button. The Tenants pane opens.
   - Select the same Tenant object you used for your Chat servers.
7. Click `Save & Close`.
8. Continue adding the Tenant to all Co-browse Node applications in the same way.

End of procedure

Website Instrumentation

You must instrument your website to enable Genesys Co-browse. This means that every page accessible by your customers must include the Co-browse JavaScript code. This code must be on the following page types:

1. Pages referred through links on the website or reachable through the address bar.
2. Pages loaded in iframes, which are hosted inside the first type of page.

The Co-browse Javascript code can be added to the web pages of any website that uses mainstream web technologies such as PHP, Java, or .NET.

document.domain

By default, the Co-browse JavaScript explicitly sets the `document.domain` property on the customer side to allow synchronization of iframes loaded from another sub-domain. Co-browse always sets the `document.domain` property to the second-level domain. You can disable Co-browse's modification of `document.domain` using the setDocumentDomain option in the JavaScript Configuration API.

If the scripts on your website also explicitly set `document.domain` and the value is different than the value set by Co-browse, one of the attempts (either from your website or Co-browse) to set `document.domain` will be overridden. This could potentially be unsafe if your site allows third-party users to create their own sub-domains because it might enable those users to get scripting access to the site.

Co-browse Proxy

You can quickly get up and running with any website by using the proxy-based approach. This approach is an easy way to test Co-browse in a lab environment without modifying your existing site; however, it has significantly lower performance in terms of page loading on the "Customer" side. For details about setting up the proxy, see Test with the Co-browse Proxy.

Document Type

Genesys Co-browse relies on modern browser features, if available, and does not work with older browsers such as Internet Explorer 8. To activate these features in browsers which support them, use the appropriate document type definition at the beginning of each page in your website.

For HTML5:

```
<!DOCTYPE html>
```

For Internet Explorer, a special meta tag can be added:

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

For more information, see the following:

- Specifying legacy document modes
- Interoperable HTML5 Quirks Mode in IE10

Basic Instrumentation

Co-browse is shipped with two JavaScript applications that each enables different functionality on your website.

- `gcb.min.js` — The default Co-browse JavaScript application, which includes Chat and Co-browse functionality.
- `genesys.min.js` — The Integrated JavaScript Application, which includes Chat, Co-browse, and Web Engagement functionality.

The rest of this section covers how to add the default Co-browse JavaScript application to your site using `gcb.min.js`. For information about how to add the Integrated JavaScript Application to your site, see the integrated instrumentation snippet.

To enable Co-browse and Chat, you must add the default Co-browse instrumentation snippet before the closing </head> tag on your web pages:

```
<script>(function(d, s, id, o) {
  var fs = d.getElementsByTagName(s)[0], e;
  if (d.getElementById(id)) return;
  e = d.createElement(s); e.id = id; e.src = o.src;
  e.setAttribute('data-gcb-url', o.cbUrl);
  fs.parentNode.insertBefore(e, fs);
})(document, 'script', 'genesys-js', {
  src: "<COBROWSE_SERVER_URL>/cobrowse/js/gcb.min.js",
  cbUrl: "<COBROWSE_SERVER_URL>/cobrowse"
});</script>
```

You can use the snippet above to enable Co-browse and Chat on your website, but make sure you update `<COBROWSE_SERVER_URL>`:

- To load the JavaScript from the Co-browse server, set the `src` parameter to the following:
  `http(s):<COBROWSE_HOST>[:<COBROWSE_PORT>]/cobrowse/js/gcb.min.js`
- To connect the JavaScript application to the Co-browse server, set the `cbUrl` parameter to the following:
  `http(s):<COBROWSE_HOST>[:<COBROWSE_PORT>]/cobrowse`

This is the URL of the Co-browse application. It may also be the URL of the load balancer or reverse proxy. To enable secure content synchronization between the Customer (end user) browser and the Co-browse Server, use an HTTPS-based URL and HTTPS port instead.

Starting with Co-browse 8.5.002+, the customer side always uses the URL in the JS instrumentation for css-proxy and url-proxy.

JavaScript does not contain private personal information and can be loaded using HTTP. There are pitfalls in both cases that must be taken into account.

> **Warning**
>
> If a website is HTTPS-based, the browser might block JavaScript loaded/executed using HTTP.

WebSockets

With WebSockets enabled, Genesys Co-browse uses either WebSockets (ws://) or secure WebSockets (wss://) depending on the protocol of your **cbUrl**. If the **cbUrl** uses `http://` then Co-browse uses `ws://` and Co-browse uses `wss://` when the **cbUrl** uses `https://`. When using `https://` in the **cbUrl**, you do not need any additonal configuration to use secure WebSockets.

Example Instrumentation

Here's an example with values set for the `src` and `cbUrl` parameters:

```
<script>(function(d, s, id, o) {
  var fs = d.getElementsByTagName(s)[0], e;
  if (d.getElementById(id)) return;
  e = d.createElement(s); e.id = id; e.src = o.src;
  e.setAttribute('data-gcb-url', o.cbUrl);
  fs.parentNode.insertBefore(e, fs);
})(document, 'script', 'genesys-js', {
  src: "http://192.168.67.39:9700/cobrowse/js/gcb.min.js",
  cbUrl: "http://192.168.67.39:9700/cobrowse"
});</script>
```

The basic instrumentation snippet in the examples above is also part of the default instrumentation for the proxies (GWM and ZAP) that are included in the Co-browse Server installation package. Note that in the proxies <COBROWSE_SERVER_URL> is set to `localhost:8700`.

> **Tip**
>
> For more information about how to test the Co-browse solution using the proxy, refer to the Test with the Co-browse Proxy.

Built-in Chat Deprecation

As of Co-browse 8.5.100.11, support for the built-in chat is deprecated and the built-in chat button is hidden by default. You should use Genesys Widgets instead of the built-in chat widget.

To enable the depracated built-in chat button, set the following:

```
<script>
var _genesys = {
  buttons: {
   chat: true
  }
};
</script>
```

Enabling Console Logs

All logging for the Co-browse JavaScript apps is turned off by default, but it can be enabled on both the customer side and agent side.

Enabling Console Logs on the Customer Side

This is done via the **debug** configuration option. Add this script before your instrumentation:

```
<script>
var _genesys = {
        debug: true
};
</script>
<script>(function(d, s, id, o) {
```

Enabling console logs on the agent side

Add the debug=1 parameter to the URL. For example:
`http://cobrowse:8700/cobrowse/slave.html#sid=123456789&debug=1`.

Advanced Instrumentation

To customize instrumentation and configuration of Co-browse, see the Co-browse JavaScript API.

Configure a Cluster of Co-browse Servers

Genesys Co-browse supports load balancing using Stickiness.

Load balancing is enabled by configuring a cluster of Co-browse Servers. Cassandra is embedded in Genesys Co-browse Server, so when you set up a cluster of Genesys Co-browse servers, each server may also act as a Cassandra node. You configure the Cassandra nodes by setting configuration options in the `cassandraEmbedded` section of the Co-browse Server application.

Complete the following steps to implement load balancing:

1. Set up 2 or more Co-browse Server nodes

> Tip
>
> To determine the how many nodes your Co-browse cluster needs, use the Genesys Co-browse Sizing Calculator.

8.5.000

For Co-browse 8.5.000, you must set up a cluster of Co-browse Servers to enable load balancing. For each Co-browse Server in your planned cluster, complete the procedures on the Install Genesys Co-browse Server page.

If the Co-browse servers reside on the same machine, you must configure the applications to use different ports for Jetty and the embedded Cassandra. This will prevent port conflicts among your Co-browse Servers.

> Important
>
> Every Co-browse Server in the cluster generally plays the same role as the others, except some embedded Cassandra nodes act as seed nodes. This means that to see consistent behavior on the cluster, regardless of which server serves requests, all Co-browse Servers should have the same options set in their application objects in Configuration Server. The rule of thumb is to configure the cluster servers the same, unless it is absolutely necessary to do otherwise (for example, a port is busy on a machine). This simplifies maintenance of production deployments.

8.5.001+

For Co-browse 8.5.001+, you must set up a cluster of Co-browse Nodes to enable load balancing. To do this, complete the procedures to create Application objects for a Co-browse Cluster and Co-browse Nodes. Follow the installation steps outlined in the **8.5.001+** tab in the Creating the Co-browse Server Application Object in Genesys Administrator section.

If the Co-browse Nodes reside on the same machine, you must configure the applications to use different ports for Jetty and the embedded Cassandra. This will prevent port conflicts among your Co-browse Nodes.

2. Configure the Cassandra cluster

Configure the Cassandra cluster

Prerequisite: You have completed Set up 2 or more Co-browse Servers.

External Cassandra Cluster Setup

> Important
>
> For supported versions of Cassandra, see Genesys Co-browse in the Supported Operating Environment Reference Guide.

External Cassandra cluster deployment is thoroughly described in the official Cassandra documentation. See the following:

- Planning a cluster deployment
- Initializing a cluster
- External Cassandra Access Configuration

Embedded Cassandra Cluster Setup

> Important
>
> Starting in 8.5.0, Embedded Cassandra mode is deprecated in Genesys Co-browse; support for this mode will be discontinued in 9.0.

An Embedded Cassandra cluster is setup similarly to an external Cassandra except that embedded Cassandra node settings are provided either through Configuration Server options or an external `cassandra.yaml` file.

Start of procedure

Complete the steps below for each Co-browse application you created in Set up 2 or more Co-browse Servers:

1. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`.
2. Select the Co-browse application and click `Edit`.
3. In the `Options` tab, locate the `cassandraEmbedded` section and update the following options:
   1. listenAddress — Set this value to the IP of the node (listenAddress in cassandra.yml).
   2. rpcAddress — Set this value to the IP of the node (rpcAddress in cassandra.ymal).
   3. seedNodes — Set this value to the IP of the first node (seedNodes in cassandra.yaml).
   4. clusterName (optional) — This name should be the same for each node (name in cassandra.yaml).
4. Click `Save & Close`.

End of procedure

Replication Strategy

By default, Co-browse server activates *NetworkTopologyStrategy* as a replication strategy. *NetworkTopologyStrategy* is recommended for production Cassandra deployments and is supported by *GossipingPropertyFileSnitch*. *GossipingPropertyFileSnitch* relies on the `cassandra-rackdc.properties` file. Make sure the data center data center names defined in this file (one for each Cassandra node) correspond to data center names defined in the replicationStrategyParams option.

The `cassandra-rackdc.properties` file location depends on the type of Cassandra cluster:

- External Cassandra—see the `conf` subdirectory of the Cassandra installation directory. For more information, see http://docs.datastax.com/en/cassandra/2.0/cassandra/architecture/architectureSnitchGossipPF_c.html.
- Embedded Cassandra—see the `resources` subdirectory of the server directory. For example, `<Co-browse Server Install Directory>/server/resources`.

Replication Factor

Replication factor configures the number of copies of data to keep in the cluster. Typically, three copies is enough for most scenarios (provided you have at least three nodes in your cluster). The replication factor can be increased to achieve higher redundancy levels. Set the replication factor in the replicationStrategyParams option to a number less than or equal to the number of nodes.

3. Verify the status of the Cassandra cluster

Prerequisite: You have a separate installation of Cassandra 2.x (2.1.3+, the same version used in Co-browse).

1. Start the first node and wait until it starts listening.
2. Start all other nodes in your cluster.
3. Open a command line and run `<cassandra home>\bin\cassandra-cli.bat -h <ip of first node> -p <cassandra rpcPort>`, where `<ip of first node>` is the IP of the first node in your cluster and `<cassandra rpcPort>` is the value you configured for rpcPort.
4. Enter the following command: `describe cluster`. The output should look similar to the following:

```
[default@unknown] describe cluster;
Cluster Information:
 Snitch: org.apache.cassandra.locator.SimpleSnitch
 Partitioner: org.apache.cassandra.dht.RandomPartitioner
 Schema versions:
 6c960880-1719-11e3-0000-242d50cf1fbf: [192.168.00.1, 192.168.00.2, 192.168.00.3]
```

The list of IP address in square brackets (`[192.168.00.1, 192.168.00.2 ...]`) should match all the nodes in your cluster.

End of procedure

4. Configure the load balancer

See Configuring a Load Balancer for Co-browse Cluster for details about configuring the load balancer and sample configurations for Nginx and Apache.

> ### Tip
> In Co-browse 8.5.002+, the agent side uses the cluster URL while the end user (master) side uses the URL in the Website Instrumentation. You can have two load balancers, an internal load balancer for agents which you specify in the cluster URL option and a public load balancer for end users to use in the JS instrumentation. Depending on your infrastructure's setup, two load balancers may benefit traffic.

5. Modify the website instrumentation

You must modify the URLs in your Co-browse instrumentation scripts to point to your configured load balancer. See Website Instrumentation for details about modifying the script.

> Important
>
> Starting with Co-browse 8.5.002+, the consumer (master) side always uses the URL in the JS instrumentation for css-proxy and url-proxy.

If you are using the Co-Browse proxy to instrument your site, you will need to modify the URLs in the in proxy's `map.xml` file. See Test with the Co-browse Proxy for details about modifying the xml file.

> Warning
>
> The Co-browse proxy should only be used in a lab environment, not in production.

6. Modify the agent side and controller configuration

Configure the Co-browse Server applications

- 8.5.000—Modify the `url` option in the `cluster` section of all your Co-browse Server applications.
- 8.5.001—Modify the `url` option in the `cluster` section of your Co-browse Cluster application.

See the `cluster` section for details.

> Tip
>
> In Co-browse 8.5.002+, the agent side uses the cluster URL while the end user (master) side uses the URL in the Website Instrumentation. You can have two load balancers, an internal load balancer for agents which you specify in the cluster URL option and a public load balancer for end users to use in the JS instrumentation. Depending on your infrastructure's setup, two load balancers may benefit traffic.

You must also set up a similar configuration for the Genesys Co-browse Plug-in for Workspace

Desktop Edition. To support this, you might consider setting up two load balancers:

- public — This load balancer should have a limited set of Co-browse resources. For example, it should not include session history resources.
- private — This load balancer should have all Co-browse resources and it should be placed in the network so that it is accessible only from the corporate intranet. It should only be used for internal applications, such as Workspace Desktop Edition.

Complete the procedure below to configure the plug-in to support the Co-browse cluster:

Configure the Co-browse Plug-in for Workspace Desktop Edition

See Configuring Workspace Desktop Edition to allow the Plug-in to work with co-browsing.

7. Configure Genesys Workspace Web Edition

If you use Workspace Web Edition on the agent side, you must configure it to work with Co-browse. For instructions, see Configure Genesys Workspace Web Edition to Work with Co-browse.

8. Launch and test

To test your set-up, create a Co-browse session, join it as an agent and do some co-browsing. If you can do this, your configuration was successful.

End of procedure

What's New in Cassandra Configuration for 8.5.0?

This page describes changes in Cassandra configuration between Genesys Co-browse 8.5.0 and 8.1.3.

Overview of Changes in Cassandra Access and Management

- Co-browse server can now be interconnected with an external Cassandra cluster.
- Co-browse server now uses a new approach to configuring embedded Cassandra.
- Genesys Co-browse 8.5.0 splits the configuration of the embedded Cassandra node and the Co-browse keyspace.
- Genesys Co-browse configuration is now similar to configuration of GWE and UCS.
- Co-browse 8.5.0 now uses Cassandra 2.X. For supported versions of Cassandra, see Genesys Co-browse in the Supported Operating Environment Reference Guide.

Co-browse Keyspace Configuration

Keyspace specific options are kept in a dedicated configuration section, cassandraKeyspace. These options apply to both embedded and external Cassandra.

Embedded Cassandra Configuration

> **Important**
>
> Starting in 8.5.0, Embedded Cassandra mode is deprecated in Genesys Co-browse; support for this mode will be discontinued in 9.0.

Co-browse server may act as a Cassandra node. The options to configure the embedded Cassandra service changed in 8.5.0. These options are now in a dedicated section, cassandraEmbedded.

Cassandra Cluster Access Configuration

Embedded Cassandra Access Configuration

When embedded Cassandra is enabled, the Co-browse server will always connect to the embedded

Cassandra node when it needs to read or write data in the database. No additional configuration needed.

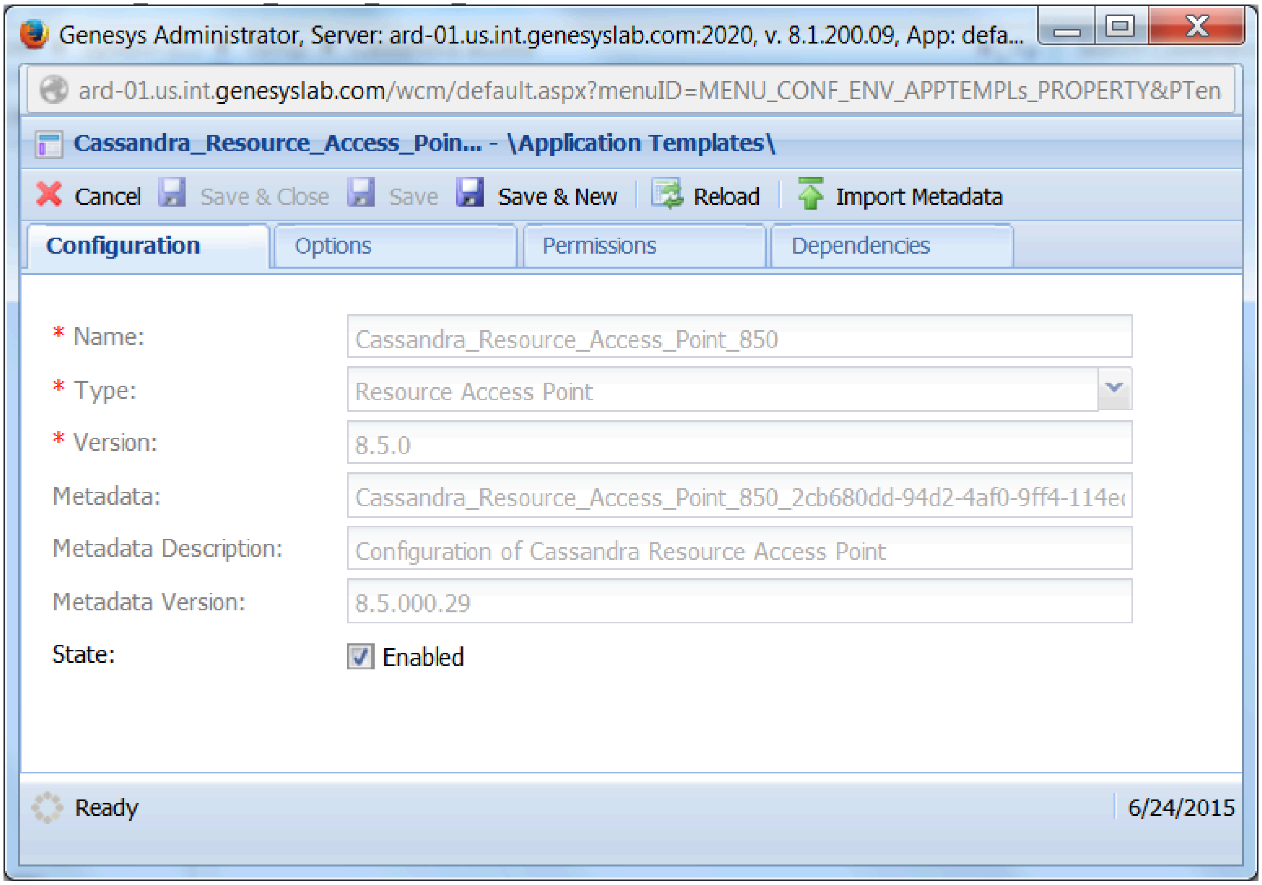
External Cassandra Access Configuration

In Co-browse 8.5.0, you can use a dedicated Cassandra Resource Access Point in Configuration Server to link a Co-browse server to an external Cassandra cluster.

Procedure: Create a Dedicated Cassandra Resource Access Point

Start of Procedure

1. Import the templates `Cassandra_Resource_Access_Point_850.apd` and `Cassandra_Resource_Access_Point_850.xml`


Genesys Administrator, Server: ard-01.us.int.genesyslab.com:2020, v. 8.1.200.09, App: defa...
ard-01.us.int.genesyslab.com/wcm/default.aspx?menuID=MENU_CONF_ENV_APPTEMPLs_PROPERTY&PTen
Cassandra_Resource_Access_Poin... - \Application Templates\
Cancel
Save & Close
Save
Save & New
Reload
Import Metadata
Configuration
Options
Permissions
Dependencies
Ready
6/24/2015

* Name: Cassandra_Resource_Access_Point_850

* Type: Resource Access Point

* Version: 8.5.0

Metadata: Cassandra_Resource_Access_Point_850_2cb680dd-94d2-4af0-9ff4-114e

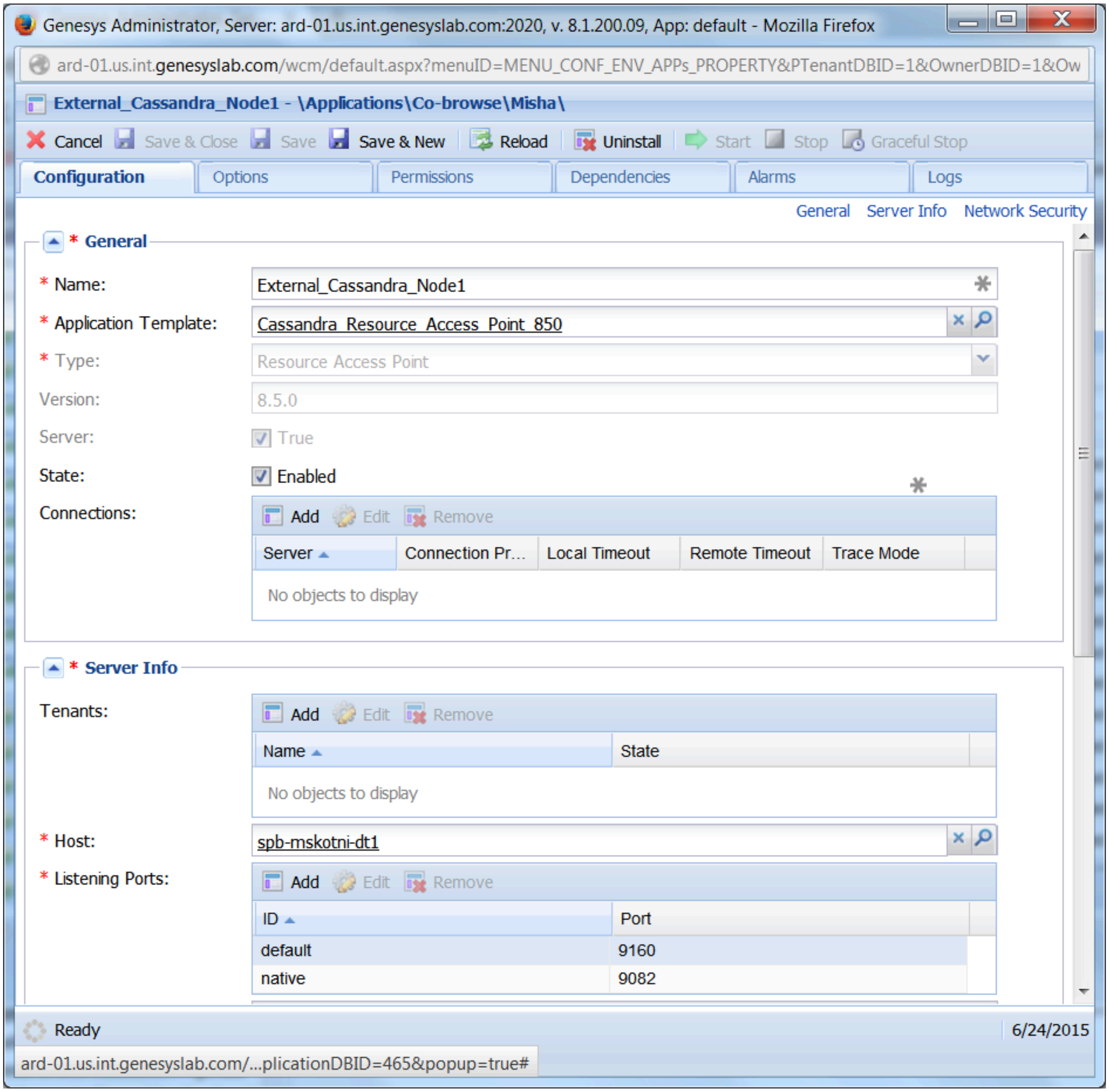
Metadata Description: Configuration of Cassandra Resource Access Point

Metadata Version: 8.5.000.29

State: ☑ Enabled

2. Using the imported application template from the previous step, create one Cassandra Resource Access Point(RAP) for each Cassandra node in an external Cassandra cluster that the Co-browse server needs to communicate with. Configure the following:
   1. For `Host`, specify the host of the external Cassandra Node
   2. Add a `default` port with the value of the rpc port the Cassandra node is using to listen for Thrift client connections. Optionally, specify `rpc` protocol for the port.

3. Add a `native` port with the value of the CQL native port the Cassandra node is using to listen for CQL client connections. Optionally, specify `native` protocol for the port.


Genesys Administrator, Server: ard-01.us.int.genesyslab.com:2020, v. 8.1.200.09, App: default - Mozilla Firefox
ard-01.us.int.genesyslab.com/wcm/default.aspx?menuID=MENU_CONF_ENV_APPs_PROPERTY&PTenantDBID=1&OwnerDBID=1&Ow
External_Cassandra_Node1 - \Applications\Co-browse\Misha\
Cancel
Save & Close
Save
Save & New
Reload
Uninstall
Start
Stop
Graceful Stop
Configuration
Options
Permissions
Dependencies
Alarms
Logs
General
Server Info
Network Security
* General
Add
Edit
Remove
* Server Info
Ready
6/24/2015
ard-01.us.int.genesyslab.com/...plicationDBID=465&popup=true#

* Name:	External_Cassandra_Node1
* Application Template:	Cassandra_Resource_Access_Point_850
* Type:	Resource Access Point
Version:	8.5.0
Server:	☑ True
State:	☑ Enabled

Connections:

Server	Connection Pr...	Local Timeout	Remote Timeout	Trace Mode
No objects to display				

Tenants:

Name	State
No objects to display	

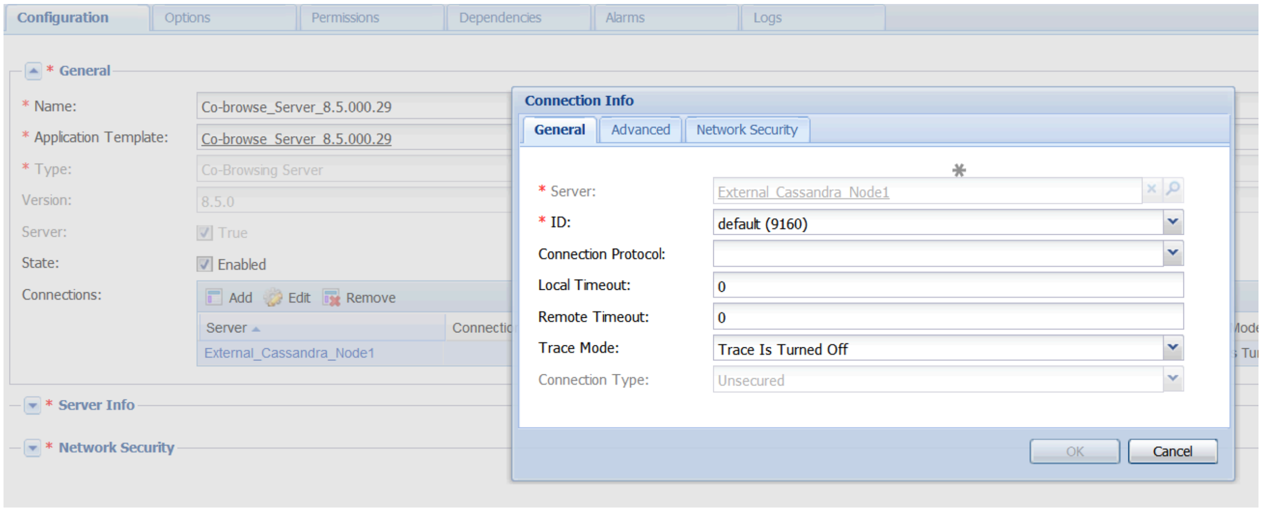
* Host: spb-mskotni-dt1

* Listening Ports:

ID	Port
default	9160
native	9082

3. Configure Cassandra RAP Connections:
    - 8.5.000:
        - Add Cassandra RAP connections to one or more of the Co-browse Server applications you created.
        - Set the `enabled` option to `false` in the cassandraEmbedded section of each Co-browse server application in the cluster.
    - 8.5.001+:

- Add Cassandra RAP connections to the Co-browse Cluster application object.
- Set the enabled option to false in the cassandraEmbedded section of the Co-browse Cluster application object.



* Name: Co-browse_Server_8.5.000.29
* Application Template: Co-browse_Server_8.5.000.29
* Type: Co-Browsing Server
Version: 8.5.0
Server: ☑ True
State: ☑ Enabled
Connections: External_Cassandra_Node1

* Server: External_Cassandra_Node1
* ID: default (9160)
Connection Protocol:
Local Timeout: 0
Remote Timeout: 0
Trace Mode: Trace Is Turned Off
Connection Type: Unsecured

Several connections with different Cassandra RAP applications ensures a redundancy of connections to the external Cassandra cluster. If one Cassandra node in the cluster fails, Co-browse server will be able to cooperate with the external Cassandra cluster through a different Cassandra node.

End of Procedure

Configuring a Cassandra Cluster

For more information on configuring a Cassandra Cluster, see Configure a Cluster of Co-browse Servers.

Configuring a Load Balancer for Co-browse Cluster

Load Balancer Requirements

When configuring a load balancer, note the following requirements:

- You must use a third-party HTTP load balancer. Genesys cannot provide or validate a third-party load balancer.
- The load balancer must support health check monitoring of each node. A failed Co-browse node cannot recover gracefully. The load balancer must detect node failure to notify the client and allow a manual restart of the session.
- The load balancer must also support cookie based session stickiness. Genesys components add the `gcbSessionServer` cookie to HTTP requests and you should configure the load balancer to distribute requests to the appropriate Co-browse node based on the cookie value.
- We highly recommend WebSocket support. See also, Sizing.
- If WebSocket support is enabled, the load balancer must be able to balance HTTP requests and WebSocket connections at the same time to properly handle scenarios where the end user's browser or your infrastructure does not support WebSockets.
- SSL decryption—Co-browse relies on application-generated cookie headers and the load balancer must have access to HTTP headers. If incoming traffic uses HTTPS, the load balancer must be able to decrypt HTTPS traffic and access the cookie headers used for session stickiness. The resulting traffic going from the load balancer to the Co-browse server can be re-encrypted (SSL Bridging/Re-encryption) or remain in HTTP. Keeping the traffic in HTTP reduces the load on the Co-browse server (SSL Offloading).

WebSocket Support

To achieve the best performance with Co-browse, Genesys highly recommends you configure WebSocket support for your load balancer. WebSockets improve performance and considerably reduce the request throughput rate of each session. If WebSockets are unavailable, Co-browse still functions but uses other transports that perform significantly slower. In some cases, WebSockets become mandatory to ensure proper order of DOM change events as Co-browse server does not provide native ordering of DOM change events.

If your load balancer does not support WebSockets and you do not want to wait for Co-browse to automatically switch to another transport, you can use the `disableWebSockets` options for the customer side and the agent side. For more information, see JavaScript Configuration API and disablewebsockets configuration.

Load Balancer Configuration

Your load balancer configuration will depend upon which load balancer you implement. See the examples below for sample configurations of Nginx and Apache.

