10 Cobrowsing

Co-browsing allows your agents to establish shared web browsing sessions with your web-site visitors. Before accurate co-browsing can be accomplished, there are a number of configuration and setup procedures that must be completed. Additionally, a review of certain aspects of your existing website may be required.

To manage your co-browse deployment, select the **Deploy : Co-browsing** menu option.

Understanding How Cobrowse Works

Velaro's co-browse server operates by placing what is technically referred to as a "reverse proxy server" between your web server and your website visitors. While a traditional reverse proxy server is simply responsible for forwarding and managing one-to-one requests from your website visitors, Velaro's server has many enhanced capabilities which allow for multiple visitors to share the same co-browse session at the same time.

By implementing co-browse in the above manner, Velaro's services are much more robust and "lightweight" than traditional web-collaboration services. For example:

- Many co-browse applications do not work if there is any private session information. For example, if your customers log in to a private area of your website, or they are placing items in a shopping cart, this information is typically not shared. Velaro's implementation works perfectly under both scenarios.
- Other co-browse applications implement "heavier" collaboration techniques, such as requiring each participant to download and install a Java application or applet. With today's ever increasing security standards, and the fact that the Java runtime is no longer installed on most operating systems by default, this can become problematic. Velaro's co-browse is "applet free" and works with almost all of today's browsers, right out of the box, with no additional add-ons.

To get a basic understanding of how Velaro's co-browse works, lets look at the following flow chart of a typical web interaction between a web browser and a web server:

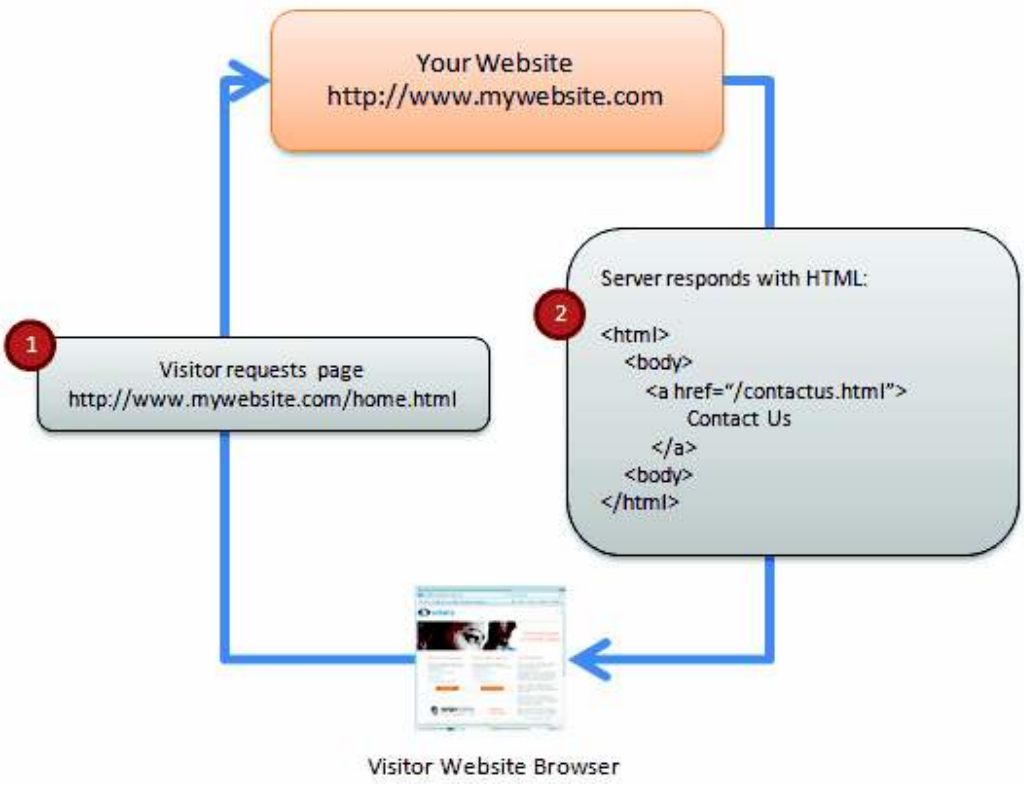


Figure 1: Standard Browsing Session

Step 1: The browser initiates a request to your web server and asks for a specific web page.

