Deployment Guide

Genesys Co-browse 8.1.3

1/8/2022

Table of Contents

Genesys Co-browse 8.1 Deployment Guide

Welcome to the *Genesys Co-browse 8.1 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 Interaction Workspace** provides co-browsing functionality for Interaction Workspace users.
- **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.
- **Integrated JavaScript Application** is a JavaScript component that includes the Chat, Co-browse, and (optionally) Web Engagement Tracker JavaScript applications. You should add this component to the pages on your website where you want to enable co-browsing.

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.
- 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.

Browser Support

Genesys Co-browse supports the following browsers:

- Internet Explorer 9 and above (Windows)
- Firefox 17 and above (Windows, Linux, and Solaris)
- Safari 6 and above (Mac)
- Google Chrome (Windows)

Genesys recommends that you use Internet Explorer 10 or above on agent machines for improved synchronization speed due to Web Sockets support and better JavaScript engine performance.

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.

Related Components

Genesys Co-browse interacts with the following Genesys Products:

- Interaction Workspace — The Genesys Co-browse Plug-in for Interaction Workspace is required to interface Genesys Interaction Workspace with Genesys Co-browse. This plug-in enables the agent to join and terminate a co-browsing session with a customer.
- Chat Server — An eServices component, Chat Server handles chat interactions between agents and web visitors.
- Genesys Web Engagement — If Genesys Web Engagement is installed, the chat widget can be integrated with Genesys Co-browse and used to initiate a co-browsing session.

Minimum Required Components

The following components are mandatory for Genesys Co-browse:

Server Name	Compliant Versions (and later)
Configuration Server	8.1.100.14
Chat Server	8.1.000.41
Interaction Workspace	8.1.401.44
Workspace Desktop Edition	8.5.100.05
Universal Contact Server	8.1.001.12
Interaction Server	8.1.000.13
Universal Routing Server	8.1.100.09
Stat Server	8.1.000.23
CCPulse+	8.0.101.34
Data Modeling Assistant	7.6.100.01

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 agent 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

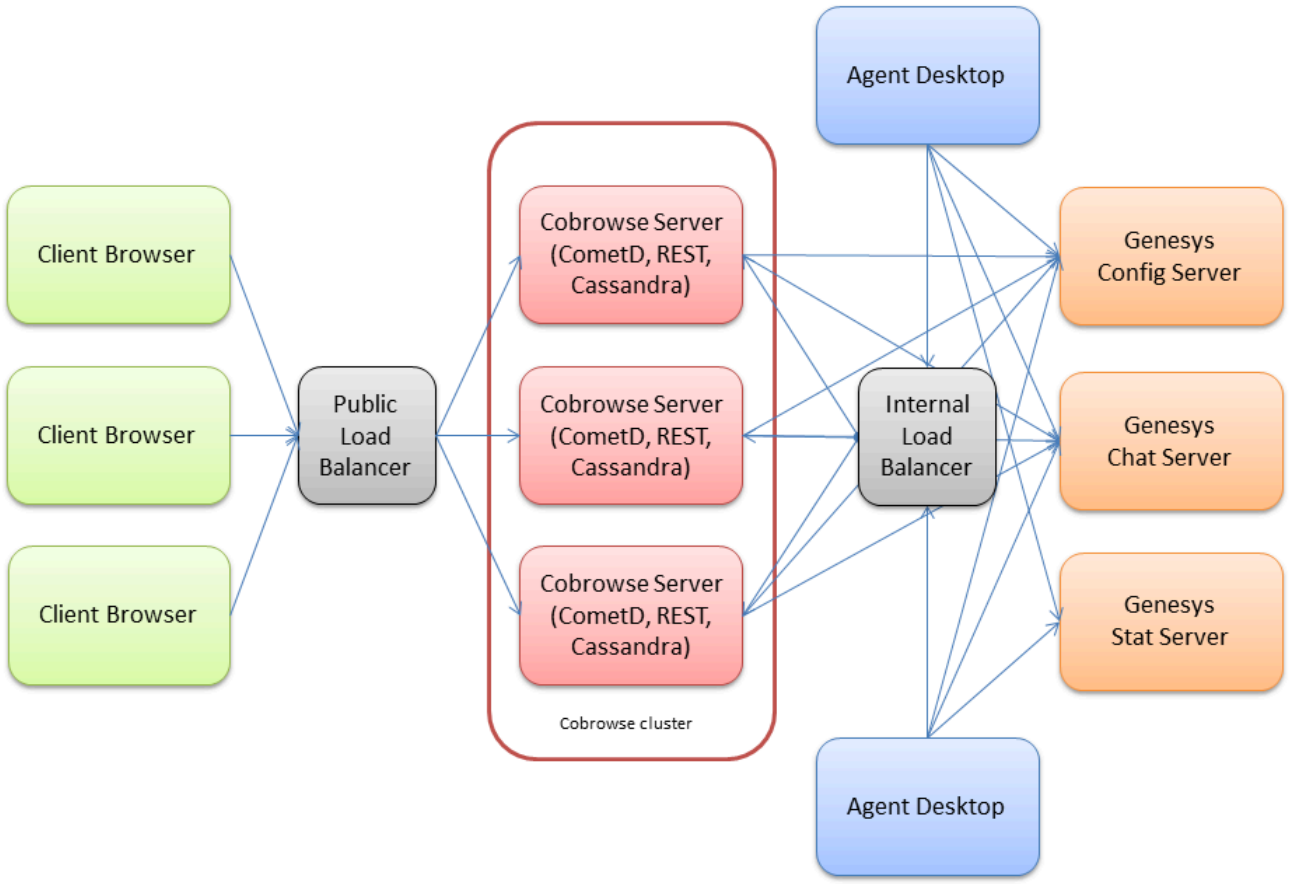
Restrictions and Known Limitations

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:

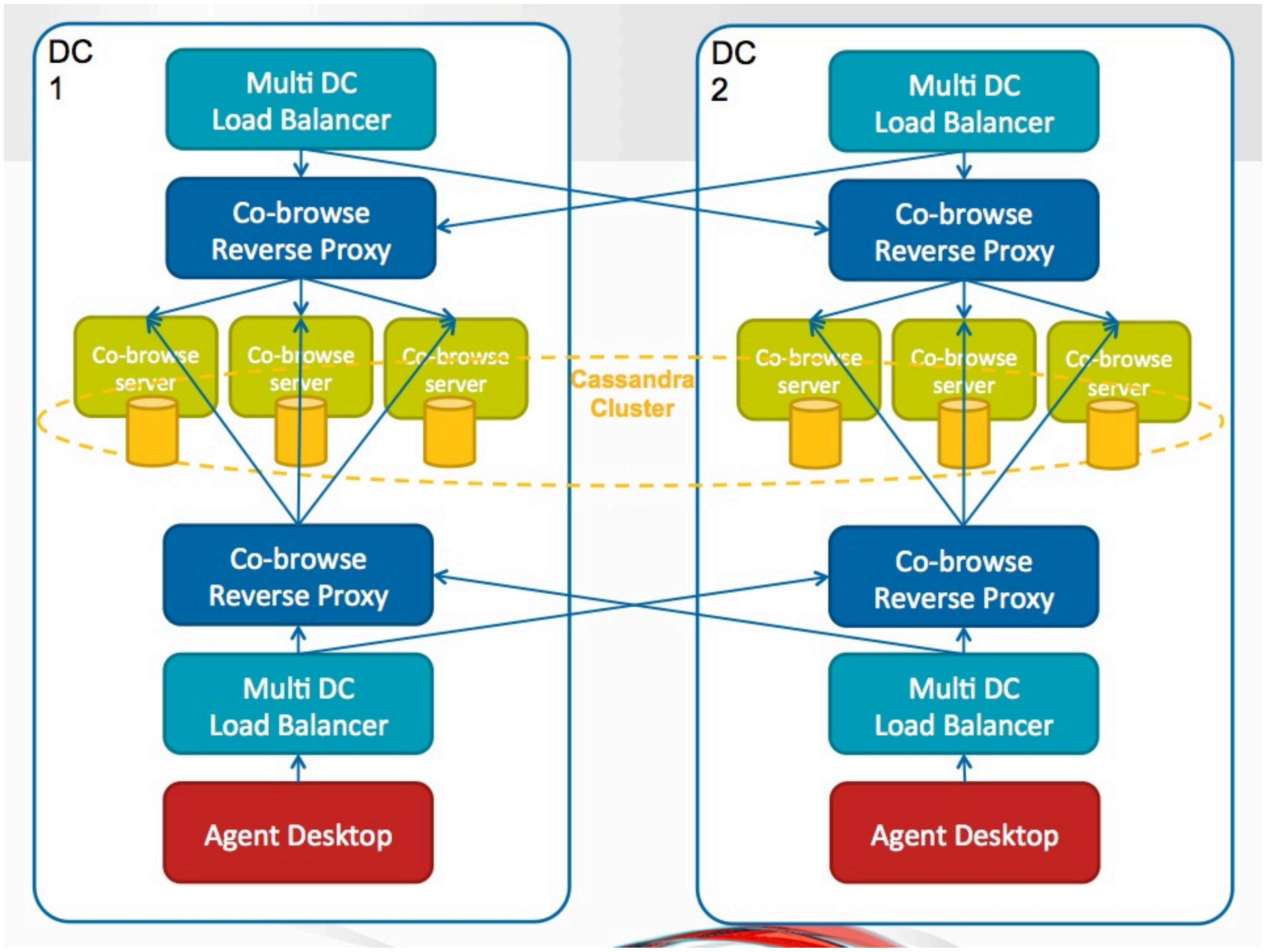


- Each Co-browse server has the same role in the cluster and must be identically configured.
- Each Co-browse server hosts the following:
  1. CometD server with Co-browse and Web Chat Services
  2. Live and Historical session REST APIs
  3. Embedded Cassandra Node
- A Co-browse cluster is formed through 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

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.

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

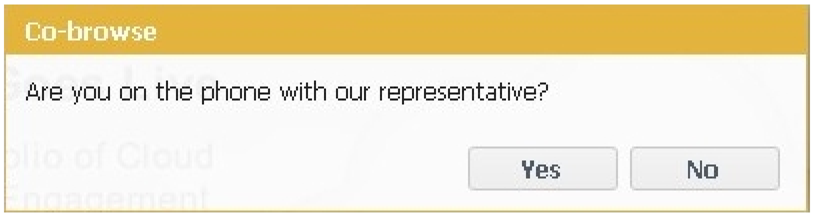
A co-browse session can only be initiated by a customer. An agent does not have the option or ability to send a co-browse request to a customer. This provides greater security to the customer. In order to initiate a co-browse session, the customer must already be engaged in an interaction with an agent, 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 (also referred to as the "Master") 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.

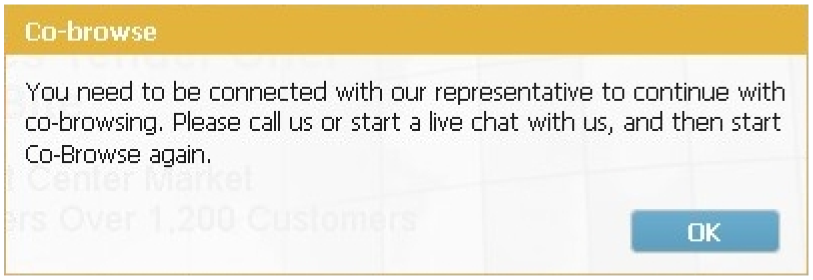
By default, there is a "Co-browsing" button on the left side of every web page that supports co-browsing. Note that the location on the page can vary, depending on configuration. When the customer clicks this button, they are presented with a message window asking them to confirm that they are engaged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



Are you on the phone with our representative?

Co-browse message

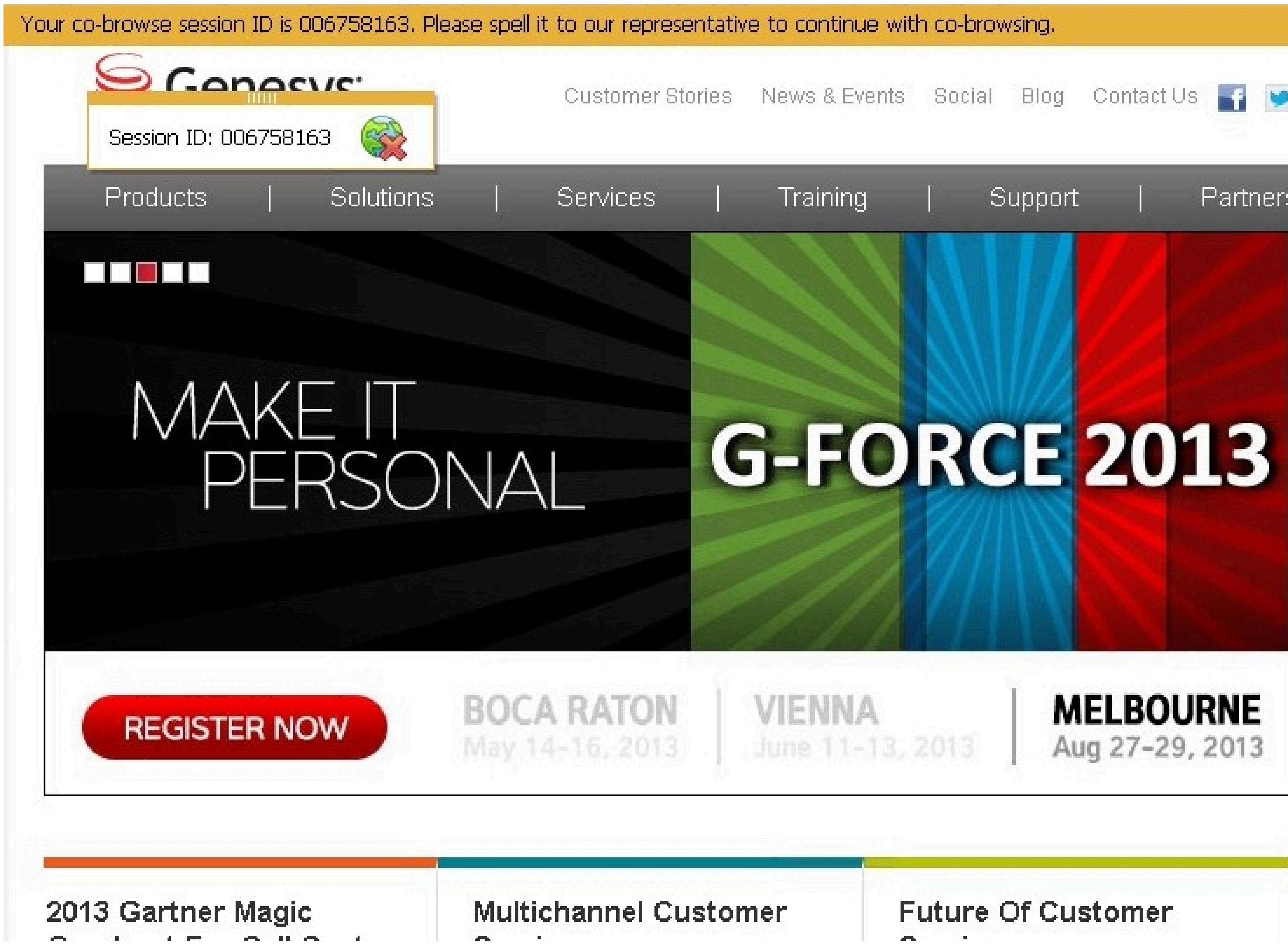
If the customer selects "No", a new message advises them to either initiate a voice call or a chat in order to co-browse.



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

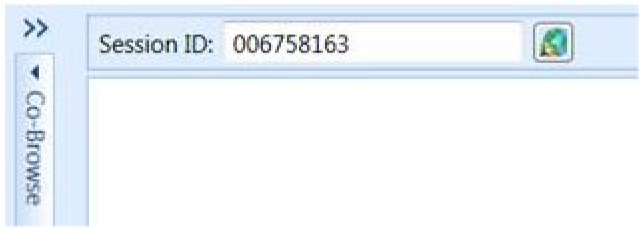
If the customer selects "Yes", an alpha-numeric session identifier appears on the customer's screen.


Your co-browse session ID is 006758163. Please spell it to our representative to continue with co-browsing.