All proposed examples assume cookie-based stickiness. If you use URL-based stickiness and the actual nodes are not publicly accessible, you may want to add logic to route publicly accessible URLs of Co-browse nodes (such as `http://<load-balancer>?co-browse-node-id={node-id}`) to the actual nodes. However, such configuration is beyond the scope of this Guide. For the details about cookie-based and URL-based stickiness supported by Co-browse Solution, see Stickiness.

> **Important**
>
> Due to 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 being from third parties and the solution will not work. Users will not be able to start chat nor begin co-browsing.

High Availability and Health Checks

Currently, there is no fail-over support for Co-browse sessions. If a Co-browse Server node becomes inaccessible, Co-browse live sessions hosted by this server terminate. To notify clients (agent desktop and end user's browser application) that a session has ended you must implement healthchecks/ fallback functionality in your Load Balancer. You must configure your LB to route all requests that go to a failed node to another node. This node will detect it does not *own* the Co-browse sessions and terminate them, sending notifications to the clients. After sessions are terminated, agents and customers can manually establish new Co-browse sessions.

> **Important**
>
> Built-in chat *does* support fail-over. If you use built-in chat, chat session stay active and agents join new Co-browse sessions automatically.

> **Important**
>
> You can use the /cobrowse/health HTTP resource for health checks.

Nginx Configuration Samples

Below are two sample configurations for load balancing with Nginx:

- The first sample keeps connections secure with HTTPS both **from browsers to load balancer** and **from load balancer to servers**.
- The second sample uses the SSL Acceleration technique, where HTTPS is used only from the browsers to the load balancers; plain HTTP is used from the load balancer to the Co-browse servers.

> **Important**
>
> These configurations are intended to be examples and might not represent best practices for Nginx configuration.

> **Important**
>
> These configurations use a five second timeout for High Availability, if a server dies, the load balancer switches the client to another server after five seconds. In production, you can eliminate this timeout by using "health checks" functionality, available in Nginx PLUS or through third-party plug-ins. See the following links for more information:
>
> - http://nginx.com/products/application-health-checks/
> - http://wiki.nginx.org/NginxHttpHealthcheckModule
> - https://github.com/cep21/healthcheck_nginx_upstreams
> - https://github.com/yaoweibin/nginx_upstream_check_module

Sample One for Nginx

```
# Basic configuration for load balancing 2 or more Co-Browse servers.
# All nodes are listed 2 times: in upstream and map directives.
# Co-browse applications are responsible for setting the "gcbSessionServer" cookie
# with one of the values listed in map directive. These values are names of
# applications in config server.
# This (default) variant uses HTTPS (if browser request is HTTPS) for connections
# both from browser to load balancer and from load balancer to Co-Browse servers.
# For another version with HTTPS only from browser to LB, see nginxSSLAccelerated.conf

# IMPORTANT!
# This configuration is not intended for production use!
# It is mere example of how this functionality can be achieved.

events {
    worker_connections  1024;
}
```

```
http {
    include       mime.types;
    default_type  application/octet-stream;
    # to handle longer names of Co-browse server applications
    map_hash_bucket_size 64;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for" "$upstream_addr"';

    access_log  logs/nginx_access.log main;
    error_log logs/nginx_error.log warn;

    upstream http_cobrowse_cluster {
        server 192.168.73.210:8700 fail_timeout=5s;
        server 192.168.73.210:8701 fail_timeout=5s;
    }
    upstream https_cobrowse_cluster {
        server 192.168.73.210:8743 fail_timeout=5s;
        server 192.168.73.210:8744 fail_timeout=5s;
    }

    map $cookie_gcbSessionServer $http_sticky_backend {
        default 0;
        .CB_Server_1   192.168.73.210:8700;
        .CB_Server_2   192.168.73.210:8701;
    }
    map $cookie_gcbSessionServer $https_sticky_backend {
        default 0;
        .CB_Server_1   192.168.73.210:8743;
        .CB_Server_2   192.168.73.210:8744;
    }

    map $http_upgrade $connection_upgrade {
        default upgrade;
        ''      close;
    }

    server {
        listen 8080;
        listen 8083 ssl;
        ssl_certificate cobrowse.unsigned.crt;
        ssl_certificate_key cobrowse.unsigned.key;

        location @fallback {
            proxy_pass http://http_cobrowse_cluster;
        }

        location /cobrowse {
            # Allow websockets, see http://nginx.org/en/docs/http/websocket.html
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection $connection_upgrade;

            # Increase buffer sizes to find room for DOM and CSS messages
            proxy_buffers 8 2m;
            proxy_buffer_size 10m;
            proxy_busy_buffers_size 10m;

            # If Co-browse server doesn't respond in 5 seconds, consider it dead
            # (a 504 will fire and be caught by error_page directive for fallback).
            # This timeout can be eliminated using "health checks" functionality
            # available in Nginx PLUS or via 3rd party plugins. See the following links:
```

```
            # http://nginx.com/products/application-health-checks/
            # http://wiki.nginx.org/NginxHttpHealthcheckModule
            # https://github.com/cep21/healthcheck_nginx_upstreams
            # https://github.com/yaoweibin/nginx_upstream_check_module
            proxy_connect_timeout 5s;

            # Fall back if server responds incorrectly
            error_page 502 = @fallback;
            # or if doesn't respond at all.
            error_page 504 = @fallback;

            # Create a map of choices
            # see https://gist.github.com/jrom/1760790
            if ($scheme = 'http') {
                set $test HTTP;
            }
            if ($scheme = 'https') {
                set $test HTTPS;
            }
            if ($http_sticky_backend) {
                set $test "${test}-STICKY";
            }

            if ($test = HTTP-STICKY) {
                proxy_pass http://$http_sticky_backend$uri?$args;
                break;
            }
            if ($test = HTTPS-STICKY) {
                proxy_pass https://$https_sticky_backend$uri?$args;
                break;
            }
            if ($test = HTTP) {
                proxy_pass http://http_cobrowse_cluster;
                break;
            }
            if ($test = HTTPS) {
                proxy_pass https://https_cobrowse_cluster;
                break;
            }



            return 500 "Misconfiguration";
        }

    }

}
```

Sample Two for Nginx

```
# Basic configuration for load balancing 2 or more Co-browser servers.
# Nodes are listed 2 times: in upstream and map directives.
# Co-browse applications are responsible for setting the "gcbSessionServer" cookie
# with one of the values listed in map directive. These values are names of
# applications in config server.
# Note that this version uses "SSL acceleration" (http://en.wikipedia.org/wiki/
SSL_Acceleration,
# http://en.wikipedia.org/wiki/Load_balancing_(computing)#Load_balancer_features):
# load balancer terminated SSL connections, passing HTTPS requests as HTTP to the servers.

# IMPORTANT!
# This configuration is not intended for production use!
# It is mere example of how this functionality can be achieved.

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  logs/nginx_access.log  main;
    error_log logs/nginx_error.log debug;

    upstream http_cobrowse_cluster {
        server 192.168.73.210:8700 fail_timeout=5s;
        server 192.168.73.210:8701 fail_timeout=5s;
    }

    map $cookie_gcbSessionServer $sticky_backend {
        default 0;
        .CB_Server_1   192.168.73.210:8700;
        .CB_Server_2   192.168.73.210:8701;
    }

    map $http_upgrade $connection_upgrade {
        default upgrade;
        ''      close;
    }

    server {
        listen 8080;
        listen 8083 ssl;
        ssl_certificate cobrowse.unsigned.crt;
        ssl_certificate_key cobrowse.unsigned.key;

        location @fallback {
            proxy_pass http://http_cobrowse_cluster;
        }

        location /cobrowse {
            # Allow websockets, see http://nginx.org/en/docs/http/websocket.html
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection $connection_upgrade;

            # Increase buffer sizes to find room for DOM and CSS messages
            proxy_buffers 8 2m;
            proxy_buffer_size 10m;
            proxy_busy_buffers_size 10m;

            # If Co-browse server doesn't respond in 5 seconds, consider it dead
            # (a 504 will fire and be caught by error_page directive for fallback)
            # This timeout can be eliminated using "health checks" functionality
            # available in Nginx PLUS or via 3rd party plugins. See the following links:
            # http://nginx.com/products/application-health-checks/
            # http://wiki.nginx.org/NginxHttpHealthcheckModule
            # https://github.com/cep21/healthcheck_nginx_upstreams
            # https://github.com/yaoweibin/nginx_upstream_check_module
            proxy_connect_timeout 5s;

            # Fall back if server responds incorrectly
            error_page 502 = @fallback;
            # or if doesn't respond at all.
            error_page 504 = @fallback;

            if ($sticky_backend) {
                proxy_pass http://$sticky_backend$uri?$args;
            }
            proxy_pass http://http_cobrowse_cluster;
        }

    }

}
```

Apache Configuration Samples

Below are two sample configurations for load balancing with Apache (both without WebSockets support):

- The first sample uses an insecure connection with HTTP both from browsers to the load balancer and from the load balancer to the servers.
- The second sample uses the SSL Acceleration technique, where HTTPS is used only from the browsers to the load balancer. Plain HTTP is used from the load balancer to the Co-browse servers.

> **Important**
>
> These configurations are intended to be examples and might not represent best practices for Apache configuration.

Prerequisites for both samples

- If you are using a proxy to inject the instrumentation snippet into your site, you must exclude the load balancer host from proxying. Otherwise Apache configuration will work incorrectly in some cases such as when IE9 is a customer browser.
- Disable WebSockets for the customer side and the agent side. For more information, see JavaScript Configuration API#disableWebSockets and disableWebSockets configuration.

Sample One for Apache

```
Listen APACHE_PORT_1

#Load Balancer of Co-browse Servers
<VirtualHost *:APACHE_PORT_1>
    ProxyRequests Off
    <Proxy balancer://CLUSTER_NAME>
```

```
        BalancerMember http://<HOST_1>:<PORT_1>/cobrowse route=<CO-BROWSE_SERVER_1_APP_NAME>
        BalancerMember http://<HOST_2>:<PORT_2>/cobrowse route=<CO-BROWSE_SERVER_2_APP_NAME>
        BalancerMember http://<HOST_3>:<PORT_3>/cobrowse route=<CO-BROWSE_SERVER_3_APP_NAME>
        ProxySet stickysession=gcbSessionServer
    </Proxy>
    ProxyPass /cobrowse balancer://CLUSTER_NAME
</VirtualHost>
```

Sample Two for Apache

```
Listen 8090
Listen 8093

#Load Balancer of Co-browse Servers
ProxyRequests Off
<Proxy balancer://cluster_cobrowse>
    BalancerMember http://co-browse_host_1:8700/cobrowse route=CB_Server_1
    BalancerMember http://co-browse_host_2:8700/cobrowse route=CB_Server_2
    BalancerMember http://co-browse_host_3:8700/cobrowse route=CB_Server_3
    ProxySet stickysession=gcbSessionServer
</Proxy>
ProxyPass /cobrowse balancer://cluster_cobrowse

NameVirtualHost *:8090
NameVirtualHost *:8093

<VirtualHost *:8090>
    ServerName apache_server_name
</VirtualHost>

<VirtualHost *:8093>
    ServerName apache_server_name

    SSLEngine on
    SSLCertificateFile "cert.crt"
    SSLCertificateKeyFile "priv_key_pkcs8.pem"
    SSLCACertificateFile "CA.crt"
</VirtualHost>
```

Test with the Co-browse Proxy

Genesys Co-browse includes the ZAProxy development tool that enable you to test Co-browse without adding the JavaScript code snippet to your website. Once you have configured the proxy, you can launch it and start the Co-browse Server.

The pages below provide details about how to configure, start, and use the proxy:

- ZAProxy — The Zed Attack Proxy is based on the OWASP Zed Attack Proxy Project.
- Security Testing with ZAProxy — In addition to acting as a proxy, the ZAProxy can also validate vulnerabilities in your website.

> **Warning**
>
> Genesys Co-browse no longer supports GWM Proxy.

ZAProxy

The Zed Attack Proxy (ZAProxy) included in the Co-browse Server installation package is based on the OWASP Zed Attack Proxy Project.

ZAProxy can run in two modes:

- UI-less ZAProxy—can only be used as a proxy injecting web site with the instrumentation snippet.
- UI-based ZaProxy—in addition to acting as a proxy, the ZAProxy also provides a UI for validating the vulnerabilities in your website. For details, see SecurityTesting#Security Testing with ZAProxy.

Start and Configure ZAProxy

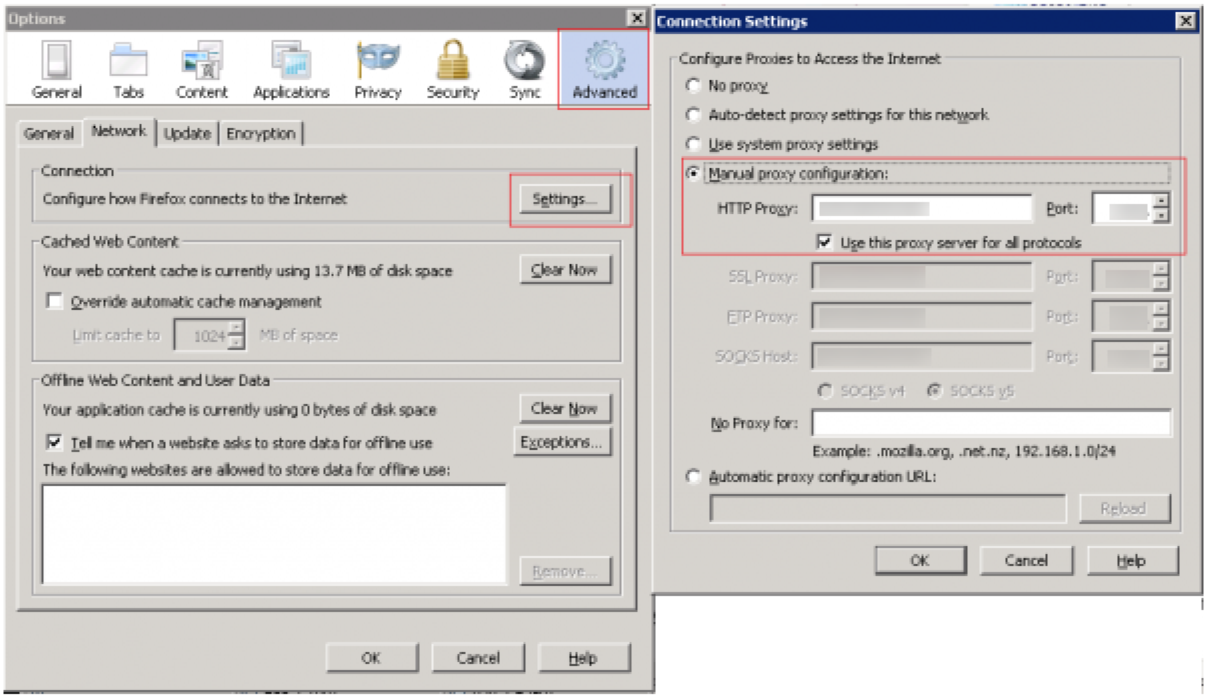
- Start and Configure UI-less ZAProxy
- Start and Configure UI-based ZaProxy

Set up your Web Browser

After you configure either UI-less ZAProxy or UI-based ZaProxy, set up your Web Browser to use ZAProxy:

Start

1. Start your web browser.
2. Open your Internet settings. For instance, in Firefox, select `Tools > Options`. The `Options` dialog window appears.
3. Select `Advanced` and in the `Network` tab, click `Settings...`. The `Connection Settings` dialog window opens.
4. Select the `Manual proxy configuration` option and do the following:
   - Enter your host IP address in the HTTP Proxy text box.
   - Enter the port used by the ZAProxy in the Port text box. This is the port you made note of in Configure ZAProxy Host and Port.
   - Select the `Use this proxy server for all protocols` option.



Connection

Configure how Firefox connects to the Internet

Cached Web Content

Your web content cache is currently using 13.7 MB of disk space

□ Override automatic cache management

Limit cache to 1024 MB of space

Offline Web Content and User Data

Your application cache is currently using 0 bytes of disk space

☑ Tell me when a website asks to store data for offline use

The following websites are allowed to store data for offline use:

Configure Proxies to Access the Internet

○ No proxy

○ Auto-detect proxy settings for this network

○ Use system proxy settings

● Manual proxy configuration:

HTTP Proxy: Port:

☑ Use this proxy server for all protocols

SSL Proxy: Port:

FTP Proxy: Port:

SOCKS Host: Port:

○ SOCKS v4 ● SOCKS v5

No Proxy for:

Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL:

ZAProxy used in Firefox

- In the "No Proxy for:" text box, list the IP address or domain name as it appears in the `data-gcb-url` attribute of the Co-browse JavaScript (see Basic Instrumentation). This ensures that communication with Co-browse server is not proxied. **Note:** If the proxy and Co-browser Server are running on the same machine, this value will be the same as the IP in the HTTP Proxy text box.

5. Click OK. Now your browser is using the ZAProxy, which will inject the Co-browse JavaScript code into all web pages except those you specified in Configure the URL Filter.

End

UI-less ZAProxy

> **Important**
>
> The ZAProxy requires JDK 1.7 or higher. If there are several Java installations and the system-wide Java is not Java 7+, you should explicitly specify the path to the required Java installation in the **zap.bat** (Windows) or **zap.sh** (Linux) file.

Start/Stop the Proxy

Start the Proxy

Navigate to your Co-browse Server installation directory and launch **proxy.bat** (on Windows) or **proxy.sh** (on Linux). The proxy starts in UI-less mode.

Stop the Proxy

To stop the ZAProxy, press **CTRL+C**.

Configure ZAProxy Host and Port

The **proxy.bat/proxy.sh** file starts the proxy using the default host name and port `15001`. If the UI-based ZAP was never started on the host, the default host name is `localhost`. Otherwise, modify the host name and port using the ZAProxy UI.

If it is necessary to change the host name or port number the proxy uses to start, you must updated the **proxy.bat/proxy.sh** file correspondingly:

Examples:

- `zap.bat -daemon -host myfavoritehost.mydomain.com -port 15001`
- `zap.bat -daemon -host 192.167.90.10 -port 15001`

Update the Instrumentation Script

If your ZAProxy is running in UI-less mode, you can update the instrumentation snippet in the configuration file used by the plug-in to inject the web pages.

Start

1. Open the file **FilterMultiReplaceResponseBody.xml** located in the `<Co-browse Server installation>/tools/proxy/plugin` folder.
2. Update the instrumentation script.
3. Save and close.
4. Restart the Proxy.

End

Set up your Web Browser

To use the proxy you need to set up your Web Browser. See ZAProxy#Set_up_your_Web_Browser

Resolving the protocol_version error

After configuring the proxy in your browser, you may encounter the following error on some HTTPS sites:

```
ZAP Error [javax.net.ssl.SSLException]: Received fatal alert: protocol_version
```

This error happens when a site only supports older versions of the TLS protocol. To fix this error you must override some of the default ZAP configuration by updating your **proxy.bat/proxy.sh** file:

```
zap.bat -daemon -port 15001 -config connection.securityProtocolsEnabled.protocol=TLSv1
```

> ### Important
> If you encounter this error on a site you want to instrument with Co-browse, update the corresponding `clientTlsProtocols` option to TLSv1

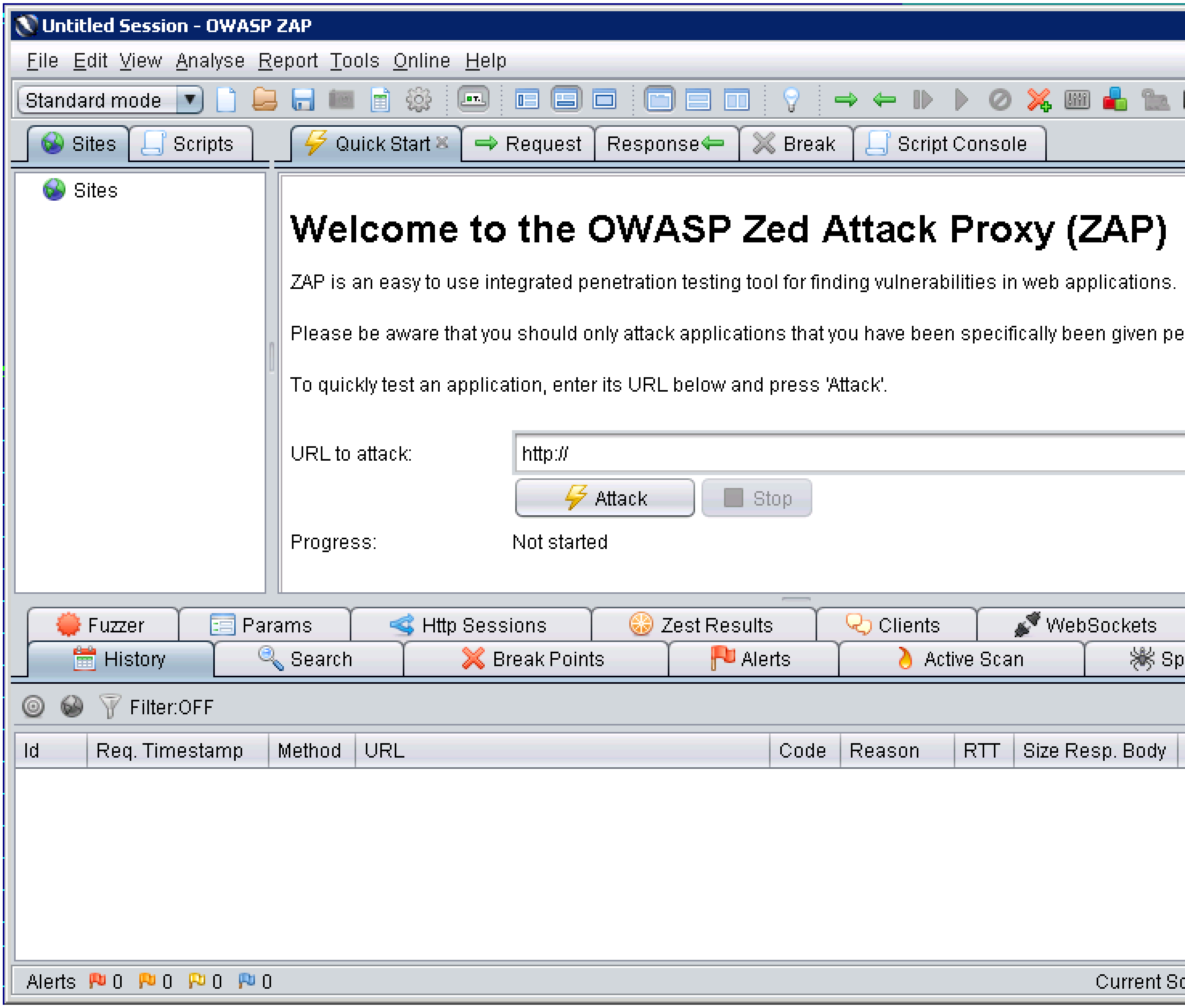
UI-based ZAProxy

> **Important**
>
> The ZAProxy requires JDK 1.7 or higher. If there are several Java installations and the system-wide Java is not Java 7+, you should explicitly specify the path to the required Java installation in the **zap.bat** (Windows) or **zap.sh** (Linux) file.

Start/Stop the Proxy

Start the Proxy

Navigate to your Co-browse Server installation directory and launch **tools\zapproxy\zap.bat** (on Windows) or **tools\zapproxy\zap.sh** (on Linux). The proxy starts and opens the UI, which you can use to configure proxy settings, update the instrumentation script, and test the security of your site.



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given pe

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://

Progress: Not started

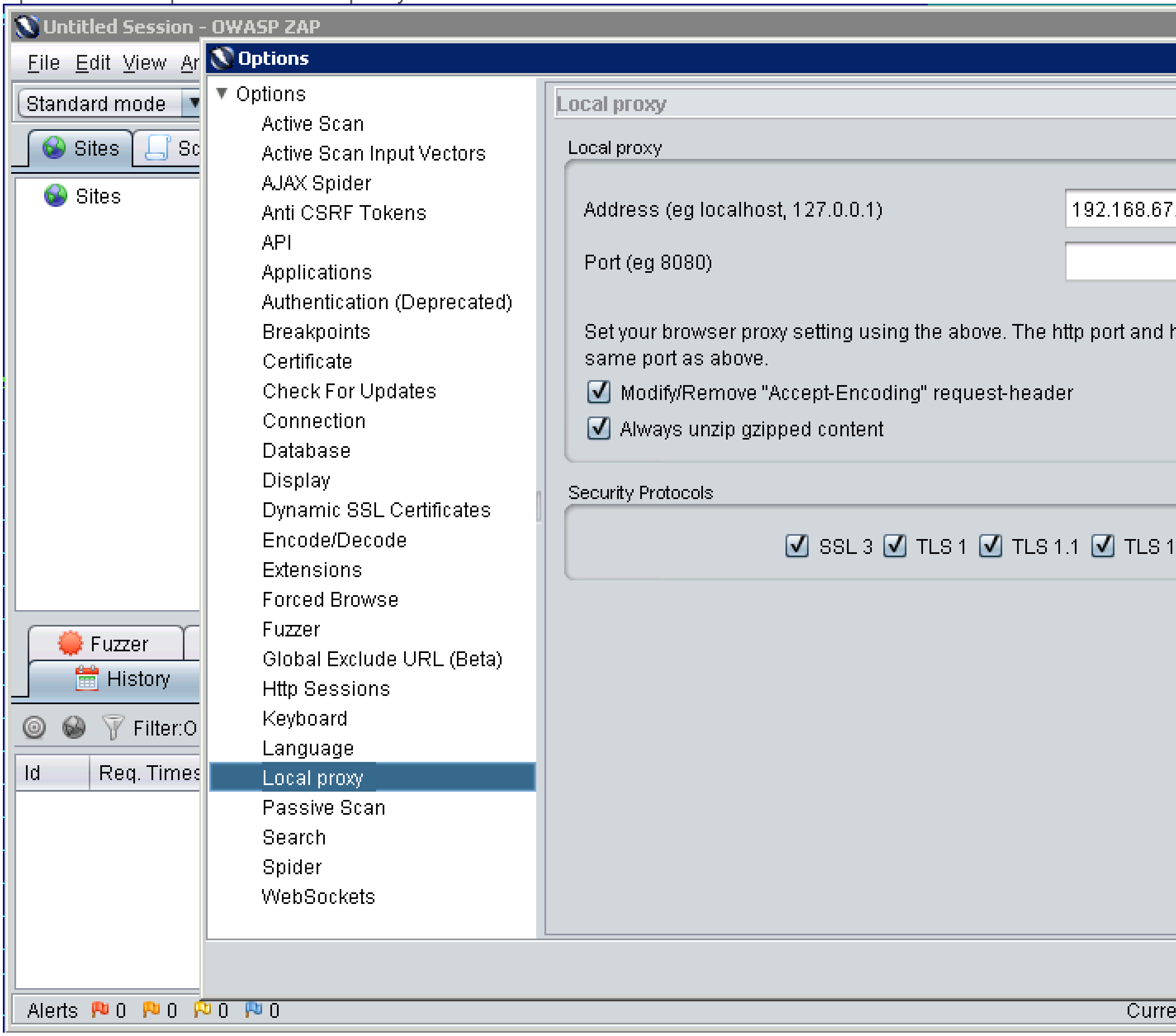
Stop the Proxy

To stop the ZAProxy, simply close the UI window.

Configure ZAProxy Host and Port

Start

1. Open `Tools > Options > Local proxy`.



Local proxy

Local proxy

Address (eg localhost, 127.0.0.1) 192.168.67

Port (eg 8080)

Set your browser proxy setting using the above. The http port and h
same port as above.

☑ Modify/Remove "Accept-Encoding" request-header

☑ Always unzip gzipped content

Security Protocols

☑ SSL 3 ☑ TLS 1 ☑ TLS 1.1 ☑ TLS 1

2. In the `Local proxy` panel, specify the host and port of this proxy. Do not use "localhost" or "127.0.0.1" for the host name.

3. Note the values of the host and port — you will use these to Set up your Web Browser.

4. If you changed the settings, restart the proxy.

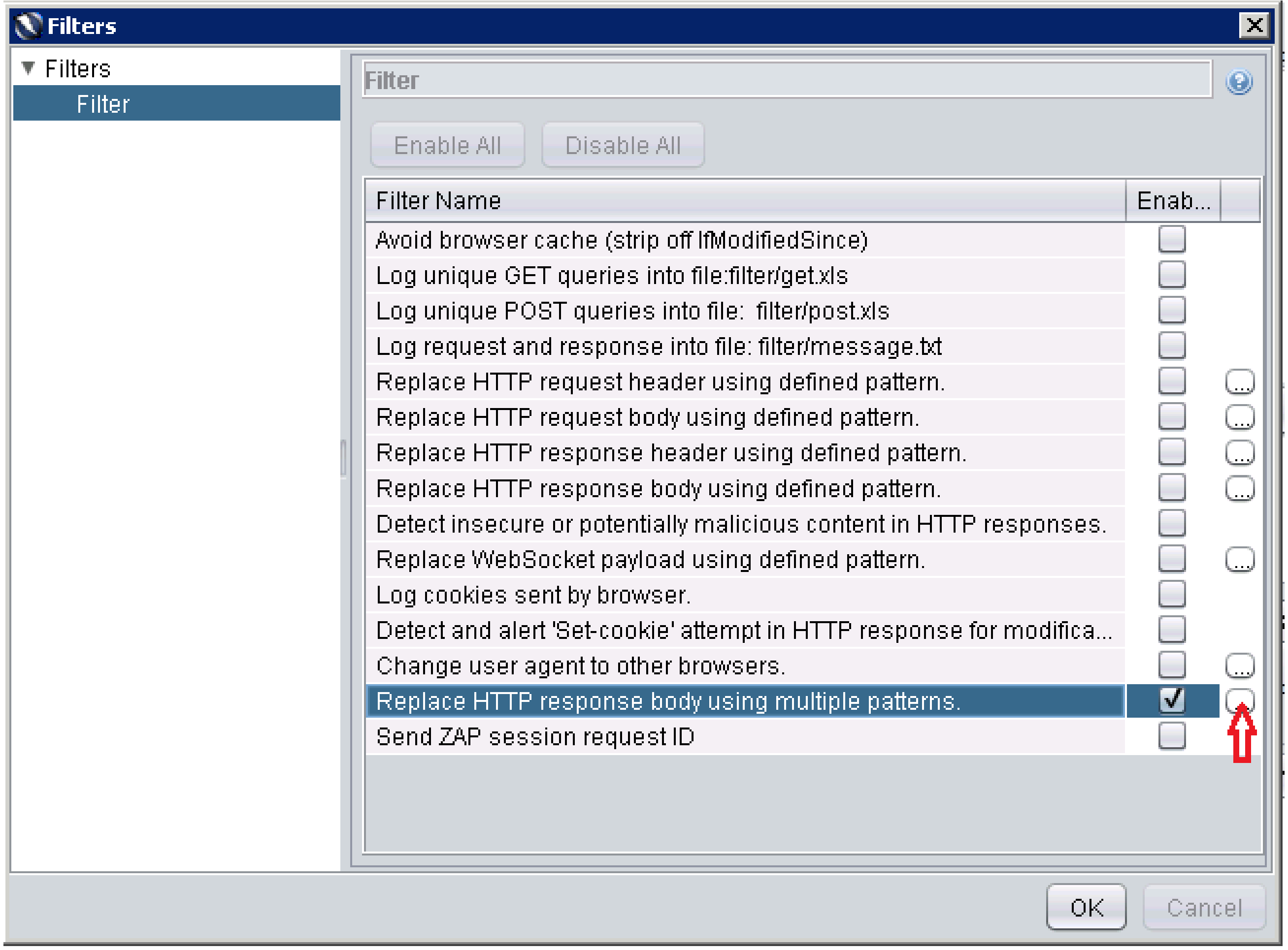
End

Update the Instrumentation Script

ZAProxy includes the default Co-browse instrumentation script, which you can view by completing the steps below.