Step 2: The web server responds by sending the HTML contents of the requested page back to the visitor.

When you are cobrowsing a web site via Velaro's co-browse server, the transaction adds a few extra steps. As in Figure 2:

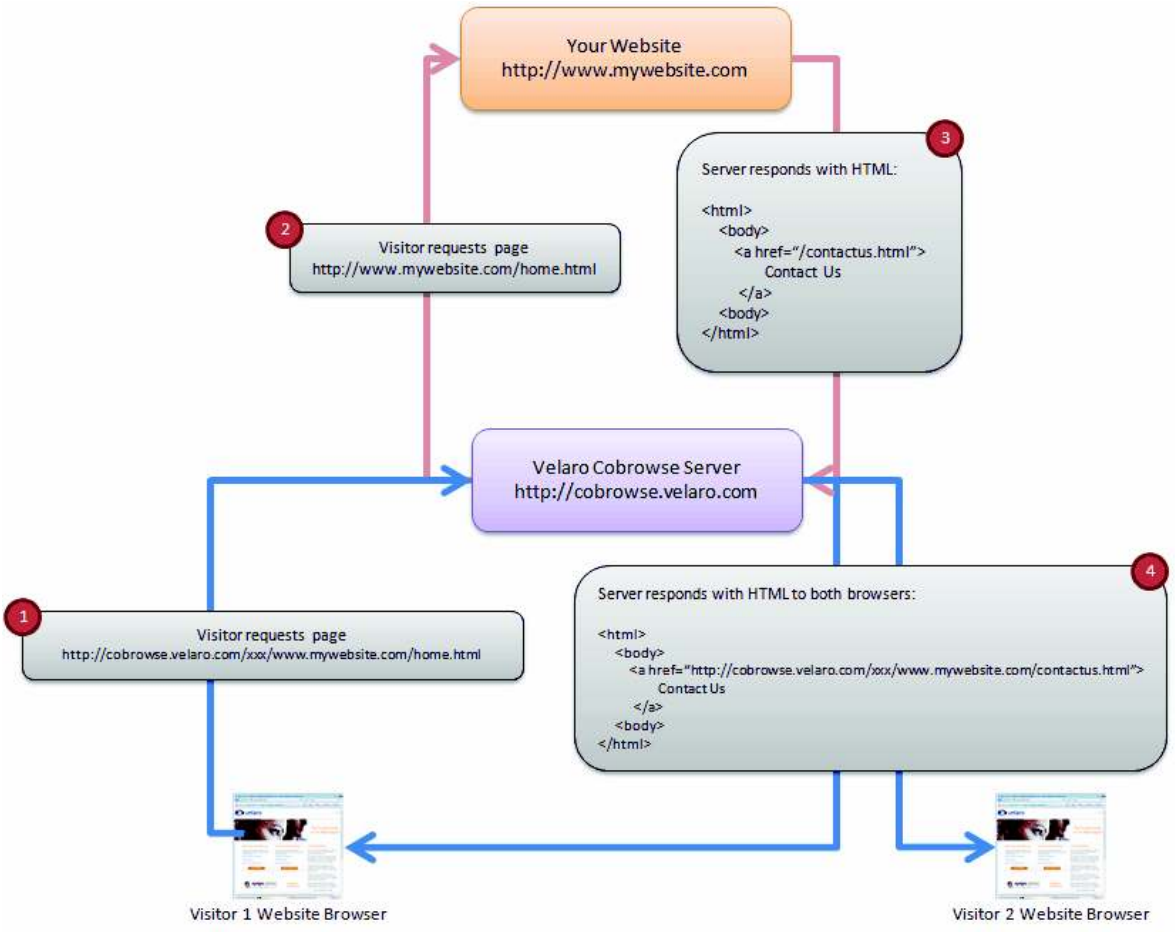


Figure 2: Velaro Cobrowse Session with two browsers

The following steps assume that a co-browse session has already been initiated:

Step 1: Visitor 1 requests a web page. However, since the session is being managed by Velaro's co-browse server, the actual URL they are trying to load has been modified. Rather than asking for http://www.mywebsite.com/home.html, they are requesting a modified URL which points to Velaro's co-browse server (http://cobrowse.velaro.com/xxx/www.mywebsite.com/home.html).

