Enterprise Edition

Replay Guide and Reference

atomiclabs™

Last update: **February 2010**

Table of Contents

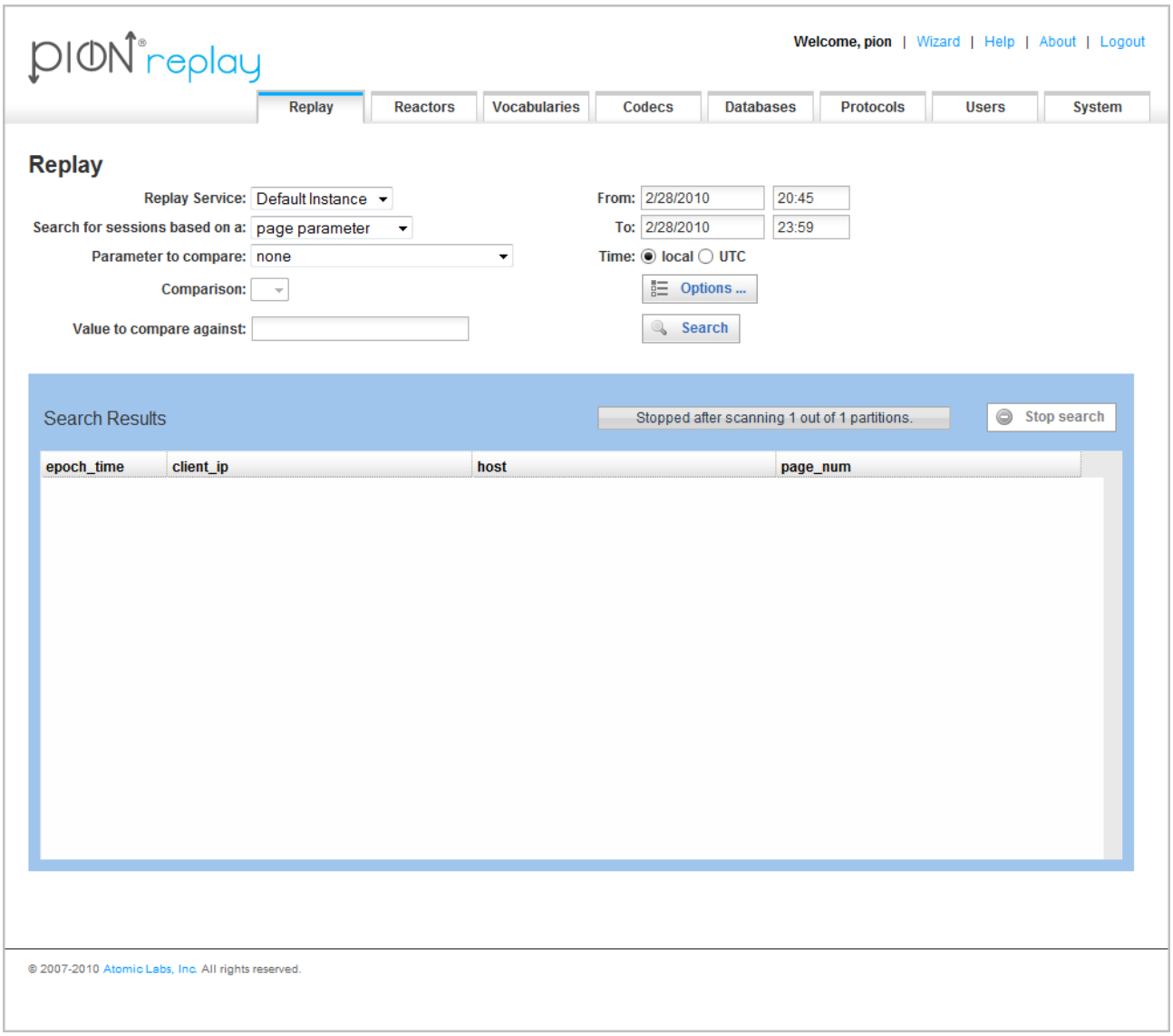
Using the Replay Feature

The Replay feature allows you to see what a visitor to your web site saw and “step through” their experience on your web site. The Replay feature captures the page views seen by a user, and **allows you to see the pages the y saw, as the user saw them**! Using Replay, you can also display information about the user's session, such as the client IP address, date and time, number of pages viewed, and more. The Replay feature:

- Uses the Content Storage Reactor and a database to store the content (pages).
- User can search sessions from "Sessions," "Pages," or "Requests" using configurable queries.
- Session contains multiple page views, which can be viewed in a browser ("you're seeing what the end-user saw").