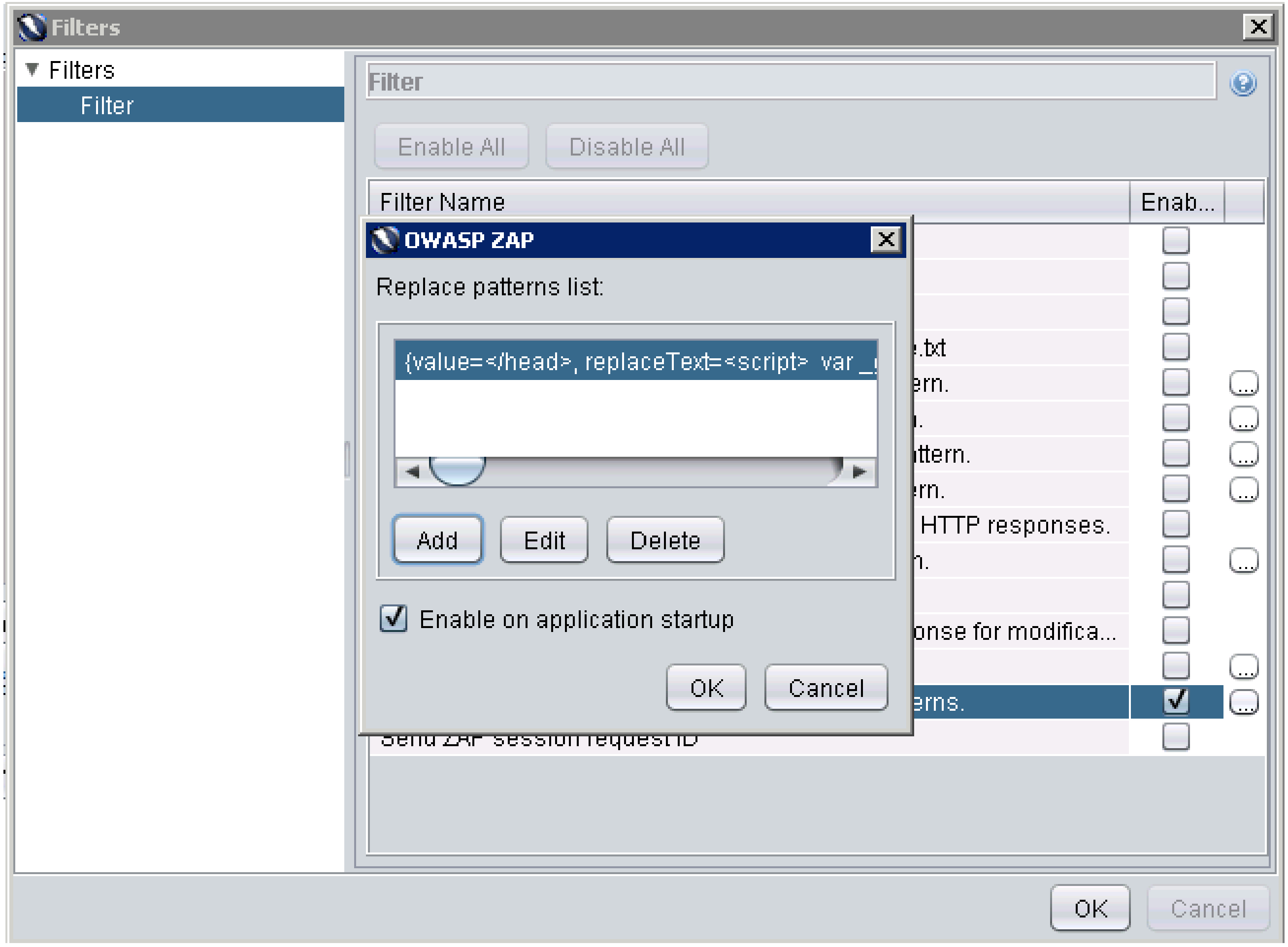
Start

1. Open `Tools > Filter`.
2. In the dialog that opens, click the small oval with the ellipses (...), located near the checked box for the "Replace HTTP response body..." item.



Filter Name	Enab...	
Avoid browser cache (strip off IfModifiedSince)	☐	
Log unique GET queries into file:filter/get.xls	☐	
Log unique POST queries into file: filter/post.xls	☐	
Log request and response into file: filter/message.txt	☐	
Replace HTTP request header using defined pattern.	☐	...
Replace HTTP request body using defined pattern.	☐	...
Replace HTTP response header using defined pattern.	☐	...
Replace HTTP response body using defined pattern.	☐	...
Detect insecure or potentially malicious content in HTTP responses.	☐	
Replace WebSocket payload using defined pattern.	☐	...
Log cookies sent by browser.	☐	
Detect and alert 'Set-cookie' attempt in HTTP response for modifica...	☐	
Change user agent to other browsers.	☐	...
Replace HTTP response body using multiple patterns.	☑	...
Send ZAP session request ID	☐	

3. In the dialog that opens, select the line and click `Edit`.

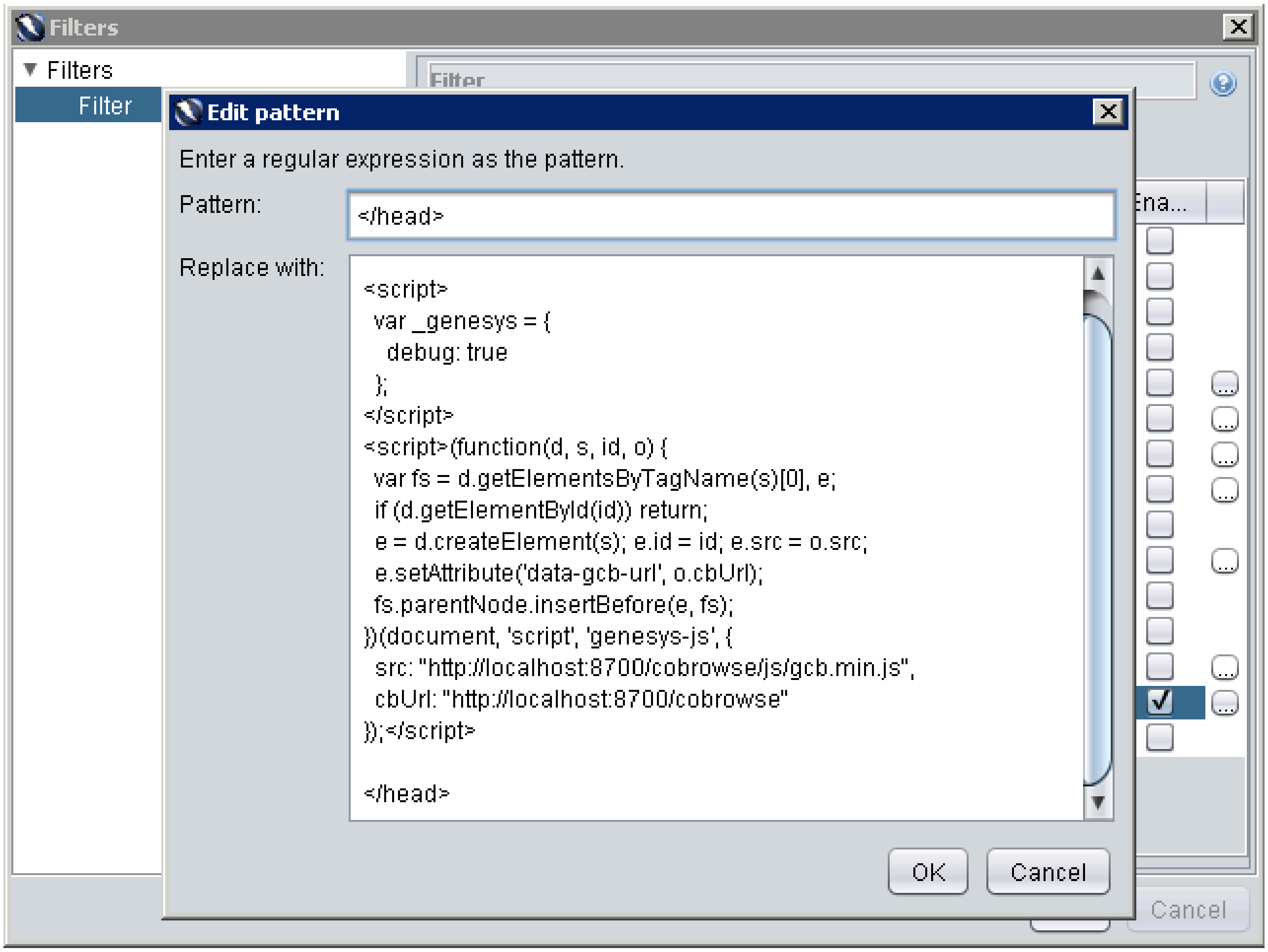


Replace patterns list:

{value=</head>, replaceText=<script> var _

☑ Enable on application startup

The `Edit pattern` dialog opens.



Enter a regular expression as the pattern.

Pattern: `</head>`

Replace with:

```
<script>
 var _genesys = {
   debug: true
 };
</script>
<script>(function(d, s, id, o) {
 var fs = d.getElementsByTagName(s)[0], e;
 if (d.getElementById(id)) return;
 e = d.createElement(s); e.id = id; e.src = o.src;
 e.setAttribute('data-gcb-url', o.cbUrl);
 fs.parentNode.insertBefore(e, fs);
})(document, 'script', 'genesys-js', {
 src: "http://localhost:8700/cobrowse/js/gcb.min.js",
 cbUrl: "http://localhost:8700/cobrowse"
});</script>

</head>
```

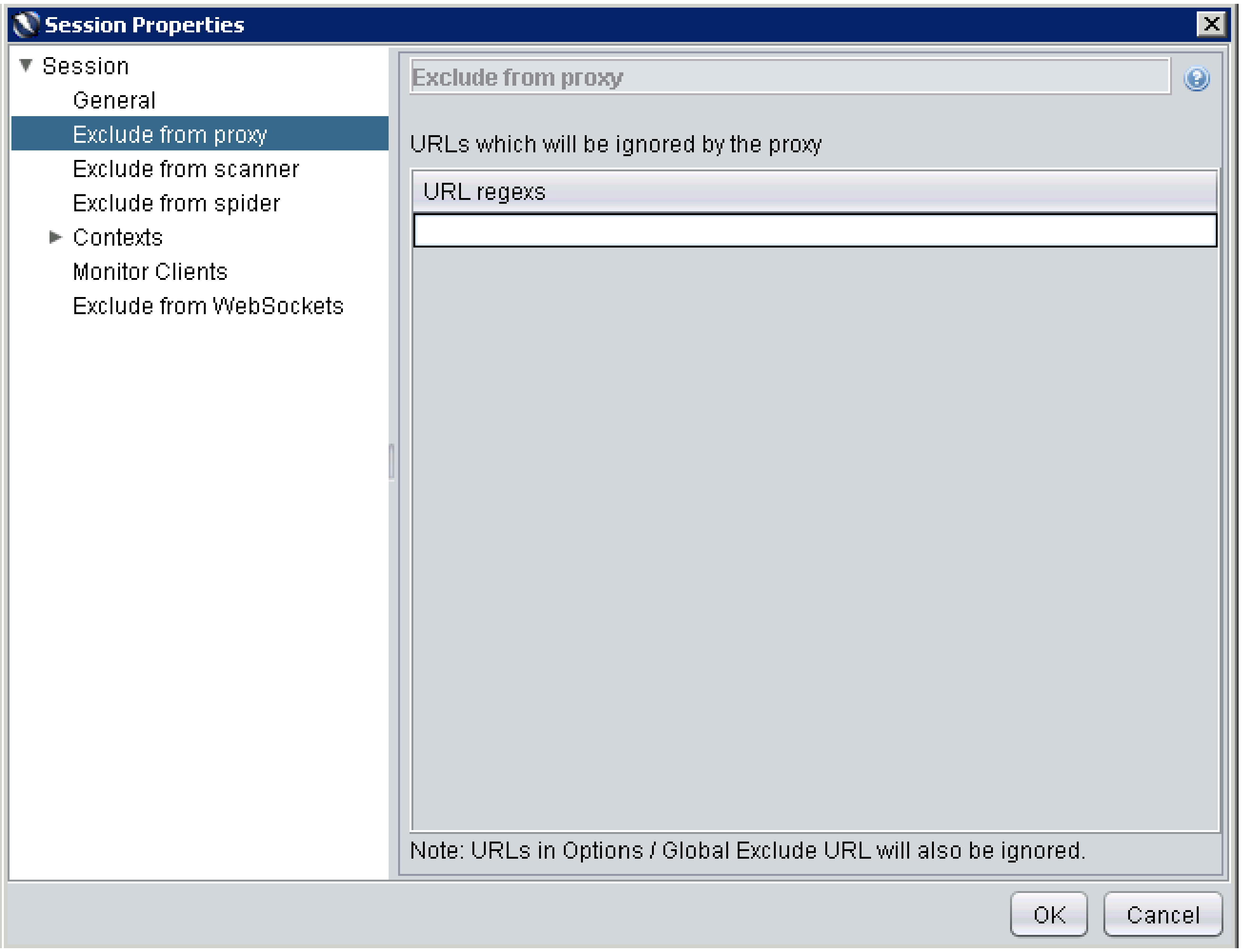
4. To save the changes, click OK on the current dialog and on the two parent dialogs.

End

Configure the URL Filter

To configure URLs that the proxy should ignore, use one of the following ways:

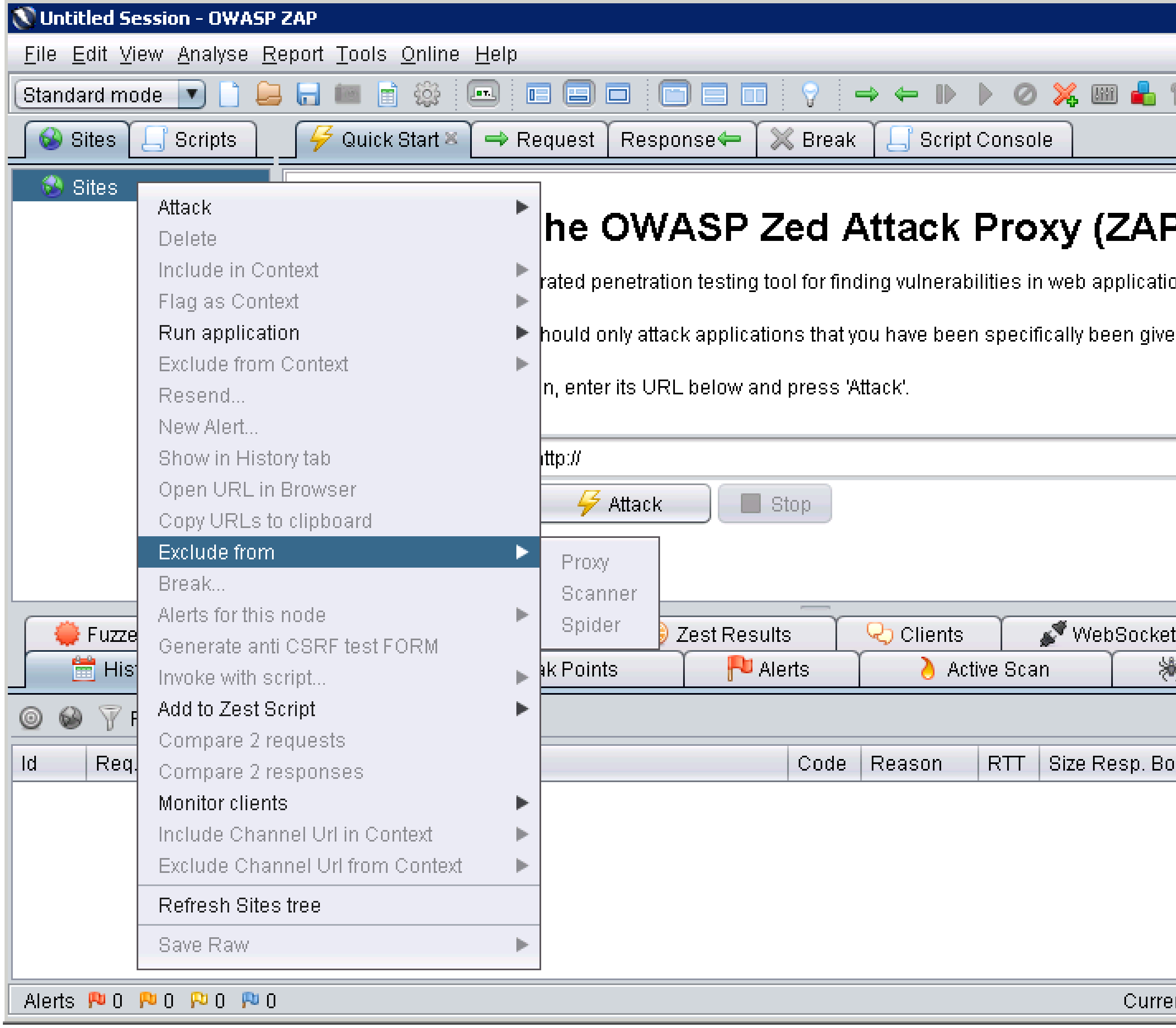
- Select `File > Session Properties`. In the Session Properties dialog, select `Exclude from proxy`, double-click `URL regexs` and add your URL. Click OK.



URLs which will be ignored by the proxy

Note: URLs in Options / Global Exclude URL will also be ignored.

- In the Sites tab, right-click on a site and select `Exclude from > Proxy`.



If you want the proxy to remember the excluded URLs beyond the current session, select `File > Persist session...` and select a file to save your session.

Set up your Web Browser

To use the proxy you need to set up your Web Browser. See ZAProxy#Set_up_your_Web_Browser

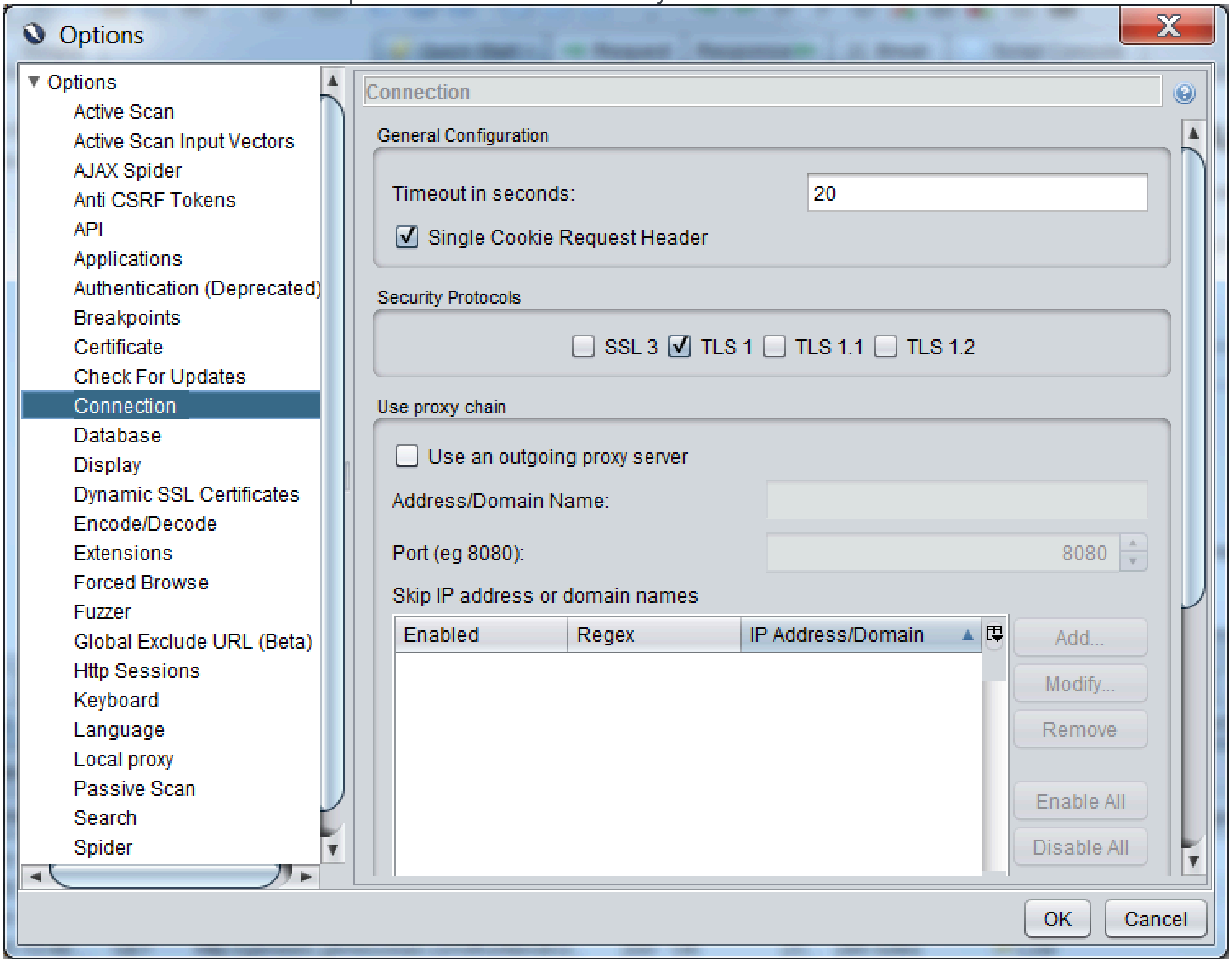
Resolving the protocol_version error

After configuring the proxy in your browser, you may encounter the following error on some HTTPS sites:

```
ZAP Error [javax.net.ssl.SSLException]: Received fatal alert: protocol_version
```

This error happens when a site only supports older versions of the TLS protocol. To fix this error:

1. Open **Tools > Options > Connection**.
2. Un-check all checkboxes except for **TLS 1** in the Security Protocols section.



Connection

General Configuration

Timeout in seconds: 20

☑ Single Cookie Request Header

Security Protocols

□ SSL 3 ☑ TLS 1 □ TLS 1.1 □ TLS 1.2

Use proxy chain

□ Use an outgoing proxy server

Address/Domain Name:

Port (eg 8080): 8080

Skip IP address or domain names

Enabled	Regex	IP Address/Domain

3. Click **OK** and reload the web page.

Important

If you encounter this error on a site you want to instrument with Co-browse, update

> the corresponding `clientTlsProtocols` option to `TLSv1`

Security Testing with ZAProxy

Genesys performs security testing with OWASP Zed Attack Proxy (ZAProxy) to make sure the Genesys Co-browse solution is invincible to known attacks.

Tip

Genesys aims to test against the latest version of ZAProxy available at the time of a release. For your convenience, this version is shipped together with the Co-browse solution. For instructions on how to obtain and use the ZAProy, see ZAProxy

Important

Some issues reported by ZAProxy security are false positives. See ZAProxy False Positives.

ZAP Overview

The ZAProxy is an easy-to-use, integrated penetration testing tool for finding vulnerabilities in websites and web applications.

Among others, ZAProxy supports the follow methods for penetration security testing:

- passive scan
- active scan

Genesys uses both methods.

Passive Scan Overview

ZAP is an Intercepting Proxy. It allows you to see all of the requests made to a website/web app and all of the responses received from it. For example, you can see AJAX calls that might not otherwise be obvious.

Once set up, ZAP automatically passively scans all of the requests to and responses from the web application being tested.

While mandatory use cases for the application that is being tested are followed (either manually or automatically), ZAProxy analyzes the requests to verify the usual operations are safe.

Active Scan Overview

Active scanning attempts to find potential vulnerabilities by using known web attacks against the selected targets. Active scanning is an attack on those targets. ZAProxy emulates known attacks when active mode is used.

Through active scanning, Genesys Co-browse is verified against the following types of attacks:

- **Spider attack** — Automatically discovers all URL links found on a web resource, sends requests, and analyzes results (including src attributes, comments, low-level information disclosure, and so on).
- **Brute browsing** (based on the Brute Force technique) — Systematically makes requests to find secure resources based on known (commonly used) rules. For example, backup, configuration files, temporary directories, and so on.
- **Active scan** — Attempts to perform a predefined set of attacks on all resources available for the web resource. You can find the default set of rules here.
- **Ajax spider** — Automatically discovers web resources based on presumed rules of AJAX control (JS scripts investigation, page events, common rules, dynamic DOM, and so on).

> Important
>
> Requests to other web applications must be excluded from scanning in order to see a report for a particular web application.

> Important
>
> Web applications that are being tested should be started on the local box because some types of verification (like active scanning) can be forbidden by network administrators.

False Positives

Some issues reported by ZAProxy security testing are not actual vulnerabilities:

- High risk security alert "Remote file inclusion".

  To allow certain types of dynamic content synchronization, Co-browse may proxy some of the website's static assets (for example, CSS files). The response content is only interpreted by browsers as a corresponding asset (like CSS), because it is retrieved through according means (for example, in case of CSS through stylesheet links). Illegible assets are skipped. The source for a proxied asset is always the web site page itself, which is by definition a legitimate resource. To limit access to Co-browse resource proxy mechanism, use the `allowedExternalDomains` option.

- Medium risk security alert "Secure page browser cache" and Low risk security alert "Incomplete or no cache-control and pragma HTTPHeader set".

  As mentioned above, Co-browse may proxy some of the website's static assets (like CSS). Browser side caching of such resources is determined by original servers that own those resources and are responsible for proper caching. In other words, Co-browse will just serve the same headers that the original website serves. In most cases, such static assets do not contain any sensitive information and can be safely cached. However, if you decide to disable all caching for these resources, you can do this on Co-browse side using the `dislabeCaching` option, without the need to modify headers on your website's side. Note that this may increase traffic load for both end users (if they opt to Co-browse), and agents.

- Medium level security alert "X-Frame-Options header not set".

  Co-browse JavaScript by design works as a third-party add-on to another web site. Moreover, it can be (and usually is) loaded from another domain and must operate in iframes while the X-Frame-Options header is specifically designed to disallow that. To prevent other websites from using your Co-browse deployment, use the `allowedOrigins` configuration option (it will not remove the "X-Frame-Options" alert).

- Low risk security alert "Cookie set without HttpOnly flag".

  This security alert means that the cookie can be accessed by JavaScript, which is a security issue if the cookie is a session cookie that can be used to hijack the session. However, the reported cookie is not such a cookie and will not allow anyone to hijack a Co-browse session.

References

If you want to examine your website against vulnerabilities in a similar way, refer to the OWASP Zed Attack Proxy Project or additional documentation to learn about security testing with ZAP.

Genesys Co-browse Reporting Templates

Co-browse comes with reporting templates for Pulse, CCPulse+, and DMA. See the following pages to set up and use the Co-browse reporting templates:

- Pulse Reporting Templates
- CCPulse+ and DMA Reporting Templates

Pulse Templates

Genesys Co-browse includes templates for Pulse. To set up and use the Pulse templates see the following pages:

- Setting Up Reporting Templates in Pulse—procedures to make the Co-browse templates available from Pulse.
- Pulse Templates Overview—description of each Co-browse Pulse template.

Setting Up Reporting Templates in Pulse

Use the procedures below to make the Co-browse templates available from Pulse.

Import Configuration Options for Stat Server

Before working with the templates, you must first import configuration options to the Stat Server application from the following files:

- `StatProfile_Pulse.cfg`—located in the directory `<Reporting Templates Root Folder>/Pulse`. This file is used for the Stat Server application object. It contains the necessary configuration options, statistics, and filters.
- `pulse_statistics.cfg`—located in the director `<Pulse installation folder>/scripts`. This file is used for the Pulse application object.

Start of procedure

1. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`.
2. Select your Stat Server Application and click `Edit`.
3. In the `Options` tab, click `Import`. The `Import Options` dialog opens.
4. Click `Yes`. The `Click 'Add' and choose a file with configuration options to import` dialog opens.
5. Click `Add`. The `Choose File to Upload` window opens.
6. Choose the `StatProfile_Pulse.cfg` file from the `<Reporting Templates Root Folder>/Pulse` directory and click `Open`. The configuration options for the Co-browse reporting templates are imported.
7. Click `Add` again. Choose the `pulse_statistics.cfg` file from the `<Pulse Installation Folder>/scripts` directory and click `Open`.
8. Click `Save & Close`.
9. Mandatory: Restart your Stat Server Application.

End of procedure

Next Steps
See Tune Pulse DB Tool to Allow Template Importing.

Tune Pulse DB Tool to Allow Template Importing

Pre-requisite: You have completed Import Configuration Options for Stat Server

Start of procedure

1. Open the folder <`Pulse installation folder`>`/dbtool`.
2. Save the file `sample_dbtool.cfg` with the name `dbtool.cfg`.
3. Now update the file `dbtool.cfg`. Specify the #<`DB type`> section. For example:

   ### MSSQL Example

   ```
   # MSSQL
    db.type=mssql
    db.url=jdbc:jtds:sqlserver://demosrv:1433/pulse
    db.user=sa
    db.password=<password>
   ```

   ### POSTGRES Example

   ```
   # POSTGRES
    db.type=postgres
    db.url=jdbc:postgresql://stage-rds-gax-1.genhtcc.com:5432/pulse851_21_41_db
    db.user=pulse851_21_41_db
    db.password=<password>
   ```

4. Save the `dbtool.cfg` file. Verify that the DB Tool is working with the command `dbtool.bat -l`. If the DB Tool is correctly configured, your console will show a list of current Layouts.

End of procedure

Next Steps
See Import Pulse Templates.

Import Pulse Templates

Pre-requisites: You have completed Import Configuration Options for Stat Server and Tune Pulse DB Tool to Allow Template Importing

Start of procedure

1. Open the folder <`Pulse installation folder`>`/dbtool` and open the file `dbtool.cfg` for editing.
2. Set `layout.override.layout_id=1` and save the file.
3. From your console, run the command:
   - **Pulse 8.5.102:**
     - Windows: `dbtool -i <path to template file>`

  - Linux: `./dbtool.sh -i <path to template file>`

For example, `dbtool -i C:\templates\21_template.txt`.

- **Pulse 8.5.103+:**
  - Windows: `dbtool -ti <path to template file>`
  - Linux: `./dbtool.sh -ti <path to template file>`

For example, `dbtool -ti C:\templates\21_template.txt`.

If the import is successful the console displays something similar to the following:

```
Advanced Data Management Utility for Pulse
 ver. 8.5.010.04 (2014-12-02), wbrt2.proto ver. 2014.10.29.00
 Copyright (c) Genesys Telecommunications Laboratories, 2013-2014. All rights reserved.
 Using configuration file 'dbtool.cfg'.
 Inserting layouts...
 Layout: (114) 'CurrentInteractions'
 Inserted 1 record(s).
```

In the example above, `Layout: (114) 'CurrentInteractions'` has the following meaning:

- 114 is the template db id, the `LAYOUT_KEY` from the `WBRT_LAYOUT` table of the Pulse database.
- `CurrentInteractions` is the template name.

End of procedure

Next Steps
See Working With Pulse and Pulse Templates Overview

Working With Pulse

After you have imported the Co-browse templates into Pulse, you can use the templates to create new Pulse widgets.

For more information on creating a widget see Pulse User Help—Widgets.

Pulse Templates Overview

Pulse templates come in two groups:

- *Current Statistics*—collected for interactions currently kept by agents. These templates are located in the `/Pulse/current` folder.
- *Total Statistics*—collected for interactions ended in the configured time interval (not more than 24 hours). These templates are located in the `/Pulse/total` folder.

To set up and use the Co-browse Pulse templates, see Setting Up Reporting Templates in Pulse.