Step 2: Velaro's co-browse server receives this request from step 1, and makes the actual request to your website for the proper page, http://www.mywebsite.com/home.html

Step 3: Your web server responds to Velaro's co-browse server and sends the proper page and HTML back to the co-browse server.

Step 4: Velaro's co-browse server analyzes the response from your webserver. During this analysis, the co-browse server's primary responsibility is to interrogate the text that was returned, look for all references within the HTML that contains links and URL's back to your website server. Once the co-browse server identifies these URL's, it converts them in to URL's that tell the web browser to respond back to the co-browse server. The examples noted in the above flowcharts are as follows:

In Figure 1, the HTML response of the document request contains an anchor tag <A> with a link back to the contactus.html web page. In Figure 1 this can be seen in step 2, as Contact Us

In Figure 2, your web server is providing the same response as in Figure 1, however, as demonstrated in step 4, the co-browse server successfully identified this link and converted it to one that will respond properly through the Velaro co-browse server, rather than going directly to your web server. The resulting link is Contact Us

The co-browse server then ensures that all web browsers that are connected to this session (typically your agent and the visitor) receive the resulting page.

Other Responsibilities

The co-browse server does a lot more than simple link conversion as demonstrated above. It is also responsible for:

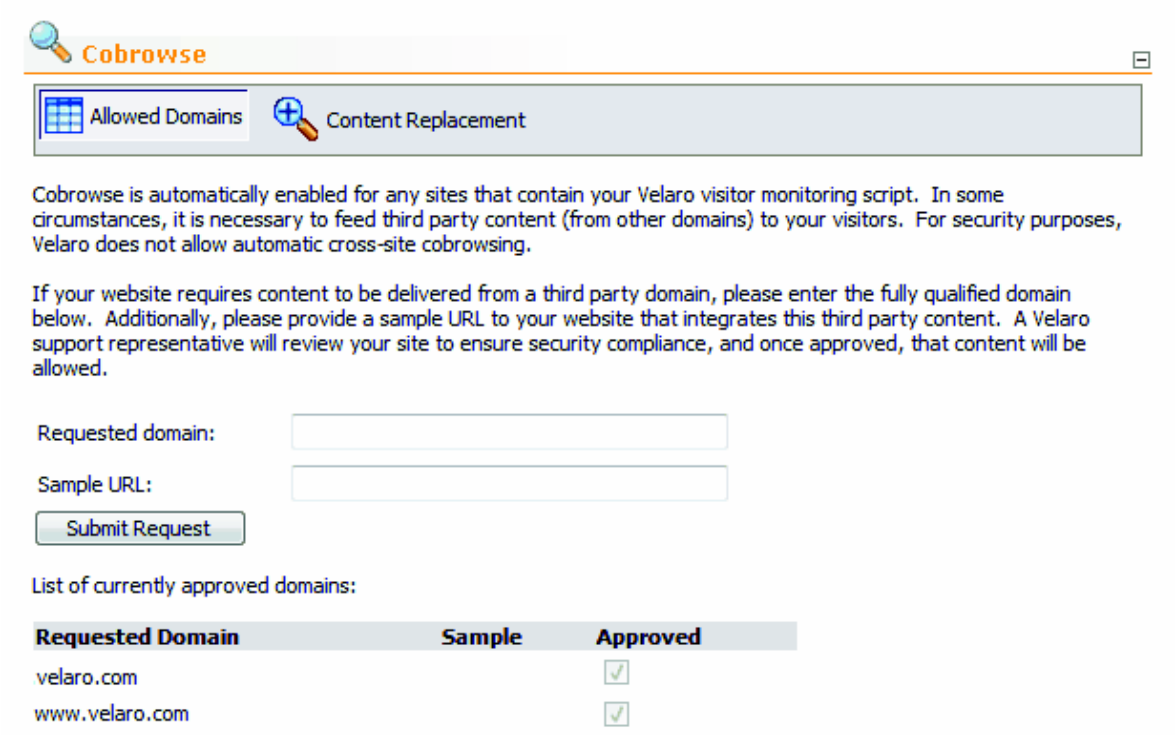
- Ensuring that cookies are shared across web sessions.
- Validating that the web site you are trying to surf is an approved domain to prevent your agent from inadvertently surfing websites other than your own.
- Injecting Javascript in to all web pages to capture mouse clicks, key presses, and other local browser events, then pushing those events to other users within the session.
- Injecting additional Javascript in to all web pages which is responsible for periodically checking the co-browse server to see if any additional data or new pages need to be navigated.
- Coordinating updates when your web site uses multiple frames.