Session ID: 006758163

Session identifier

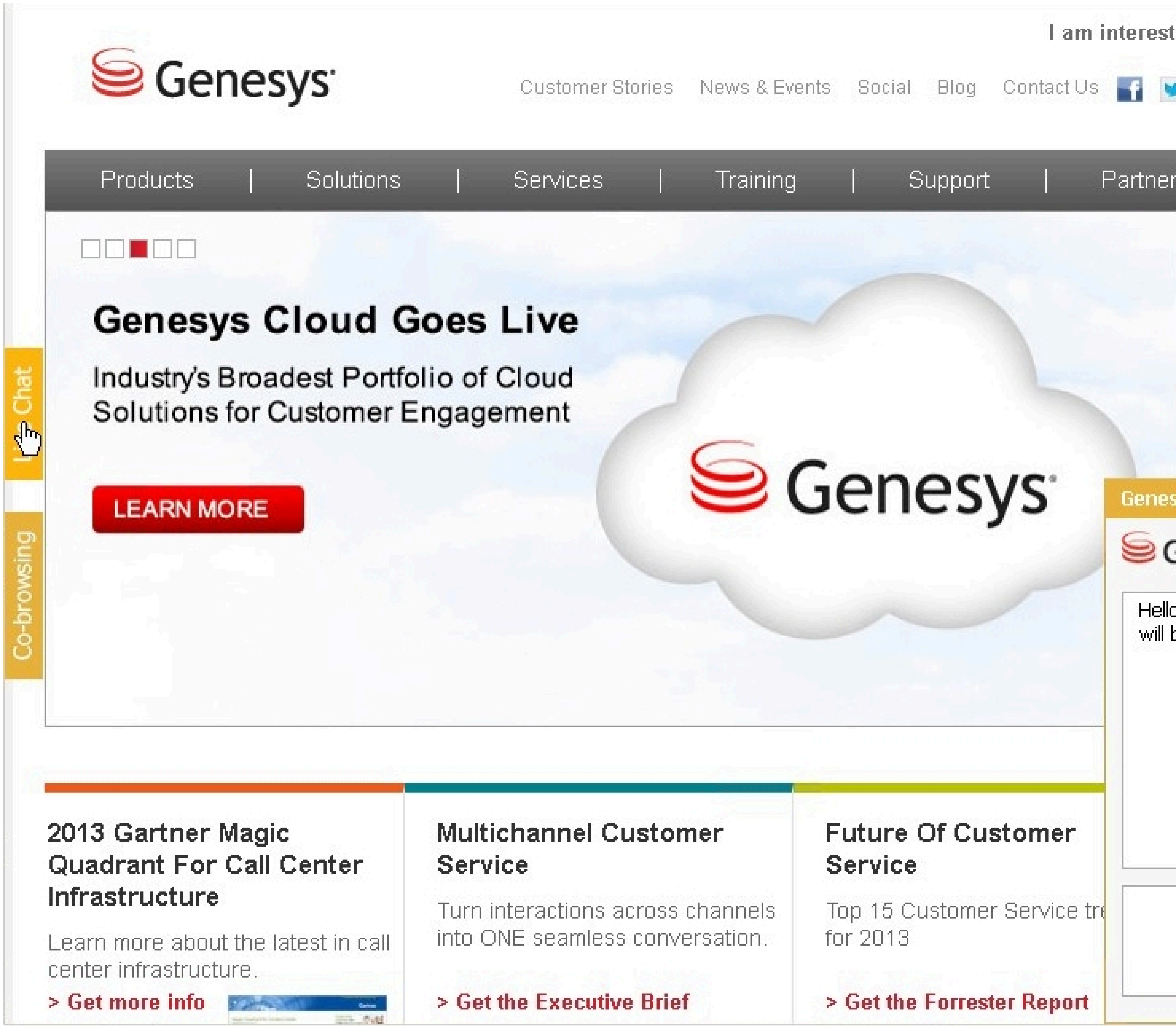
This identifier can then be read to the agent over the phone or the customer might have to send the session ID through their external chat window. The agent enters the session identifier in the appropriate field in Interaction Workspace, and then the customer's browser is displayed in the agent's view. 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.


Session ID: 006758163

Session identifier, Agent view

Initiating a co-browse session from an integrated chat

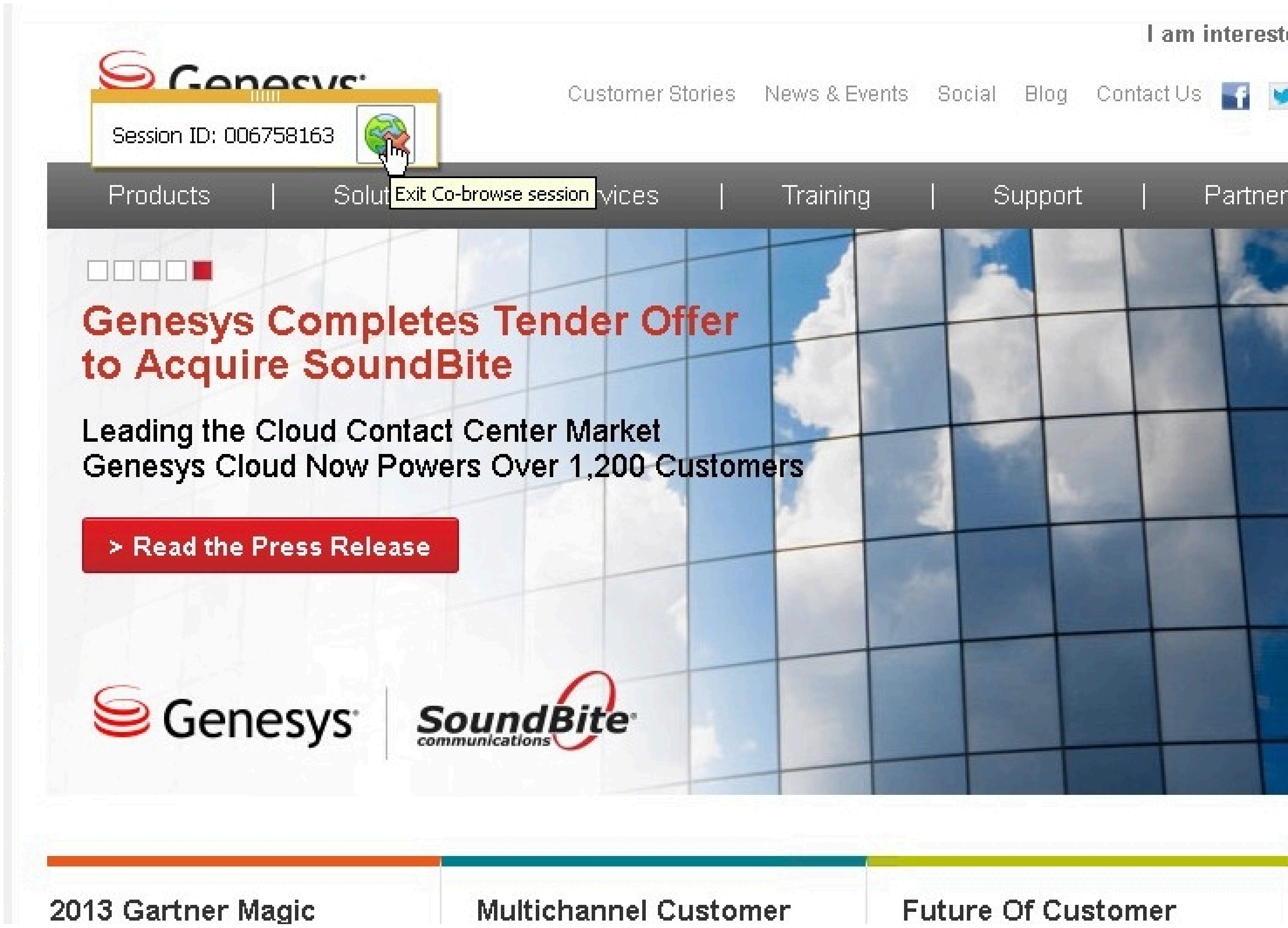
A co-browse session can also be initiated by the customer if the customer and agent are engaged in a chat. Genesys Co-browse has a pre-built html chat client that is embedded directly into the website, so that the chat window is integrated into the browser window the customer is viewing. The customer can use this chat client by clicking the default "Live Chat" 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 initiate a chat session with a representative.


I am interest
Genesys
Customer Stories
News & Events
Social
Blog
Contact Us
Products
Solutions
Services
Training
Support
Partner
Genesys Cloud Goes Live
Industry's Broadest Portfolio of Cloud Solutions for Customer Engagement
LEARN MORE
Chat
Co-browsing
Hello
will b
2013 Gartner Magic Quadrant For Call Center Infrastructure
Learn more about the latest in call center infrastructure.
> Get more info
Multichannel Customer Service
Turn interactions across channels into ONE seamless conversation.
> Get the Executive Brief
Future Of Customer Service
Top 15 Customer Service tr
for 2013
> Get the Forrester Report
Starting Live Chat

The chat window might include a "Co-browse" button or the customer might click the default "Co-browsing" button (name and location can vary, depending on configuration). As in the previous example, a session identifier will be assigned and displayed. Depending on the configuration, the customer might have to send the session ID to the agent through their chat window, or the session ID might be sent to the agent's desktop automatically. Once the session ID has been entered in the agent's desktop (either manually by the agent or programmatically by the application), the session is established.

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).



Exit 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 preconfigured timeout expires. Likewise, if the agent closes their browser, or navigates to a third-party website, the session will terminate if agent does not return back to the session page within the preconfigured 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

When a co-browse session is active, both the agent and the customer have the right to perform conventional user actions at the same time. Both the agent and the customer are in "write mode" during the session. This means that both the agent and the customer can enter text, click buttons, and so on.

All actions (mouse clicks, key presses, and so on) are actually performed on the master (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.

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 8.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?

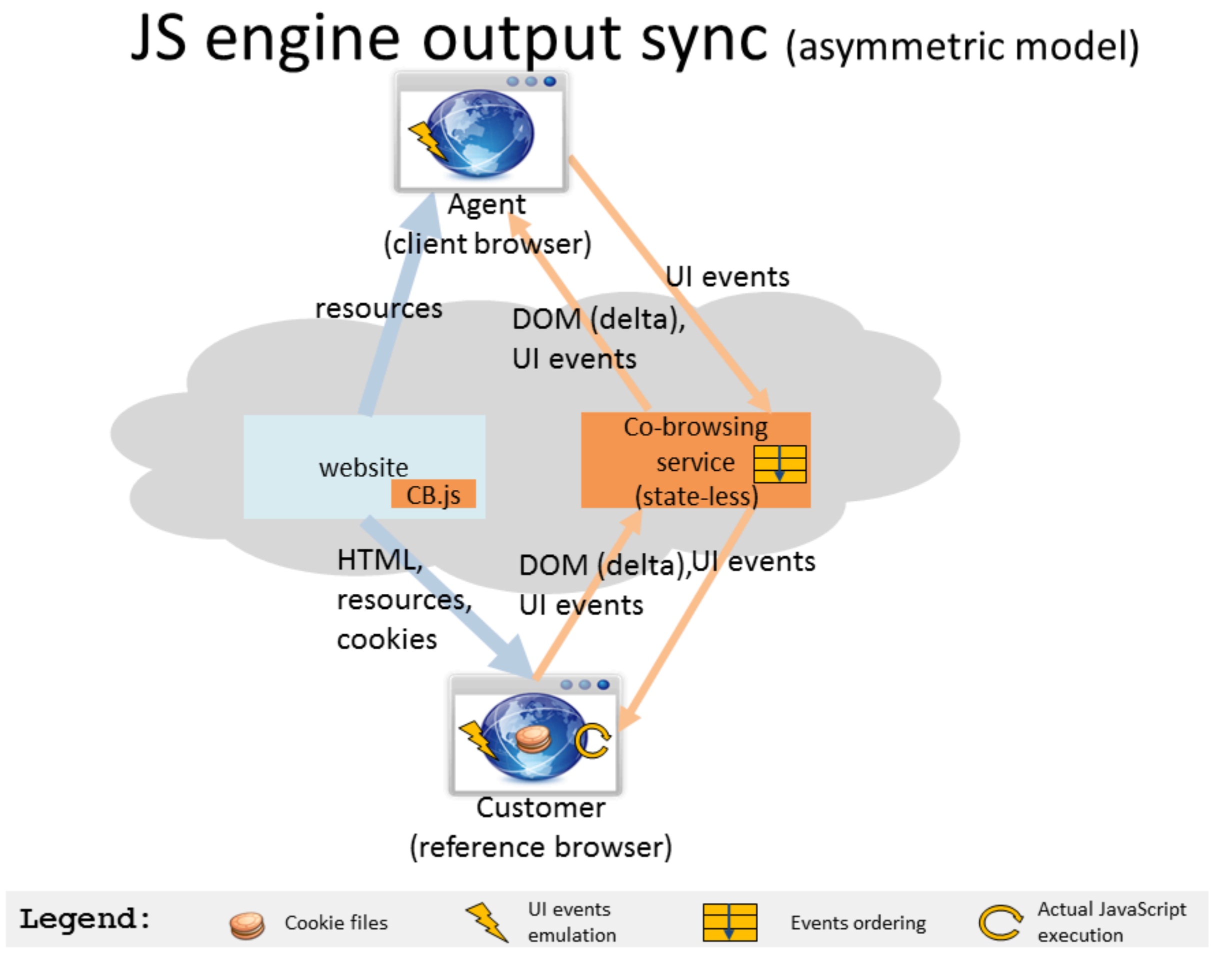
A: Daily builds are verified by Quality Assurance. The main use cases are automated.

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

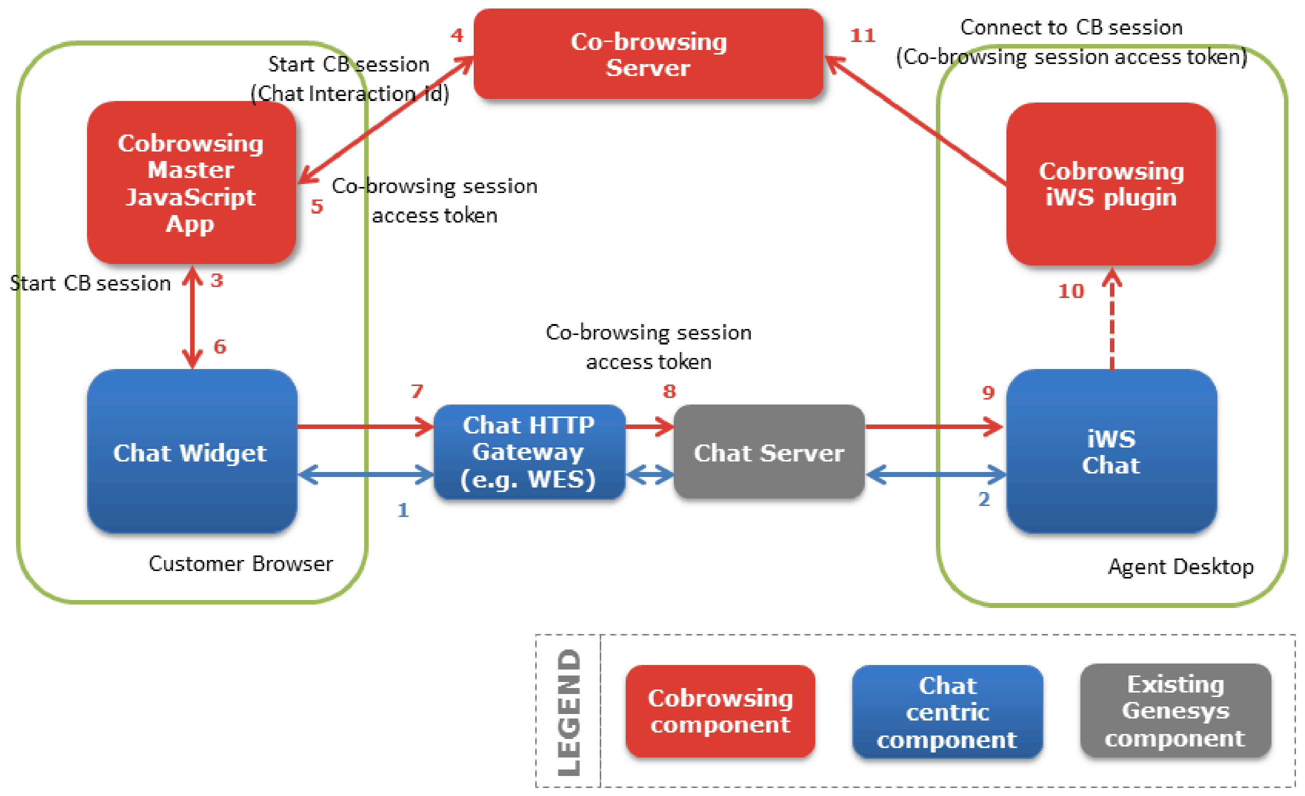
A: Similar principles are implemented by competitors.

Q: What are the basic principles of system inter-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 web site 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: Co-browse is fully integrated with IWS and supports integration with custom agent applications.

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.

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 the Minimum Required Components.
2. Install Genesys Co-browse Server.	See Install Genesys Co-browse Server for details.
3. Install the related plug-in for Interaction Workspace.	Install the Genesys Co-browse Plug-in for Interaction Workspace to enable co-browsing features in Interaction Workspace. See Install the Genesys Co-browse Plug-in for Interaction Workspace for installation details.
4. Load the certificate and private keys into the Java and Jetty keystores.	See Loading Certificate for SSL for details.
5. Configure a Cluster of Chat Servers.	Complete the procedures on the Configure a Cluster of Chat Servers if your solution includes more than one Chat Server.
6. Add the Co-browse JavaScript snippet to your website.	See Website Instrumentation for details.
7. Configure a cluster of Co-browse servers.	See Configure a Cluster of Co-browse Servers for details.
8. Start and stop Genesys Co-browse Server.	See Start and Stop Genesys Co-browse Server for details.
9. 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.
10. 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+
Chat Server	8.1.000.41+
Interaction Workspace	8.1.401.44+
Workspace Desktop Edition	8.5.100.05+
Workspace Web Edition	8.5.200.80+
Universal Contact Server	8.1.001.12+
Interaction Server	8.1.000.13+
Universal Routing Server	8.1.100.09+
Stat Server	8.1.000.23+

Install Genesys Co-browse Server

Configure the JAVA_HOME environment variable

Before installing Genesys Co-browse Server, you must first make sure your JAVA_HOME environment variable is configured correctly. The value for this variable must not have a slash ("\", "/") at the end. For example, `C:\java` (Windows) or `C:/java` (Unix).

For Unix operating systems, the JAVA_HOME environment variable must be defined at the system level to allow Co-browse Server to be started from Genesys Administrator and Solution Control Interface.

For Windows operating systems there is a known bug in Java (http://www.duckware.com/tech/java6msvcr71.html) that you can avoid by completing the following steps:

1. Add `%JAVA_HOME%\bin` to the beginning of the PATH environment variable.
2. Make sure `java.exe` is launched from the `%JAVA_HOME%\bin` folder.