<tabber> Current Statistics=

Current Co-browse Agents

Important

Deprecated in **8.5.003**

File: `/Pulse/current/CurrentCo-browseAgents.txt`
Description: Number of agents working on Co-browse interactions
Allowed Object Types: Agent Group, Place Group

Template Statistics:

Pulse Statistic	Description
Chat Agents	Number of agents working on Chat
Chat with CB	Number of agents working on Chat with Co-browse
Inbound Voice	Number of agents working on inbound Voice with Co-browse
Inb Voice with CB	Number of agents working on Inbound Voice with CB

Current Co-browse Agents Number

Important

New in **8.5.003**

File: `/Pulse/current/CurrentCo-browseAgentsNumber.txt`
Description: Number of agents working on Co-browse interactions
Allowed Object Types: Agent Group, Place Group

Template Statistics:

Pulse Statistic	Description
CB Agents Number	Number of agents working with Co-browse
Agents with CB denial	Number of agents working on Chat/Voice with CB session denial
Agents that have allowed CB after denial	Number of agents working on Chat/Voice with allowed CB after denial
Current Group State	Hidden Auxiliary statistic needed for user data retrieval

Current Chat Interactions

File: `/Pulse/current/CurrentChatInteractions.txt`
Description: Number of chat interactions currently handled by agents
Allowed Object Types: Agent, Place, Agent Group, Place Group

Template Statistics:

Pulse Statistic	Description
Chat	Number of chat interactions currently handled by agents
CB in Chat	Number of chat with Co-browse interactions currently handled by agents
(CB in Chat)/Chat, %	Ratio of current number of chat with Co-browse interactions and total number of chat interactions
Web Chat	Number of web chat interactions currently handled by agents
CB in Web Chat	Number of web chat with Co-browse interactions currently handled by agents
(CB in Web Chat)/Web Chat, %	Ratio of current number of web chat with Co-

Pulse Statistic	Description
	browse interactions and total number of web chat interactions

Current Voice Interactions

File: `/Pulse/current/CurrentVoiceInteractions.txt`
Description: Number of inbound voice interactions currently handled by agents
Allowed Object Types: Agent, Place, Agent Group, Place Group

Template Statistics:

Pulse Statistic	Description
Inbound Voice	Number of inbound voice interactions currently handled by agents
CB in Inbound Voice	Number of inbound voice with Co-browse interactions currently handled by agents
(CB on InbVoice)/InbVoice, %	Ratio of current inbound voice with Co-browse interactions and total number of inbound voice interactions

Current Co-browse Denials

Important

New in **8.5.003**

File: `/Pulse/current/CurrentCo-browseDenials.txt`
Description: Number of chat/voice interactions currently handled by agents with allowed co-browse after denial
Allowed Object Types: Agent, Place, Agent Group, Place Group

Template Statistics:

Pulse Statistic	Description
Allowed CB after denial	Number of chat/voice interactions currently handled by agents with allowed co-browse after denial

Pulse Statistic	Description
CB denial in Voice	Number of voice interactions currently handled by agents with co-browse session denial
CB denial in Web Chat	Number of web chat interactions currently handled by agents with co-browse session denial
CB denial in Chat	Number of chat interactions currently handled by agents with co-browse session denial

Co-browse Sessions State

> **Important**
>
> Updated in **8.5.003**

File: `/Pulse/current/Co-browseSessionsState.txt`
Description: Current Co-browse Session State:

- Agent State
- Co-browse State
- Start Time
- ID
- Quantity

Allowed Object Types: Agent

Template Statistics:

Pulse Statistic	Description
Current Agent State	Hidden auxiliary statistic needed for user data retrieval
CB Session State	Co-browse session state. Possible values: • alive • finished • denied (added in 8.5.003)
Co-browse Session Start Time	Session start time
Co-browse Session ID	Session ID

Pulse Statistic	Description
Co-browse Session Quantity	Session quantity

|-| Total Stastics=

Total Co-browse Agents

> **Important**
>
> Deprecated in **8.5.003**

File: `/Pulse/total/TotalCo-browseAgents.txt`
Description: Total amount of agents who worked on Co-browse interactions
Allowed Object Types: Agent Group, Place Group
Template Statistics:

Pulse Statistic	Description
Chat Agents	Number of agents who worked on Chat
Chat with CB	Number of agents who worked on Chat with Co-browse
Inbound Voice	Number of agents who worked on Inbound Voice
Inb Voice with CB	Number of agents who worked on Inbound voice with Co-browse

Total Chat Interactions

File: `/Pulse/total/TotalChatInteractions.txt`
Description: Total number and total duration of chat interactions handled by agents
Allowed Object Types: Agent, Place, Agent Group, Place Group
Template Statistics:

Pulse Statistic	Description
Chat	Total number of chat interactions handled by agents
CB in Chat	Total number of chat with Co-browse interactions handled by agents
(CB in Chat)/Chat, %	Ratio of total number of chat with Co-browse interactions and chat interactions
Chat Duration	Duration of chat interactions handled by agents

Pulse Statistic	Description
CB in Chat Duration	Duration of chat with Co-browse interactions handled by agents
(CB in Chat)/Chat, % Duration	Ratio of total duration of chat with Co-browse interactions and chat interactions

Total Web Chat Interactions

File: `/Pulse/total/TotalWebChatInteractions.txt`
Description: Total number and total duration of web chat interactions handled by agents
Allowed Object Types: Agent, Place, Agent Group, Place Group
Template Statistics:

Pulse Statistic	Description
Web Chat	Total number of web chat interactions handled by agents
CB in Web Chat	Total number of web chat with Co-browse interactions handled by agents
(CB in Web Chat)/Web Chat, %	Ratio of total number of web chat with Co-browse interactions and Web Chat interactions
Web Chat Duration	Duration of web chat interactions handled by agents
CB in Web Chat Duration	Duration of web chat with Co-browse interactions handled by agents
(CB in Web Chat)/Web Chat, % Duration	Ratio of total duration of web chat with Co-browse interactions and web chat interactions

Total Voice Interactions

File: `/Pulse/total/TotalVoiceInteractions.txt`
Description: Total number and total duration of voice interactions handled by agents
Allowed Object Types: Agent, Place, Agent Group, Place Group
Template Statistics:

Pulse Statistic	Description
Inbound Voice	Total number of inbound voice interactions handled by agents
CB in Inbound Voice	Total number of inbound voice with Co-browse interactions handled by agents
(CB in inVoice)/inboundVoice, %	Ratio of total number of inbound voice with Co-

Pulse Statistic	Description
	browse interactions and inbound voice interactions
Inbound Voice Duration	Duration of inbound voice interactions handled by agents
CB in Inb Voice Duration	Duration of inbound voice with Co-browse interactions handled by agents
(CB in InbVoice)/InbVoice, % Duration	Ratio of total duration of inbound voice with Co-browse interactions and inbound voice interactions

Total Co-browse Denials

> **Important**
> New in **8.5.003**

File: `/Pulse/total/TotalCo-browseDenials.txt`
Description: Total amount of chat/voice interactions handled by agents with allowed co-browse after denial
Allowed Object Types: Agent, Place, Agent Group, Place Group
Template Statistics:

Pulse Statistic	Description
CB denial in Chat	Total number of chat interactions handled by agents with co-browse session denial
CB denial in Web Chat	Total number of web chat interactions handled by agents with co-browse session denial
CB denial in Voice	Total number of voice interactions handled by agents with co-browse session denial
Allowed CB after denial	Total number of chat/voice interactions handled by agents with allowed co-browse after denial

CCPulse+ Templates

Genesys Co-browse includes templates for CCPulse+. To set up and use the CCPulse+ templates see the following pages:

- Setting Up Reporting Templates in CCPulse+—procedures to make the Co-browse templates available from CCPulse+.
- CCPulse+ Templates Overview—description of each Co-browse CCPulse+ template.

Setting Up Reporting Templates in CCPulse+

Genesys Co-browse includes templates for real-time and historical reporting. Before working with the templates, you must first import configuration options from the following files, located in the Genesys Co-browse Sample Reporting Templates root directory:

- `StatProfile.cfg` — used for the Stat Server application object. It contains the necessary configuration options, statistics, and filters.
- `CCPulseProfile.cfg` — used for the CCPulse+ application object.

Import Configuration Options for Stat Server and CCPulse+

Start of procedure

1. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`.
2. Select your Stat Server Application and click `Edit`.
3. In the `Options` tab, click `Import`. The `Import Options` dialog opens.
4. Click `Yes`. The `Click 'Add' and choose a file with configuration options to import` dialog opens.
5. Click `Add`. The `Choose File to Upload` window opens.
6. Choose the `StatProfile.cfg` from the root directory of Genesys Co-browse Sample Reporting Templates and click `Open`. The configuration options for the Co-browse reporting templates are imported.
7. Click `Save & Close`.
8. Mandatory: Restart your Stat Server Application.
9. Mandatory: Reopen your Data Modeling Assistant.
10. Select your CCPulse+ Application and click `Edit`.
11. Complete steps 3-7. Be sure to import the `CCPulseProfile.cfg` in step 6.
12. Mandatory: Reopen your CCPulse+.

End of procedure

Next Steps
See CCPulse+ Templates Overview.

CCPulse+ Templates Overview

CCPulse+ templates come in two groups, *Real-time Reporting* and *Historical Reporting*. See the tabs below for descriptions of the templates in each group. To set up and use the Co-browse CCPulse+ templates, see Setting Up Reporting Templates in CCPulse+.

Real-time Reporting

Genesys Co-browse includes the following real-time reporting templates that allow you to:

- `Co-browseAgents.xtpl`—see the current number of agents participating in Co-browse sessions.
- `Co-browseInteractions.xtpl`—see the current and daily total number of Co-browse sessions.
- `Co-browseIntsDuration.xtpl`—see the total number and total duration of Co-browse sessions.
- `Co-browseInteractionsExt.xtpl`—see the user data (Co-browse session ID, Co-browse start time, Co-browse sessions quantity) against certain Co-browse session.

CCPulse+ Templates

> ### Important
> For CCPulse+ templates to be imported correctly, you must rename each template using the Import/Export Wizard before applying the import procedure.

Co-browseAgents.xtpl

Object to apply: Tenant, Agent Group, Place Group.
Available CCPulse+ Views:

- Create Real-Time View.

> ### Important
> For the tenant to be applied correctly, you should select the tenant object together with agent group(s) or place group(s) for the CCPulse+ Real-Time View.

<Object>	Statistic	Values
CurrentNumber	Agents working on Chat with CB	*Current number of agents working on Chat with Co-browse*
	Agents working on Chat	*Current number of agents working on Chat*
	Agents working on inbound Voice with CB	*Current number of agents working on Inbound Voice with Co-browse*
	Agents working on inbound Voice	*Current number of agents working on Inbound Voice*
	Agents working on Voice with CB	*Current number of agents working on Voice (Inbound, Internal, Consult) with Co-browse*
	Agents working on Voice	*Current number of agents working on Voice*
	Agents working on inbound CB	*Current number of agents working on Inbound Voice and Chat with Co-browse*
	Agents working on CB	*Current number of agents working on Voice and Chat with Co-browse*

Co-browseInteractions.xtpl

Object to apply: Tenant, Agent Group, Place Group, Place, Agent.
Available CCPulse+ Views:

- Create Real-Time View
- Create Real-Time View for Members
- Create Real-Time View V/AG Dynamic Membership.

> ### Important
> For the tenant to be applied correctly, you should select the tenant object together with any other valid object(s) (agent group, place group, place, agent) for the CCPulse+ Real-Time View.

<Object>	Statistic	Values
Current Interactions	Chat with CB Handling	*Current number of Chat with Co-browse interactions*
	Chat Handling	*Current number of Chat interactions*
	(Chat with CB)/Chat, %	*Ratio of current number of Chat with Co-browse interactions as opposed to Chat interactions*

<Object>	Statistic	Values
	Inbound Voice with CB Handling	*Current number of Inbound Voice with Co-browse interactions*
	Inbound Voice Handling	*Current number of Inbound Voice interactions*
	(Voice with CB)/Voice, Inbound, %	*Ratio of current number of Inbound Voice with Co-browse interactions as opposed to Inbound Voice interactions*
	Voice with CB Handling	*Current number of Voice (Inbound, Internal, Consult) with Co-browse interactions*
	Voice Handling	*Current number of Voice interactions*
	(Voice with CB)/Voice, %	*Ratio of current number of Voice with Co-browse interactions as opposed to Voice interactions*
	CB Inbound Handling	*Current number of Chat and Inbound Voice with Co-browse interactions*
	CB/(Chat and Voice), Inbound, %	*Ratio of current number of Chat and Inbound Voice with Co-browse interactions as opposed to Chat and Inbound Voice interactions*
	CB Handling	*Current number of Chat and Voice with Co-browse interactions*
	CB/(Chat and Voice), %	*Ratio of current number of Chat and Voice with Co-browse interactions as opposed to Chat and Voice interactions*
Total Interactions	Chat with CB Total	*Total number of Chat with Co-browse interactions*
	Chat Total	*Total number of Chat interactions*
	(Chat with CB)/Chat,Total, %	*Ratio of Total number of Chat with Co-browse interactions as opposed to Chat interactions*
	Inbound Voice with CB Total	*Total number of Inbound Voice with Co-browse interactions*
	Inbound Voice Total	*Total number of Inbound Voice interactions*
	(Voice with CB)/Voice, Inbound Total, %	*Ratio of total number of Inbound Voice with Co-browse interactions as opposed to Inbound Voice interactions*
	Voice with CB Total	*Total number of Voice (Inbound, Internal, Consult) with Co-browse interactions*

<Object>	Statistic	Values
	Voice Total	*Total number of Voice interactions*
	(Voice with CB)/Voice,Total, %	*Ratio of total number of Voice with Co-browse interactions as opposed to Voice interactions*
	CB Inbound Total	*Total number of Chat and Inbound Voice with Co-browse interactions*
	CB/(Chat and Voice), Inbound Total, %	*Ratio of total number of Chat and Inbound Voice with Co-browse interactions as opposed to Chat and Inbound Voice interactions*
	CB Total	*Total number of Chat and Voice with Co-browse interactions*
	CB/(Chat and Voice), Total,%	*Ratio of total number of Chat and Voice with Co-browse interactions as opposed to Chat and Voice interactions*

Co-browseIntsDuration.xtpl

Object to apply: Agent Group, Place Group, Place, Agent.
Available CCPulse+ Views:

- Create Real-Time View
- Create Real-Time View for Members
- Create Real-Time View V/AG Dynamic Membership.

<Object>	Statistic	Values
Total Interactions	Chat with CB	*Total number of Chat with Co-browse interactions*
	Chat	*Total number of Chat interactions*
	(Chat with CB)/Chat, %	*Ratio of Total number of Chat with Co-browse interactions as opposed to Chat interactions*
	Inbound Voice with CB	*Total number of Inbound Voice with Co-browse interactions*
	Inbound Voice	*Total number of Inbound Voice interactions*
	(Voice with CB)/Voice, Inbound, %	*Ratio of total number of Inbound Voice with Co-browse interactions as opposed to Inbound Voice interactions*
	Voice with CB	*Total number of Voice (Inbound, Internal, Consult) with Co-browse interactions*

<Object>	Statistic	Values
	Voice	*Total number of Voice interactions*
	(Voice with CB)/Voice, %	*Ratio of total number of Voice with Co-browse interactions as opposed to Voice interactions*
	CB Inbound	*Total number of Chat and Inbound Voice with Co-browse interactions*
	CB/(Chat and Voice), Inbound, %	*Ratio of total number of Chat and Inbound Voice with Co-browse interactions as opposed to Chat and Inbound Voice interactions*
	CB	*Total number of Chat and Voice with Co-browse interactions*
	CB/(Chat and Voice), %	*Ratio of total number of Chat and Voice with Co-browse interactions as opposed to Chat and Voice interactions*
Total Duration	CB Duration in Chat, hh:mm:ss	*Total duration of Co-browse sessions in Chat interactions*
	Chat Duration, hh:mm:ss	*Total duration of Chat interactions*
	CB Duration in Chat/Chat Duration, %	*Ratio of total duration of Co-browse sessions in Chat as opposed to Chat interactions duration*
	CB Duration in Voice, hh:mm:ss	*Total duration of Co-browse sessions in Voice interactions*
	Voice Duration, hh:mm:ss	*Total duration of Voice interactions*
	CB Duration in Voice/Voice Duration, %	*Ratio of total duration of Co-browse sessions in Voice as opposed to Voice interactions duration*

Co-browseInteractionsExt.xtpl

Object to apply: Agent Group, Place Group, Place, Agent.
Available CCPulse+ Views:

- Create Real-Time View
- Create Real-Time View for Members (Agent Group)
- Create Real-Time View V/AG Dynamic Membership (Agent Group)

	CoBrowseStartTime	CoBrowseSessionId	CoBrowseSessionsQuantity
<Agent name>	*Co-browse session start time*	*Co-browse session ID*	*Co-browse sessions quantity*
...	...	...	...

<Agent>	Statistic	Values
Current Interactions	CoBrowseStartTime	*Co-browse session start time*
	CoBrowseSessionId	*Co-browse session ID*
	CoBrowseSessionsQuantity	*Co-browse sessions quantity*

Historical Reporting

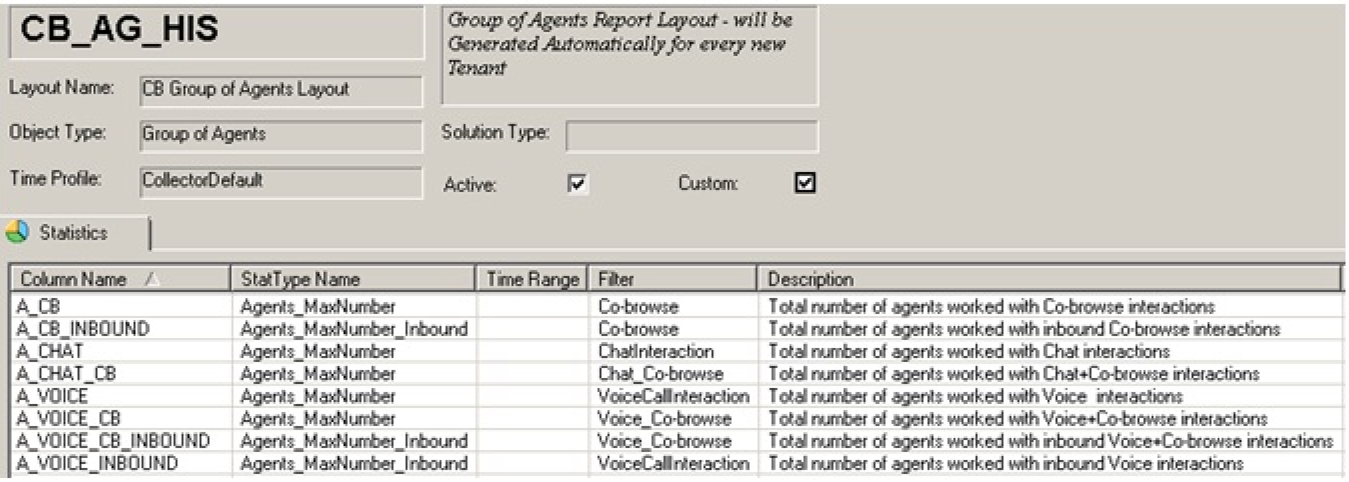
Genesys Co-browse includes the following historical reporting templates that allow you to:

- `CB_AG_HIS.xml` — create and activate historical Report layout to collect Co-browse statistics against Agent Group(s) in the reporting databases for the CCPulse+ views using `Co-browseAgentsHist.xtpl`.
- `CB_INT_HIS.xml` — create and activate historical Report layout to collect Co-browse statistics against Agent(s) in the reporting databases for the CCPulse+ views using `Co-browseInteractionsHist.xtpl`.
- `CB_INT_AG.xml` — create and activate historical Report layout to collect Co-browse statistics against Agent Group(s) in the reporting databases for the CCPulse+ views using `Co-browseInteractionsHist.xtpl`.
- `CB_INT_PG.xml` — create and activate historical Report layout to collect Co-browse statistics against Place Group(s) in the reporting databases for the CCPulse+ views using `Co-browseInteractionsHist.xtpl`.
- `CB_INT_PL.xml` — create and activate historical Report layout to collect Co-browse statistics against Place(s) in the reporting databases for the CCPulse+ views using Co-browseInteractionsHist.xtpl.
- `Co-browseAgentsHist.xtpl` — see the total number of agents participated in Co-browse sessions.
- `Co-browseInteractionsHist.xtpl` — see the total number and total duration of Co-browse sessions.

Data Modeling Assistant Templates

CB_AG_HIS.xml

Object to apply: Agent Group.



CB_AG_HIS

Group of Agents Report Layout - will be Generated Automatically for every new Tenant

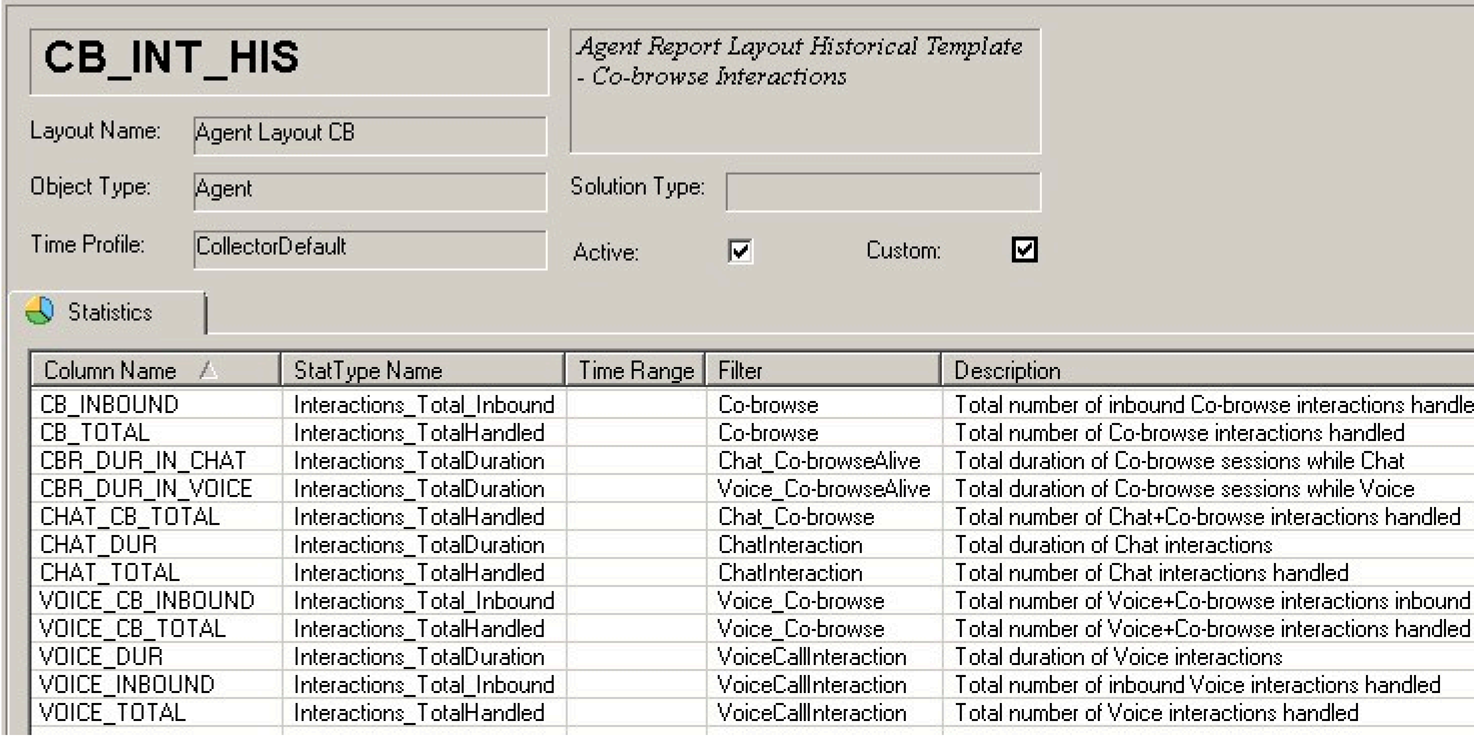
Layout Name:	CB Group of Agents Layout		
Object Type:	Group of Agents	Solution Type:	
Time Profile:	CollectorDefault	Active: ☑	Custom: ☑

Statistics

Column Name	StatType Name	Time Range	Filter	Description
A_CB	Agents_MaxNumber		Co-browse	Total number of agents worked with Co-browse interactions
A_CB_INBOUND	Agents_MaxNumber_Inbound		Co-browse	Total number of agents worked with inbound Co-browse interactions
A_CHAT	Agents_MaxNumber		ChatInteraction	Total number of agents worked with Chat interactions
A_CHAT_CB	Agents_MaxNumber		Chat_Co-browse	Total number of agents worked with Chat+Co-browse interactions
A_VOICE	Agents_MaxNumber		VoiceCallInteraction	Total number of agents worked with Voice interactions
A_VOICE_CB	Agents_MaxNumber		Voice_Co-browse	Total number of agents worked with Voice+Co-browse interactions
A_VOICE_CB_INBOUND	Agents_MaxNumber_Inbound		Voice_Co-browse	Total number of agents worked with inbound Voice+Co-browse interactions
A_VOICE_INBOUND	Agents_MaxNumber_Inbound		VoiceCallInteraction	Total number of agents worked with inbound Voice interactions

CB_INT_HIS.xml

Object to apply: Agent.



CB_INT_HIS

Agent Report Layout Historical Template - Co-browse Interactions

Layout Name:	Agent Layout CB		
Object Type:	Agent	Solution Type:	
Time Profile:	CollectorDefault	Active: ☑	Custom: ☑

Statistics

Column Name	StatType Name	Time Range	Filter	Description
CB_INBOUND	Interactions_Total_Inbound		Co-browse	Total number of inbound Co-browse interactions handle
CB_TOTAL	Interactions_TotalHandled		Co-browse	Total number of Co-browse interactions handled
CBR_DUR_IN_CHAT	Interactions_TotalDuration		Chat_Co-browseAlive	Total duration of Co-browse sessions while Chat
CBR_DUR_IN_VOICE	Interactions_TotalDuration		Voice_Co-browseAlive	Total duration of Co-browse sessions while Voice
CHAT_CB_TOTAL	Interactions_TotalHandled		Chat_Co-browse	Total number of Chat+Co-browse interactions handled
CHAT_DUR	Interactions_TotalDuration		ChatInteraction	Total duration of Chat interactions
CHAT_TOTAL	Interactions_TotalHandled		ChatInteraction	Total number of Chat interactions handled
VOICE_CB_INBOUND	Interactions_Total_Inbound		Voice_Co-browse	Total number of Voice+Co-browse interactions inbound
VOICE_CB_TOTAL	Interactions_TotalHandled		Voice_Co-browse	Total number of Voice+Co-browse interactions handled
VOICE_DUR	Interactions_TotalDuration		VoiceCallInteraction	Total duration of Voice interactions
VOICE_INBOUND	Interactions_Total_Inbound		VoiceCallInteraction	Total number of inbound Voice interactions handled
VOICE_TOTAL	Interactions_TotalHandled		VoiceCallInteraction	Total number of Voice interactions handled

CB_INT_AG.xml

Object to apply: Agent Group.
The Statistics and Time Profile are the same as in CB_INT_HIS.xml.

CB_INT_PG.xml

Object to apply: Place Group.
The Statistics and Time Profile are the same as in CB_INT_HIS.xml.

CB_INT_PL.xml

Object to apply: Place.
The Statistics and Time Profile are the same as in CB_INT_HIS.xml.

CCPulse+ Templates

> **Important**
>
> For CCPulse+ templates to be imported correctly, you must rename each template using the Import/Export Wizard before applying the import procedure.

Co-browseAgentsHist.xtpl

Object to apply: Agent Group.
Available CCPulse+ Views:

- Create Historical View.

<Object>	Statistic	Values
TotalNumber	Agents worked on Chat with CB	*Total number of agents worked on Chat with Co-browse*
	Agents worked on Chat	*Total number of agents worked on Chat*
	Agents worked on inbound Voice with CB	*Total number of agents worked on Inbound Voice with Co-browse*
	Agents worked on inbound Voice	*Total number of agents worked on Inbound Voice*
	Agents worked on Voice with CB	*Total number of agents worked on Voice (Inbound, Internal, Consult) with Co-browse*
	Agents worked on Voice	*Total number of agents worked on Voice*
	Agents worked on inbound CB	*Total number of agents worked on Inbound Voice and Chat with Co-browse*
	Agents worked on CB	*Total number of agents worked on Voice and Chat with Co-browse*

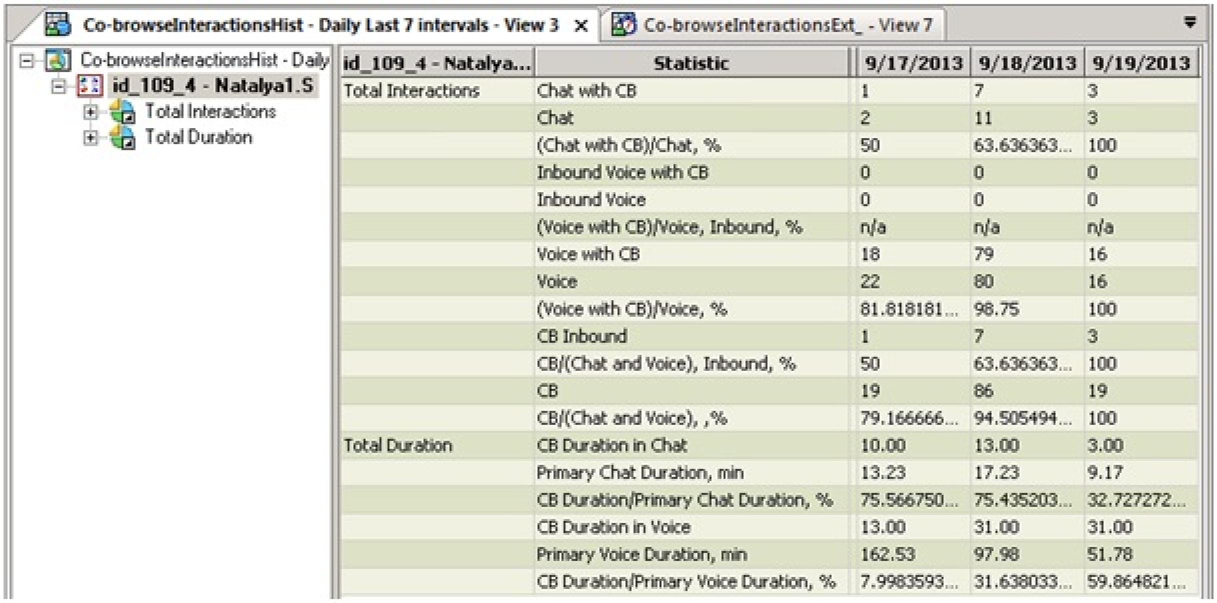
Co-browseInteractionsHist.xtpl

Object to apply: Agent Group, Place Group, Place, Agent.
Available CCPulse+ Views:

- Create Historical View
- Create Historical View for Members (Agent Group).