Approving Domains

Before you can use Velaro's co-browse function, the web pages that you want to allow your agents and visitors to navigate must be authenticated. Authentication happens in two ways:

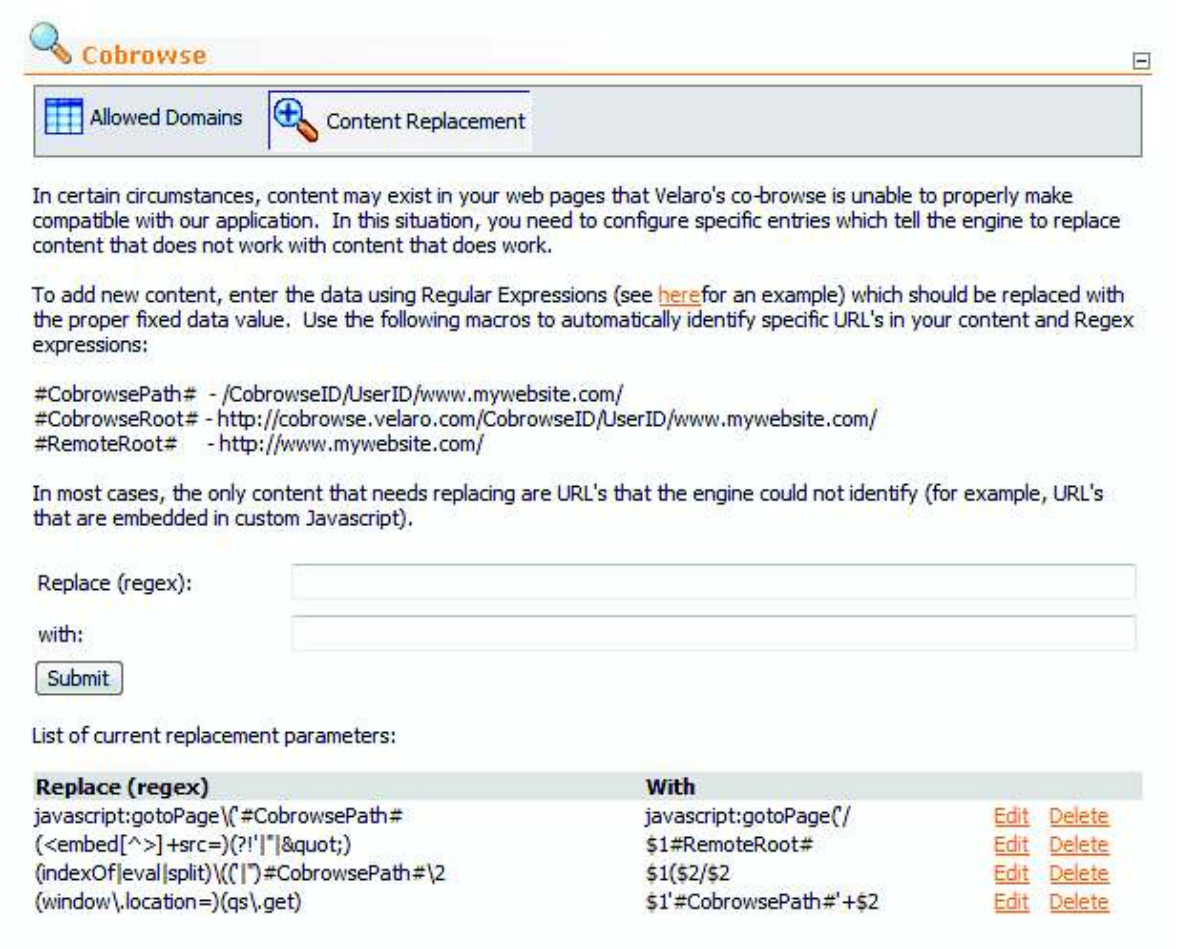
1. If the web page has Velaro's visitor monitoring script deployed, the page is automatically approved and may be co-browsed with out explicit approval by Velaro's administrators.