Next Step

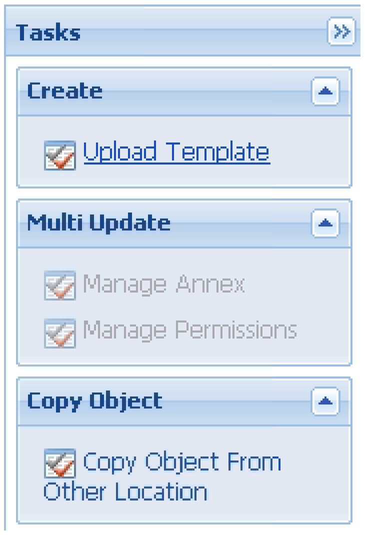
Creating the Co-browse Server Application Object in Genesys Administrator

Creating the Co-browse Server Application Object in Genesys Administrator

Importing the Application Template for the Co-browse Server

Start of Procedure

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 the Add button of the `Click 'Add' and choose application template (APD) file to import` dialog box.
4. Browse to the `Co-browse_Server_813.apd` file, available in the `templates` directory of your installation CD. The `New Application Template` panel opens.
5. Click `Save & Close`.

End of Procedure

Next Step

Creating the Co-browse Server Application

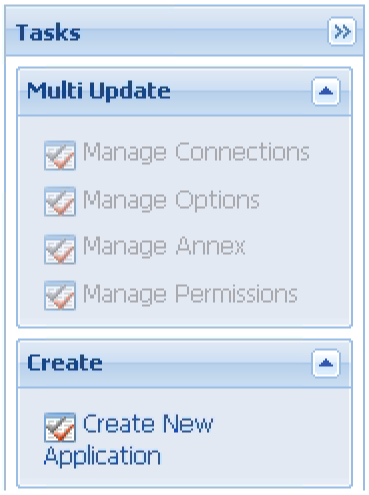
Creating the Co-browse Server Application

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.

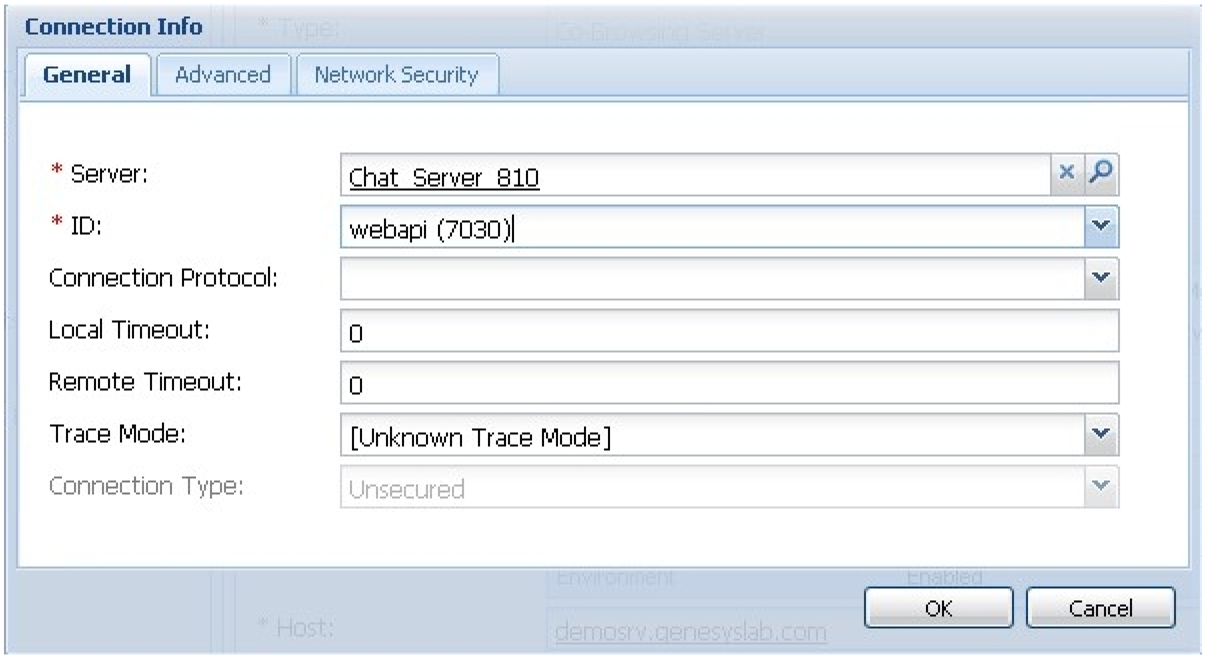
Start of Procedure

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_813.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.

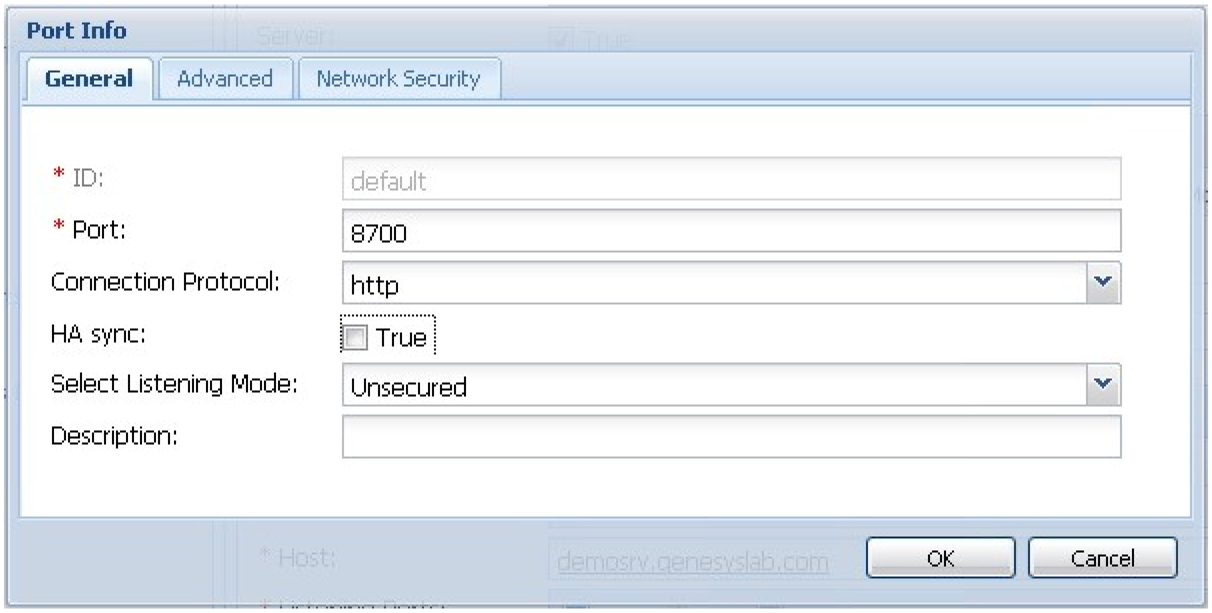


* Server:	Chat_Server_810
* ID:	webapi (7030)
Connection Protocol:	
Local Timeout:	0
Remote Timeout:	0
Trace Mode:	[Unknown Trace Mode]
Connection Type:	Unsecured

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.



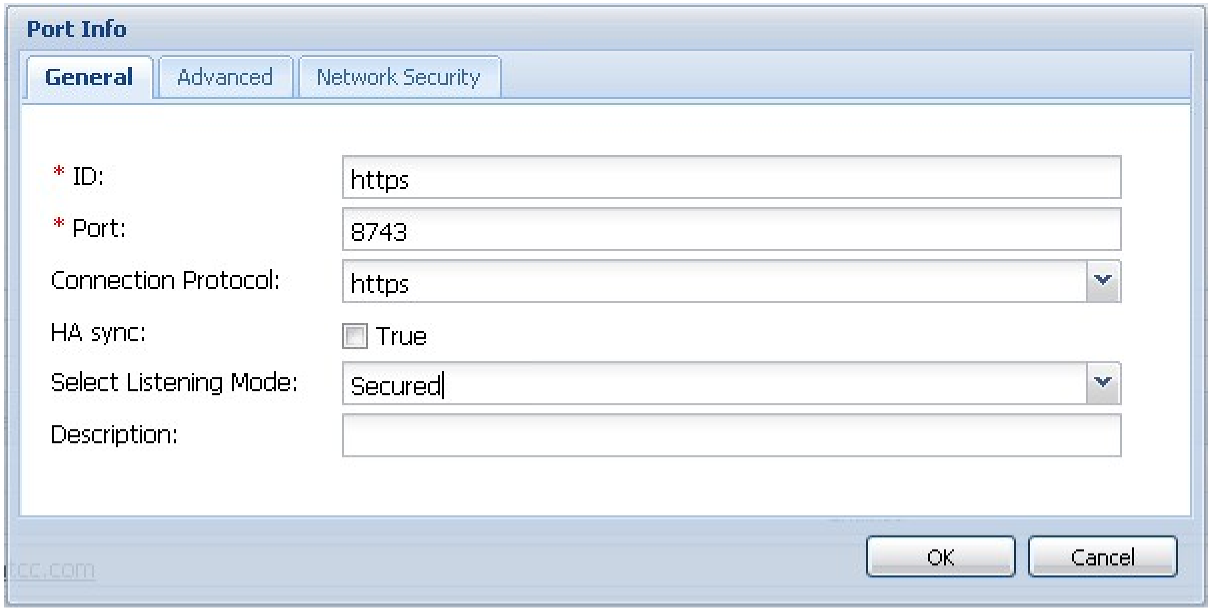
Port Info — General

* 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.


Port Info

General | Advanced | Network Security

* 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 of Procedure

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 are ready to complete the installation process.

Important

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

Prerequisites: You completed Creating the Co-browse Server Application.

Start of Procedure

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 the welcome screen.
2. Click *Next*. The *Connection Parameters to the Configuration Server* screen appears.
3. Under *Host*, specify the host name and port number where Configuration Server is running. (This is the main "listening" port entered in the Server Info tab for Configuration Server.)
4. Under *User*, enter the user name and password for logging on to Configuration Server.
5. Click *Next*. The *Select Application* screen appears.
6. Select the Co-browse Server Application that 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 browse for the desired installation location.

Important

The GWM Proxy that is included in the installation package cannot work if the full path contains spaces. As a workaround, specify the full path to the Co-browse Server applications without spaces during the installation process.

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*. The Genesys Installation Wizard indicates it is performing the requested operation for Co-browse Server. When through, the Installation Complete screen appears.
13. Click *Finish* to complete your installation of Co-browse Server.

End of Procedure

Next Steps

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

Configuring the Co-browse Server

Purpose: 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 of Procedure

1. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`.
2. Select the Co-browse Server Application that you created previously.
3. In the `Options` tab, locate the `sessions` section and update the following option:
   - `domRetrictionsURL`—a URL that points to the XML file that contains DOM restrictions. **Note:** 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 options in the `chat` section:
   - `queuekey`—specifies the endpoint configured in Chat Server, in the format `tenantid:endpointname`.
6. 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.
7. Click `Save & Close`.

End of Procedure

Next Steps

- Install the Genesys Co-browse Plug-in for Interaction Workspace

Install the Genesys Co-browse Plug-in for Interaction Workspace

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 Interaction Workspace in Application Mode

Prerequisites

- You have installed Interaction Workspace in Application mode.
- You have installed Internet Explorer 9 or above.

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 Interaction Workspace 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 Interaction Workspace. When done, the **Installation Complete** screen appears.
5. Click **Finish** to complete your installation of the Genesys Co-browse Plug-in for Interaction Workspace.

End of procedure

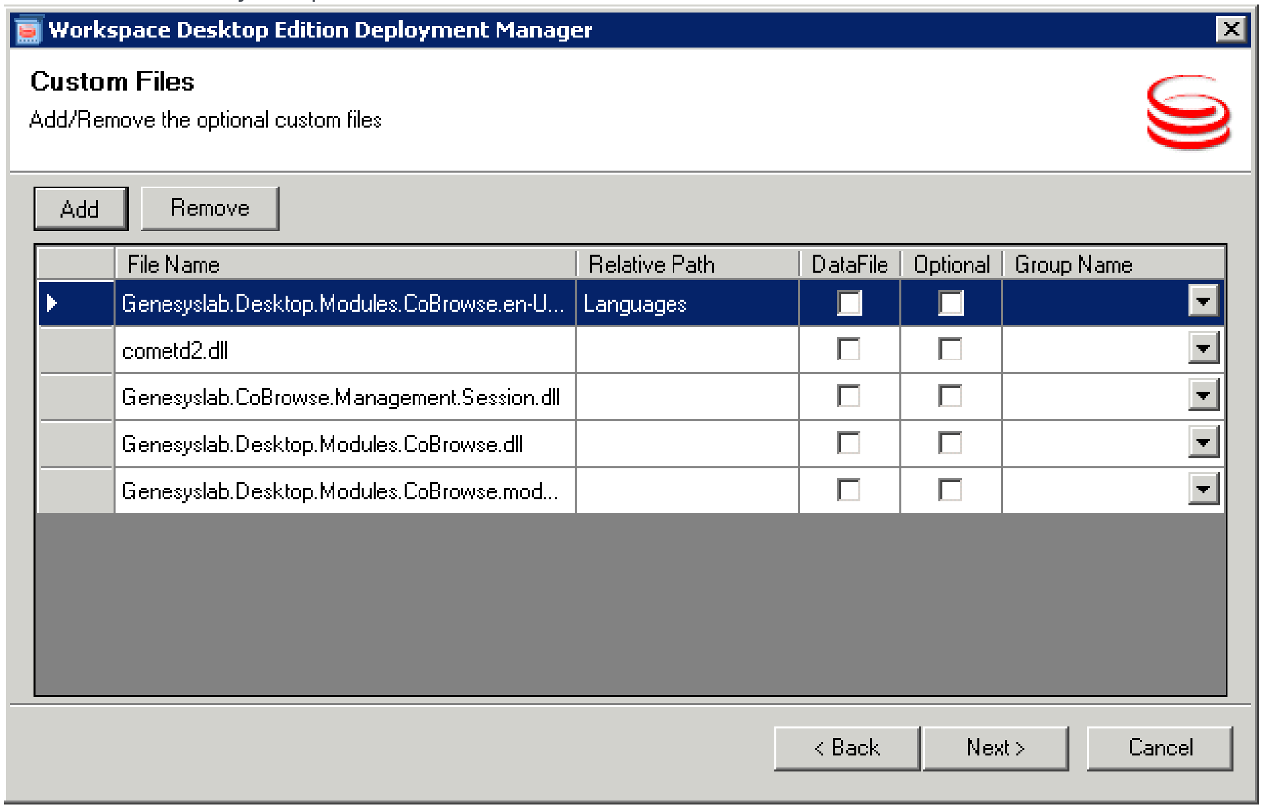
Installing the Genesys Co-browse Plug-in for ClickOnce/ Developers Toolkit Interaction Workspace

Prerequisites

- You have Installed Interaction Workspace in ClickOnce or Developers Toolkit mode.
- You have installed Internet Explorer 9 or above

Start of procedure

1. Install the Co-browse IWS Plug-in in your IWS installation as described in Installing the Genesys Co-browse Plug-in.
2. From the `Start` menu, open **Interaction Workspace- 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 IWS 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
Add
Remove
< Back
Next >
Cancel

	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 IWS **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 IWS main window.

End of procedure

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

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

Prerequisites

- You have installed Interaction Workspace.
- You have installed Genesys Co-browse Server.
- You have installed the Genesys Co-browse Plug-in for Interaction Workspace.

Configuring co-browsing using the Connection to Co-browse Server

Configuring Connection to Co-browse Server

Start of procedure

1. In Genesys Administrator, navigate to `PROVISIONING > Environment > Applications`.
2. Select the Interaction Workspace Application.
3. In the Application's **Connections** section, click **Add**.
4. Select the Co-browse Server and click **OK**.
5. Edit the connection and set the `ID` to the `default` or `https` listening port of your Co-browse Server application.
6. Click **OK**.
7. Click **Save & Close**.

End of procedure

Setting Read Permissions for Co-browse Server application

You should grant Interaction Workspace read access to the Co-browse Server.

Start of procedure

1. Open Genesys Administrator.
2. Select the necessary tenant.
3. Open the properties of the Co-browse Server.
4. Open the **Permissions** tab.
5. Add **Read** permissions for the access group or particular agent:
    1. Click **Add Access Group OR Add User**.
    2. Double click on just added item.
    3. Check **Read (R)** permission.
6. Save changes.

End of procedure

Configuring co-browsing for Co-browse Server Cluster

To configure the IWS plug-in for Co-browse Server Cluster, refer to Configure a Cluster of Co-browse Servers: Modify the slave and controller configuration.

Next Steps

- Loading Certificate for SSL

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.

Important

You must use the Java Development Kit version 1.6.0_29 or higher to support the JSSE keystore.

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/`

Next Steps

- Configure a Cluster of Chat Servers

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 TLS Options

Genesys Co-browse Server includes the following TLS-related configuration options in its security section.

Option	Default Value	Mandatory	Changes Take Effect	Description
trusted-ca-type	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:

Option	Default Value	Mandatory	Changes Take Effect	Description
				valid file name (including path)
trusted-ca-pwd	none	no	after restart	Password for the JKS trusted storage. Valid values: any string

See Configuring Trusted Stores below for details about configuration for a specific type of store (PEM, JKS, MSCAPI).

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 `trusted-ca-type` 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 `trusted-ca-type` 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 `trusted-pwd` option to the password defined for your keystore in Step 2.
8. Click `Save & Close`.