<Object>	Statistic	Values
Total Interactions	Chat with CB	*Total number of Chat with Co-browse interactions*
	Chat	*Total number of Chat interactions*
	(Chat with CB)/Chat, %	*Ratio of Total number of Chat with Co-browse interactions as opposed to Chat interactions*
	Inbound Voice with CB	*Total number of Inbound Voice with Co-browse interactions*
	Inbound Voice	*Total number of Inbound Voice interactions*
	(Voice with CB)/Voice, Inbound, %	*Ratio of total number of Inbound Voice with Co-browse interactions as opposed to Inbound Voice interactions*
	Voice with CB	*Total number of Voice (Inbound, Internal, Consult) with Co-browse interactions*
	Voice	*Total number of Voice interactions*
	(Voice with CB)/Voice, %	*Ratio of total number of Voice with Co-browse interactions as opposed to Voice interactions*
	CB Inbound	*Total number of Chat and Inbound Voice with Co-browse interactions*
	CB/(Chat and Voice), Inbound, %	*Ratio of total number of Chat and Inbound Voice with Co-browse interactions as opposed to Chat and Inbound Voice interactions*
	CB	*Total number of Chat and Voice with Co-browse interactions*
	CB/(Chat and Voice), %	*Ratio of total number of Chat and Voice with Co-browse interactions as opposed to Chat and Voice interactions*
Total Duration	CB Duration in Chat, hh:mm:ss	*Total duration of Co-browse sessions in Chat interactions*
	Chat Duration, hh:mm:ss	*Total duration of Chat interactions*

<Object>	Statistic	Values
	CB Duration in Chat/Chat Duration, %	*Ratio of total duration of Co-browse sessions in Chat as opposed to Chat interactions duration*
	CB Duration in Voice, hh:mm:ss	*Total duration of Co-browse sessions in Voice interactions*
	Voice Duration, hh:mm:ss	*Total duration of Voice interactions*
	CB Duration in Voice/Voice Duration, %	*Ratio of total duration of Co-browse sessions in Voice as opposed to Voice interactions duration*


Co-browseInteractionsHist - Daily Last 7 intervals - View 3
Co-browseInteractionsExt_ - View 7
Co-browseInteractionsHist - Daily
id_109_4 - Natalya1.S
Total Interactions
Total Duration

id_109_4 - Natalya...	Statistic	9/17/2013	9/18/2013	9/19/2013
Total Interactions	Chat with CB	1	7	3
	Chat	2	11	3
	(Chat with CB)/Chat, %	50	63.636363...	100
	Inbound Voice with CB	0	0	0
	Inbound Voice	0	0	0
	(Voice with CB)/Voice, Inbound, %	n/a	n/a	n/a
	Voice with CB	18	79	16
	Voice	22	80	16
	(Voice with CB)/Voice, %	81.818181...	98.75	100
	CB Inbound	1	7	3
	CB/(Chat and Voice), Inbound, %	50	63.636363...	100
	CB	19	86	19
	CB/(Chat and Voice), ,%	79.166666...	94.505494...	100
Total Duration	CB Duration in Chat	10.00	13.00	3.00
	Primary Chat Duration, min	13.23	17.23	9.17
	CB Duration/Primary Chat Duration, %	75.566750...	75.435203...	32.727272...
	CB Duration in Voice	13.00	31.00	31.00
	Primary Voice Duration, min	162.53	97.98	51.78
	CB Duration/Primary Voice Duration, %	7.9983593...	31.638033...	59.864821...

Example of Co-browseInteractionsHist View.

Configuration Options

Important

For Co-browser Server clusters, every Co-browse Server in the cluster generally plays the same role as the others, except some embedded Cassandra nodes act as seed nodes. This means that to see consistent behavior on the cluster, regardless of which server serves requests, all Co-browse Servers should have the same options set in their application objects in Configuration Server. The rule of thumb is to configure the cluster servers the same, unless it is absolutely necessary to do otherwise (for example, a port is busy on a machine). This simplifies maintenance of production deployments.

Co-browse Server

You can set the following configuration options on your Co-browse Server application in Genesys Administrator:

Section Name	Options
cassandraEmbedded *Configure embedded Cassandra to support the Co-browse Server cluster*	enabled clusterName seedNodes commitLogDirectory dataDirectory savedCachesDirectory listenAddress rpcAddress rpcPort nativeTransportPort storagePort sslStoragePort configFile
cassandraKeyspace *Configure Cassandra keyspace*	dataCompression name readConsistencyLevel writeConsistencyLevel replicationStrategy replicationStrategyParams retention.entity.all retention.entity.chat_session retention.entity.live_sessions retention.entity.session_history retention.entity.window_history retention.time-unit
cross-origin *Configure list of websites allowed to access the Co-browse server*	allowedOrigins disableHttpOptionsRequest

Section Name	Options
chat *Settings for chat*	connectionTimeout queueKey useChat refreshTaskPeriod refreshPoolSize sessionRestorationTimeout
cluster *Configure the Co-browse Server cluster*	url serverUrl
cometd *Settings for CometD*	logLevel maxInterval
forward-proxy *Configure a forward proxy*	host port user password
http-proxy *Configures Co-browse Server's HTTP proxy functionality*	allowedExternalDomains clientTlsProtocols
http-security *Configure HTTP security for Co-browse resources*	disableCaching
log *Configure the logs generated by the Co-browse Server*	all expire segment time_convert time_format trace verbose
metrics *Configure metrics tracked by the Co-browse Server*	reporter.jmx.enabled reporter.log.enabled reporter.log.logFrequency reporter.messageServer.enabled reporter.messageServer.logFrequency reporter.console.enabled reporter.console.logFrequency HeapMemoryUsage.threshold GcFrequency.threshold GcLatency.threshold <metricName>.threshold <metricName>.slidingWindowSize ServerResponseTime.slidingWindowSize ServerResponseTime.threshold SlaveRenderLatency.threshold JettyThreadPoolUsage.threshold InactiveSessions.threshold
security *Enable TLS on connections with other Genesys servers*	provider trusted-ca truststore-password
session *Configure DOM restrictions*	domRestrictionsURL

Section Name	Options
	inactivityDuration
slave *Configure localization for the agent side UI*	localization cssPatchUrl theme disableWebSockets externalJS wweOrigins allowedThirdPartyDomains
static-web-resources *Configure static web resources*	browserHardCacheDuration

Co-browse Plug-in for Interaction Workspace

You can set the following configuration options for the Co-browse plug-in on your Interaction Workspace application in Genesys Administrator:

Section Name	Options
cobrowse *Configure the Co-browse Plug-in for Workspace Desktop Edition*	url disableCertificateValidation useBrowserLogging agentSessionsLimit

cassandraEmbedded Section

> **Important**
>
> Starting in 8.5.0, Embedded Cassandra mode is deprecated in Genesys Co-browse; support for this mode will be discontinued in 9.0.

The cassandraEmbedded section configures embedded Cassandra support for the Co-browse Server cluster.

enabled

Default Value: `true`
Valid Values: `true` or `false`
Changes Take Effect: After Co-browse server restart

Specifies whether or not Co-browse server should act as a Cassandra cluster node.

clusterName

Default Value: `Cluster`
Valid Values: Any string
Changes Take Effect: After Co-browse server restart

The name of the embedded Cassandra cluster node. This option is mainly used to prevent machines in one logical cluster from joining another. For more information, see http://docs.datastax.com/en/cassandra/2.1/cassandra/configuration/configCassandra_yaml_r.html?scroll=reference_ds_qfg_n1r_1k__cluster_name

seedNodes

Default Value: `localhost`
Valid Values: Comma-delimited list of IP addresses
Changes Take Effect: After Co-browse server restart

When a node joins a cluster, it contacts the seed node(s) listed in this option to determine the ring topology and get gossip information about the other nodes in the cluster.

Every node in the cluster should have the same list of seeds specified as a comma-delimited list of IP addresses. In multiple data center clusters, the seed list should include at least one node from each data center (replication group). For more information, see http://docs.datastax.com/en/cassandra/2.1/cassandra/configuration/configCassandra_yaml_r.html?scroll=reference_ds_qfg_n1r_1k__seed_provider.

This option is only applicable when embedded Cassandra service is activated.

commitLogDirectory

Default Value: `./storage/commitLog`
Valid Values: Valid directory path. The directory may not exist.
Changes Take Effect: After Co-browse server restart

Specifies the directory where Cassandra's commitlog directories will be located or created. If left empty, the Co-browse Server web application assumes it is running within a Jetty web container and the storage directory will be a storage sub-directory of the Jetty home directory.

This option is only applicable when embedded Cassandra service is activated.

dataDirectory

Default Value: `./storage/data`
Valid Values: Valid directory path. The directory may not exist.
Changes Take Effect: After Co-browse server restart

Specifies the directory where Cassandra's data will be located or created. If left empty, the Co-browse Server web application assumes it is running within a Jetty web container and the storage directory will be a storage sub-directory of the Jetty home directory.

This option is only applicable when embedded Cassandra service is activated.

savedCachesDirectory

Default Value: `./storage/saved_cache`
Valid Values: Valid directory path. The directory may not exist.
Changes Take Effect: After Co-browse server restart

Specifies the directory where Cassandra's saved_caches directories will be located or created. If left empty, the Co-browse Server web application assumes it is running within a Jetty web container and the storage directory will a "storage" sub-directory of Jetty home directory.

The option is applicable only when embedded Cassandra service is activated.

listenAddress

Default Value: `localhost`
Valid Values: Blank or valid address
Changes Take Effect: After Co-browse server restart

Specifies the address to bind to and to tell other Cassandra nodes to connect to. You *must* change this if you want multiple nodes to be able to communicate.

Leaving this option blank lets `InetAddress.getLocalHost()` set the address. If the node is properly configured (hostname, name resolution), the address will resolve to the address associated with the hostname.

rpcAddress

Default Value: `localhost`
Valid Values: Valid IP address or hostname.

Changes Take Effect: After Co-browse server restart

Specifies the listen address for remote procedure calls (client connections). This option is also used to configure Co-browse server as a client. See http://docs.datastax.com/en/cassandra/2.1/cassandra/configuration/configCassandra_yaml_r.html?scroll=reference_ds_qfg_n1r_1k__rpc_address. If the address is invalid, Co-browse server will not be able to connect to the embedded Cassandra service.

rpcPort

Default Value: `9160`
Valid Values: Any free TCP port
Changes Take Effect: After Co-browse server restart

Specifies the port for remote procedure calls (client connections) and the Thrift service.
http://docs.datastax.com/en/cassandra/2.1/cassandra/configuration/configCassandra_yaml_r.html?scroll=reference_ds_qfg_n1r_1k__rpc_address

nativeTransportPort

Default Value: `9042`
Valid Values: Any free TCP port
Changes Take Effect: After Co-browse server restart

Specifies the port for the CQL native transport to listen for clients.

storagePort

Default Value: `7000`
Valid Values: Any free TCP port
Changes Take Effect: After Co-browse server restart

Specifies the TCP port for commands and data.

sslStoragePort

Default Value: `7001`
Valid Values: Any free TCP port
Changes Take Effect: After Co-browse server restart

Specifies the SSL port for encrypted communication.

configFile

Default Value: none
Valid Values: Valid path to the `*.yaml` cassandra configuration file
Changes Take Effect: After Co-browse server restart

Specifies the Embedded Cassandra external configuration YAML file path. It overrides all Cassandra settings in the section.

endpointSnitch

Default Value: `GossipingPropertyFileSnitch`
Valid Values: `SimpleSnitch`, `GossipingPropertyFileSnitch`, `PropertyFileSnitch`, `Ec2Snitch`, `Ec2MultiRegionSnitch`, or `RackInferringSnitch`
Changes Take Effect: After Co-browse server restart

A snitch determines which nodes belong to which data centers and racks. They inform Cassandra about the network topology so Cassandra can route requests efficiently. They also allow Cassandra to distribute replicas by grouping machines into data centers and racks. Specifically, the replication strategy places the replicas based on the information provided by the new snitch. Also see, http://docs.datastax.com/en/cassandra/2.1/cassandra/architecture/architectureSnitchesAbout_c.html.

Additional options not included in the template

You can also configure the following options which are not included in the template:

> **Important**
>
> All options in this section are applied only after application restart.

[+] Click to view table

Option name	Mandatory	Default Value	Possible Values	Description
partitioner	No	org.apache.cassandra.dht.Murmur3Partitioner	org.apache.cassandra.dht.ByteOrderedPartitioner, org.apache.cassandra.dht.RandomPartitioner, org.apache.cassandra.dht.Murmur3Partitioner	A partitioner determines how data is distributed across the nodes in the cluster (including replicas). Basically, a partitioner is a function for deriving a token representing a row from its partition key, typically by hashing. Each row of data is then distributed across the cluster by the value of the token. http://docs.datastax.com/en/cassandra/2.1/cassandra/architecture/architecturePartitionerAbout_c.html
commitFailurePolicy	No	stop	stop,	Policy for commit disk

Option name	Mandatory	Default Value	Possible Values	Description
			stop_commit, ignore, die	failures: - *die* - Shut down gossip and Thrift and kill the JVM, so the node can be replaced. - *stop* - Shut down gossip and Thrift, leaving the node effectively dead, but can be inspected using JMX. - *stop_commit* - Shut down the commit log, letting writes collect but continuing to service reads - *ignore* - Ignore fatal errors and let the batches fail
diskFailurePolicy	No	stop	best_effort, stop, ignore, stop_paranoid, die	Sets how Cassandraresponds to disk failure. Recommend settings are *stop* or *best_effort*. - *die* - Shut down gossip and Thrift and kill the JVM for any file system errors or single SSTable errors, so the node can be replaced. - *stop_paranoid* - Shut down gossip and Thrift even for single SSTableerrors.

Option name	Mandatory	Default Value	Possible Values	Description
				- *stop* - Shut down gossipand Thrift, leaving the node effectively dead, but available for inspection using JMX. - *best_effort* - Stop usingthe failed disk and respond to requests based on the remaining available SSTables. This means you will see obsolete data at consistency level of ONE. - *ignore* - Ignores fatal errors and lets the requests fail; all file system errors are logged but otherwise ignored.
autoBootstrap	No	true	true, false	This setting has been removed from default configuration. It makes new (non-seed) nodes automatically migrate the right data to themselves. When initializing a fresh cluster without data, set this option to *false*
batchSizeWarnThreshold	No	5	Valid integer	Log WARN on any batch size exceeding this value in kilobytes. Caution should be

Option name	Mandatory	Default Value	Possible Values	Description
				taken on increasing the size of this threshold as it can lead to node instability
concurrentReads	No	32	Valid ineteger	For workloads with more data than can fit in memory, the bottleneck is ads fetching data from disk. Setting to 16×number_of_drives allows operations to queue low enough in the stack so that the OS and drives can reorder them. The default setting applies to both logical volume managed (LVM) and RAID drives
concurrentWrites	No	32	Valid ineteger	Writes in Cassandra are rarely I/O bound, so the ideal number of concurrent writes depends on the number of CPU cores in your system. The recommended value is 8×number_of_cpu_cores
concurrentCounterWrites	No	32	Valid ineteger	Counter writes read the current values before incrementing and writing them back. The recommended value is 16×number_of_drives
streamThroughputOutbound	No	200	Valid integer	Throttles all outbound streaming file transfers on a node to the specified throughput (Megabits/ seconds).

Option name	Mandatory	Default Value	Possible Values	Description
				Cassandra does mostly sequential I/O when streaming data during bootstrap or repair, which can lead to saturating the network connection and degrading client (RPC) performance.
interDCStreamThroughputOutbound	No		Valid integer	Throttles all streaming file transfer between the data centers (Megabits/ seconds).. This setting allows throttles streaming throughput betweens data centers in addition to throttling all network stream traffic as configured with **streamThroughputOutbound**
trickleFsync	No	false	true, false	When doing sequential writing, enabling this option tells fsync to force the operating system to flush the dirty buffers at a set interval **trickleFsyncInterval** . Enable this parameter to avoid sudden dirty buffer flushing from impacting read latencies. Recommended to use on SSDs, but not on HDDs.
trickleFsyncInterval	No	10240	Valid integer	Sets the size of the fsync in kilobytes
autoSnapshot (NODE ONLY)	No	true	true, false	Enable or disable whether a snapshot is taken

Option name	Mandatory	Default Value	Possible Values	Description
				of the data before keyspace truncation or dropping of tables. To prevent data loss, using the default setting is strongly advised. If you set to false, you will lose data on truncation or drop
incrementalBackups	No	false	true, false	Backs up data updated since the last snapshot was taken. When enabled, Cassandra creates a hard link to each SSTable flushed or streamed locally in a backups/ subdirectory of the keyspace data. Removing these links is the operator's responsibility
snapshotBeforeCompation	No	false	true, false	Enable or disable taking a snapshot before each compaction. This option is useful to back up data when there is a data format change. Be careful using this option because Cassandra does not clean up older snapshots automatically
commitLogSync	No	periodic	periodic, batch	The method that Cassandra uses to acknowledge writes http://docs.datastax.com/en/cassandra/2.1/cassandra/dml/dml_durability_c.html
commitLogSyncPeriod	No	10000	Valid integer	The period that Cassandra uses to acknowledge writes in

Option name	Mandatory	Default Value	Possible Values	Description
				milliseconds
commitLogSegmentSize	No	32	Valid integer	Sets the size (in Mb) of the individual commitlog file segments. A commitlog segment may be archived, deleted, or recycled after all its data has been flushed to SSTables. This amount of data can potentially include commitlog segments from every table in the system. The default size is usually suitable for most commitlog archiving, but if you want a finer granularity, 8 or 16 MB is reasonable.
commitLogTotalSpace	No	8192	Valid integer	Total space used for commitlogs. If the used space goes above this value,Cassandra rounds up to the next nearest segment multiple and flushes memtables to disk for the oldest commitlog segments, removing those log segments. This reduces the amount of data to replay on start-up, and prevents infrequently-updated tables from indefinitely keeping commitlog segments. A small total commitlog space tends to cause more flush activity on less-

Option name	Mandatory	Default Value	Possible Values	Description
				active tables
concurrentCompators	No		Valid integer	Sets the number ofconcurrent compaction processes allowed to runsimultaneously on a node, not including validation compactions for anti-entropy repair. Simultaneouscompactions help preserve read performance in amixed read-write workload by mitigating the tendencyof small SSTables to accumulate during a single long-running compaction. If your data directoriesare backed by SSD, increase this value to the numberof cores. If compaction running too slowly or too fast, adjust **compactionThroughput** first. If not set the value will be calculated: Smaller of number of disks or number of cores,with a minimum of 2 and a maximum of 8 per CPU core
sstablePreemptiveOpenInterval	No	50	Valid integer	When compacting, the replacement opens SSTables before they arecompletely written and uses in place of the prior SSTables for any range previously written (in Mb). This setting helps to smoothly transfer reads between the SSTables by reducing page cache churn and keeps hot rows hot.
compactionThroughput	No	16	Valid integer	Throttles compaction to the specified total throughput across the entire system (in Mb/seconds). The faster you insert data, the faster you need to compact in order to keep the SSTable count down. The

Option name	Mandatory	Default Value	Possible Values	Description
				recommended value is 16 to 32 times the rate of write throughput (in MB/second). Setting the value to 0 disables compaction throttling.
compactionLargePartitionWarningThreshold	No	100	Valid integer	Logs a warning when compaction partitions larger than the set value in Mb
numTokens	No	256	Valid integer	Defines the number of tokens randomly assigned to this node on the ring when using virtual nodes (vnodes). The more tokens, relative to other nodes, the larger the proportion of data that the node stores.
memtableAllocationType	No	heap_buffers	unslabbed_heap_buffers, heap_buffers, offheap_buffers, offheap_objects	Specify the way Cassandra allocates and manages memtable memory. See Off-heap memtablesin Cassandra 2.1.
memtableCleanupThreshold	No		Valid float	Ratio of occupied non-flushing memtable size to total permitted size for triggering a flush of the largest memtable. Larger values mean larger flushes and less compaction, but also less concurrent flush activity, which can make it difficult to keep your disks saturated under heavy write load. If not set the value will be calculated as 1/(1 + **memtableFlushWriters**)
memtableFlushWriters	No		Valid integer	Sets the number of memtable flush writer threads. These threads

Option name	Mandatory	Default Value	Possible Values	Description
				are blocked by disk I/O, and each one holds a memtable in memory while blocked. If your data directories are backed by SSD, increase this setting to the number of cores. If not set the value will be calculated as (Smaller of number of disks or number of cores with a minimum of 2 and a maximum of 8)
memtableHeapSize	No		Valid integer	Total permitted memory (in Mb) to use for memtables. Triggers a flush based on **memtableCleanupThreshold**. Cassandra stops accepting writes when the limit is exceeded until a flush completes If not set the value will be calculated as (1/4 heap)
memtableOffheapSpace	No		Valid integer	If not set the value will be calculated as (1/4 heap)
fileCacheSize	No		Valid integer	Total memory to use for SSTable-reading buffers. If not set the value will be calculated as (Smaller of 1/4 heap or 512)
authenticator	No	org.apache.cassandra.auth.AllowAllAuthenticator	org.apache.cassandra.auth.AllowAllAuthenticator, org.apache.cassandra.auth.PasswordAuthenticator	The authentication backend http://docs.datastax.com/en/cassandra/2.1/cassandra/security/secure_about_native_authenticate_c.html
authorizer	No	org.apache.cassandra.auth.AllowAllAuthorizer	org.apache.cassandra.auth.AllowAllAuthorizer, org.apache.cassandra.auth.CassandraAuthorizer	The authorization backend http://docs.datastax.com/en/cassandra/2.1/

Option name	Mandatory	Default Value	Possible Values	Description
				cassandra/security/ secure_about_native_authorize_c.html
permissionsValidity	No	2000	Valid integer	How long (in milliseconds) permissions in cache remain valid. Depending on the authorizer, such as *org.apache.cassandra.auth.Cassa* fetching permissions can be resource intensive. This setting disabled when set to 0 or when *org.apache.cassandra.auth.AllowA* is set.
permissionsUpdateInterval	No		Valid integer	Refresh interval (in milliseconds) for permissions cache (if enabled). After this interval, cache entries become eligible for refresh. On next access, an async reload is scheduled and the old value is returned until it completes. If **permissionsValidity**, then this property must benon-zero If not set the value will be the same like **permissionsValidity**
writeTimeout	No	2000	Valid long	The time that the coordinator waits for write operations to complete
readTimeout	No	5000	Valid long	The time that the coordinator waits for read operations to complete
rangeTimeout	No	10000	Valid long	The time that the coordinator waits for sequential or index scans to complete
counterWriteTimeout	No	5000	Valid long	The timethat the coordinator waits for counter writes to complete

Option name	Mandatory	Default Value	Possible Values	Description
casContentionTimeout	No	1000	Valid long	The time that the coordinator continues to retry a CAS (compare and set) operation that contends with other proposals for the same row.
truncateTimeout	No	60000	Valid long	The time that the coordinator waits for truncates (remove all data from a table) to complete. The long default value allows for a snapshot to be taken before removing the data. If **autoSnapshot** is disabled (not recommended), you can reduce this time.
requestTimeout	No	10000	Valid long	The default time for other miscellaneous operations
encryption.server.internode	No	none	none, all, dc, rack	Enable or disable inter-node encryption. You must also generate keys and provide the appropriate key and trust store locations and passwords. No custom encryption options are currently enabled http://docs.datastax.com/en/cassandra/2.1/cassandra/security/secureSSLNodeToNode_t.html
encryption.server.keystore	No	conf/.keystore	Valid path	The location of a Java keystore (JKS) suitable for use with Java Secure Socket Extension (JSSE), which is the Java version of the Secure Sockets Layer (SSL), and Transport Layer Security (TLS)

Option name	Mandatory	Default Value	Possible Values	Description
				protocols. The keystore contains the private key used to encrypt outgoing messages
encryption.server.keystorePassword	No	cassandra		Password for the keystore
encryption.server.truststore	No	conf/.truststore	Valid path	Location of the truststore containing the trusted certificate for authenticating remote servers
encryption.server.truststorePassword	No	cassandra		Password for the truststore
encryption.client.enabled	No	false	true, false	Enable or disable client-to-node encryption. You must also generate keys and provide the appropriate key and trust store locations and passwords. No custom encryption options are currently enabled http://docs.datastax.com/en/cassandra/2.1/cassandra/security/secureSSLClientToNode_t.html
encryption.client.keystore	No	conf/.keystore	Valid path	The location of a Java keystore (JKS) suitable for use with Java Secure Socket Extension (JSSE), which is the Java version of the Secure Sockets Layer (SSL), and Transport Layer Security (TLS) protocols. The keystore contains the private key used to encrypt outgoing messages
encryption.client.keystorePassword	No	cassandra		Password for the keystore. This must match the password used when generating the keystore and

Option name	Mandatory	Default Value	Possible Values	Description
				truststore.
encryption.client.truststore	No	conf/.truststore	Valid path	Set if encryption.client.clientAuth is true
encryption.client.truststorePassword	No	*<truststore_password>*		Set if encryption.client.clientAuth is true
encryption.client.clientAuth	No	false	true, false	Enables or disables certificate authentication

cassandraKeyspace Section

dataCompression

Default Value: `lz4`
Valid Values: `lz4`, `snappy`, `deflate`
Changes Take Effect: Applied when the keyspace is created

Specifies data compression algorithm used when data is stored on the disk. Compression maximizes the storage capacity of Cassandra nodes by reducing the volume of data on the disk and disk I/O, particularly for read-dominated workloads. Cassandra quickly finds the location of rows in the SSTable index and decompresses the relevant row chunks. See http://docs.datastax.com/en/cassandra/2.0/cassandra/operations/ops_config_compress_t.html

name

Default Value: `Cobrowse`
Valid Values: string
Changes Take Effect: Applied when the keyspace is created

Specifies the Cassandra keyspace name where Co-browse server data will be stored. If the keyspace with this name does not exist in the cluster, it will be created automatically.

readConsistencyLevel

Default Value: `LOCAL_QUORUM`
Valid Values: `ALL` , `EACH_QUORUM` , `QUORUM` , `LOCAL_QUORUM` , `ONE` , `TWO` , `THREE` , `LOCAL_ONE` , `ANY`
Changes Take Effect: After Co-browse server restart

Specifies the consistency level. Determines the number of replicas on which the read must succeed before returning any data to the client application.

writeConsistencyLevel

Default Value: `LOCAL_QUORUM`
Valid Values: `ALL` , `EACH_QUORUM` , `QUORUM` , `LOCAL_QUORUM` , `ONE` , `TWO` , `THREE` , `LOCAL_ONE` , `ANY`
Changes Take Effect: After Co-browse server restart

Specifies the consistency level. Determines the number of replicas on which the write must succeed before returning an acknowledgment to the client application.

replicationStrategy

Default Value: `NetworkTopologyStrategy`
Valid Values: `SimpleStrategy` , `NetworkTopologyStrategy`
Changes Take Effect: Applied when the keyspace is created.

Specifies the keyspace replica placement strategy. See http://docs.datastax.com/en/cassandra/2.1/cassandra/architecture/architectureDataDistributeReplication_c.html.

> **Warning**
>
> Genesys strongly recomends **not** using `SimpleStrategy`. If you use `SimpleStrategy`, you cannot use more than one data center and KeySpaces you create will be difficult to migrate to `NetworkTopologyStrategy` once they contain a lot of data. Also see https://docs.datastax.com/en/cassandra/2.0/cassandra/architecture/architectureDataDistributeReplication_c.html

If you set your replication strategy to `SimpleStrategy` you must also:

- Configure a replication factor in the **replicationStrategyParams** option.
- For embedded Cassandra, set the **endpointSnitch** option of the **cassandraEmbedded** section to `SimpleSnitch`.
- For external Cassandra, set `endpoint_snitch: SmipleSnitch` in your **cassandra.yaml** file.

replicationStrategyParams

Default Value: `'OperationalDC':1`
Valid Values:

For `NetworkTopologyStrategy` set the value to comma separated pairs of `'[Some_Data_Center_Name]':[replication factor number]`

When the replication strategy is `SimpleStrategy`, value should contain either a number or `'replication_factor':<number of replication factor>`. For example, 3 or `'replication_factor':3`.

Changes Take Effect: After Co-browse server restart.

Comma separated parameters which define how many replicas you want per data center.

retention.entity.all

Default Value: 1
Valid Values: Positive integer in time-units. Must be equal to or greater than any other `retention.entity` value.
Changes Take Effect: After Co-browse server restart.

Specifies the default duration in time-units (days by default) to keep data in a column family if a special duration is not present for the column family.

retention.entity.chat_session

Default Value: 1
Valid Values: Positive integer in time-units. Must be greater than the amount of time expressed by the `cometd/maxInterval` option.
Changes Take Effect: After Co-browse server restart.

> **Warning**
>
> This option is not yet in the template and will be added in the 8.5.1 release.

Specifies duration to keep data in the chat_session column family. This column family stores chat session data.

retention.entity.live_sessions

Default Value: 1
Valid Values: Positive integer in time-units. Must be greater than the amount of time expressed by the `cometd/maxInterval` option.
Changes Take Effect: After Co-browse server restart.

> **Warning**
>
> Starting with 8.5.003.04, the **retention.entity.livesessionentity** opton is now **retention.entity.live_sessions**. As documented in this known issue, the old option name is still in the configuration option template.

Specifies the duration in time-units (days by default) to keep data in the `live_sessions` column family. The column family stores the core Co-browse session state shared in the Co-browse cluster. In a normal situation, data from this column family is removed automatically shortly after a session is deactivated (when cometd/maxInterval elapses), but the retention policy mechanism guarantees that the data will be removed anyway. Default is 1 time-unit.

retention.entity.session_history

Default Value: 1
Valid Values: Positive integer in time-units. Must be greater than the amount of time expressed by the `cometd/maxInterval` option.
Changes Take Effect: After Co-browse server restart.

> **Warning**
>
> Starting with 8.5.003.04, the **retention.entity.sessionhistoryentity** option is now **retention.entity.session_history**. As documented in this known issue, the old option name is still in the configuration option template.

Specifies the duration in time-units (days by default) to keep data in the `session_history` column family. The column family stores session historical data accessible through the REST API.

retention.entity.window_history

Default Value: 1
Valid Values: Positive integer in time-units. Must be greater than the amount of time expressed by the `cometd/maxInterval` option.
Changes Take Effect: After Co-browse server restart.

> ### Warning
> Starting with 8.5.003.04, the **retention.entity.windowhistoryentity** option is now **retention.entity.window_history**. As documented in this known issue, the old option name is still in the configuration option template.

Specifies the duration in time-units (days by default) to keep data in the `window_history` column family. The column family is a temporary store for page navigation information.

retention.time-unit

Default Value: `day`
Valid Values: `sec`, `min`, `hour`, `day`, or `month`
Changes Take Effect: After Co-browse server restart.

Specifies the retention time unit to define `retention.entity` values.

cross-origin Section

allowedOrigins

Default Value: `"*"`
Valid Values: List of origins in the format `<scheme>://<domain>[:<port>]`

For example:

- `"http://*.genesys.com"`
- `"http://intranet.domain.com:8700,http://<host>:<port>,http://<ip_address>"`

Changes Take Effect: Immediately

A comma separated list of origins, such as instrumented web sites, allowed to access the Co-browse Server. If an allowed origin contains one or more `"*"` characters (for example, http://*.domain.com), then `"*"` characters are converted to `".*"`. Any `"."` characters are converted to `"\."`. The resulting allowed origin can be interpreted as a regular expression.

disableHttpOptionsRequest

Default Value: `false`
Valid Values: `true` or `false`
Changes Take Effect: Immediately

Available starting with Co-browse Server 8.5.003.07

If set to `true`, disables all HTTP OPTIONS requests.

Disabling HTTP OPTIONS may slightly affect performance as it blocks the long-polling transport, which needs `OPTIONS` requests. Generally, Genesys does not advise you disable `HTTP OPTIONS` unless required by your security policies.

chat Section

> **Important**
>
> Starting with the 8.5.100.11 release of Genesys Co-browse, Genesys is deprecating the Built-in Chat Widget and its APIs in preparation for discontinuing support in the upcoming 9.0 release.
>
> This functionality is now available through a single set of consumer-facing digital channel APIs that are part of Genesys Mobile Services (GMS), and through Genesys Widgets, a set of productized widgets that are optimized for use with desktop and mobile web clients, and which are based on the GMS APIs.
>
> Genesys Widgets provide for an easy integration with Co-browse, allowing you to proactively serve these widgets to your web-based customers.
>
> Although the deprecated APIs and Built-in Chat Widget will be supported for the life of the 8.5 release of Co-browse, Genesys recommends that you move as soon as you can to the new APIs and to Genesys Widgets to ensure that your functionality is not affected when you migrate to the 9.0 release.

connectionTimeout

Default Value: `10000`
Valid Values: Any positive integer
Changes Take Effect: After restart

Specifies the connection timeout, in milliseconds, when Co-browse Server communicates with Chat Server.

queueKey

Default Value: None
Valid Values: `<tenant id>:<chat access point name>`
Changes Take Effect: After restart

Specifies the access point that is used to place submitted chat interactions. For example, `1:default` or `101:chat_queue`. This option must be specified if the value of `useChat` is `true`.

useChat

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: After restart

Specifies whether Co-browse Server uses the built-in Chat Server functionality. If `true`, Co-browse Server acts as a Chat Server client and HTTP "gateway" between the customer browser and Chat Server. If `false`, chat-related functions are disabled on the Co-browse Server.

refreshTaskPeriod

Default Value: `3000`
Valid Values: Positive numeric
Changes Take Effect: After restart

Period of time before Co-browse is pinged for new tasks. Period should be small enough for fast replies but not too large to overload ChatServer with requests. Suggested time is around 5 seconds.

refreshPoolSize

Default Value: `10`
Valid Values: Positive numeric
Changes Take Effect: After restart

Amount of working threads fetching updates of chat session transcripts. The following formula can be used to calculate option value:

(<expected count of simultaneuosly chatting agent> * <average time of single Refresh request processing in milliseconds>) / (<count of servers in cluster> * <`refreshTaskPeriod` in milliseconds>)

Example:

- 1000 expected agents (peak loading)
- 5 Co-browse servers with chat components
- `refreshTaskPeriod` value of 5000 milliseconds
- Average time of processing Refresh command of 100 milliseconds

If the customer expects the values above, the estimated pool size is (1000 * 100) / (5 * 5000) = 4. If `refreshTaskPeriod` is 2000, the formula results in `10`.

sessionRestorationTimeout

Default Value: `10000`
Valid Values: Positive numeric
Changes Take Effect: After restart

Period of time that client tries to establish connection with Chat Server for a particular chat session. After timeout, session terminates. This value should be big enough to cover unexpected short term network problems but small enough not to annoy visitors with frozen chat window. Values from 10 to 30 seconds are recommended.

cluster Section

url

Important

The **url** option is mandatory.