2. If your web pages do not have Velaro's visitor monitoring script, you must make a request to Velaro for co-browse approval. To request a domain, click the **Allowed Domain** toolbar button. Enter the requested domain, and a sample URL that Velaro's administrators can use to investigate your request. Once a request has been approved, your domain is listed in the table at the bottom of this module.



Customizing Co-browse Server Output

In most cases, Velaro's co-browse server works out-of-the-box. However, there are instances where your website pages contain combinations of HTML and Javascript that it can not interpret properly. If the co-browse server is unable to make all the automatic content replacements in your web pages, you may be required to force them. To inject custom content replacements, select the **Content Replacement** toolbar button. You are then presented with the following screen:



Before making custom content replacements, a complete analysis of your web pages is required. You should understand specifically what part of your web pages are rendering incorrectly. To do this, you should be familiar with your existing web pages and how they function. You should then create a co-browse session and investigate those pages as well. In most cases, a thorough understanding of HTML and Javascript is required.

Once the specific changes that need to be made have been identified, you configure the co-browse server to make them using Regular Expressions (RegEx).

Use the **Replace (regex)** field to enter the content that Velaro's co-browse server should look for within your web pages.

Use the **with:** field to enter the content that Velaro's co-browse server should replace the content that was found in the **Replace** field.

A list of all currently active content replacements is listed at the bottom of this module.

Velaro provides three macros that are responsible for generating the proper co-browse content. These macros may be used in both the Replace and With fields. Since certain portions of the co-browse server's URLs contain unique session information that changes, it is critical that you use these macros rather than hard-coded URLs that you may have seen while analyzing your content. The three macros are displayed in the above screenshot.

In the simplest form of content replacement, no RegEx expressions are required. For example, you may configure the co-browse server with the following rule:

Replace: Michael **With:** Jeffrey

This rule would change all occurrences of the word "Michael" with "Jeffrey" within your HTML pages.

For more complicated content replacements, a complete knowledge of Regular Expression rules is required. Velaro suggests the following websites for more information on RegEx:

http://www.regular-expressions.info/tutorialcnt.html

and http://www.sellsbrothers.com/tools/#regexd

One important note about RegEx expressions: Since the co-browse server is using Regex for content replacement, you may use the parameterized search/replace options provided by RegEx. This is done by enclosing content within the **Replace** field with parenthesis. For example:

Replace: ()Michael() **With:** $1Jeffrey$2

This rule will change all occurrences of the text "Michael" with "Jeffrey" within your HTML pages. Note how the replacement text contains $1 and $2. The content for those values is determined by the content within first and second group of parenthesis within the **Replace** field.

Cobrowse Limitations

If your web site is not working correctly out of the box and you believe content replacement rules must be created, you should observe the following limitations as guidelines for generating your content replacements:

- In most cases, if you have custom Javascript that is managing URL's within Javascript strings, this code can not be picked up automatically by the co-browse server and will require content replacement.
- If you are programmatically creating Javascript event handlers to your HTML elements, you may be inadvertently un-hooking Velaro's handlers, or Velaro's handlers may not be able to catch your Javascript events for proper replication. These cases may also need content replacement.
- Flash interaction cannot be replicated.

- If your web pages have existing Javascript errors, co-browse will not work.
- Nested frames behavior is indeterminate. The system was only built to handle one-level deep of frames.
- Currently, only the following events are replicated across the co-browse sessions:
    - Navigation to another page
    - Focus on any “input” element. The focus event may cause extraneous “extra” events if any person within the session is also using other web-browser windows as toggling from one browser window to another automatically fires this event.
    - Keypress and text change on any text input
    - Focus, keypress, and text change on any textarea
    - Checking/unchecking radio buttons and check boxes
    - Selection changed on a drop-down list
    - Javascript within an <A href=”javascript:”> tag
    - Any javascript click event
    - Any javascript change event
    - Page or frame scrolling. Frame scrolling cannot be replicated from Internet Explorer (but page scrolling can).
    - Closing a pop-up window
    - Javascript mouse-over events
- Cookies cannot be replicated from outside the co-browse to inside the co-browse or vice versa. This means that the user is essentially initiating a new session with your web server when they enter a co-browse session. Any cookies that we set before the session started do not exist within the co-browse session.
- Popup window replication may be blocked by popup blockers because they require no user action by the second party. If this happens, and the second user clicks the link themselves, each popup window will have a different FrameID, therefore co-browse between those two new windows will fail. As a solution to this problem, web visitors should be directed to turn off popup blockers prior to starting a co-browse session if your website uses them frequently.
- If your website is using certain forms of HTTP compression, Velaro's co-browse server will not be able to decompress it.
- Only HTTP and HTTPS protocols are supported.