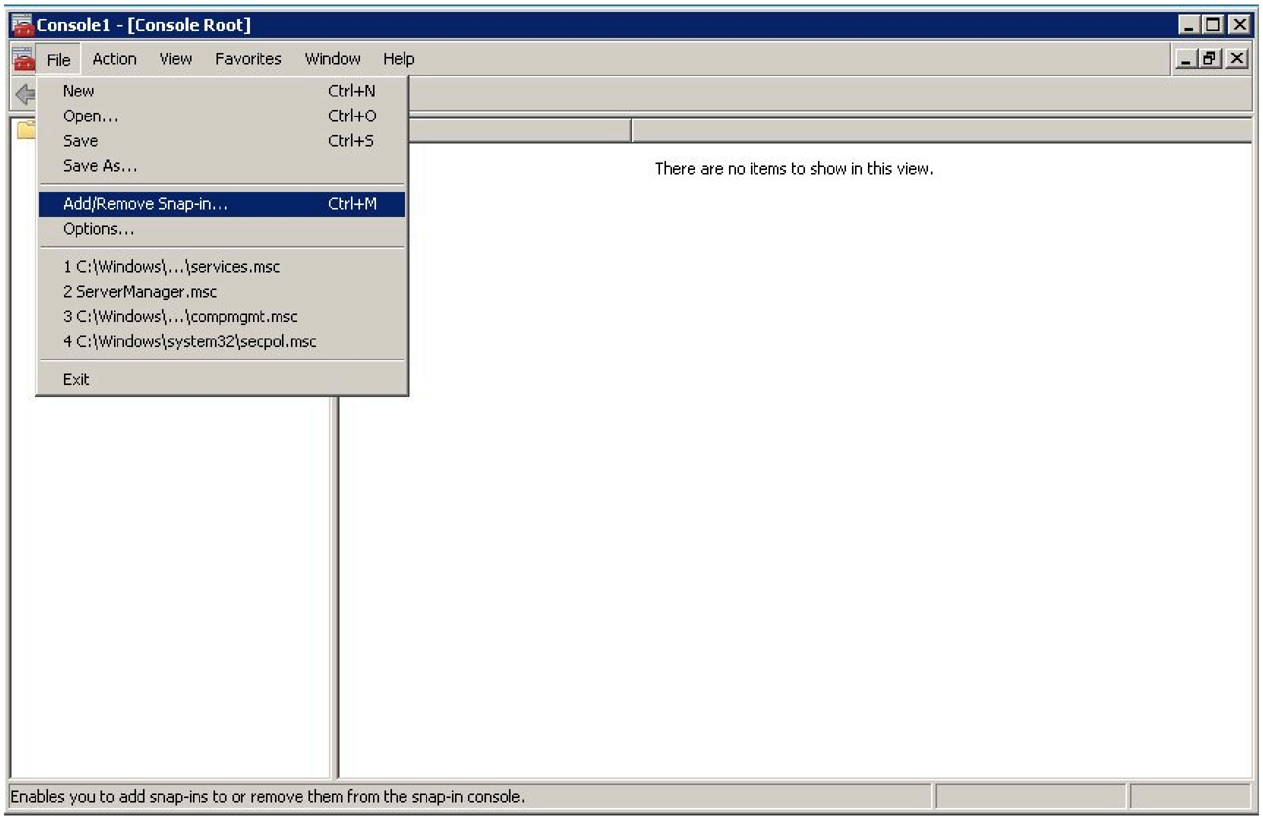
End

MSCAPI Trusted Store

Complete the steps below to work with the MSCAPI certificate trusted store:

Start

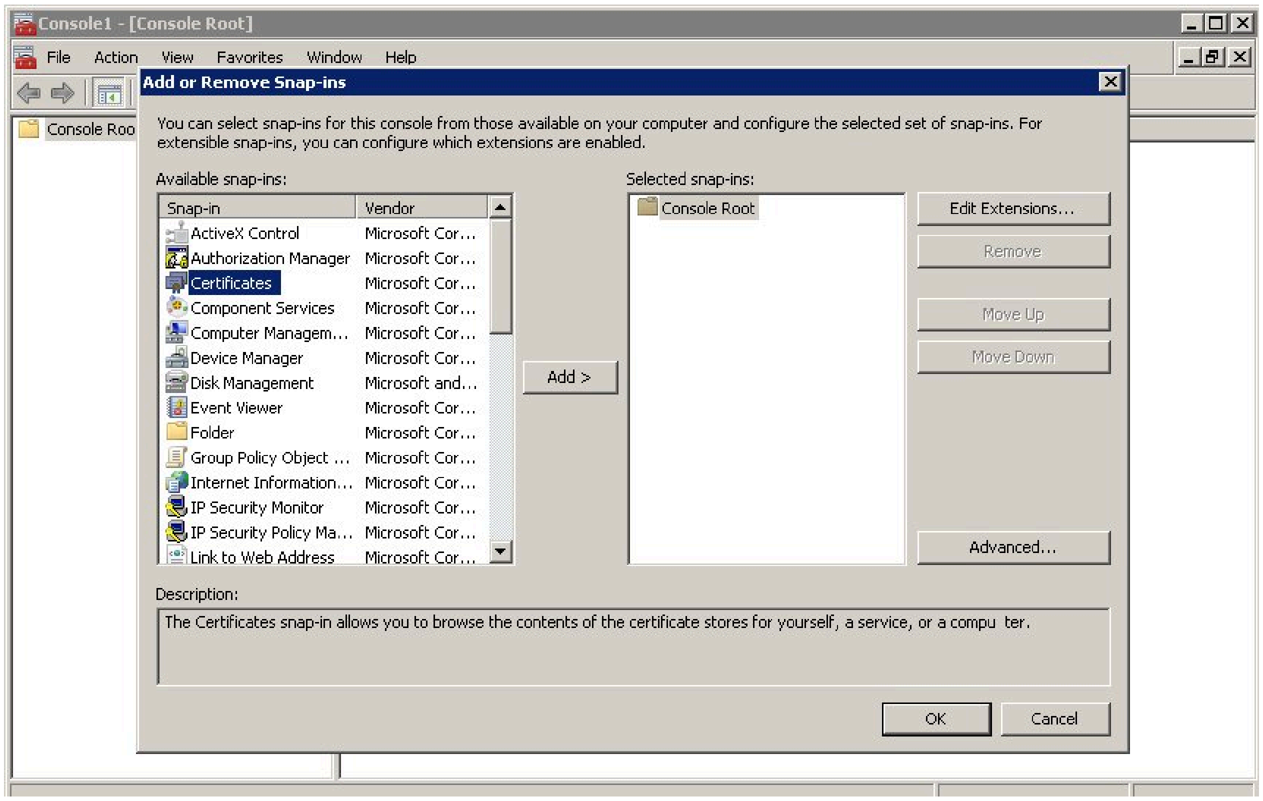
1. Configure and tune TLS for Genesys servers to use certificates signed by the same CA.
2. If the Co-browse Server is running on a different host, copy the trusted CA certificate to this host.
3. Import the CA certificate to WCS via Certificates Snap-in on the Co-browse Server host by launching the 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.

Available snap-ins:

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...

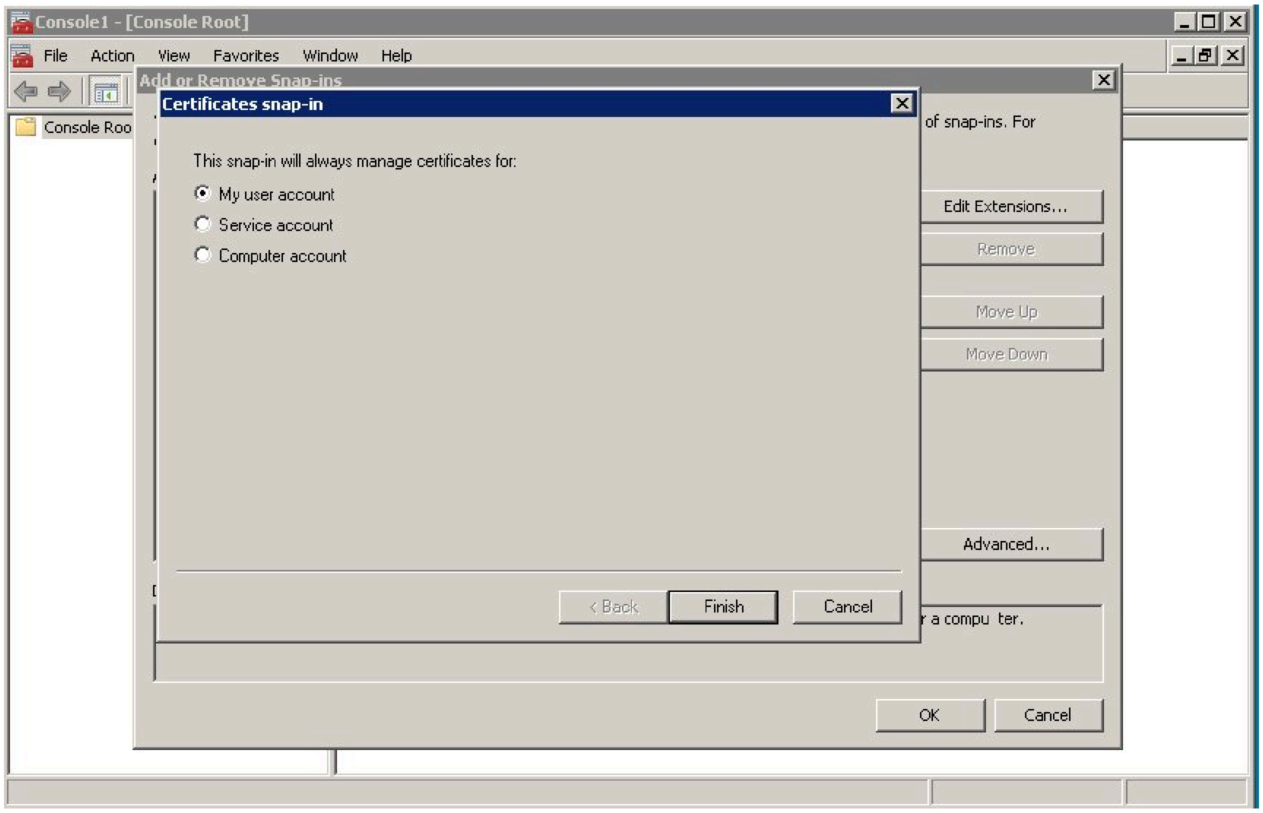
Selected snap-ins:

Console Root

Description:

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a compu ter.

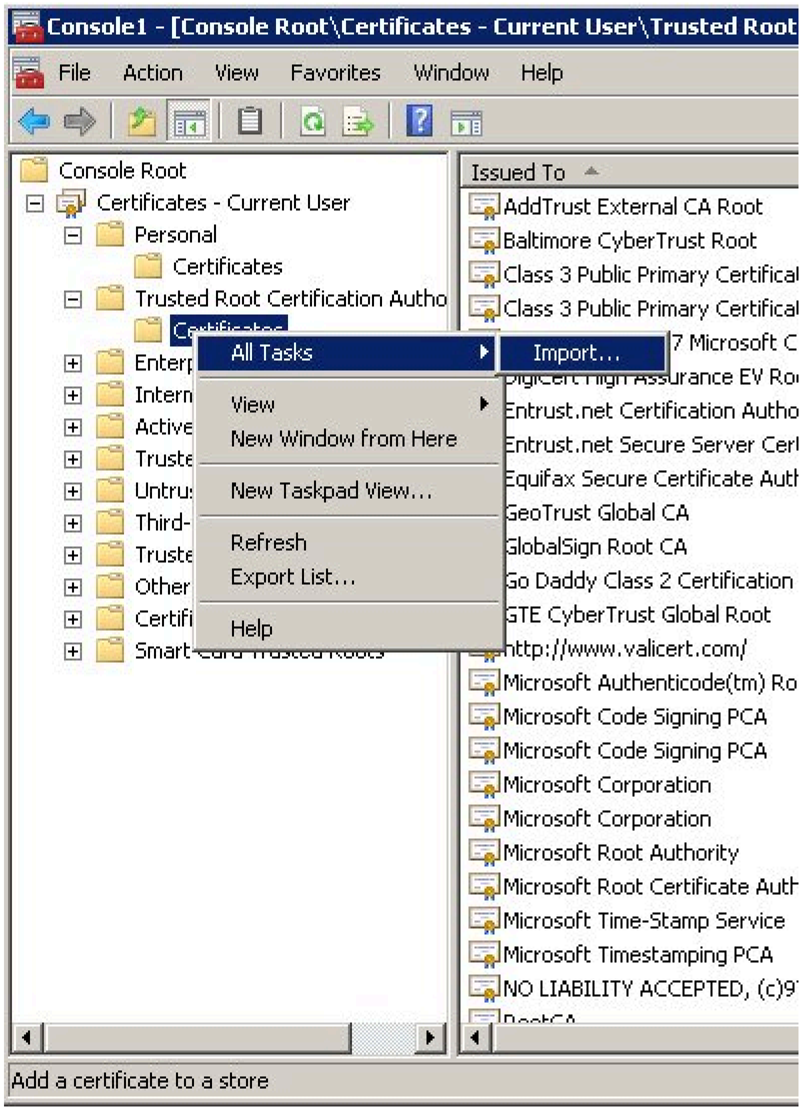
6. Select the account to manage certificates for and click `Finish`. It is important to place certificates under the correct Windows account. Some applications are run as services under the Local Service or System account, while others are run under user accounts. The account chosen in MMC must be the 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.


Console1 - [Console Root\Certificates - Current User\Trusted Root
File Action View Favorites Window Help
Console Root
Certificates - Current User
Personal
Certificates
Trusted Root Certification Autho
Certificates
Enterp
Intern
Active
Truste
Untru
Third-
Truste
Other
Certifi
Smart Card Trusted Roots
All Tasks
Import...
View
New Window from Here
New Taskpad View...
Refresh
Export List...
Help
Issued To
AddTrust External CA Root
Baltimore CyberTrust Root
Class 3 Public Primary Certificat
Class 3 Public Primary Certificat
7 Microsoft C
DigiCert High Assurance EV Ro
Entrust.net Certification Autho
Entrust.net Secure Server Cert
Equifax Secure Certificate Auth
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 `trusted-ca-type` option to `MSCAP`.
12. Click `Save & Close`.

End

Configure a Cluster of Chat Servers

Genesys Co-browse Server 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 Server 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 Server.

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 Server 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 `Tenants` section, click the `Add` button. The `Tenants` panel opens.
   - Select the same tenant object you used for your Chat servers.
4. 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>`
5. Click `Save & Close`

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

The Co-browse JavaScript explicitly sets the document.domain property on the master to allow sychronization of iframes loaded from another sub-domain. The document.domain property is always set by Co-browse to the second-level domain.

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 "Master" 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>/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 Master (end user) browser and the Co-browse Server, use an HTTPS-based URL and HTTPS port instead.

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.

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]]]><![CDATA[, 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.

Enabling Console Logs

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

Enabling Console Logs on the Master

See Integrated JavaScript Application#debug.

Enabling console logs on the slave

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 the "sticky cookies" technique. The Co-browse application sets the `gcbSessionServer` cookie every time:

- A Co-browse session is created
- A chat session is created
- A slave joins an existing session

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 also acts as a Cassandra node. You configure the Cassandra nodes by setting configuration options in the `cbdb` section of the Co-browse Server application.

Important

One agent cannot have more than 1 simultaneous Co-browse sessions against a cluster of Co-browse servers.

Complete the following steps to implement load balancing:

1. Set up 3 or more Co-browse Servers

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

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.

2. Configure the Cassandra cluster

Configure the Cassandra cluster

Prerequisite: You have completed Set up 3 or more Co-browse Servers.

Important

To work correctly, you must configure your Cassandra Cluster according to these rules:

1. Only one Cassandra node per IP allowed.
2. All **rpcPort**, **nativeTransportPort**, **storagePort** and **sslStoragePort** values must be same across all nodes in the Cassandra Cluster.
3. We recommend that you include at least 3 nodes in your Cassandra cluster. For embedded Cassandra, this means you must have at least 3 Co-browse servers to comply with this recommendation.

Start of procedure

Complete the steps below for each Co-browse application you created in Set up 3 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 cbdb section and update the following options:
   1. listenAddress — Set this value to the IP of the node.
   2. rpcAddress — Set this value to the IP of the node.
   3. seedNodes — Set this value to the IP of the first node.
   4. replicationFactor — Set this value to a number less than the total number of nodes. Replication factor is the number of copies of data to keep in the cluster. Typically 3 copies is enough for most scenarios (provided you have more than three nodes in your cluster), but this can be increased to achieve higher consistency levels.
   5. cassandraClusterName (optional) — This name should be the same for each node.
4. Click `Save & Close`.

End of procedure

Configure Initial Tokens for Cassandra

Prerequisite: You have calculated tokens for the nodes in your cluster. See http://www.datastax.com/docs/1.0/initialize/token_generation#calculating-tokens-for-a-single-data-center for details.

You can configure the initial token by putting the token directly in the `cassandra.yaml` file **or** by adding a placeholder in the `cassandra.yaml` file.

Complete **one** of the following procedures:

Add Initial Token to the cassandra.yaml File

Start of procedure

1. For each Co-browse Server in your cluster, open the `<co-browse_installation_directory>/server/etc/cassandra.yaml` file with a text editor.
2. Set `initial_token` to the token value.
3. Save the file.

End of procedure

Add a Placeholder to the cassandra.yaml File

Important

For more information on placeholders and the `cassandra.yaml` template, see the cassandraYamlTemplateUrl option description.

Start of procedure

1. Open the `<co-browse_installation_directory>/server/etc/cassandra.yaml` file for one of your Co-browse servers with a text editor.
2. Add a placeholder for the `initial_token` value, such as $initialToken$.
3. Save the file.
4. Copy the modified `cassandra.yaml` template to all your Co-browse servers.
5. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`.
6. Complete the followings steps for each Co-browse application:
   1. Select the Co-browse application and click `Edit`.
   2. In the `Options` tab, locate the cbdb section and add the new option. The option name is the name of your placeholder, such as `initialToken`, and the value is the token generated for the server.
   3. Click `Save & Close`.