Default Value: None
Valid Values: Valid HTTP or HTTPS absolute URL
Changes Take Effect: Immediately

Specifies the HTTP(S) URL of the Co-browse cluster, for example, `http://[host]:[port]/cobrowse`. Typically, the value is the URL of the load balancer. Set this value when you create your Co-browse Cluster Application. Since Co-browse 8.5.0, you can set the **url** value to an HTTP or HTTPS based URL.

Important

In Co-browse 8.5.002+, only the agent side directly uses the cluster URL. The end user (customer) side uses the URL provided in the Website Instrumentation. You can have two load balancers, an internal load balancer for agents which you specify in this option and a public load balancer for end users to use in the JS instrumentation. Depending on your infrastructure's setup, two load balancers may benefit traffic.

Tip

The **secureURL** and **useSecureConnection** options were discontinued starting with Co-browse 8.5.0 because HTTPS URLs can now be configured from the **url** option.

serverUrl

Important

- The **serverUrl** option is **not** mandatory in most cases while the url option above is always mandatory.
- You must leave **serverUrl** empty when using Workspace *Desktop* Edition as the agent application.

- You must specify **serverUrl** when using Worspace *Web* Edition as the agent application.

Default Value: None
Valid Values: Any valid public URL. The URL must point to a Co-browse web application such as `http(s)://<host>:<port>/cobrowse` and should not include a trailing slash.
Changes Take Effect: For new Co-browse sessions

This option is used to configure URL-Based Stickiness. For more information on routing all requests from the customer and agent sides to the same Co-browse node within a given session (that is, *sticking* to the same node), see Stickiness.

cometd Section

logLevel

Default Value: `Info`
Valid Values: `Off`, `Config`, `Info`, `Debug`
Changes Take Effect: After restart

Sets the CometD (Bayeux) server logging level.

maxInterval

Default Value: `600000`
Valid Values: Any positive integer
Changes Take Effect: After restart

Specifies the period of time (in milliseconds) after which CometD clients (mainly browsers) that do not send CometD connect requests are considered lost. If the customer side or agent side Co-browse session clients are disconnected, the session automatically ends.

forward-proxy Section

Configures forward proxy options to let the Co-browse server obtain public web resources in an environment where the Internet is accessed through a forward proxy (for example, DMZ or local intranet).

> **Important**
>
> The Co-browse server accesses public web resources when it proxies CSS and other co-browsed web site resources.

host

Default value:
Valid Values: Either a domain name or IP address (IPv4 or IPv6)
Changes Take Effect: After Co-browse server restart

The forward proxy host. If the host option is not specified (default), the Co-browse server makes direct connections to the target web servers.

port

Default value:
Valid Values: Valid TCP port
Changes Take Effect: After Co-browse server restart

The forward proxy port. If the `host` option is specified, the port **must also** be specified.

user

Default value:
Valid Values: Valid user name
Changes Take Effect: After Co-browse server restart

User name used in HTTP Basic authentication if the forward proxy requires authentication.

password

Default value:
Valid Values: Valid password
Changes Take Effect: After Co-browse server restart

Password used in HTTP Basic authentication if the forward proxy requires authentication. If the `user` option is specified, the password **must also** be specified.

http-proxy Section

Configures Co-browse Server's HTTP proxy functionality.

allowedExternalDomains

Default value: *
Valid Values: List of any valid domains or wild cards. For example, ***.mydomain.com**,*.net, **mydomain-*.com**.
Changes take effect: Immediately

List of domains from which resources are allowed to be proxied through Co-browse server. This option enforces an additional level of control of what can be included on the web page during a Co-browse session.

clientTlsProtocols

Default value: No value
Valid Values: A comma separated list of values from the following: `TLSv1, TLSv1.1, TLSv1.2`
Changes take effect: After Co-browse server restart

Explicitly lists TLS protocol versions Co-browse server should use when using HTTPS to communicate with proxied resource target servers. Co-browse server does not work with SSL protocol due to its security vulnerabilities. If a target server supports only a specific protocol (for example, TLSv1), specify only this protocol.

http-security Section

Configures HTTP security for Co-browse resources.

disableCaching

Default value: `false`
Valid Values: `true`, `false`
Changes take effect: Immediately

Disables HTTP client / proxy caching for all resources including static resources. When the option value is set to `true`, ZAP proxy does *not* generate the following warnings:

- `Incomplete or no cache-control and pragma HTTPHeader set (Low Risk)`
- `Secure page browser cache (Medium Risk)`

When the option is set to `false` or has no set value:

- Static resources are cached according to the option values set in the static-web-resources section.
- Dynamic data exchange is not cached.
- Proxied resources are cached in accordance with caching policy of the original resources.

More information can be found here:

- https://www.owasp.org/index.php/Session_Management_Cheat_Sheet#Web_Content_Caching
- https://blog.42.nl/articles/securing-web-applications-using-owasp-zap-passive-mode/

log Section

all

Default Value: stdout
Valid Values:

stdout	Log events are sent to the Standard output (stdout).
stderr	Log events are sent to the Standard error output (stderr).
network	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.Setting the all log level option to the network output enables an application to send log events of the Standard, Interaction, and Trace levels to Message Server. Debug-level log events are neither sent to Message Server nor stored in the Log Database.
[filename]	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. Important For remote logging in Windows, use forward slashes in the log path instead of back slashes.

Changes take effect: Immediately

Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example: all = stdout, logfile

expire

Default Value: 10
Valid Values:

false	No expiration; all generated segments are stored.
<number> file or <number>	Sets the maximum number of log files to store. Specify a number from 1—1000.

Changes Take Effect: After server restart

Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) before the files are removed. This option is ignored if log output is not configured to be sent to a log file.

> **Warning**
>
> If this option's value is incorrectly set an out of the range of value it will be automatically reset to `10`.

segment

Default Value: `100 MB`
Valid Values:

`false`	No segmentation is allowed.
`<number> KB or <number>`	Sets the maximum segment size, in kilobytes. The minimum segment size is `100 KB`.
`<number> MB`	Sets the maximum segment size, in megabytes.
`<number> hr`	Sets the number of hours for the segment to stay open. The minimum number is 1 hour.

Changes Take Effect: After server restart

Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.

time_convert

Default Value: `utc`
Valid Values:

`local`	The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the application's host computer is used.
`utc`	The time of log record generation is expressed as Coordinated Universal Time (UTC).

Changes Take Effect: Immediately

Specifies the system in which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since the Epoch time (00:00:00 UTC, January 1, 1970).

time_format

Default Value: `time`
Valid Values:

`time`	The time string is formatted according to the `HH:MM:SS.sss` (hours, minutes, seconds, and milliseconds) format.
`locale`	The time string is formatted according to the system's locale.
`ISO8601`	The date in the time string is formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds.

Changes Take Effect: Immediately

Specifies how to represent, in a log file, the time when an application generates log records. A log record's time field in the ISO 8601 format looks like this: `2001-07-24T04:58:10.123`

trace

Default Value: `stdout`
Valid Values:

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels). The log outputs must be separated by a comma when more than one output is configured. For example: `trace = stderr, network`

verbose

Default Value: `trace`
Valid Values:

`all`	All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated.
`debug`	The same as `all.`
`trace`	Log events of the Trace level and higher (that is,

	log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated.
`interaction`	Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated.
`standard`	Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated.
`none`	No output is produced.

Changes Take Effect: Immediately

Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug.

metrics Section

reporter.jmx.enabled

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Enables or disables the JMX reporter.

reporter.log.enabled

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Enables or disables metrics reporting to a file.

reporter.log.logFrequency

Default Value: `30min`
Valid Values: A positive integer and time unit such as `ms`, `s`, `min`, `h` or `d`. For example, `30min` or `50s`.
Changes Take Effect: Immediately

Defines the reporting frequency for logging to a file.

reporter.messageServer.enabled

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Enables or disables the Message Server reporter.

reporter.messageServer.logFrequency

Default Value: `30min`
Valid Values: A positive integer and time unit such as `ms`, `s`, `min`, `h` or `d`. For example, `30min` or `50s`.
Changes Take Effect: Immediately

Defines reporting frequency for the Message Server reporter..

reporter.console.enabled

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Enables or disables metrics reporting to the **stdout** console.

reporter.console.logFrequency

Default Value: `30min`
Valid Values: A positive integer and time unit such as `ms`, `s`, `min`, `h` or `d`. For example, `30min` or `50s`.
Changes Take Effect: Immediately

Defines the reporting frequency for logging to the **stdout** console.

HeapMemoryUsage.threshold

Default Value: `0.8`
Valid Values: A decimal fraction between 0 and 1
Changes Take Effect: Immediately

Defines the heap memory usage threshold value. This is the ratio of used heap memory to the maximum heap memory.

GcFrequency.threshold

Default Value: 24
Valid Values: A positive numeric value
Changes Take Effect: Immediately

Defines how many times garbage collection can occur within a given hour.

GcLatency.threshold

Default Value: `1000`
Valid Values: The number of milliseconds
Changes Take Effect: Immediately

Defines the garbage collection latency threshold value, in milliseconds, in relation to the last time the garbage was collected within the configured time interval.

<metricName>.threshold

Default Value:

Valid Values: Non-negative number
Changes Take Effect: Immediately

Defines the threshold value for a particular metric.

<metricName>.slidingWindowSize

Default Value: `10`
Valid Values: Any positive integer
Changes Take Effect: After server restart

Defines the sliding window size for a metric, the number of last measurements applied for a particular metric calculation.

ServerResponseTime.slidingWindowSize

Default Value: `1000`
Valid Values: Any positive integer
Changes Take Effect: After server restart

Defines the sliding window size for the ServerResponseTime metric, the number of last measurements applied to metric calculation.

ServerResponseTime.threshold

Default Value: `100`
Valid Values: Number in milliseconds
Changes Take Effect: Immediately

Defines, in milliseconds, the maximum value allowed for the `ServerResponseTime` metric. The metric is calculated as the average time for the latest N routings of data from customer to agent where N is defined by the `ServerResponseTime.slidingWindowSize` value.

SlaveRenderLatency.threshold

Default Value: `10000`
Valid Values: Number in milliseconds
Changes Take Effect: Immediately

Defines, in milliseconds, the **SlaveRenederLatency** metric's threshold value for the configured time interval. Agent side rendering latency shows if the reported agent side rendering is too slow.

JettyThreadPoolUsage.threshold

Default Value: `0.9`
Valid Values: Number between `0` and `1`

Changes Take Effect: Immediately

Defines the Jetty thread pool usage threshold value which is the ratio of the used Jetty thread pool size to the maximum available. This value helps you determine if too few free threads are allowed to handle http requests.

InactiveSessions.threshold

Default Value: `0.2`
Valid Values: Number between `0` and `1`
Changes Take Effect: Immediately

Defines the ratio of inactive sessions to all sessions in the configured time interval. Shows how many Co-browse sessions created by customer side but never joined by an agent.

security Section

provider

Default value: none
Valid Values:

- `MSCAPI` - MS-CAPI keystore
- `JKS` - Java keystore
- `PEM` - PEM keystore

Changes Take Effect: After restart
Specifies the type of trusted storage. If empty, TLS support is disabled for connections between Co-browse Server and other Genesys servers.

> **Tip**
>
> The `provider` option was named `trusted-ca-type` in Co-browse 8.1.3 releases.

trusted-ca

Default value: none
Valid Values: Valid file name
Changes Take Effect: After restart
File name for JKS trusted storage type or trusted CA in PEM format.

truststore-password

Default value: none
Valid Values: Any sequence of characters
Changes Take Effect: After restart
Password for the JKS trusted storage.

> **Tip**
>
> The `truststore-password` option was named `trusted-ca-pwd` in Co-browse 8.1.3 releases.

session Section

domRestrictionsURL

Default Value: None
Valid Values:

- HTTP-based URLs, such as `http://localhost/cobrowse/my-dom-restrictions.xml`
- File-based based URLs, such as `file:C:/my-dom-restrictions.xml`

Changes Take Effect: For new Co-browse sessions

Specifies the URL to the DOM restrictions XML file. If nothing is specified, the default DOM restrictions policy is applied, which prevents agents from clicking submit buttons. For more information, see DOM Restrictions in the Developer's Guide.

Important

Do not use the /static folder for storing the DOM restrictions XML file.

Important

Data masking is enabled in the system for all password inputs and cannot be changed by the DOM restrictions XML file.

Important

HTTP hosting of a DOM restrictions XML resource requires additional considerations:

1. The XML web resource should return a `Last-Modified` header.
2. If the XML web resource is hosted on a web server that is only accessible through a proxy, the proxy settings (http://docs.oracle.com/javase/7/docs/api/java/net/doc-files/net-properties.html) must be set using JVM system properties in `setenv.bat/sh`. For example:

```
set JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyHost=<host> -Dhttp.proxyPort=<port>
```

inactivityDuration

Default Value: `600`

Valid Values: Any positive integer
Changes Take Effect: For new Co-browse sessions

Specifies, in seconds, the period of inactivity during a Co-browse session before the agent joins. Once the agent joins, the session becomes active. If a session does not become active within this period, it is automatically deactivated.

writeModeAllowed

Default Value: `true`
Valid Values:`true` or `false`
Changes Take Effect: For new Co-browse sessions

Specifies if agents are allowed to send customers a request to enter Write Mode. If `false`, the UI button to request Write Mode will not be available to agents.

slave Section

localization

Default value:
Valid Values: String containing a valid URL
Changes Take Effect: Immediately

URL used to load external localization file. This file should be a JSON file hosted on a server with JSONP support (For example, the Co-browse server. See Serving JSONP). By default, the built-in English localization is used. For more information about localization, see Localization—Localizing the agent UI.

cssPatchUrl

Default value:
Valid Values: String containing a valid URL
Changes Take Effect: Immediately

URL used to load an external CSS file that is applied to the agent side representation of the page seen by the user. May be used to solve CSS synchronization issues.

theme

Default value: `wde`
Valid Values:

- `iws`—theme matching the look and feel of Interaction Workspace 8.1.
- `wde`—theme matching the look and feel of Workspace Desktop Edition.
- `wde-hc`—theme matching the **High Contrast** theme in Workspace Desktop Edition.

Changes Take Effect: Immediately

Name of theme applied to the agent UI.

disableWebSockets

Default value: `false`
Valid Values:

- `true`—disable WebSockets
- `false`—do not disable WebSockets

Changes Take Effect: Immediately

This option will disable WebSocket communication.

> **Important**
>
> Use of this option in production is **not** recommended as it may have a significant impact on performance. See Public JavaScript API#disableWebSockets for the analogous option for the customer side and more details.

externalJS

Default value:
Valid Values: String containing a valid URL

Changes Take Effect: Immediately (after agent page reloads)

This option specifies the URL of an additional JavaScript file that will be loaded and executed on the agent side.

wweOrigins

Default value:
Valid Values: A comma-separated list of origins. For example, `http://my-web-server-1,http://my-web-server-2`.
Changes Take Effect: Immediately (after agent page reloads)

Available since Co-browse Server **8.5.003.04**.

Configures the list of Workspace Web Edition origins used by agents. This option enables communication with Workspace Web Edition so an agent does not automatically become **inactive** when using the Co-browse iframe.

An origin consists of a protocol and domain. Optionally, you may include the port, username, and password in an origin. For example, if agents open Workspace Web Edition from `https://htcc.genhtcc.com/ui/ad/v1/index.html`, then you should set this option to `https://htcc.genhtcc.com`.

allowedThirdPartyDomains

Default Value: Empty
Valid Values: Empty, *, or a comma-separated list of origins. Example value: `https://site.com,http://test.site.org:8080`
Changes Take Effect: For new Co-browse sessions

Use this option to enable iframes from specific third-party domains for agents. By default, all third-party iframes in a website are disabled for agents. For example, if your website contains an iframe pointing to `https://third-party-site.com/a-page.html`, the iframe does not load for agents unless you list `https://third-party-site.com` in this option. You can also leave this option empty or set it to *:

- Empty—disables all third-party domains for agents.
- *—allows all third-party domains. Note that even if all third-party domains are allowed, JavaScript execution is always disabled in third-party iframes for the agent's browser.

When configuring third-party domains, you must list each subdomain seperately. For example, `https://third-party-site.com` does not include `https://subdomain.third-party-site.com`. You must list both to enable them.

password

Default Value: None
Valid Values: String with 16 characters
Changes Take Effect: Immediately for co-browse sessions started after change

Added in: Co-browse Server 8.5.102.02

Specifies the token used to authenticate communication between Co-browse Server and Workspace Desktop Edition.

static-web-resources Section

browserHardCacheDuration

Default value: `0`
Valid Values: Positive integer or zero
Changes take effect: Immediately
Specifies the duration in seconds of hard caching for Co-browse static resources like JavaScript and CSS files. If the value is `0`, hard caching is not applied (the headers are not set).

Note that setting a hard cache duration above 0 may lead to problems when upgrading Co-browse. If a browser uses a cached version of JavaScript not compatible with the upgraded server, Co-browse may not function properly until the cache duration expires.

cobrowse Section

> **Important**
>
> The options below provide configuration for the Co-browse Plug-in for Workspace Desktop Edition. To use these options, you must add them to the `cobrowse` section of your Interaction Workspace application in Genesys Administrator.

url

Default Value: None
Valid Values: Valid HTTP(S) URL
Changes Take Effect: After Interaction Workspace restart

Specifies the HTTP(S) URL (such as http://[host]:[port]/cobrowse) of the Co-browse cluster. Typically, this value is the URL for the load balancer. Since Co-browse 8.5.0, you can set the url value to an HTTP or HTTPS based URL.

> **Tip**
>
> The `secureURL` and `useSecureConnection` options have been discontinued starting with Co-browse 8.5.0 because HTTPS URLs can now be configured from the `url` option.

disableCertificateValidation

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After Interaction Workspace restart

Disables certificate validation for the connection between Interaction Workspace and the Co-browse Server.

useBrowserLogging

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After Interaction Workspace restart

Enables the embedded Internet Explorer (agent web UI) logs to redirect into the Interaction Workspace log.

agentSessionsLimit

Default Value: 1
Valid Values: Positive integer or `0`
Changes Take Effect: Immediately for co-browse sessions started after change
Added In: Co-browse Plug-in for Workspace Desktop Edition **8.5.003.04**.

If set to greater than 1, disables one-session agent limitations and configures the number of simultaneous co-browsing sessions an agent can participate in. A value of `0` sets an unlimited amount of simultaneous co-browse sessions for agents.

extendedAttachedData

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately for co-browse sessions started after change
Added In: Co-browse Plug-in for Workspace Desktop Edition **8.5.102.01**.

Setting the value to `true` enables extended attached data for the voice or chat interaction.

password

Default Value: None
Valid Values: String with 16 characters
Changes Take Effect: Immediately for co-browse sessions started after change
Added in: Co-browse Plug-in for Workspace Desktop Edition **8.5.102.01**

Specifies the token used to authenticate communication between Co-browse Server and Workspace Desktop Edition.

Testing and Troubleshooting the Co-browse Solution

Use the following procedures to test that a Genesys Co-browse solution is configured correctly.

Testing Co-browse Without Chat

Required Components

The following components are the minimum required to test Co-browse without chat:

- Local Control Agent
- Configuration Server
- Solution Control Server
- Genesys Administrator
- Co-browse Server

See compatible versions in Related Components.

Preparing for Testing

Prerequisites

- Genesys Framework is running.
- Co-browse Server is installed and configured to work without chat. See Installation Procedures.
- To allow Co-browse Server to work without chat, set the Co-browse Server option useChat in the chat section to `false`.

Start of Procedure

1. Start the required servers.
2. After each server starts, check its trace log for errors.

End of Procedure

Testing the Co-browse Solution Without Proxy

Instrument your website with the Co-browse JavaScript snippet. See Basic Instrumentation.

Testing the Co-browse Solution With Proxy

To learn how to use the proxies included in the Co-browse installation package, see Test with the Co-browse Proxy.

Testing Co-browse Instrumentation

Prerequisites

- The Co-browse JavaScript snippet is on your website. See Basic Instrumentation.

Start of Procedure

1. Open an instrumented page in a supported browser — this page is referred to as the Customer page.

End of Procedure

Successful result: The `Live Chat` and `Co-browsing` buttons are present on the page.

Problem: The buttons are absent.

Possible causes of the problem

1. The page is incorrectly instrumented. To verify, open the page source and confirm the instrumentation script is present and correct. For details, see Website Instrumentation. If you use the proxy for testing Co-browse, it might be a proxy problem. Refer to Troubleshooting the "proxy instrumentation problem" for details and a workaround.
2. Co-browse Server is not running or not working properly. Check the Co-browse Server `trace` log.
3. Localization settings for the Customer page are incorrectly specified in the instrumentation script. For details about localization settings, see Localization.
4. The network has a problem.

> **Important**
>
> To further investigate a problem, enable the Customer browser console log in your instrumentation script. For details, see Enabling console logs.

Testing Co-browse Session

Opening the Agent Page

Prerequisites: You have successfully completed Preparing for Testing.

Start of Procedure

1. Open the Agent page using `http://<Co-browse_Server_host>:<port>/cobrowse/slave.html`.
   For example: `http://localhost:8700/cobrowse/slave.html`

End of procedure

Successful result: The Agent page opens and has an edit box for the Session ID.

Problem: The edit box does not appear.

Possible causes of problem:

1. The URL is incorrect.
2. The localization settings for the Agent side are incorrectly specified in the `slave` section of the Co-browse Server application. For details about localization settings, see Localizing the agent UI.
3. The network has a problem.
4. If it is not clear what the problem is, enable debug logging on the Agent browser, open the Developer Console, and reload the page. You will see debug logs in the Developer Console.

> **Important**
>
> To enable the debug log in the Agent browser, insert debug=1 into the Agent URL and reload the page. For details, see Enabling Console Logs.

Next Step

- Starting a Co-browse Session

Starting a Co-browse Session

Prerequisites: You have successfully completed Opening the Agent Page.

Start of procedure

1. In the Customer page, click `Co-browsing`.

End of procedure

Successful result: The Co-browse session starts and the Session ID appears on the page.

Problem: The Co-browse session does not start.

Possible causes of problem:

1. It could be an intranet problem if the Customer page is viewed on Internet Explorer (IE) 10 or IE 11. For details, see Troubleshooting the intranet problem in IE10 and IE11
2. Co-browse Server is not responding or not working. Check the Co-browse Server debug log.
3. The network has a problem.

Next Step

- Joining the Agent to the Co-browse Session

Joining the Agent to the Co-browse Session

Prerequisites: There are no problems when Opening the Agent Page and Starting a Co-browse Session.

Start of procedure

1. Copy the Session ID from the Customer page and paste it into the edit box on the Agent page.
2. Join the session.

End of procedure

Successful result: The Agent user successfully joins the session.

Problem: The Agent user cannot join the session.

Possible causes of problem:

1. Co-browse Server is not responding or not working. Check the Co-browse Server debug log.
2. The network has a problem.

Next Step

- Testing Co-browse With Chat

Testing Co-browse With Chat

Required Components

The following components are the minimum required, in addition to the components listed above, to test Co-browse with chat:

- Stat Server
- Universal Routing Server
- Interaction Server
- Contact Server
- Chat Server
- Workspace Desktop Edition (WDE)
- Co-browse Plug-in for WDE
- Chat strategy activated on necessary queue

See compatible versions in Related Components.

You must also have at least one agent that can log in and go ready for Chat using Interaction Workspace or Workspace Desktop Edition.

Preparing for Testing

Prerequisites

- Genesys Framework is running.
- Both Universal Contact Server and Interaction Server have connections to Stat Server.
- Both Stat Server, Universal Routing Server, and Chat Server have connections to Interaction Server.
- Interaction Server has a connection to Chat Server through the ESP port.
- Co-browse Server is installed and fully configured for working with Chat. See Installation Procedures.
- Interaction Workspace is configured to work with Co-browse Server. See Installing the Plug-in for Interaction Workspace.
- You have installed Internet Explorer 9 or above.

Start of Procedure

1. Start the required servers.
2. After each server starts, check its trace log for errors.
3. In Interaction Workspace, Log in as an agent and go ready for Chat.

End of Procedure

Testing Chat with Co-browse

Initiating a Chat Session

Prerequisites: You have successfully completed all previous test procedures.

Start of Procedure

1. Confirm that an agent is logged in and ready for Chat in Interaction Workspace.
2. On the Customer page, click `Live Chat`.

End of Procedure

Successful result: The Chat Widget opens and the agent receives the Chat interaction.

Problem: The agent does not receive the Chat interaction.

Possible Causes of the Problem:

1. The Chat inbound strategy is not activated or activated on an improper Interaction Server or Universal Routing Server.

2. Stat Server is not working properly. See the Stat Server debug log.
3. Universal Routing Server is not working properly. Universal Routing Server may not be processing your Chat inbound strategy or your Chat inbound strategy is not working as expected. See the Universal Routing Server debug log.
4. The necessary eServices components do not work properly. See the Chat Server, Interaction Server, and Universal Contact Server debug logs.
5. The agent cannot initiate a Chat session. Verify that the proper Capacity Rule is assigned to the agent.
6. Interaction Workspace cannot handle Chat interactions. See the Interaction Workspace debug log.
7. The network has a problem.

Next Step

- Starting the Co-browse Session

Starting the Co-browse Session

Prerequisites: You have successfully completed Initiating a Chat Session.

Start of Procedure

1. As an agent, accept the Chat interaction.
2. Open the `CO-BROWSE` tab of the Chat interaction.
3. From the Customer page, click `Co-browsing`.

End of Procedure

Successful Results:

- The Co-browse session starts.
- The Session ID appears on the Customer page.
- The Co-browse page automatically opens in the `CO-BROWSE` tab for the agent.

Problem 1: The Co-browse session does not start. The Session ID does not appear on the Customer page.

Possible causes of problem 1:

1. It could be an intranet problem if the Customer page is viewed in IE10 or IE11. For details, see Troubleshooting the intranet problem in IE10 and IE11.
2. Co-browse Server is not responding. See the Co-browse server debug log.
3. The network has a problem.

Problem 2: The Co-browse session starts but the Co-browse page does not automatically open for the agent.

Possible causes of problem 2:

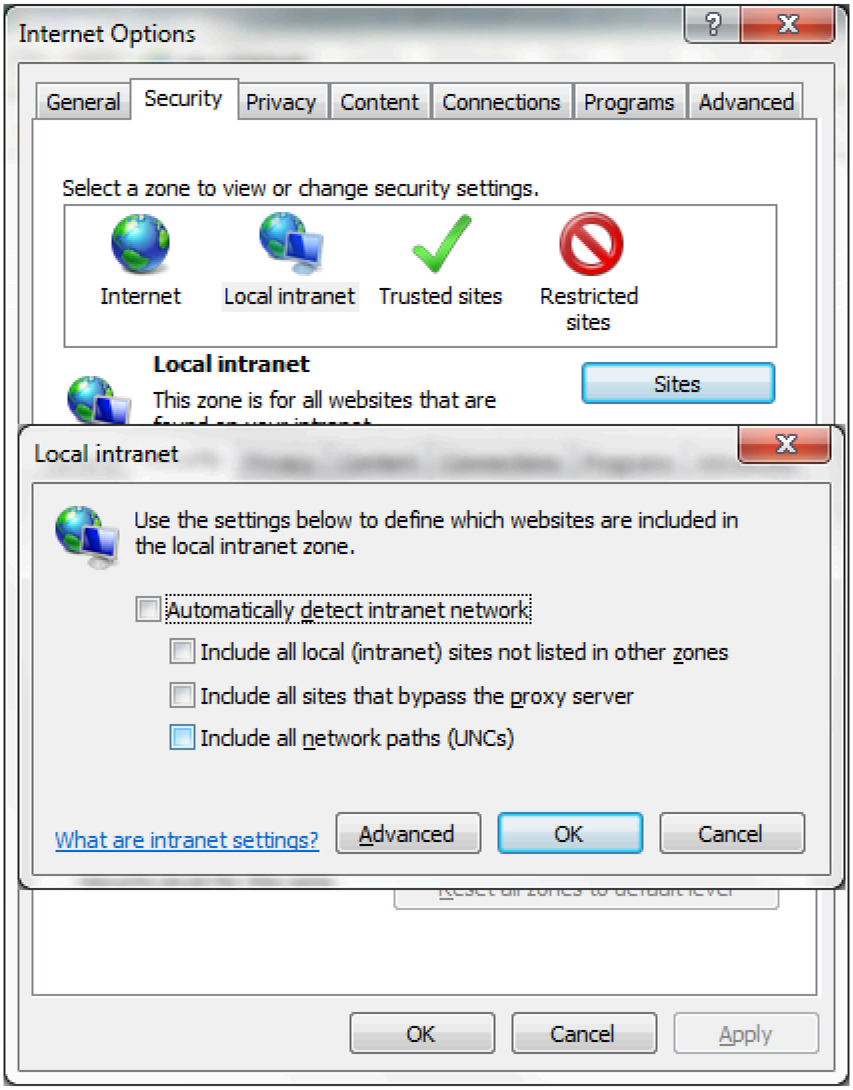
1. Interaction Workspace is incorrectly configured to work with Co-browse Server. You must configure Interaction Workspace's relationship with Co-browse Server:
   - For a standalone Co-browse Server, verify that Interaction Workspace's Connection List contains the Co-browser Server application. See Installing the Plug-in for Interactive Workspace.
   - For a cluster of Co-browse Servers, verify the options in the `cobrowse` section of the Interaction Workspace Application. See Configure the Co-browse Plug-in for Interaction Workspace for details. Also, see the Interaction Workspace debug log.
2. Co-browse Server is not responding. See the Co-browse Server debug log.
3. The network has a problem.
4. If it is not clear what the problem is, enable browser debug logging on the Interaction Workspace log, end the chat, and try to start a new chat with co-browsing again. In this case, the browser logs will be stored in the Interaction Workspace log, starting with the word `BROWSER`. Open the log and try to investigate the problem.

> **Important**
>
> To enable the debug log in the Agent browser, set the `useBrowserLogging` option in the `cobrowse` section of the Interaction Workspace application to `true`.

Troubleshooting the Intranet Problem in IE10 and IE11

IE10 and IE11 do not allow WebSockets on local domains. Internet Explorer uses a built-in algorithm to determine if the domain is local and falls under IE's `Local intranet` security zone rules. This algorithm is affected by several factors, one of which is the browser's proxy settings. If your domain is listed as "excluded" from proxying, then IE treats your domain as local and does not allow WebSockets to be opened.