- Velaro's co-browse server does not support the use of specified port numbers within URL's. For example: http://www.velaro.com:8081 is invalid.

Special note about sites that use AJAX

Many of today's modern websites are beginning to implement new ways to improve the end-user's visual experience. One of the primary technologies driving these new interfaces is typically called "AJAX". AJAX is a combination of different components that allow web pages to be partially updated rather than having the entire page reload.

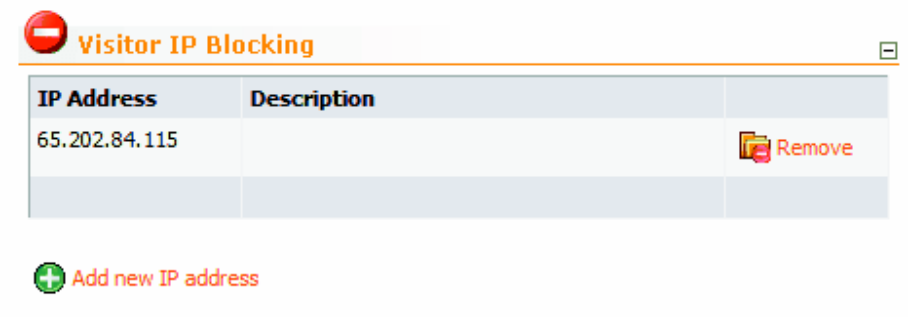
Partial page updating is done via Javascript that uses a built-in browser object which is capable of making direct connections to web-servers "behind the scenes" and parsing the results easily using XML. Since these updates are done in a non-traditional process, the standard page-updates used by Velaro's co-browse server are not valid and do not work.

AJAX calls are typically initiated on the browser via user events, such as mouse clicks, key presses, and button clicks. AJAX libraries trigger off these events and use the XmlHttpRequest object to contact the server and receive new content to display on the screen. In a traditional co-browse session, this causes a problem as the Velaro co-browse server receives the new content and pushes it to all users within the co-browse session. However, since none of the browsers other than the one that made the initial request are properly "listening" for this new data via the XmlHttpRequest object; they simply update the **entire** page with the resulting content.

Velaro's co-browse server circumvents this problem by identifying all requests made via the **XmlHttpRequest.Open** method and automatically assigns them their own context so they are not replicated to other browsers. In this manner, the resulting data received from your web server is only returned to the original requesting web browser. To ensure that all other browsers in the co-browse session have the same content displayed, the co-browser server replicates the **event** responsible for creating the AJAX call. In this way, each browser essentially creates their own AJAX call and manages their own result.

11 Visitor IP Blocking

Visitor IP Blocking allows you to restrict chat access from specified web site visitors. To manage IP blocking, select the **Setup : IP Blocking** menu option.



Managing IP Addresses

To restrict access to live help from specific website visitors, click the **Add new IP address** option. You are then prompted to enter the host IP address of the visitor to block. Note that from that point forward, any visitor who is coming from this IP address will not be able to chat, nor will they be displayed in your agent's visitor monitoring section of the Velaro Desktop application.

When adding an IP address, Velaro provides a descriptive field to let you comment and remember why you have blocked a specific address.

To remove an IP address from the blocked list, select the **Remove** option next to the IP address you want to allow.

Warning: Blocking an IP address from the control panel, as specified above, prevents all your agents from viewing the visitor in real-time or chatting with them. Individual agents may block visitors on their machine directly from within the Velaro Agent Desktop, but this has no effect on other agents as does the above procedure.