End of procedure

3. Verify the status of the Cassandra cluster

Prerequisite: You have a separate installation of Cassandra 1.0.6 (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

Important

To achieve the best performance with Co-browse, Genesys highly recommends that you configure web sockets support for your load balancer. If web sockets are unavailable, Co-browse still functions, but it uses other transports that perform significantly slower. If your load balancer does not support web sockets and you do not want to wait for Co-browse to automatically switch to another transport, you can use the `disableWebSockets` options for the master and the slave. For more information, see JavaScript Configuration API#disableWebSockets and Slave Configuration Section#disableWebSockets

Important

Due to Safari's strict cookie policy, 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.

Your load balancer configuration will depend upon which load balancer you implement. 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**. It also shows an example of High Availability configuration.
- 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.

Sample 1

Important

This configuration uses a 5 second timeout for High Availability (if a server dies, the load balancer switches the client to another server only after 5 seconds). In production, this timeout can be eliminated using "health checks" functionality, available in Nginx PLUS or via 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

```
# 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_Egor   192.168.73.210:8700;
        .CB_Server_Egor_2 192.168.73.210:8701;
    }
    map $cookie_gcbSessionServer $https_sticky_backend {
        default 0;
        .CB_Server_Egor   192.168.73.210:8743;
        .CB_Server_Egor_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 2

```
# 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.

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 cobrowse_cluster {
 server 192.168.73.95:8700;
 server 192.168.73.95:8701;
 }

 map $cookie_gcbSessionServer $sticky_backend {
 default 0;
 .CB_Server_1 192.168.73.95:8700;
 .CB_Server_2 192.168.73.95:8701;
 }

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

 }

}
```

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.

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 slave and controller configuration

Configure the Co-browse Server applications

Modify the url or secureUrl option in the `cluster` section of all your Co-Browse Server applications. If you use the `secureUrl` option, you must also set the useSecureConnection option to `true` See cluster Section for details.

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

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

Configure the Co-browse Plug-in for Interaction Workspace

Prerequisites: You have installed the Genesys Co-browse Plug-in for Interaction Workspace.

Start of procedure

1. Open Genesys Administrator and navigate to `PROVISIONING > Environment > Applications`.
2. Select the Interaction Workspace Application.
3. In the `Options` tab, add the following options in the `cobrowse` section:
   - url
   - secureUrl
   - useSecureConnection

   For each option in the list above:

   1. Click `New`.
   2. In the `New Option` window, enter `cobrowse` in the `Section` text area and complete the `Name` and `Value` fields.
   3. Click `OK`.
4. Click `Save & Close`.

End of procedure

7. Launch and test

Complete this procedure to launch your servers and validate that load balancing is working correctly.

Start of procedure

1. Start your load balancer.
2. Start the Co-Browse servers.
3. Check the value of the `gcbSessionServer` cookie for the master:
   1. Open a web browser and clear the cookies.
   2. Go to the website where you instrumented Genesys Co-browse and start a Co-browse or Chat session.
   3. Check your cookies and make sure that the value of the `gcbSessionServer` cookie is the name of one of your Co-browse applications.
4. Check the value of the `gcbSessionServer` cookie for the slave:
   1. Open a new browser and clear the cookies.
   2. Open the slave from the load balancer. For example, `http://<load_balancer_IP>:<load_balancer_listening_port>/cobrowse/slave.html`.
   3. Enter the session ID and join.
   4. Make sure that the `gcbSessionServer` cookie is set to the same name and value as the master.

End of procedure

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.

Test with the Co-browse Proxy

Genesys Co-browse includes two development tools, GWM Proxy and ZAProxy, 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 proxies:

- GWM Proxy — This simple proxy server does not have a UI and is based on the Paros proxy. You must restart it to apply the instrumentation changes.
- ZAProxy — The Zed Attack Proxy has a UI and is based on the OWASP Zed Attack Proxy Project.
- Security Testing with ZAProxy — In addition to acting as a proxy, the ZAProxy also provides a UI for validating the vulnerabilities in your website.

GWM Proxy

The GWM Proxy included in the Co-browse Server installation package is a development tool that enables you to test Co-browse without adding the instrumentation JavaScript code snippet to your website.

Complete the procedures below to configure and run the GWM Proxy.

Configure GWM Proxy Host and Port

Start

1. Navigate to `C:\Users\<current user>\GWMProxy`. If this folder does not exist, this means that the proxy server has never been started. Navigate to `<Co-browse_installation>\tools\proxy` and start the server with `startserver.bat`. This will automatically create the `GWMProxy` folder.
2. In the `GWMProxy` folder, open the `config.xml` file for editing.
3. Find the `<proxy>` tag and make sure the value of its child `<ip>` tag is set to your host IP address.

> **Important**
> This value must **not** be 127.0.0.1 or localhost.

4. Note the value of the `<port>` tag (usually 15001) — you will use this to Set up your Web Browser.

```
<proxy>
        <ip>192.168.44.83</ip>
        <port>15001</port>
        <reverseProxy>
        ...
 </proxy>
```

End

Instrument the Site Through the Proxy

Set the Domain Names

Start

1. Navigate to `<co-browse installation>\tools\proxy\` and open the `map.xml` file for editing.
2. In the `<map>` tag, set the value of the `domains` attribute to the domains of the site you would like to use

the proxy. Separate each value in this list with a semi-colon.

```
<?xml version="1.0"?>
  <mapping>
         <map replace="..."
domains="www.genesys.com;genesys.com;programs.genesys.com;content.genesys.com">
```

End

Adjust the Instrumentation Script

You can modify the default script included in the **map.xml** file according to your needs:

- update the host and port for Co-browse Server
- add JS Configuration API objects

> **Important**
> Do not forget to restart GWM Proxy after making modifications to the **map.xml** file so that your changes will take effect.

Start the Proxy

Navigate to your Co-browse installation directory and then launch `tools\proxy\startserver.bat`. The GWM Proxy starts.

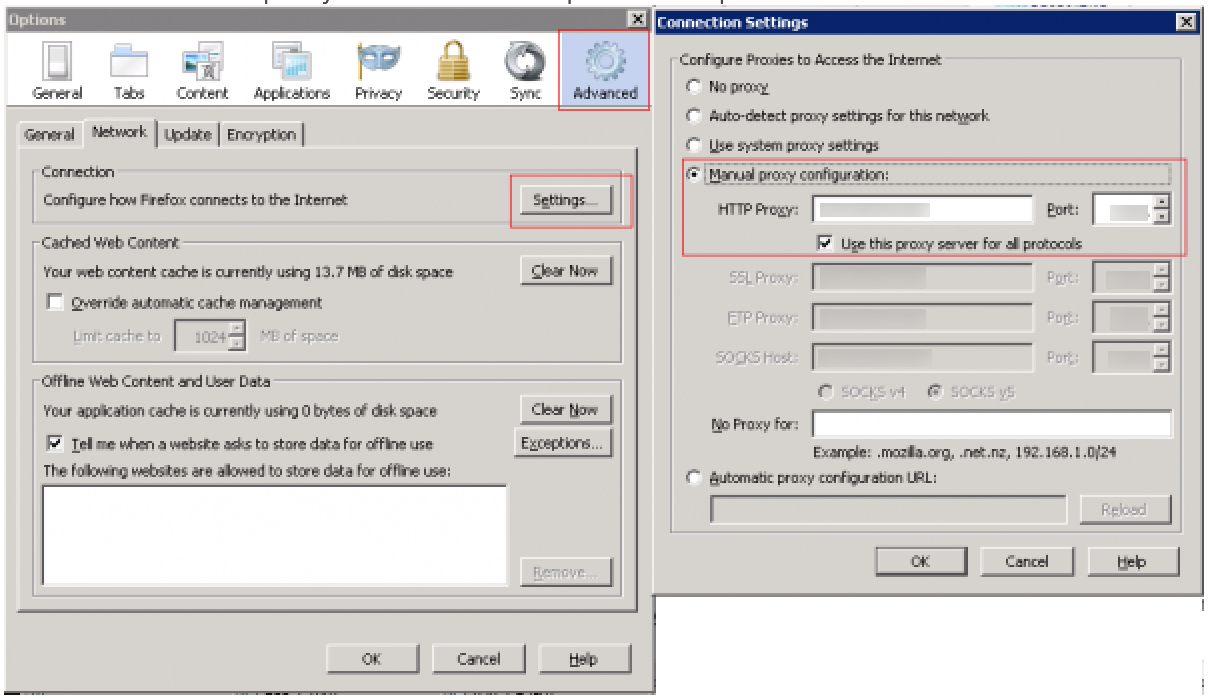
> **Important**
> The GWM Proxy included in the Co-browse installation package cannot work if the full path contains spaces.

Set up your Web Browser

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 GWM Proxy in the Port text box. This is the port you made note of in Configure GWM Proxy 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:

GWM Proxy 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 GWM Proxy, which will inject the Co-browse JavaScript code into the web pages of the monitored domain you specified in Set the Domain Names.

End

ZAProxy

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

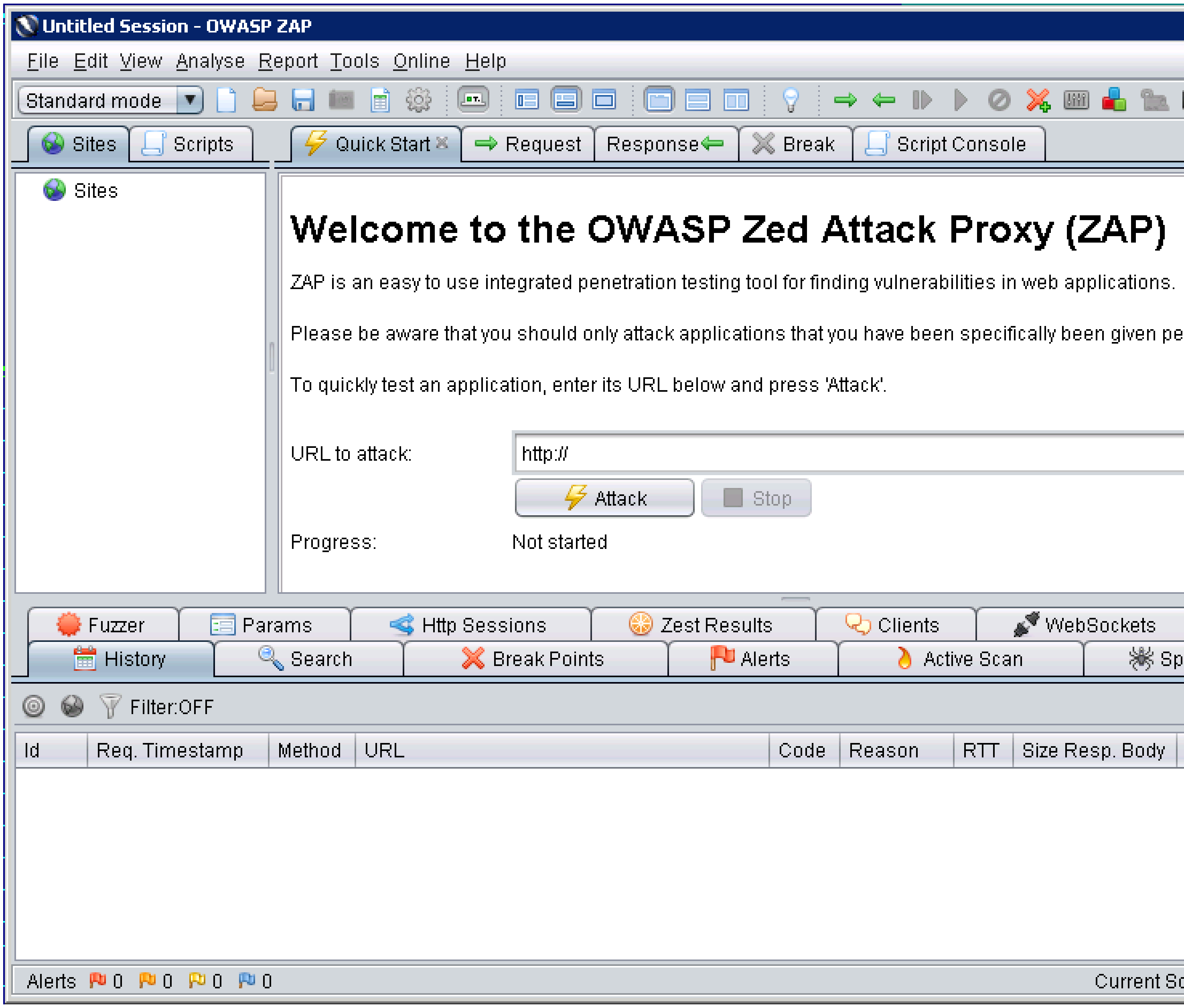
Complete the procedures below to configure and run the ZAProxy.

Important
While Genesys Co-browse requires a minimum of Java version 1.6, 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.


Untitled Session - OWASP ZAP
File Edit View Analyse Report Tools Online Help
Standard mode
Sites Scripts
Quick Start Request Response Break Script Console
Sites

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://

Attack Stop

Progress: Not started

Fuzzer Params Http Sessions Zest Results Clients WebSockets
History Search Break Points Alerts Active Scan Sp
Filter:OFF
Id Req. Timestamp Method URL Code Reason RTT Size Resp. Body
Alerts 0 0 0 0 Current Sc

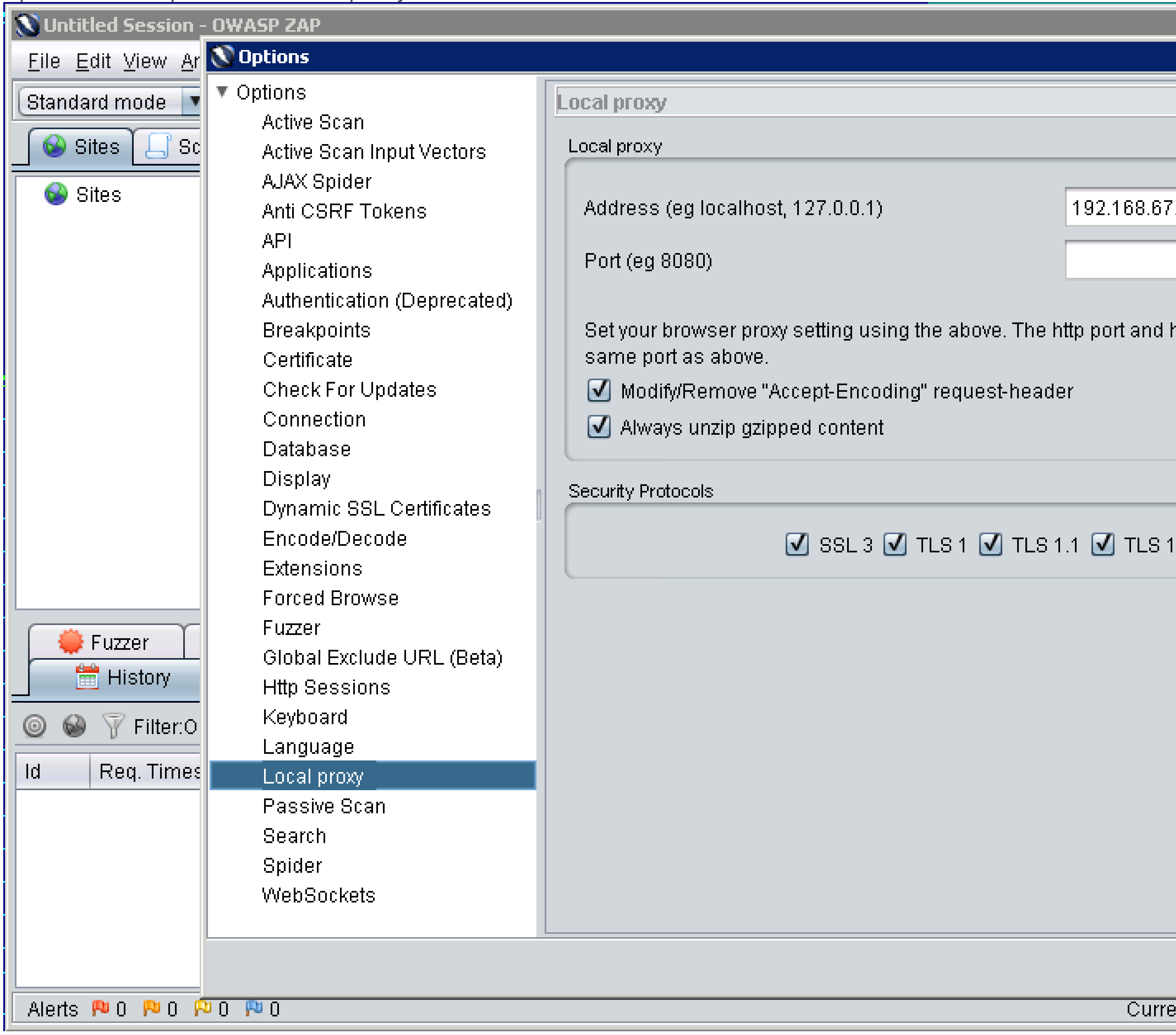
Stop the Proxy

To stop the ZAProxy, simply close the UI window.

Configure ZAProxy Host and Port

Start

1. Open `Tools > Options > 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

☑ 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.

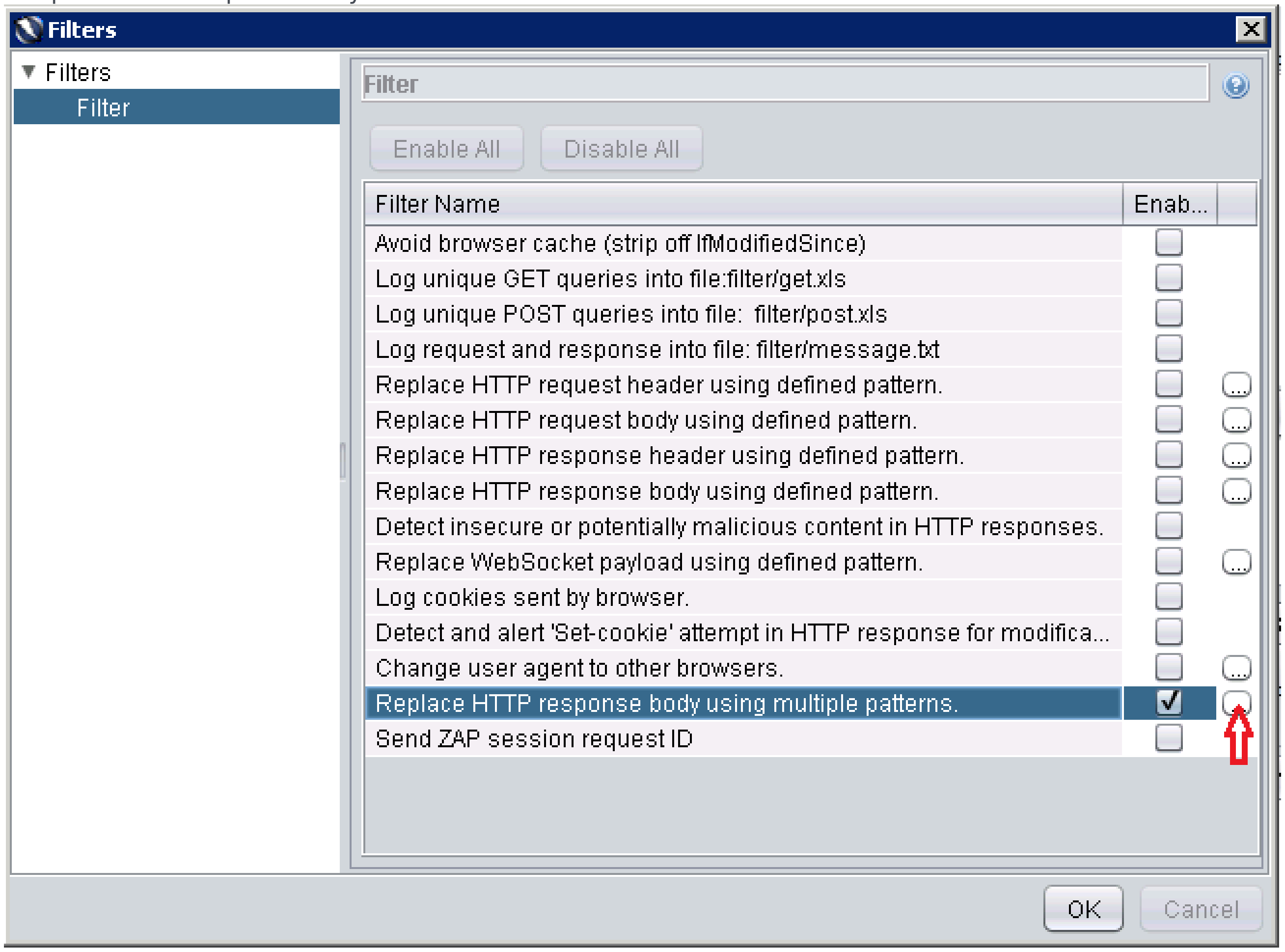
End

Update the Instrumentation Script

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

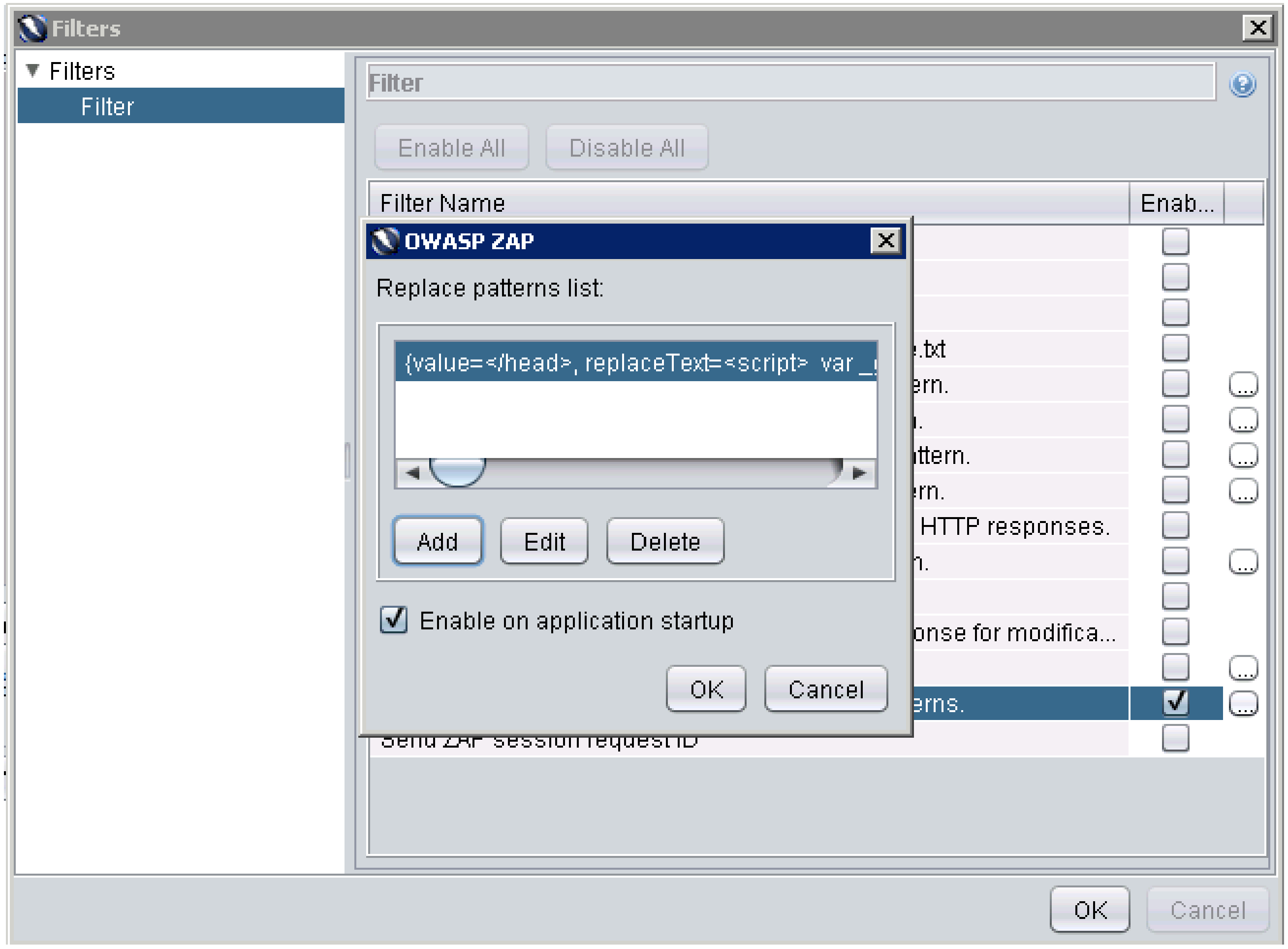
Start

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.

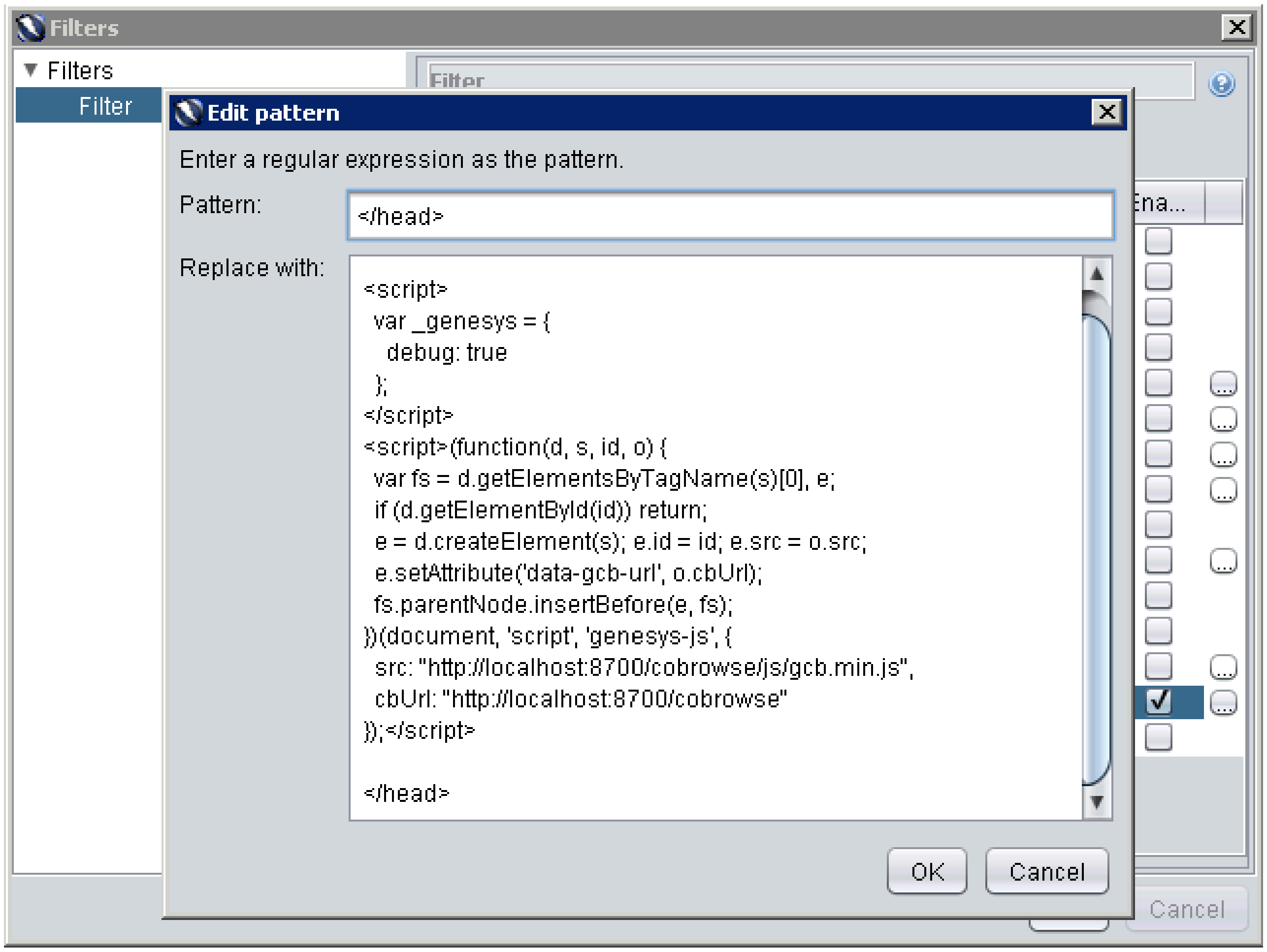


Filter Name	Enab...	
Avoid browser cache (strip off IfModifiedSince)	☐	
Log unique GET queries into file:filter/get.xls	☐	
Log unique POST queries into file: filter/post.xls	☐	
Log request and response into file: filter/message.txt	☐	
Replace HTTP request header using defined pattern.	☐	...
Replace HTTP 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`.



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>
```