To overcome this, you can disable IE's intranet network settings. Go to `Tools > Internet Options > Security > Local intranet > Sites` and deselect each checkbox:



Select a zone to view or change security settings.

Local intranet

This zone is for all websites that are found on your intranet.

Use the settings below to define which websites are included in the local intranet zone.

□ Automatically detect intranet network

□ Include all local (intranet) sites not listed in other zones

□ Include all sites that bypass the proxy server

□ Include all network paths (UNCs)

Disable the intranet network settings.

If you reach Co-browse Server by an internal IP address (such as 192.168.XX.XX), you can overcome the problem by adding a "fake" domain in the `hosts` file. For example:

```
192.0.2.10    cobrowse.com
```

Next, modify your website instrumentation to use the "fake" domain (in this case, `cobrowse.com`) for the URL of your Co-browse application.

```
<script>(function(d, s, id, o) {
  var fs = d.getElementsByTagName(s)[0], e;
  if (d.getElementById(id)) return;
  e = d.createElement(s); e.id = id; e.src = o.src;
  e.setAttribute('data-gcb-url', o.cbUrl);
  fs.parentNode.insertBefore(e, fs);
})(document, 'script', 'genesys-js', {
  src: "http://192.0.2.10:8700/cobrowse/js/gcb.min.js",
  cbUrl: "http://cobrowse.com:8700/cobrowse"
});</script>
```

> **Important**
>
> This problem is unlikely to happen in production environments because Co-browse Server is in Internet Explorer's `Internet` zone for end users. It may occur when testing the Co-browse solution if Co-browse is deployed in a local network and used exclusively within a company.

Troubleshooting the "proxy instrumentation problem"

Prerequisites

- Co-browse instrumentation is done via the Co-browse proxy (most likely to happen in a development or or demo environments - it is impossible in production).

Symptoms

- The website's JavaScript fails completely or partially. This might lead to different problems — the most likely is that some areas of the site are unresponsive or do not render at all (most likely the dynamic areas, such as tabs, accordions, submenus, and so on).
- Errors in the browser console (or error alerts in older versions of Internet Exporer).

Troubleshooting

1. Open developer tools and examine console logs for errors that happen outside of `gcb.min.js`.
2. Remove instrumentation, reload the page with `Ctrl+F5` (to clean the cache), and see if the same errors are still there.
3. If errors are there with and without Co-browse instrumentation, it is not a proxy issue.
4. If errors are there only with Co-browse instrumentation, it is probably a proxy issue.

Root cause

The Co-browse proxy works by examining all requests made to the website and replacing a certain sequence of characters with Co-browse instrumentation *AND* this sequence of characters. For example, the following:

```
....
</head>
<body>
....
```

becomes:

```
....
<COBROWSE_INSTRUMENTATION>
</head>
<body>
....
```

after the </head> sequence of characters is replaced by the proxy.

However, if any of the site's JavaScript files contains this sequence, it is *ALSO* "instrumented" and will most likely be broken.

Treatment

1. Find the sequence of characters that appears ONLY in the website's HTML code and does not appear in any of its JS files.
2. Modify the proxy's `map.xml` file to use the new character sequence. For example it may be:

   ```
   <map replace="%s </body>" domains=....
   ```

   or

   ```
   <map replace="%s <meta charset" domains=....
   ```

   > ## Important
   > All special characters in the `replace` attribute should be converted to HTML entities.

3. Restart the proxy.

Troubleshooting CSS Synchronization

If some or all of the content of your website is not properly rendered on the agent side, it is most likely a CSS synchronization problem.

> ## Warning
> If your website is using conditional comments for Internet Explorer, also see Limitations on IE Conditional Comments.

First, try different CSS synchronization settings. The possible settings are `server` (default setting), `browser`, or both.

The most reliable CSS synchronization mode is `server` but there may be edge cases where `browser` mode or both modes together produce better results.

If the problem persists, try the following:

1. Server mode works by proxying your site's CSS. Co-browse Server makes an HTTP request to get your site's CSS. Make sure requests from Co-browse Server are not blocked.
2. When using `browser` mode and sometimes with `server` mode, requests for your site's CSS are made from the agent browser. Make sure that requests from the agent browser are not blocked.

3. Server mode sometimes stalls on invalid CSS. When it does, a message appears in the Co-browse server logs.

Example:

```
19:16:34,454 [ WARN] LoggingCSSParseErrorHandler    - [1:30]-[1:43] Encountered text ' {'
corresponding to token <LBRACE>. Skipped until token }. Was expecting one of: <S>, ":"
```

Depending on the kind of error, the parser will do one of the following:

- Skip the stylesheet completely and let the agent browser load the stylesheet as is. CSS `:hover` effects will not be synchronized for this stylesheet.
- Supply a corrupt version of the stylesheet to the agent browser. The agent side user may end up with something visually different than the customer side user.

To avoid CSS synchronization issues, you should **validate your CSS using the freely available** CSS Lint Tool. Use the tool to avoid CSS errors. CSS with warnings from the tool is usable.

In cases where CSS synchronization issues continue, you can manually fix agent representation by providing additional CSS using the `cssPatchUrl` server configuration option.

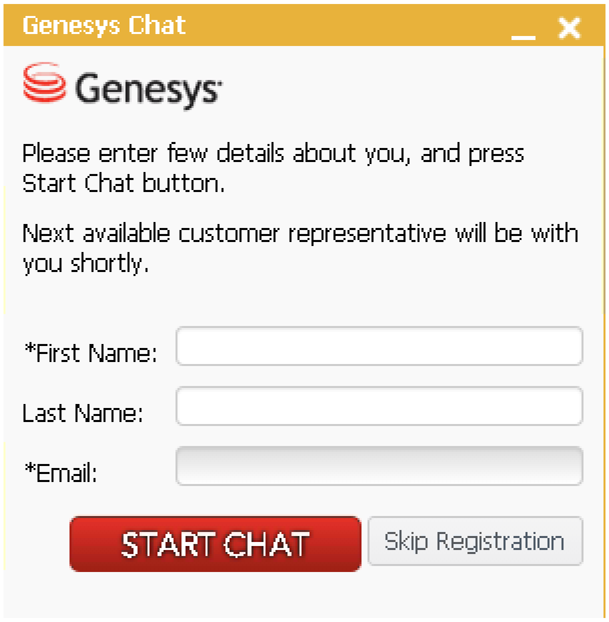
Troubleshooting Chat Widget Rendering Issues

Prerequisites:

- Built in chat widget uses embedded mode.

Symptoms:

Chat widget renders incorrectly or has unexpected styling. For example:



Please enter few details about you, and press Start Chat button.

Next available customer representative will be with you shortly.

In the figure above, the `Start Chat` button and `Email` field do not render as expected.

Possible Causes of the Problem:

In embedded mode, chat HTML and CSS is added directly to your site's page. Your site's CSS will affect Chat Widget styling. Generally, all the chat widget's styling is sandboxed, but some of your site's CSS may leak into the widget and create unwanted styling effects.

Treatment:

Find out which CSS rules create unwanted style effects in the chat widget and create a CSS patch that overrides these rules. You can use your browser's developer tools to find the CSS rules. For more information on CSS customization of the chat widget, see the **CSS-based** tab in the Customizing the User Interface section.

Co-browse Restrictions and Known Limitations

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.

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

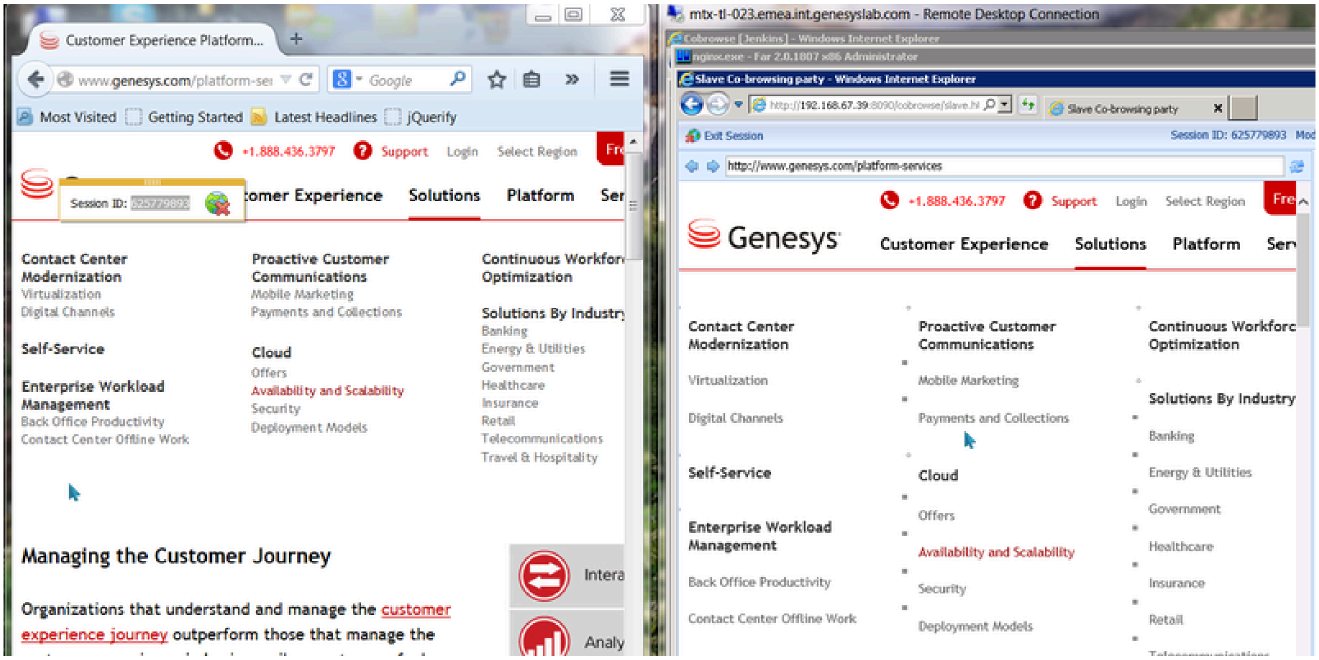
Mouse positions may differ slightly on the customer and agent-side if websites render differently in different browsers.

Representation of Dynamically Shown List Items May be Partially Broken on

Agent Browsers Using IE10 or IE11

Issues may arise when IE10/11 is used as a agent browser on a website with dynamically shown/ hidden sub-menus.

Example:


Customer Experience Platform...
www.genesys.com/platform-ser
Google
Most Visited
Getting Started
Latest Headlines
jQuerify
+1.888.436.3797
Support
Login
Select Region
Session ID: 625779893
tomer Experience
Solutions
Platform
Ser
Contact Center Modernization
Virtualization
Digital Channels
Self-Service
Enterprise Workload Management
Back Office Productivity
Contact Center Offline Work
Proactive Customer Communications
Mobile Marketing
Payments and Collections
Cloud
Offers
Availability and Scalability
Security
Deployment Models
Continuous Workforce Optimization
Solutions By Industry
Banking
Energy & Utilities
Government
Healthcare
Insurance
Retail
Telecommunications
Travel & Hospitality
Managing the Customer Journey
Organizations that understand and manage the customer experience journey outperform those that manage the
Intera
Analy
mtx-tl-023.emea.int.genesyslab.com - Remote Desktop Connection
Cobrowse [Jenkins] - Windows Internet Explorer
nginx.exe - Far 2.0.1807 x86 Administrator
Slave Co-browsing party - Windows Internet Explorer
http://192.168.67.39:8090/cobrowse/slave.ht
Slave Co-browsing party
Exit Session
Session ID: 625779893
http://www.genesys.com/platform-services
+1.888.436.3797
Support
Login
Select Region
Genesys
Customer Experience
Solutions
Platform
Ser
Contact Center Modernization
Virtualization
Digital Channels
Self-Service
Enterprise Workload Management
Back Office Productivity
Contact Center Offline Work
Proactive Customer Communications
Mobile Marketing
Payments and Collections
Cloud
Offers
Availability and Scalability
Security
Deployment Models
Continuous Workforce Optimization
Solutions By Industry
Banking
Energy & Utilities
Government
Healthcare
Insurance
Retail

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 somwhere that is accessible via HTTP
3. Specify the URL of the file in the slave.cssPatchUrl option in Config Server.

IE Conditional Comments

IE Conditional comments are used to create CSS targeting specific versions of IE or any version of IE. This technique works on IE versions 9 and below. We recommend that you avoid IE conditional comments. This technique is deprecated and support has been dropped by Microsoft since IE10.

Example:

```
<!--[if IE]>
<link href="ie.css" rel="stylesheet" />
<![endif]-->
```

```
<!--[if !IE]> -->
<link href="non-ie.css" rel="stylesheet" />
<!-- <![endif]-->
```

If your website is using this technique, we strongly advise that the agent's machine uses IE10 or above for the Co-browse agent side. In this case, CSS synchronization issues are possible if the site visitor (customer user) uses IE9.

Co-browse Fails Silently on Internet Explorer 8

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

DOM Restrictions

DOM Restrictions do not support images.

Static resources behind authentication are not supported

In the current architecture, agents receive static resources (such as CSS or images) either directly from the co-browsed website or through the Co-browse Server acting as a proxy (see CSS Synchronization). If such a resource is available only for authorized users, the agent or Co-browse Server will not be able to fetch it, and it will not be displayed to the agent. This may result in an image not shown to the agent, or in the co-browsing being almost dysfunctional, depending on how heavily your website relies on authentication for static content.

Support for such content will be addressed in a future release.

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.

Security

Genesys Co-browse is part of a solution deployment, and security should be considered at the solution level. For example, Genesys Co-browse takes measures to make sure hidden attacks in DOM do not make it to agent desktops. Meanwhile, you must consider other areas, like only exposing Genesys Co-browse on HTTPs ports, hardening intermediate proxies so as to suppress or add certain HTTP headers, and so on. The Open Web Application Security Project provides excellent guidelines to help.

Genesys Co-browse supports the following ways to protect data over the web:

- **Encryption of co-browsing data**—Co-browsing related data passed between the user, the Co-browse Server and the agent is encrypted through the HTTPS connection:
  - **Configure Security Certificate**—the Jetty web server supplied with the Co-browse solution includes a pre-configured, self-signed certificate. This allows you to use HTTPS out of the box in a lab or demo environment. For a production environment, you should use a certificate issued by a third-party Certificate Authority.
    Related documentation: Load SSL certificates and configure Jetty.
  - **Configuring Cipher Suites**—To configure specific cipher suites to include or exclude, see the Disabling/Enabling Specific Cipher Suites section of the Jetty TLS documentation.
  - **HTTPS connection for Jetty**—A Co-browse Server application defined in Configuration Server can have both HTTP and HTTPS connections for Jetty supplied with Co-browse. Related documentation:
    - *Add the secure port* section in Creating the Co-browse Server Application Object in Genesys Administrator
    - *Edit the connection and set the ID to the HTTPS listening port* in the Configuring Connection to Co-browse Server/Node Application for Workspace Desktop Edition
  - **HTTPS connection for Co-browse cluster**—A Co-browse Server application supports both HTTP and HTTPS connections for Co-browse cluster. Related documentation:
    - `url` option in the cluster section of the Co-browse Server application configuration.
    - `url` option in the cobrowse section of the Workspace Desktop Edition application configuration.
  - **HTTPS website instrumentation**—to work with Co-browse, the web page must include the Co-browse JavaScript code that provides the access to Co-browse resources. Co-browse resources can be loaded through HTTPS.
    Related documentation: Website Instrumentation.

    > **Warning**
    >
    > For Co-browse cluster to work correctly, specify HTTPS access to the Co-browse resources through the Load Balancer. In case there is a single Co-browse Server node, the instrumentation snippet should include HTTPS access to single node resources.

- **Access the internet through a forward proxy**—If HTTP connections must go through an internal proxy server (for example, DMZ or local intranet), you must configure forward proxy options to let the Co-browse server obtain public web resources.

Related documentation: See the forward-proxy section in the Co-browse Server application configuration.

- **Role-based control (RBAC) for Workspace Desktop Edition**—starting in version 8.5.001.09, the Co-browse WDE plug-in supports the `Agent—Can Monitor Co-browse` privilege. This privilege allows the agents to work with Co-browse sessions.
  Related documentation: Configuring Role-Based Access Control for Co-browse.
- **DOM restrictions**—Genesys Co-browse allows you to hide sensitive data and restrict web elements control from agents in a Co-browse session.
  Related documentation: Configure DOM Restrictions
- **CORS control**—Co-browse Server supports CORS control for websites. You may specify the list of origins allowed to access the Co-browse Server.
  Related documentation: cross-origin section in the Co-browse Server application configuration.
- **Transport Layer Security (TLS)**—all connections to the Genesys servers can be secured. TLS is supported above Java containers and Jetty. The user data submitted from the browser tier is always sent through secure connections.
  Related documentation: Configuring TLS
- **Security with External Cassandra**—Starting from 8.5.1, Genesys Co-browse supports secure access interfaces through authentication and authorization and secure network traffic through TLS.
  Related documentation: Cassandra Security
- **Static resources proxying**—Co-browse server proxies some static assets of your website like CSS, images, and fonts. While this is generally safe since your website is the only source of these assets in a Co-browse session, you may enforce the security using the following configuration options:
  - The `allowedExternalDomains` option in the http-proxy section allows you to list all the domains resources of which are allowed to be proxied through Co-browse server. Use this to prevent unauthorized parties from abusing the Co-browse server proxy.
  - The `disableCaching` option in the http-security section. Sometimes caching of resources loaded via HTTPS is considered not fully secure. While this is not so in 99% of cases because only static assets such as images or CSS are cached, you can force all caching to be disabled using this option.

> ## Important
>
> Genesys performs security testing with the OWASP Zed Attack Proxy (ZAProxy) to protect the Genesys Co-browse solution against known OWASP vulnerabilities. For details, see Security Testing with ZAProxy.

Configuring Security Certificates for Jetty

Loading Certificate for SSL

The Jetty web server supplied with the Co-browse solution includes a pre-configured, self-signed certificate. This allows you to use HTTPS out of the box in a lab or demo environment, with the restrictions described in Basic Instrumentation.

For a production environment, you should use a certificate issued by a third-party Certificate Authority. The procedures on this page provide examples of ways to load SSL certificates and configure Jetty. These examples may vary depending on your environment.

Load an SSL Certificate and Private Key into a JSSE keystore

> **Important**
>
> In a development environment, you can use self-signed certificates, but in a production environment you should use a certificate issued by a third-party Certificate Authority, such as VeriSign.

Prerequisites

- An SSL certificate, either generated by you or issued by a third-party Certificate Authority. For more information on generating a certificate, see http://wiki.eclipse.org/Jetty/Howto/Configure_SSL.

Start of procedure

1. Depending on your certificate format, do **one** of the following:
   - If your certificate is in PEM form, you can load it to a JSSE keystore with the keytool using the following command:
     ```
     keytool -keystore <keystore> -importcert -alias <alias> -file <certificate_file>
      -trustcacerts
     ```
     Where:

     `<keystore>` is the name of your JSSE keystore.

     `<alias>` is the unique alias for your certificate in the JSSE keystore.

     `<certificate_file>` is the name of your certificate file. For example, `jetty.crt`.
   - If your certificate and key are in separate files, you must combine them into a PKCS12 file before loading it to a keystore.

1. Use the following command in openssl to combine the files:
   `openssl pkcs12 -inkey <private_key> -in <certificate> -export -out <pkcs12_file>`

   Where:

   `<private_key>` is the name of your private key file. For example, `jetty.key`.

   `<certificate>` is the name of your certificate file. For example, `jetty.crt`.

   `<pkcs12_file>` is the name of the PKCS12 file that will be created. For example, `jetty.pkcs12`.

2. Load the PKCS12 file into a JSSE keystore using keytool with the following command:
   `keytool -importkeystore -srckeystore <pkcs12_file> -srcstoretype <store_type> -destkeystore <keystore>`

   Where:

   `<pkcs12_file>` is the name of your PKCS12 file. For example, `jetty.pkcs12`.

   `<store_type>` is the file type you are importing into the keystore. In this case, the type is PKCS12.

   `<keystore>` is the name of your JSSE keystore.

> Important
>
> You will need to set two passwords during this process: keystore and truststore. Make note of these passwords because you will need to add them to your Jetty SSL configuration file.

End of procedure

Next Steps

- Configure Jetty

Configure Jetty

Prerequisites

- You have completed Load an SSL Certificate and Private Key into a JSSE keystore

Start of procedure

1. Open the Jetty SSL configuration file in a text editor: `<jetty_installation>/etc/jetty-ssl.xml`.
2. Find the `<New id="sslContextFactory" class="org.eclipse.jetty.http.ssl.SslContextFactory">` element and update the passwords:

```
<New id="sslContextFactory" class="org.eclipse.jetty.http.ssl.SslContextFactory">
    <Set name="KeyStore"><Property name="jetty.home" default="." />/etc/keystore</Set>
    <Set name="KeyStorePassword">OBF:<obfuscated_keystore_password></Set>
```

```
        <Set name="KeyManagerPassword">OBF:<obfuscated_keymanager_password></Set>
        <Set name="TrustStore"><Property name="jetty.home" default="." />/etc/keystore</Set>
        <Set name="TrustStorePassword">OBF:<obfuscated_truststore_password></Set>
    </New>
```

Note: You can run Jetty's password utility to obfuscate your passwords. See http://wiki.eclipse.org/Jetty/Howto/Secure_Passwords.

3. Save your changes.

End of procedure

Choosing a Directory for the Keystore

The keystore file in the example above is given relative to the Jetty home directory. For production, you should keep your keystore in a private directory with restricted access. Even though the keystore has password, the password may be configured into the runtime environment and is vulnerable to theft.

You can now start Jetty the normal way (make sure that jcert.jar, jnet.jar and jsse.jar are on your classpath) and SSL can be used with a URL, such as `https://<your_IP>:8743/`

Configuring TLS

Genesys Co-browse supports the Transport Layer Security (TLS) protocol to secure data exchanged with other Genesys components. For details about TLS, see the Genesys 8.1 Security Deployment Guide. You can configure TLS for Co-browse by completing the procedures on this page.

Configuring TLS for Genesys Servers

To configure the TLS parameters for Genesys servers like Configuration Server, Message Server, or Chat Server, see Configuring TLS Parameters in Configuration Manager.

Configuring TLS for Co-browse Server

To enable TLS support for Co-browse Server, you must:

1. Have properly installed trusted certificates for the Genesys servers.
2. Configure TLS options for the Co-browse Server application.
3. Configure the appropriate connections between the Co-browse server application and the necessary Genesys servers through secure ports.

Configuring Trusted Stores

PEM Trusted Store

PEM stands for "Privacy Enhanced Mail", a 1993 IETF proposal for securing e-mail using public-key cryptography. That proposal defined the PEM file format for certificates as one containing a Base64-encoded X.509 certificate in specific binary representation with additional metadata headers.

PEM certificate trusted store works with CA certificate from an X.509 PEM file. It is a recommended trusted store to work on Linux systems.

Complete the steps below to work with the PEM certificate trusted store:

Start

1. Configure TLS for Genesys servers to use certificates signed by CA certificate **certificateCA.crt**.
2. Place the trusted CA certificate in PEM format on the Co-browse Server application host. To convert a certificate of another format to .pem format you can use the OpenSSL tool. For example:
   - Convert a DER file (.crt .cer .der) to PEM:
     ```
     openssl x509 -inform der -in certificateCA.crt -out certificateCA.pem
     ```
   - Convert a PKCS#12 file (.pfx .p12) containing a private key and certificates to PEM:

```
openssl pkcs12 -in certificateCA.pfx -out certificateCA.pem -nodes
```

You can add `-nocerts` to only output the private key or add `-nokeys` to only output the certificates.

3. In Genesys Administrator, navigate to `Provisioning > Environment > Applications` and open your Co-browse Server application.
4. Click the `Options` tab and navigate to the security section.
5. Set the `provider` option to PEM.
6. Set the `trusted-ca` option to the path and file name for your trusted CA in PEM format on the Co-browse Server application host.
7. Click `Save & Close`.

End

JKS Trusted Store

A Java KeyStore (JKS) is a repository of security certificates used, for instance, in SSL/TLS encryption. The Java Development Kit provides a tool named keytool to manipulate the keystore.

Complete the steps below to work with the JKS certificate trusted store:

Start

1. Configure TLS for Genesys servers to use certificates signed by CA certificate **certificateCA.crt**.
2. Import the CA certificate to an existing Java keystore using keytool:
   - Run the `keytool` command with option `-alias` set to `root`:
     ```
     keytool -import -trustcacerts -alias root -file certificateCa.crt -keystore
     /path/to/keysore/keystore.jks
     ```
   - Enter the keystore password in command line prompt - for example:
     ```
     Enter keystore password: somepassword
     ```
3. In Genesys Administrator, navigate to `Provisioning > Environment > Applications` and open your Co-browse Server application.
4. Click the `Options` tab and navigate to the security section.
5. Set the `provider` option to JKS.
6. Set the `trusted-ca` option to the path and file name for your JKS trusted storage type on the Co-browse Server application host.
7. Set the `truststore-password` option to the password defined for your keystore in Step 2.
8. Click `Save & Close`.

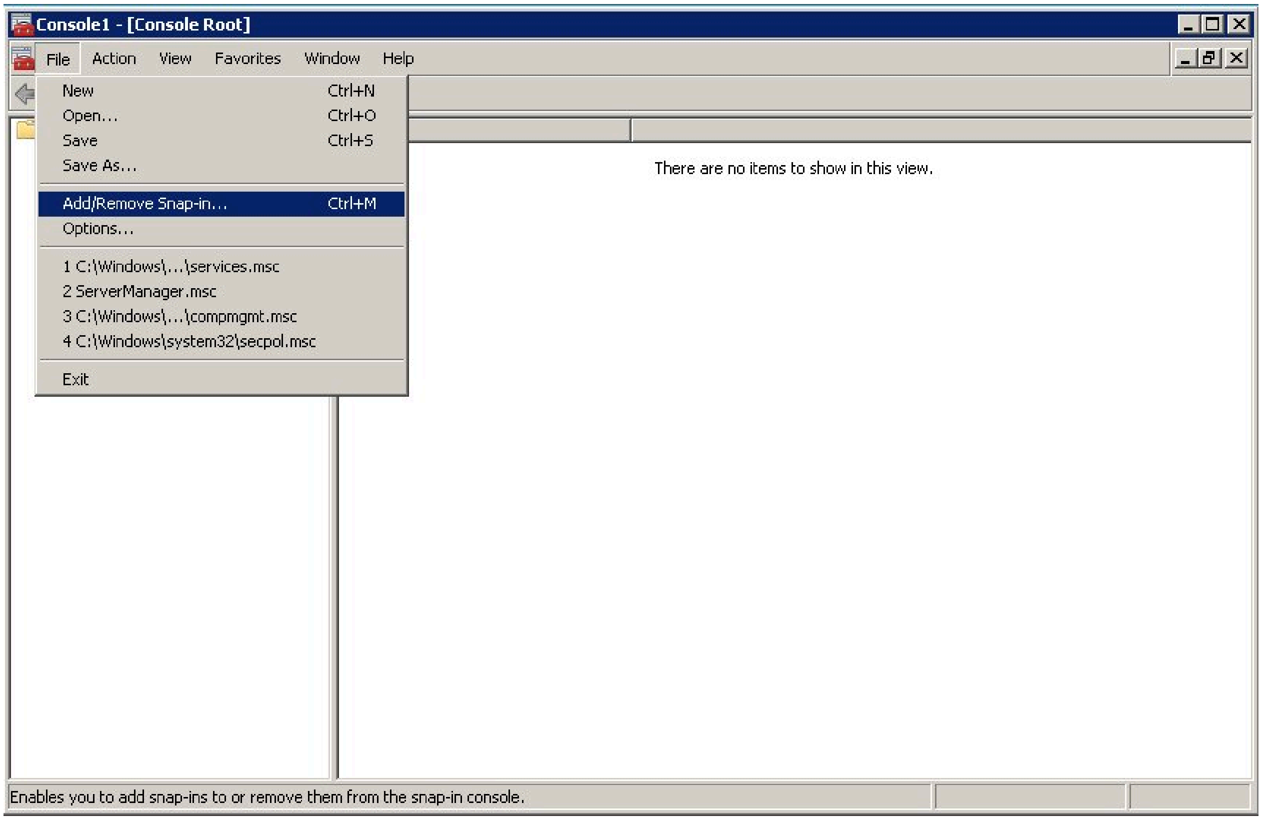
End

MSCAPI Trusted Store

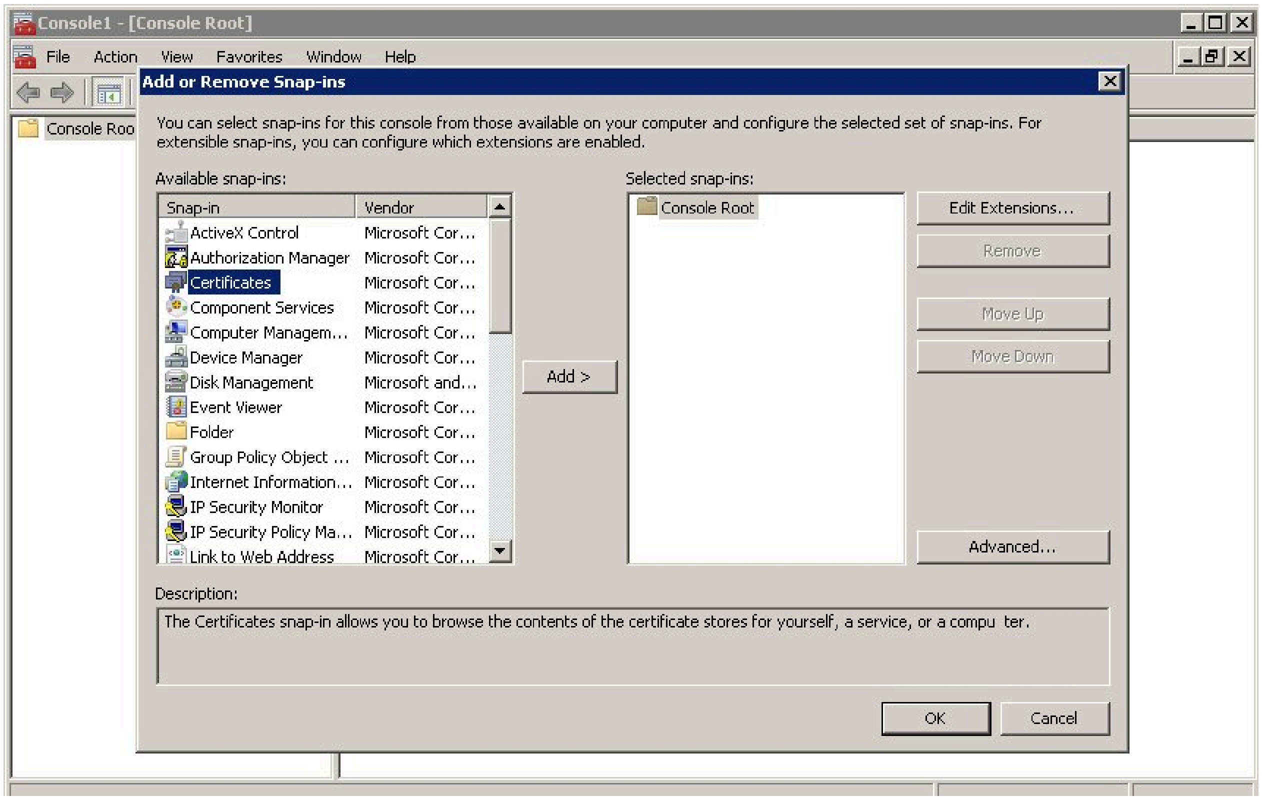
Complete the steps below to work with the MSCAPI certificate trusted store:

Start

1. Configure and tune TLS for Genesys servers to use certificates signed by the same CA.
2. If the Co-browse Server is running on a different host, copy the trusted CA certificate to this host.
3. Import the CA certificate to WCS via Certificates Snap-in on the Co-browse Server host by launching the MMC console. Enter `mmc` at the command line.
4. Select `File > Add/Remove Snap-in...` from the main menu.


Console1 - [Console Root]
File
Action
View
Favorites
Window
Help
New Ctrl+N
Open... Ctrl+O
Save Ctrl+S
Save As...
Add/Remove Snap-in... Ctrl+M
Options...
1 C:\Windows\...\services.msc
2 ServerManager.msc
3 C:\Windows\...\compmgmt.msc
4 C:\Windows\system32\secpol.msc
Exit
There are no items to show in this view.
Enables you to add snap-ins to or remove them from the snap-in console.

5. Select `Certificates` from the list of available snap-ins and click Add.

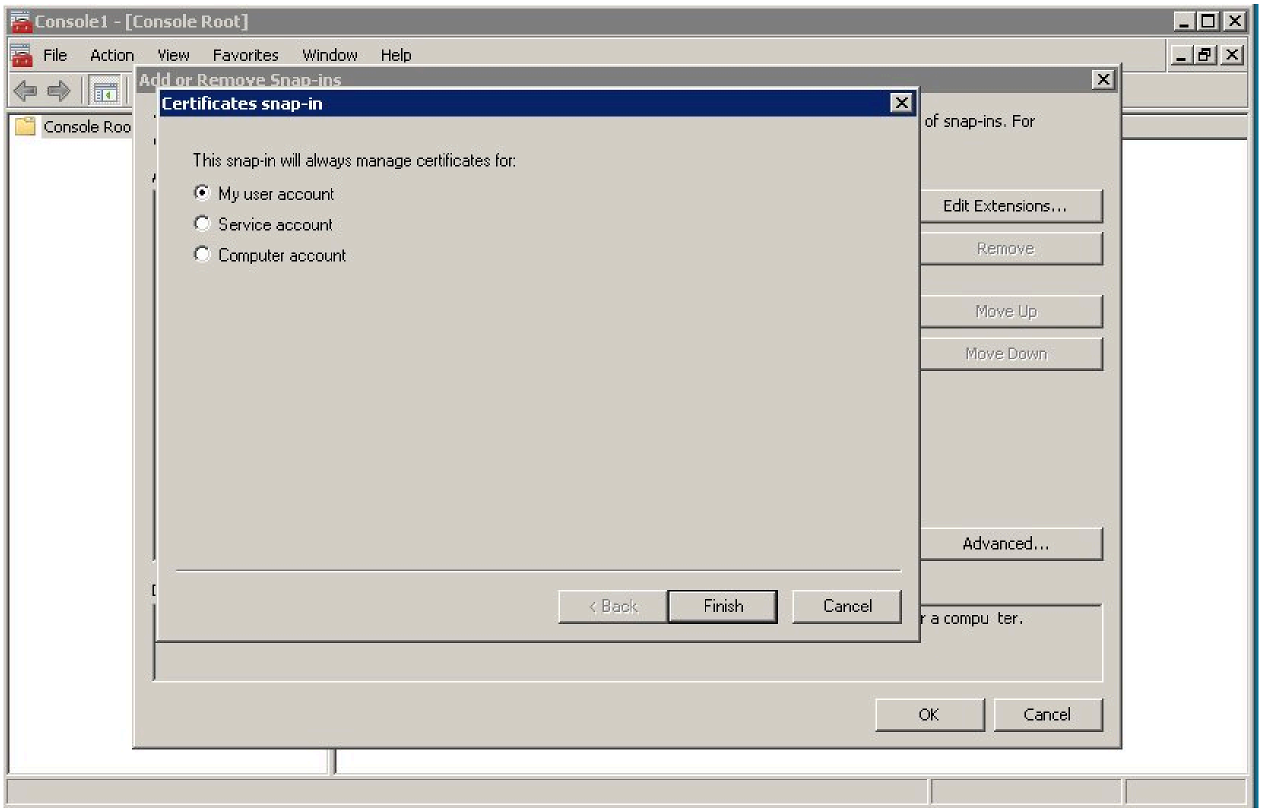


You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
Internet Information...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy Ma...	Microsoft Cor...
Link to Web Address	Microsoft Cor...

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a compu ter.

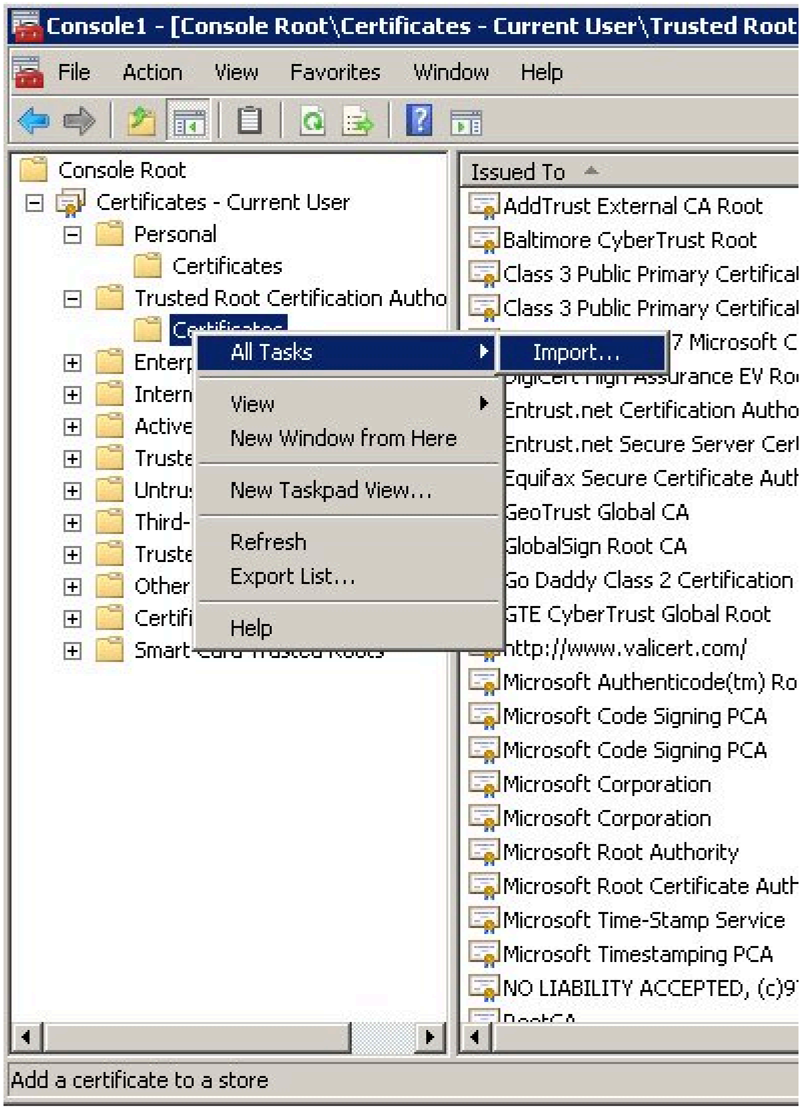
6. Select the account to manage certificates for and click `Finish`. It is important to place certificates under the correct Windows account. Some applications are run as services under the Local Service or System account, while others are run under user accounts. The account chosen in MMC must be the same as the account used by the application which certificates are configured for, otherwise the application will not be able to access this WCS storage.



This snap-in will always manage certificates for:

● My user account
○ Service account
○ Computer account

7. Click OK.
8. Import a certificate. Right-click the "Trusted Root Certification Authorities/Certificates" folder and choose `All Tasks > Import...` from the context menu. Follow the steps presented by the Certificate Import Wizard, and once finished the imported certificate appears in the certificates list.



Issued To
AddTrust External CA Root
Baltimore CyberTrust Root
Class 3 Public Primary Certifical
Class 3 Public Primary Certifical
7 Microsoft C
DigiCert High Assurance EV Roo
Entrust.net Certification Autho
Entrust.net Secure Server Cerl
Equifax Secure Certificate Autl
GeoTrust Global CA
GlobalSign Root CA
Go Daddy Class 2 Certification
GTE CyberTrust Global Root
http://www.valicert.com/
Microsoft Authenticode(tm) Ro
Microsoft Code Signing PCA
Microsoft Code Signing PCA
Microsoft Corporation
Microsoft Corporation
Microsoft Root Authority
Microsoft Root Certificate Auth
Microsoft Time-Stamp Service
Microsoft Timestamping PCA
NO LIABILITY ACCEPTED, (c)9
RootCA

Add a certificate to a store

9. In Genesys Administrator, navigate to `Provisioning > Environment > Applications` and open your Co-browse Server application.
10. Click the `Options` tab and navigate to the security section.
11. Set the `provider` option to `MSCAP`.
12. Click `Save & Close`.

End

Configuring TLS Options

Important

In Co-browse Server 8.5.0, the procedure for configuring TLS changed. You must configure TLS-related options for Configurations Server differently from other Genesys servers such as Message Server, Chat Server, and Solution Server.

- For Configuration Server, configure TLS in the **setenv.bat/setenv.sh** file located in the **server** directory.
- For other Genesys servers, configure TLS in the **security** section of the Co-browse Server (8.5.000), Co-browse Node (8.5.001+), or Co-browse Cluster (8.5.003+) application object.
- If you use Solution Control Server, you cannot configure the security section of the Co-browse Cluster application object and must configure the security section of the Co-browse Node application object.

Genesys Co-browse Server includes the following TLS-related configuration options:

Option	Default Value	Mandatory	Changes Take Effect	Description
provider	none	no	after restart	Type of trusted storage Valid values: MSCAPI, PEM or JKS If empty, TLS support is disabled.
trusted-ca	none	no	after restart	Specifies the name of the trusted store file which holds the public certificate to verify the server. Applicable for PEM and JKS trusted storage types only. Valid values: valid file name (including path)
truststore-password	none	no	after restart	Password for the JKS trusted storage. Valid values: any string

See Configuring Trusted Stores above for details about configuration for a specific type of store (PEM, JKS, MSCAPI).

Configuring TLS Connections

In Co-browse Server 8.5.0, the procedure for configuring TLS connections changed. You must configure TLS-related connections between the Co-browse Server application and the Genesys server in the following way:

- For Configuration Server, configure connection in **setenv.bat/setenv.sh** located in the **server** directory.
- For connections with other Genesys servers, configure **Connections** of the Co-browse Server (8.5.000) or Co-browse Cluster (8.5.001+) application through secure ports.

Public JMX Authorization

By default, JMX is not enabled. If you want to enable JMX, you must protect yourself from the Java deserialization vulnerability and other vulnerabilities. You should secure JMX by using the configuration below or by deploying into a secure zone like a DMZ.

Enabling Remote JMX

To enable remote JMX complete the following:

1. Enabling Remote JMX Configuration
2. Setting Remote JMX Authentication

Enabling Remote JMX Configuration

You can enable remote JMX configuration for Co-browse Server (including embedded Cassandra) or external Cassandra.

Enabling Remote JMX for Co-browse Server

To enable remote JMX for Co-browse Server, open your **setenv.bat/sh** file and uncomment the JMX settings below this line:

```
Uncomment to enable JMX Remote
```

Enabling Remote JMX for External Cassandra

If you use external Cassandra and want to monitor Co-browse column family attributes, open your **cassandra.bat** (Windows) or **cassandra-env.sh** (UNIX) and enable these JMX settings:

In **cassandra.bat**, enable the settings below the line:

```
... JMX REMOTE ACCESS SETTINGS ...
```

In **cassandra-env.sh** add:

```
JMX_PORT="7199"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.port=$JMX_PORT"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.rmi.port=$JMX_PORT"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.ssl=false"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.password.file=/etc/cassandra/
jmxremote.password"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.access.file=/etc/cassandra/
jmxremote.access"
```

Now that you enabled remote JMX, you can set remote authentication.

Setting Remote JMX Authentication

1. In the JMX remote settings you enabled above, set the following:

   ```
   -Dcom.sun.management.jmxremote.authenticate=true
   ```

2. Specify your credentials:

   a. Find the **jmxremote.access** and **jmexremmote.password.template** files in the **<JAVA_HOME>/[jre]/lib/management** directory.

   b. Rename **jmxremote.password.template** to **jmxremote.password**.

   c. By default, the JMX remote settings you enabled in **setenv.bat/sh** use the **jmxremote.access** and **jmxremote.password** files for authentication.

   To enable default roles, uncomment the role/password settings at the bottom of the **jmxremote.password** file. For example:

   ```
   monitorRole  QED
    controlRole  R&D
   ```

   > ### Tip
   > You can edit the role names and passwords but you should first make sure the defaults work.

   d. If you use the same credentials for all host applications, you can use the default password and access files. Otherwise, do the following:

      i. Copy a pair of **jmxremote.access** and **jmxremote.password** files to the path related to each application.

      ii. Add paths to the access and password files in your JMX remote configuration. For example, in Windows:

      ```
      set JMX_PORT=7199
       set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote ^
       -Dcom.sun.management.jmxremote.port=%JMX_PORT% ^
       -Dcom.sun.management.jmxremote.ssl=false ^
       -Dcom.sun.management.jmxremote.authenticate=true ^
       -Dcom.sun.management.jmxremote.password.file=<Path>\jmxremote.password ^
       -Dcom.sun.management.jmxremote.access.file=<Path>\jmxremote.access
      ```

3. Set the owner of the **jmxremote.password** file to the owner of the application process:

   - In Windows, open the **jmxremote.password** file properties and set the owner in **Security Tab > Advanced > Owner**.
   - In UNIX, run this command:

   ```
   chown <username> <path to jmxremote.password>
   ```

4. Update the permissions of the **jmxremote.password** file:

   - In Windows, open the **jmxremote.password** file's **Permissions**:

     1. Add read permissions, if absent.

2. Remove any write permissions, for example, remove **create files/write data** and **create folders/append data**.

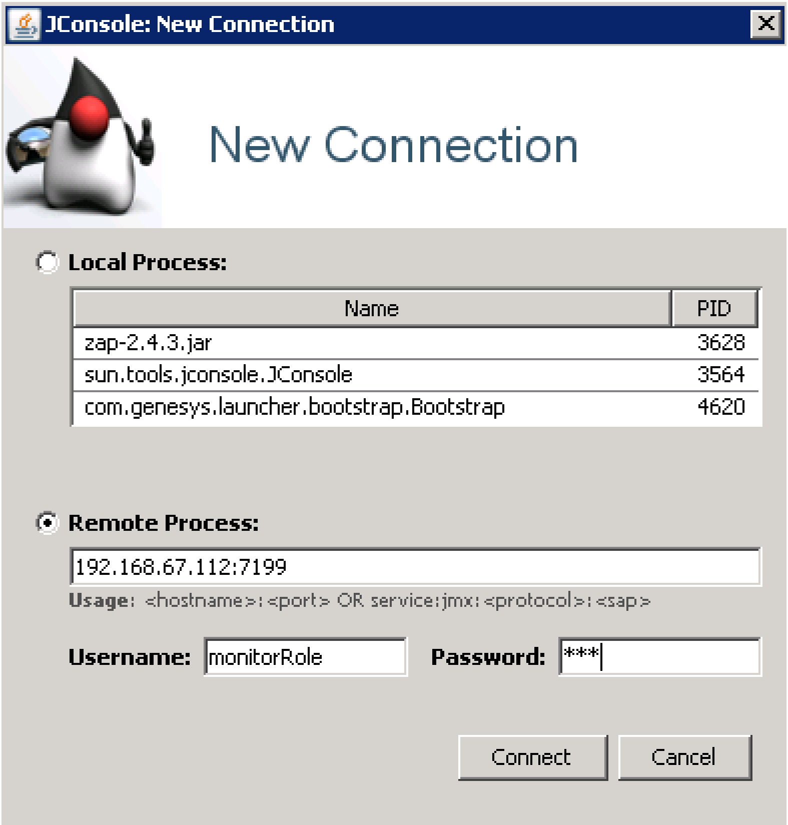
- In Unix, run this command:

```
chmod 444 <path to jmxremote.password>
```

After enabling remote JMX, you can test your authentication using the procedure below.

Testing Remote JMX Authentication

1. Start the application server.
2. Run JConsole or another JMX visual tool and log in with read-only credentials (**monitorRole** by default).



JConsole: New Connection

New Connection

○ Local Process:

Name	PID
zap-2.4.3.jar	3628
sun.tools.jconsole.JConsole	3564
com.genesys.launcher.bootstrap.Bootstrap	4620

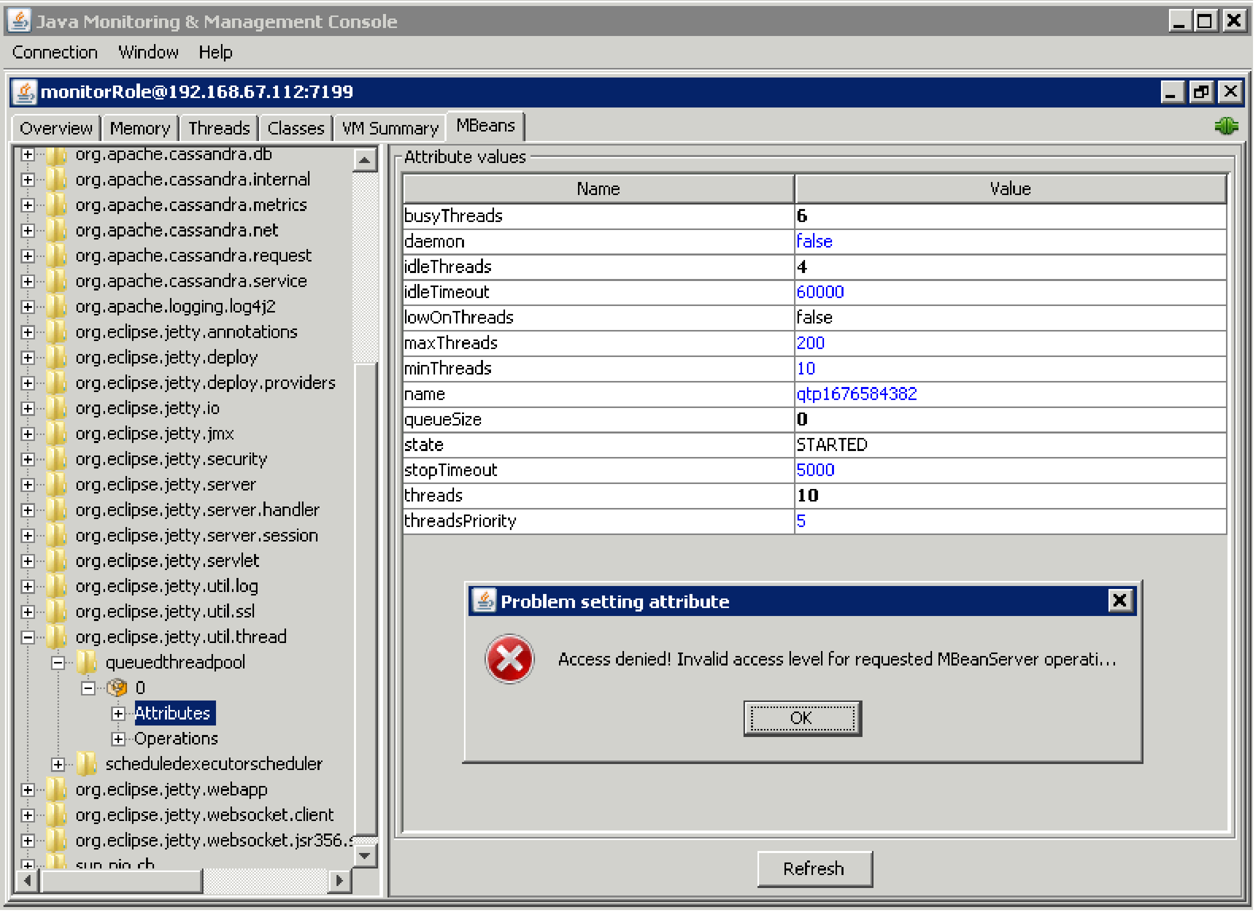
● Remote Process: 192.168.67.112:7199

Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap>

Username: monitorRole

Password: ***

3. Go to **Mbeans**. Expand **org.eclipse.jetty.util.thread** and the **queuethreadpool** attributes. Try to change the **maxThreads**. You should see an access denied error.


Java Monitoring & Management Console
Connection
Window
Help
monitorRole@192.168.67.112:7199
Overview
Memory
Threads
Classes
VM Summary
MBeans
org.apache.cassandra.db
org.apache.cassandra.internal
org.apache.cassandra.metrics
org.apache.cassandra.net
org.apache.cassandra.request
org.apache.cassandra.service
org.apache.logging.log4j2
org.eclipse.jetty.annotations
org.eclipse.jetty.deploy
org.eclipse.jetty.deploy.providers
org.eclipse.jetty.io
org.eclipse.jetty.jmx
org.eclipse.jetty.security
org.eclipse.jetty.server
org.eclipse.jetty.server.handler
org.eclipse.jetty.server.session
org.eclipse.jetty.servlet
org.eclipse.jetty.util.log
org.eclipse.jetty.util.ssl
org.eclipse.jetty.util.thread
queuedthreadpool
0
Attributes
Operations
scheduledexecutorscheduler
org.eclipse.jetty.webapp
org.eclipse.jetty.websocket.client
org.eclipse.jetty.websocket.jsr356.s
sun.nio.ch
Attribute values
Refresh

Name	Value
busyThreads	**6**
daemon	false
idleThreads	**4**
idleTimeout	60000
lowOnThreads	false
maxThreads	200
minThreads	10
name	qtp1676584382
queueSize	**0**
state	STARTED
stopTimeout	5000
threads	**10**
threadsPriority	5

Problem setting attribute

Access denied! Invalid access level for requested MBeanServer operati...

OK

4. Repeat the previous steps with read-write access (**controlRole** by default) and verify you do *not* see an access denied error.

Important

As an additional security measure, you can add an SSL certificate to prevent JMX passwords from being passed as plain text. For details, see Enabling remote JMX with password authentication and SSL.

Cassandra Security

This article describes how to tune secure access from your Co-browse Server to external Cassandra. Starting from 8.5.1, you can secure the following when using external Cassandra:

- Secure the access interfaces using authentication and authorization.
- Secure network traffic using TLS.

Securing Access Interfaces

You can secure your access interfaces based on an authentication and authorization scheme. In other words, Cassandra needs to know:

- **Authentication**—who is trying to access the system?
- **Authorization**—is the user allowed to access the system and what data can the user access?

With the default setup, anybody can access any data. To secure access interfaces from Co-browse Server to external Cassandra, you must:

1. Turn on authentication and authorization in your Cassandra configuration.
2. Set up a new Cassandra user to access the Co-browse keyspace.
3. Specify Cassandra user settings in the Resource Access Point configuration.

Configure Cassandra to Use Authentication and Authorization

Configure Cassandra by editing **<Cassandra installation directory>/conf/cassandra.yaml**.

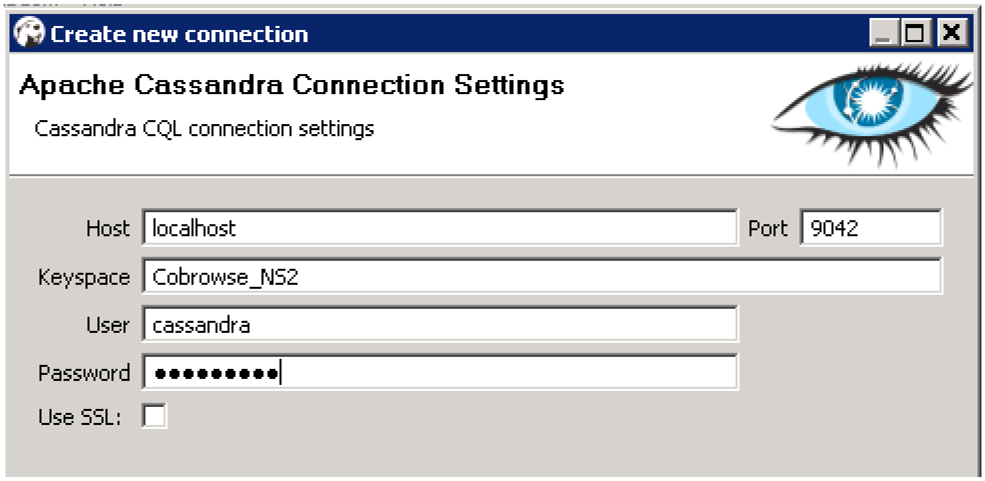
1. Set the **authenticator** option to `PasswordAuthenticator`. It's set to `AllowAllAuthenticator` by default.
2. Set the **authorizer** option to `CassandraAuthorizer`. It's set to `AllowAllAuthorizer` by default.
3. Optionally, tune your **sytem_auth** keyspace replication according to the DataStax system_auth documentation. Note that the validity period for permisions caching is 2000 ms. For more information about Cassandra permissions, see the DataStax Object permissions documentation.
4. Restart your Cassandra node.

Set Up a New Cassandra User

To set up a new Cassandra user, use a Cassandra client tool like **dbeaver** or **cqlsh**:

1. Start by connecting to Cassandra using the default superuser name and password, **cassandra/cassandra**. The following examples use dbeaver and cqlsh as examples but you can use a different Cassandra client:
    - **dbeaver**:

Navigate to **New connection > Cassandra CQL > Appache Cassandra Connection Settings**. Specify the **Host** and **Keyspace**. Use your superuser login for **User** and **Password**.



Apache Cassandra Connection Settings

Cassandra CQL connection settings

Host: localhost Port: 9042

Keyspace: Cobrowse_NS2

User: cassandra

Password: ●●●●●●●●●●

Use SSL: □

- **cqlsh**:

  Start cqlsh using the default superuser name and password:

  ```
  ./cqlsh -u cassandra -p cassandra
  ```

2. Use the CREATE USER CQL statement to create another superuser. For example:

   ```
   CREATE USER IF NOT EXISTS <new_cobrowse_user> WITH PASSWORD 'new_password' SUPERUSER
   ```

3. Use the GRANT CQL statement to grant access permisions. For example:

   ```
   GRANT ALL PERMISSIONS  ON <cobrowse_keyspace> TO <new_cobrowse_user>
   ```

   CQL also supports the authorization statements GRANT, LIST PERMISSIONS, and REVOKE.

Deactivate Default Superuser

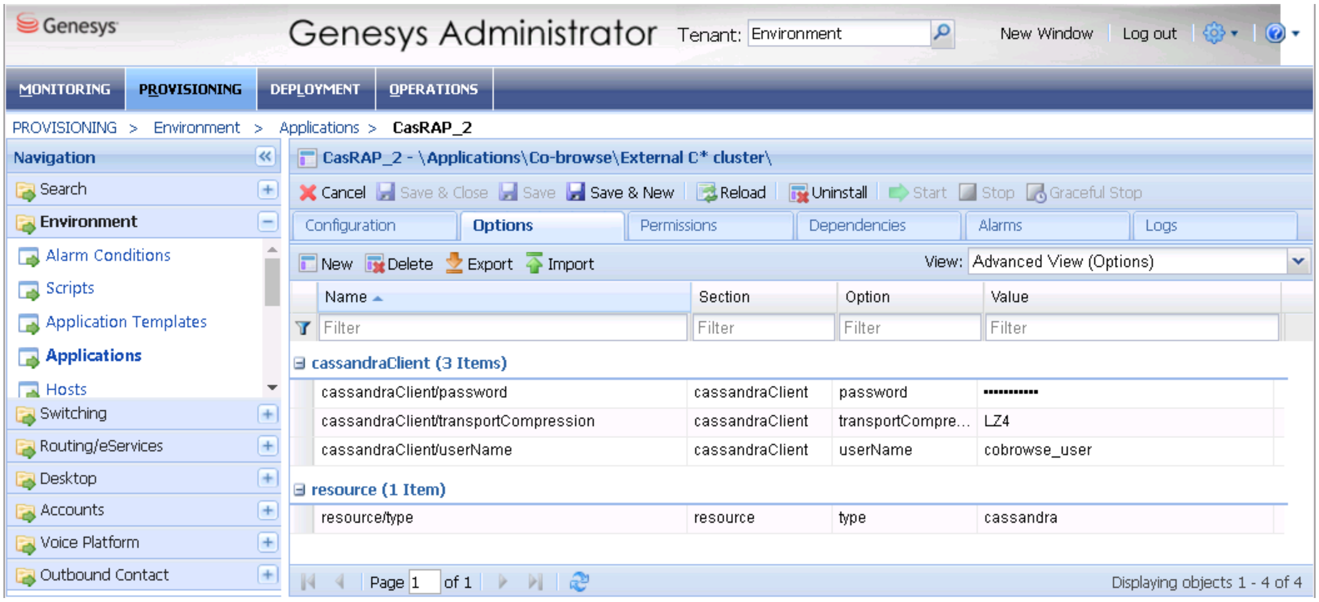
Optionally, you can now deactivate the default superuser **cassandra**:

1. Login as your new superuser.
2. Change the password for the **cassandra** user.
3. Turn off the superuser status for the **cassandra** user.

Configure Resource Access Point

Use the login information of the superuser you created to configure the Cassandra Resource Access Point:

1. Open or create a **cassandraClient** configuration options section.
2. Set the **userName** and **password** to your superuser's login.



PROVISIONING > Environment > Applications > CasRAP_2

CasRAP_2 - \Applications\Co-browse\External C* cluster\

View: Advanced View (Options)

Name	Section	Option	Value
cassandraClient (3 Items)			
cassandraClient/password	cassandraClient	password	••••••••••••
cassandraClient/transportCompression	cassandraClient	transportCompre...	LZ4
cassandraClient/userName	cassandraClient	userName	cobrowse_user
resource (1 Item)			
resource/type	resource	type	cassandra

Agent Authentication

This article describes how to configure token-based agent authentication between Co-browse server and Workspace Desktop Edition. When enabled, Co-browse checks for a valid token for all communication between server and desktop. For security, tokens are stored in the database using AES128 encryption.

Retrieving a Token

By default, token-based agent authentication is disabled.

To get started setting up authentication, first you need to retrieve a token. Using curl, your browser, or any other http client, enter this URL:

```
<host:port>/cobrowse/rest/authtoken
```

This generates a random 26-digit token. Enter this token in two places:

- Co-browse Cluster application
- Workspace Desktop Edition application in Genesys Administrator

> **Important**
>
> Make sure you configure the token in both applications; otherwise, the session will fail and generate an error message in the Co-browse agent interface.

Configuring the Co-browse Cluster Application

1. Open Configuration Options for the Co-browse Cluster application.
2. Select `Options`
3. Select `slave`
4. Select `password`
5. In the `Option Value` field, enter the 26-digit token.

Configuring the Workspace Desktop Edition Application

After you've entered the token in the Co-browse Cluster application, you now need to enter the token in Workspace Desktop Edition.

1. Open Genesys Administrator and navigate to **PROVISIONING > Environment > Applications**.
2. Select the Workspace Desktop Edition application.
3. Go to `Options` > cobrowse section.
4. Select `password`.
5. In the `Option Value` field, enter the 26-digit token.

Token Validation

The following validation scenarios apply:

- The server checks the validity of a token when the server starts.
- The server checks the validity of a token when the `password` configuration option is updated in the Co-browse Cluster application or the Workspace Desktop Edition application in Genesys Administrator.
- If a token is not set on Co-browse Cluster application and not present in the request, the Co-browse session proceeds without agent authentication.
- If a token is present in a request but not set in the Co-browse Cluster application, the Co-browse session proceeds without agent authentication.

Invalid Tokens

Scenario: Token is Invalid

If a token is invalid, the following occurs:

- An error message displays on the agent interface.
- A warning appears on the Co-browse server log.

In this case, the Co-browse session does not start.

Scenario: Token is Not Configured in the Co-browse Cluster Application

If a token is configured in the Workspace Desktop Edition application but not configured in the Co-browse Cluster application, the following occurs:

- A warning appears on the Co-browse server log.

In this case, the Co-browse session is established successfully but without agent authentication.