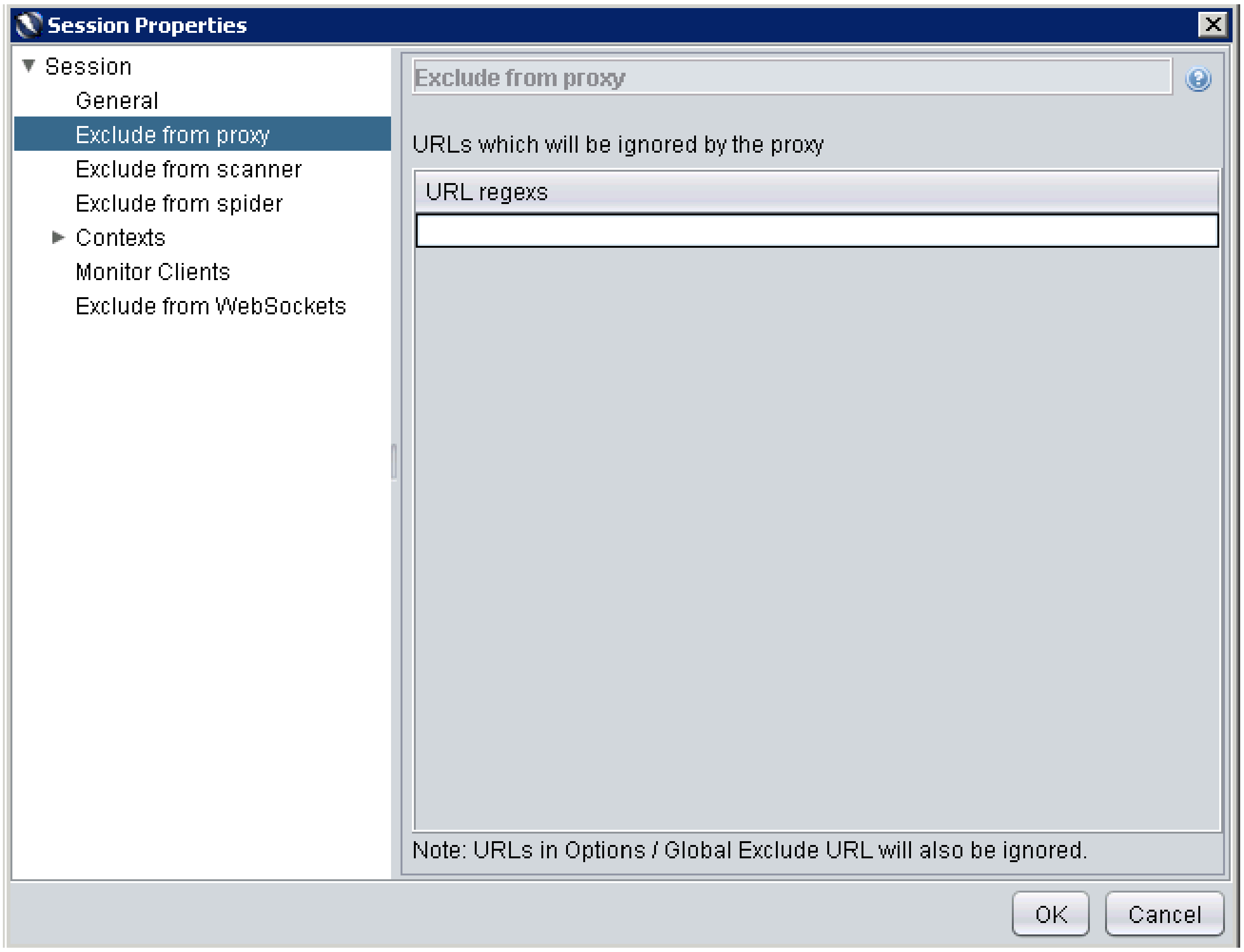
4. To save the changes, click OK on the current dialog and on the two parent dialogs.

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.

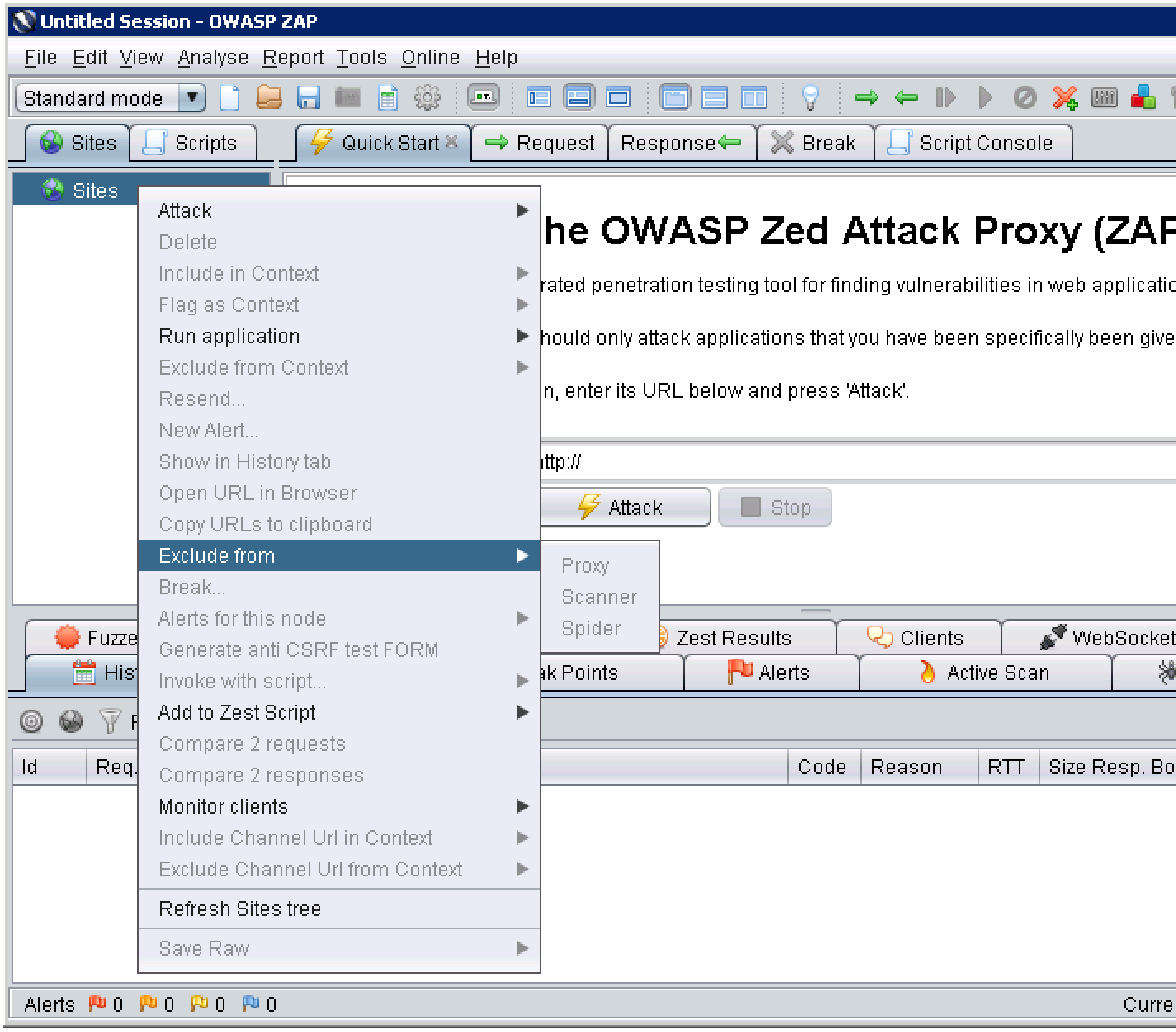


URLs which will be ignored by the proxy

URL regexs

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`.



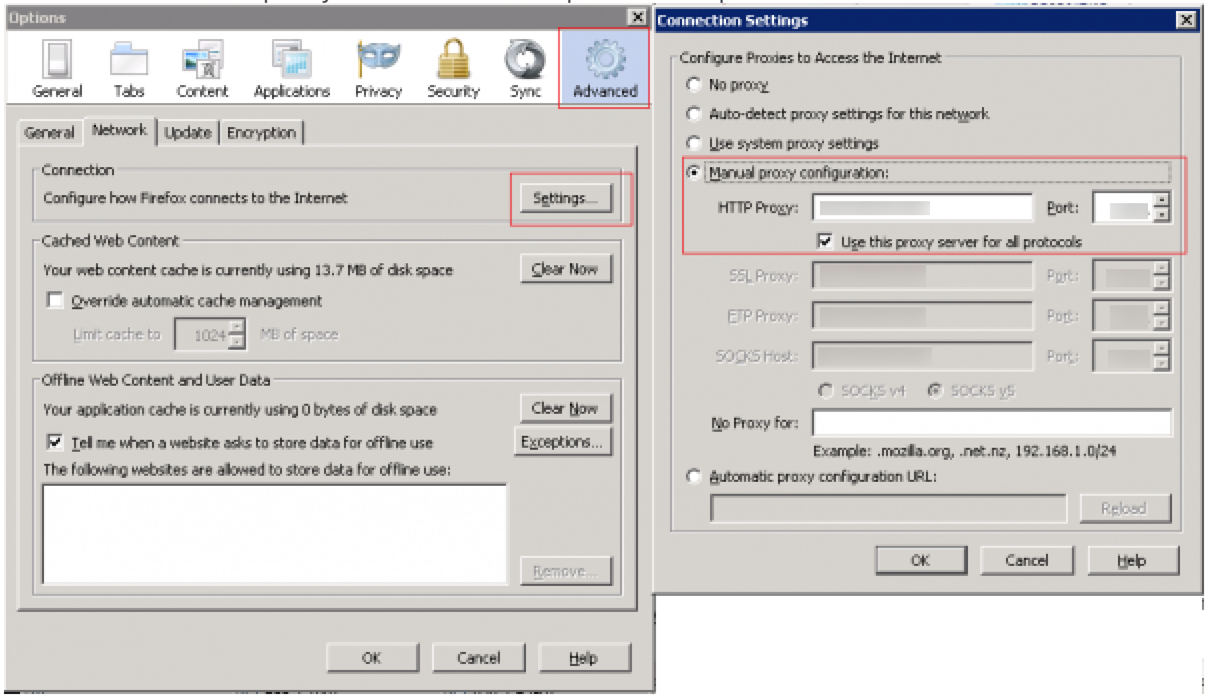
If you want the proxy to remember the excluded URLs beyond the current session, select `File > Persist session...` and select a file to save your session.

Set up your Web Browser

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.


Options
General
Tabs
Content
Applications
Privacy
Security
Sync
Advanced
General
Network
Update
Encryption
Connection
Configure how Firefox connects to the Internet
Settings...
Cached Web Content
Your web content cache is currently using 13.7 MB of disk space
Clear Now
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
Clear Now
Tell me when a website asks to store data for offline use
Exceptions...
The following websites are allowed to store data for offline use:
Remove...
OK
Cancel
Help
Connection Settings
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:
Reload
OK
Cancel
Help
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

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.

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.

References

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

Genesys Co-browse Reporting Templates

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
Review the available templates for Real-time Reporting and Historical Reporting.

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-browseInteractionsExt.xtpl` — see the user data (Co-browse session ID, Co-browse start time, Co-browse end 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*

<Object>	Statistic	Values
	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*
	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*

<Object>	Statistic	Values
		(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*
	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*

<Object>	Statistic	Values
	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-browseInteractionsExt.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.

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

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

Historical Reporting

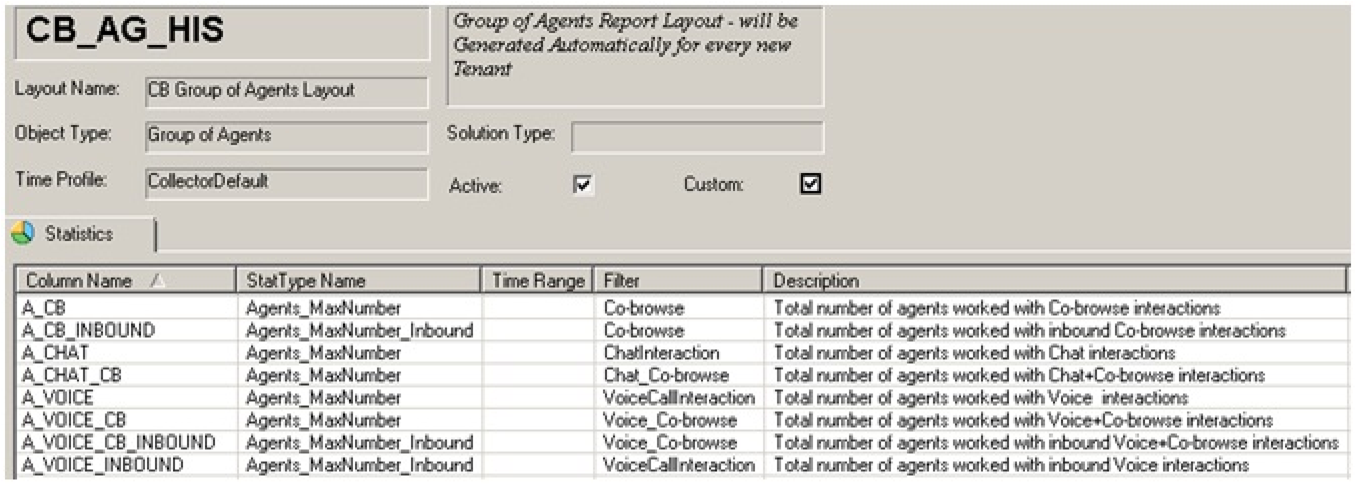
Genesys Co-browse includes the following historical reporting templates that allow you to:

- `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

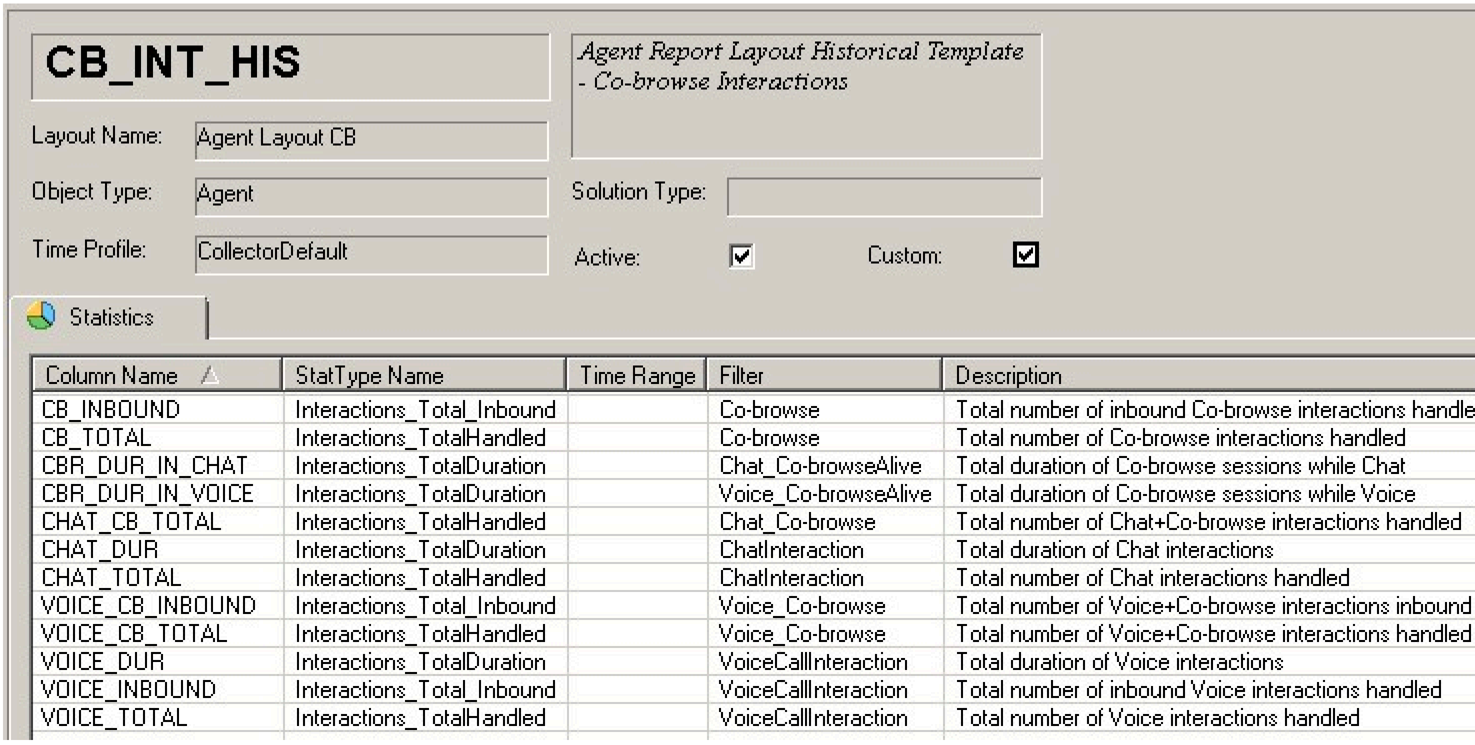
Layout Name:	CB Group of Agents Layout
Object Type:	Group of Agents
Time Profile:	CollectorDefault
Solution Type:	
Active:	✓
Custom:	✓

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

Time Profile: CollectorDefault

Solution Type:

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*

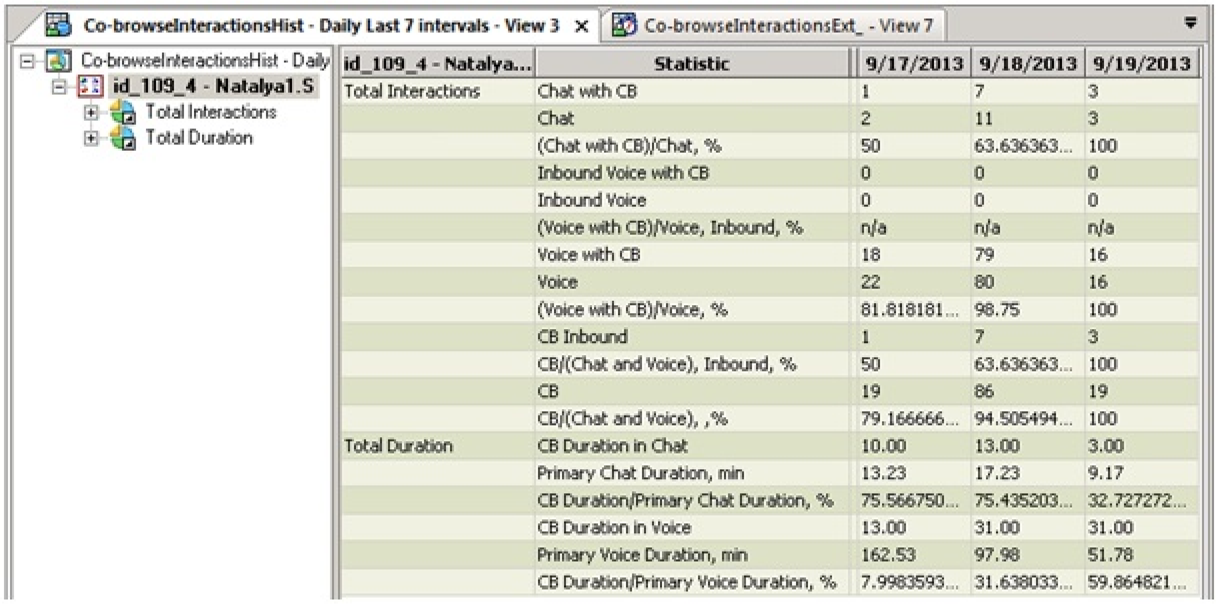
Co-browseInteractionsHist.xtpl

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*
	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*

<Object>	Statistic	Values
	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
cbdb *Configure Cassandra to support the Co-browse Server cluster*	cassandraClusterName cassandraYamlTemplateUrl embeddedCassandra keyspaceName listenAddress nativeTransportPort replicationFactor rpcAddress rpcPort seedNodes sslStoragePort storageDirectory storagePort strategyName
cbdb-retention-policy *Configure the data retention policy for Cassandra*	livesessionentity sessionhistoryentity windowhistoryentity
cross-origin *Configure list of websites allowed to access the Co-browse server*	allowedOrigins
chat *Settings for chat*	connectionTimeout queueKey useChat refreshTaskPeriod refreshPoolSize sessionRestorationTimeout

Section Name	Options
cluster *Configure the Co-browse Server cluster*	url useSecureConnection secureUrl
cometd *Settings for CometD*	logLevel maxInterval
forward-proxy *Configure a forward proxy*	host port user password
log *Configure the logs generated by the Co-browse Server*	all buffering expire segment time_convert time_format trace verbose
security *Enable TLS on connections with other Genesys servers*	trusted-ca-type trusted-ca trusted-ca-pwd
session *Configure DOM restrictions*	domRestrictionsURL inactivityDuration
slave *Configure localization for the slave UI*	localization cssPatchUrl theme disableWebSockets
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 Interaction Workspace*	url secureUrl useSecureConnection disableCertificateValidation useBrowserLogging

cbdb Section

cassandraClusterName

Default Value: `Cluster`
Valid Values: Any string
Changes Take Effect: Cannot be changed without removing the Cassandra system files.

Specifies the name of the embedded Cassandra cluster node. This is mainly used to prevent machines in one logical cluster from joining another. For more information, see http://www.datastax.com/docs/1.0/configuration/node_configuration#cluster-name.

cassandraYamlTemplateUrl

Default Value: None
Valid Values: Path to an existing `cassandra.yaml` template file.
Changes Take Effect: After restart

Specifies the path to the `cassandra.yaml` template file in the URI format. For example, `file:///E:/CB_Server/server/etc/cassandra.yaml`. If not specified, the default `cassandra.yaml` template file is taken from `[jetty-home]/etc/cassandra.yaml`. The method of filling the `cassandra.yaml` template is flexible and can be extended by the administrator. Placeholders names between `$` signs (such as `$myValuePlaceholder$`) correspond to option names in the `cbdb` section of the Co-browse Server configuration options, so the custom option value replaces the placholder. For more information about the `cassandra.yaml` template file, see http://www.datastax.com/docs/1.0/configuration/node_configuration.

Important
This option is applicable only when the value of embeddedCassandra is `true`.

embeddedCassandra

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

Specifies whether Co-browse Server acts as a Cassandra cluster node. If `true`, the embedded Cassandra service is activated. If `false`, Co-browse Server acts only as a Cassandra client.

keyspaceName

Default Value: `Cobrowse`
Valid Values: Any string
Changes Take Effect: When the keyspace is created

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

listenAddress

Default Value: No default (placed in cassandra.yaml template as it is.)
Valid Values: An IP address
Changes Take Effect: After restart

Specifies the address to bind to and tell the other Cassandra nodes to connect to. You must set this value if you want multiple nodes to be able to communicate. See [1] for details. If you leave the value blank, this will leave it up to InetAddress.getLocalHost(). This will always do the "right thing" if the node is properly configured (hostname, name resolution, and so on) and the "right thing" is to use the address associated with the hostname.

nativeTransportPort

Default Value: 19042
Valid Values: Any free TCP port
Changes Take Effect: After restart
Specifies the port on which CQL native transport listens for clients.

replicationFactor

Default Value: 1
Valid Values: An integer value less than the number of nodes in the Cassandra cluster
Changes Take Effect: When the keyspace is created

Specifies total number of replicas across the cluster. For more information, see http://www.datastax.com/docs/1.0/cluster_architecture/replication.

rpcAddress

Default Value: `localhost`
Valid Values: An IP address, a hostname, or leave unset to resolve the address using the hostname configuration of the node.
Changes Take Effect: After restart

Specifies the listen address for remote procedure calls (client connections). For more information, see http://www.datastax.com/docs/1.0/configuration/node_configuration#rpc-address. To listen on all configured interfaces, set the value to `0.0.0.0`. If this option is empty, you must have hostname resolution correctly configured on all nodes in your cluster so that the hostname resolves to the correct IP address for this node (using `/etc/hostname`, `/etc/hosts` or DNS). This option is also used to configure Co-browse Server as a Cassandra client.

rpcPort

Default Value: `29160`
Valid Values: Any free TCP port
Changes Take Effect: After restart

Specifies the port for remote procedure calls (client connections) and the Thrift service. For more information, see http://www.datastax.com/docs/1.0/configuration/node_configuration#rpc-port.

seedNodes

Default Value: `127.0.0.1`
Valid Values: A comma-delimited list of IP addresses
Changes Take Effect: After restart

Specifies the comma-delimited list of IP addresses representing the list of seeds. When a node joins a cluster, it contacts the seed node(s) to determine the ring topology and obtain gossip information about the other nodes in the cluster. Every node in the cluster should have the same list of seeds. 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://www.datastax.com/docs/1.0/configuration/node_configuration#seeds.

Important
This option is applicable only when the value of embeddedCassandra is `true`.

sslStoragePort

Default Value: 17101
Valid Values: Any free TCP port
Changes Take Effect: After restart
Specifies the SSL port for encrypted communication.

storageDirectory

Default Value: None
Valid Values: A valid directory path. This directory may not exist.
Changes Take Effect: After restart

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

Important
This option is applicable only when the value of embeddedCassandra is `true`.

storagePort

Default Value: 17100
Valid Values: Any free TCP port
Changes Take Effect: After restart
Specifies the TCP port for commands and data.

strategyName

Default Value: None
Valid Values: `SimpleStrategy` or `NetworkTopologyStrategy`
Changes Take Effect: When the keyspace is created

Specifies the keyspace replica placement strategy. For more information, see http://www.datastax.com/docs/1.0/cluster_architecture/replication#replication-strategy.

cbdb-retention-policy Section

Important

The options in the cbdb-retention-policy section manage the retention policy for the Co-browse column families stored in Cassandra. Each option corresponds to one column family (the names do not match one-to-one). If any of the options are omitted, Co-browse Server instructs Cassandra to keep data for 14 days.

livesessionentity

Default Value: `86400` (1 day)
Valid Values: Positive integer greater than number of seconds set in the maxInterval option.
Changes Take Effect: After restart

Specifies the how long, in seconds, to keep data in the `live_sessions` column family. This column family stores the core Co-browse session state shared in the Co-browse cluster. In normal situations, data from this column family is removed automatically shortly after a session is deactivated (when maxInterval elapses), but the retention policy guarantees that data will always be removed.

sessionhistoryentity

Default Value: `1209600` (14 days)
Valid Values: Positive integer greater than number of seconds set in the maxInterval option.
Changes Take Effect: After restart

Specifies the how long, in seconds, to keep data in the `session_history` column family. This column family stores session historical data that is accessible through the REST API.

windowhistoryentity

Default Vlue: `86400` (1 day)
Valid Values: Positive integer greater than number of seconds set in the maxInterval option.
Changes Take Effect: After restart

Specifies how long, in seconds, to keep data in the `window_history` column family. This column family is a temporary store for page navigation information. In normal situations, data from this column family is removed automatically when a session is deactivated, but the retention policy guarantees that data will always be removed.

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.

chat Section

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: `true`
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 master 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

Default Value: None
Valid Values: Valid HTTP URL
Changes Take Effect: After restart

Specifies the HTTP URL (such as `http://[host]:[port]/cobrowse`) of the Co-browse cluster. Typically, this value is the URL for the load balancer. If not specified, Co-browse server assumes a single node configuration and takes the host and port from the Server Info section of the Co-browse Server application.

secureUrl

Default Value: None
Valid Values: Valid HTTPS URL
Changes Take Effect: After restart

Specifies the HTTPS URL (such as `https://[host]:[port]/cobrowse`) of the Co-browse cluster. Typically, this value is the URL for the load balancer. If not specified, Co-browse server assumes a single node configuration and takes the host and port from the Server Info section of the Co-browse Server application.

> **Important**
> If the value of `useSecureConnection` is `false`, the value of the `secureUrl` option will not be used.

useSecureConnection

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

Specifies the connection type for REST and CometD connections from the slave and controller (agent desktop) to Co-browse Server. If `true`, Co-browse Server uses HTTPS; if `false`, the server uses HTTP. The URL and protocol for the master browser are managed through website instrumentation. For more information, see Website Instrumentation.

> **Important**
> If the value of `useSecureConnection` is `false`, the value of the `secureUrl` option will not be used.

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 master or slave 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.

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.
`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 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`

buffering

Default Value: `true`
Valid Values:

`true`	Enables buffering.
`false`	Disables buffering.

Changes Take Effect: Immediately

Turns on/off operating system file buffering. The option is applicable only to the `stderr` and `stdout` output. Setting this option to `true` increases the output performance.

Warning

When buffering is enabled, there might be a delay before log messages appear at the console.

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.

security Section

trusted-ca-type

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.

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.

trusted-ca-pwd

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

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

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.

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. By default, the built-in English localization is used. For more information about localization, see Localization—Localizing the slave 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 slave representation of the page seen by the user. May be used to solve CSS synchronization issues.

theme

Default value: `iws`
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 slave 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 master and more details.

static-web-resources Section

browserHardCacheDuration

Default value: `1800`
Valid Values: Positive integer or zero
Changes take effect: Immediately
Specifies the duration (in seconds) of hard caching. If the value is `0`, hard caching is not applied (the headers are not set).

cobrowse Section

Important

The options below provide configuration for the Co-browse Plug-in for Interaction Workspace. 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 URL
Changes Take Effect: After Interaction Workspace restart

Specifies the HTTP URL (such as http://[host]:[port]/cobrowse) of the Co-browse cluster. Typically, this value is the URL for the load balancer.

secureUrl

Default Value: None
Valid Values: Valid HTTPS URL
Changes Take Effect: After Interaction Workspace restart

Specifies the HTTPS URL (such as https://[host]:[port]/cobrowse) of the Co-browse cluster. Typically, this value is the URL for the load balancer.

useSecureConnection

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

Specifies the connection type for the connection between the Interaction Workspace and the Co-browse Server.

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 (slave web UI) logs to redirect into the Interaction Workspace log.

Testing and Troubleshooting the Co-browse Solution

Use the following procedures to test that a Genesys Co-browse solution is configured correctly.

> **Tip**
> Also see Restrictions and Known Limitations.

Testing Co-browse Without Chat

Required Components

The following components are the minimum required to test Co-browse without chat: 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 Master 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 Master 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 Master browser console log in your instrumentation script. For details, see Enabling console logs.

Testing Co-browse Session

Opening the Slave Page

Prerequisites: You have successfully completed Preparing for Testing.

Start of Procedure

1. Open the Slave 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 Slave 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 Slave are incorrectly specified in the `slave` section of the Co-browse Server application. For details about localization settings, see Localizing the slave UI.
3. The network has a problem.
4. If it is not clear what the problem is, enable debug logging on the Slave 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 Slave browser, insert debug=1 into the Slave 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 Slave Page.

Start of procedure

1. In the Master 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 Master 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 Slave to the Co-browse Session

Joining the Slave to the Co-browse Session

Prerequisites: There are no problems when Opening the Slave Page and Starting a Co-browse Session.

Start of procedure

1. Copy the Session ID from the Master page and paste it into the edit box on the Slave page.
2. Join the session.

End of procedure

Successful result: The Slave user successfully joins the session.

Problem: The Slave 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.

Problem: The Slave user cannot join the session and sees the message, `Session ID is invalid or has expired`.

Possible causes of problem:

1. The **slave.html** and **data-gcb-url** addresses (see Instrumentation) are different. Make sure the addresses are correct and correspond to each other.
2. You have a case where configuration options in the cluster section are not specified. There might be a difference between the **slave.html** address, **data-gcb-url** address and **<host>:<port>** in the Co-browse Server configuration. Make sure each is correct and corresponds to each other.
3. You have a case where configuration options from the cluster section are specified but incorrect. Make sure the options are correct and correspond to the **data-gcb-url** and **slave.html** addresses.

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: 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
- Interaction Workspace (IWS)/Workspace Desktop Edition (WDE)
- Co-browse Plug-in for IWS or 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 Master 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 Master page, click `Co-browsing`.

End of Procedure

Successful Results:

- The Co-browse session starts.
- The Session ID appears on the Master 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 Master page.

Possible causes of problem 1:

1. It could be an intranet problem if the Master 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.

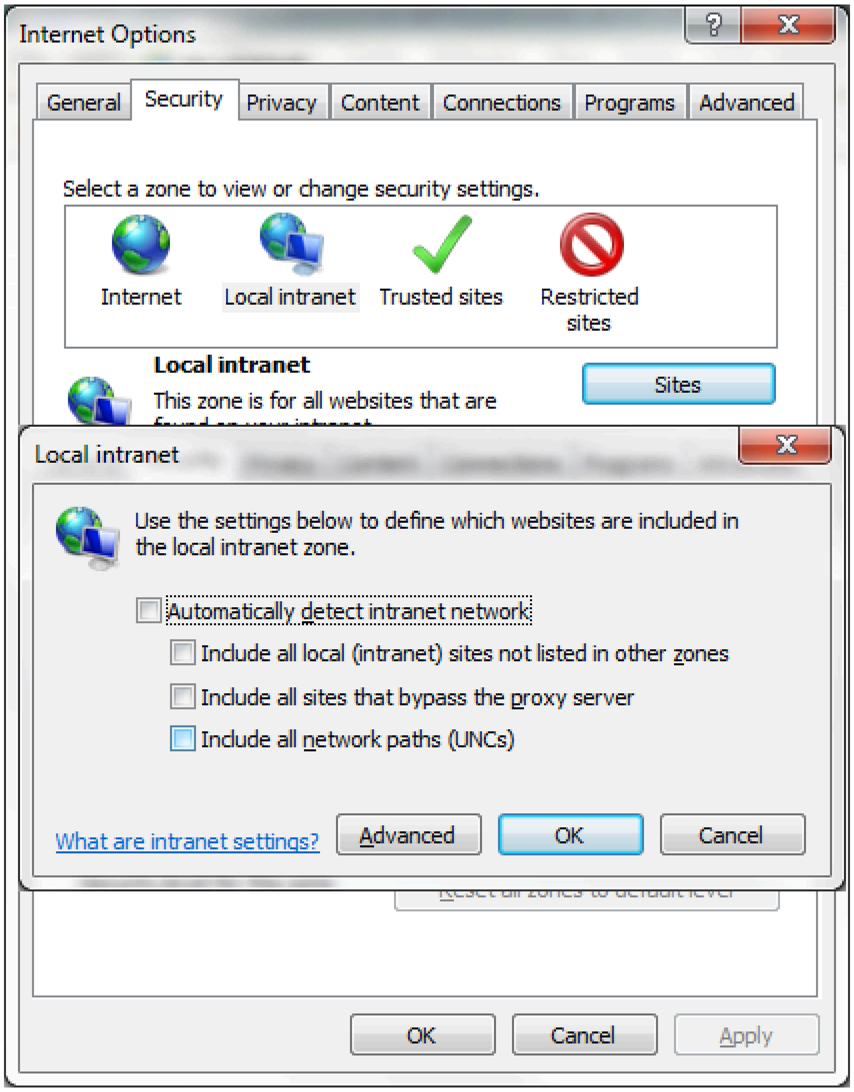
Important

To enable the debug log in the Slave 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(co, b, r, o, w, s, e) {
  e = co.getElementsByTagName(b)[0];
  if (co.getElementById(r)) return;
  s = co.createElement(b); s.id = r; s.src = o;
  s.setAttribute('data-gcb-url', w);
  e.parentNode.insertBefore(s, e);
})(document, 'script', 'gcb-js',
 'http://192.0.2.10:8700/cobrowse/js/masterApp-build.js',
 '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 slave (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 slave browser. Make sure that requests from the slave 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 slave 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 slave browser. The slave user may end up with something visually different than the master 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 slave representation by providing additional CSS using the `cssPatchUrl` server configuration option.

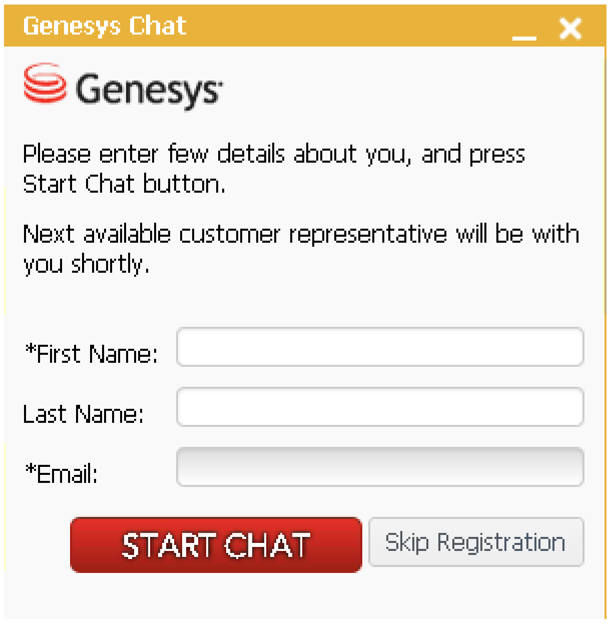
Troubleshooting Chat Widget Rendering Issues

Prerequisites:

- Built in chat widget uses embedded mode.

Symptoms:

Chat widget renders incorrectly or has unexpected styling. For example:


Genesys Chat

Genesys

Please enter few details about you, and press Start Chat button.

Next available customer representative will be with you shortly.

*First Name:

Last Name:

*Email:

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

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—Co-browse requires a valid doctype. Pages in quirks 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
- SVG
- HTML5 audio and video—HTML markup is synchronized. Synchronization of playing, pausing, etc. is not supported.

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.

Synchronization of Scrolling Position for HTML Elements is Not Supported

Important

Scrolling of main window and instrumented iframes **is** supported.

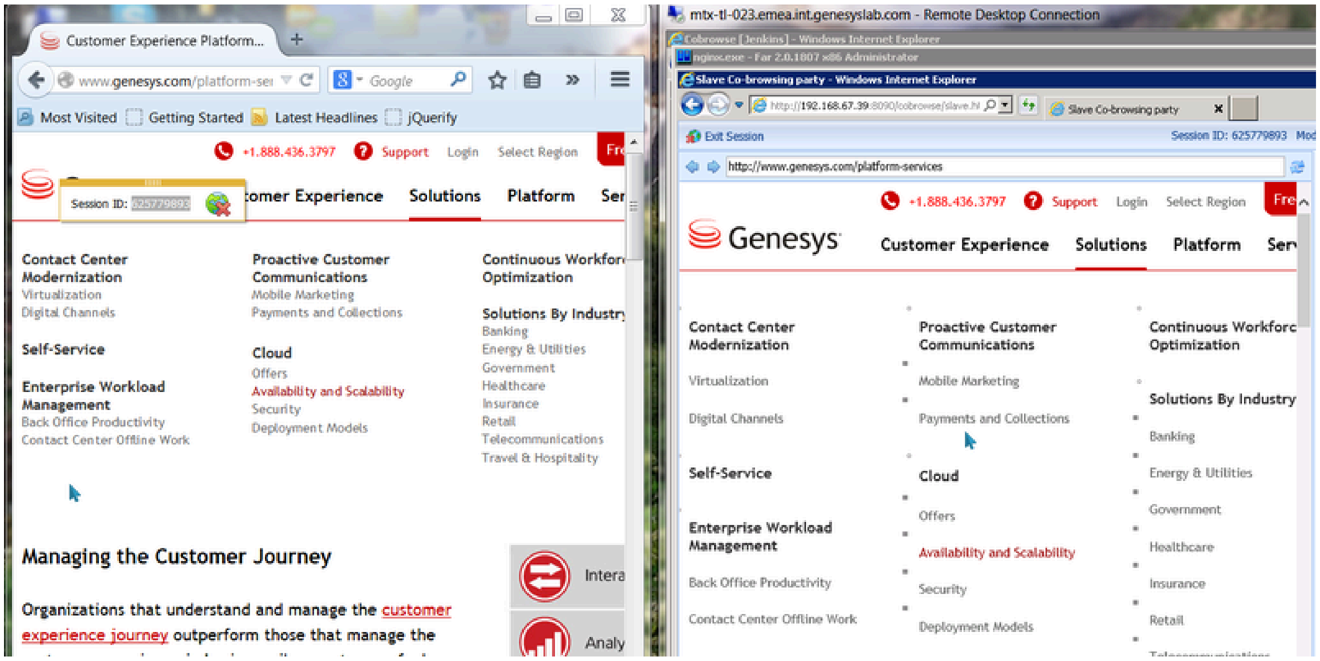
Customer Representative Can Handle Only One Co-Browse Session at a Time

Due to high interactivity, it is not possible for a customer representative to work in two Co-browse sessions with different users. This limitation is related to load balancing and session stickiness based on cookies. In addition to the technical limitation, it would be difficult for a customer representative to handle more than one Co-browse session at the same time.

Representation of Dynamically Shown List Items May be Partially Broken on Slave Browsers Using IE10 or IE11

Issues may arise when IE10/11 is used as a slave browser on a website with dynamically shown/ hidden sub-menus.

Example:


Customer Experience Platform...
www.genesys.com/platform-se
Most Visited
Getting Started
Latest Headlines
jQuerify
+1.888.436.3797
Support
Login
Select Region
Session ID: 625779893
Customer Experience
Solutions
Platform
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
mtx-tl-023.emea.int.genesyslab.com - Remote Desktop Connection
Slave Co-browsing party - Windows Internet Explorer
Exit Session
Session ID: 625779893
http://www.genesys.com/platform-services
Genesys

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 slave. In this case, CSS synchronization issues are possible if the site visitor (master user) uses IE9.

Security

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:
  - **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
    - *Edit the connection and set the ID to the HTTPS listening port* in Configuring Connection to Co-browse Server for Interaction Workspace
  - **HTTPS connection for Co-browse cluster**—A Co-browse Server application supports both HTTP and HTTPS connections for Co-browse cluster. Related documentation:
    - `secureUrl` and `useSecureConnection` options in the cluster section of the Co-browse Server application configuration.
    - `secureUrl` and `useSecureConnection` options 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.
- **Secure Sockets Layer (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. For a production environment, you should use a certificate issued by a third-party Certificate Authority.
  Related documentation: Load SSL certificates and configure Jetty.
- **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
- **Static resources proxying**—Co-browse server proxies some static assets of your website like CSS, images, and fonts. This is generally safe since your website is the only source of these assets in a Co-browse session.

Important

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