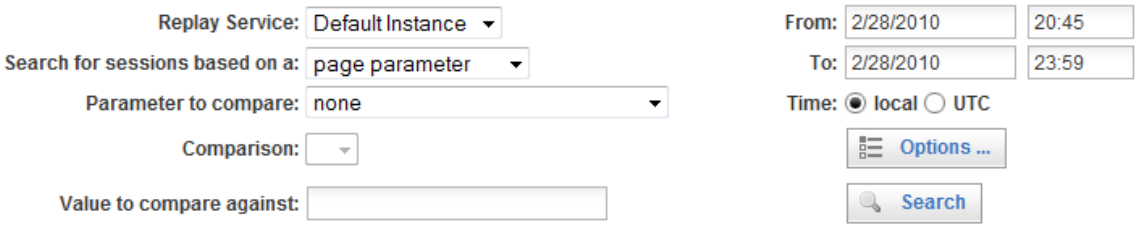
To use Replay, Pion must be configured to monitor the web server, and the Content Storage Reactor and the Database Output Reactors must be properly configured.

Note: Currently, the only supported database for Replay is the embedded SQLite database. In future releases, additional databases may be supported.



To replay a visitor's session:

1. Go to the Replay page of the Pion application.
2. Select the Replay Service (the Pion instance that is monitoring the web server that hosted the sessions you want to review).
3. Using the drop-down list boxes, specify the search criteria for the session(s) you want to review:



4. Select the **Search for session based on** criteria. You can select one of the following:

   - **Session parameters**, which include session-based information, such as server IP address, client IP address, user agent (the visitor's browser), referrer, cookie ID, and other factors relating to the session itself.

     Because Pion 'sessionizes' web visits and stores that information in a database, using the **Session parameters** will be the fastest way to find particular sessions. Finding a particular session using **Session parameters** is more efficient than the other search criteria parameters because the database storing session information is smaller and will therefore be processed faster than the others. However, because there is a built-in delay to allow sessions to time-out (1800 seconds by default), you may want to use page or request parameters to locate the session.

     There are two reasons to use the page or request parameters:

     - To see sessions that have not yet timed out (a way to handle the sessionizing timeout).
     - To be able to search based on parameters that are only available in page or request metadata. For example: status (to find sessions that had page-not-found requests) or page_title (to find sessions that hit a particular page title).

   - **Page parameters**, which include page-based data, such as server IP address, host, page number, page status, URI query (the name of the called resource), and other parameters related to the page the visitor(s) viewed.

     Finding a session using page parameters involves grouping all page views together into a Session view, which slows searches.

   - **Request parameters**, which include details about the incoming client request, such as date, time, session ID, client IP, and other client-related details.

     Finding a session using request parameters involves searching through all the individual requests, grouping all requests together, and other processing steps, making this the slowest/most resource intensive search type.

5. Select the **Parameter to compare** criteria.

   Depending on the selection you made in the **Search for session based on** criteria, the list of possible parameters to compare against changes. Select the parameter that will identify the user session(s) in which you have an interest. (You can select only one parameter at a time.)

Note: In the **Parameter to compare** field, selecting "none" (which is the default) provides a simple list/view of the events (with very little effort). Based on this list, you can then formulate your query more easily.

Select the **Comparison** to make:

- **is-not-null**, which means there must be a value (any type of value) to compare against. This will result in a list of requests/pages/sessions that have a non-empty parameter of the type you specified in the **Parameter to compare** criteria.

- **is-null**, which means there must **not** be a value to compare against. This will result in a list of requests/pages/sessions that have no value specified for the parameter of the type you specified in the **Parameter to compare** criteria.

- **exact-match**, which means an exact match to the string you will specify in the next step. This will result in a list of requests/pages/sessions whose **Parameter to compare** values exactly match the string you will specify in the next step.

- **not-exact-match**, which means “anything except an exact match” to the string you will specify in the next step. This will result in a list of requests/pages/sessions whose **Parameter to compare** values **are anything other than** the string you will specify in the next step.

- **like**, which means you will specify a character string, which will provide the pattern to compare against. Note that wildcards, such as "%" (a multi-character wild-card) and "_" (a single-character wildcard). This will result in a list of requests/pages/sessions whose **Parameter to compare** values match the pattern you will specify in the next step.

- **not-like**, which means you will specify a character string, which will provide the pattern to compare against. Note that wildcards, such as "%" (a multi-character wild-card) and "_" (a single-character wildcard). This will result in a list of requests/pages/sessions whose **Parameter to compare** values **do not** match the pattern you will specify in the next step.

- **ordered-before**, which means you will specify a string containing some form of sequence identifier, which will be compared against (a date, a numerical or alphabetic value). This will result in a list of requests/pages/sessions whose **Parameter to compare** values come before (are lower or earlier) in the sequence than the string you will specify in the next step.

- **not-ordered-before**, which means you will specify a string containing some form of sequence identifier, which will be compared against (a date, a numerical or alphabetic value). This will result in a list of requests/pages/sessions whose **Parameter to compare** values come after (are higher or later) in the sequence than the string you will specify in the next step.

- **ordered-after**, which means you will specify a string containing some form of sequence identifier, which will be compared against (a date, a numerical or alphabetic value). This will result in a list of requests/pages/sessions whose **Parameter to compare** values come after (are higher or later) in the sequence than the string you will specify in the next step.

- **not-ordered-after**, which means you will specify a string containing some form of sequence identifier, which will be compared against (a date, a numerical or alphabetic value). This will result in a list of requests/pages/sessions whose **Parameter to compare** values come before (are lower or earlier) in the sequence than the string you will specify in the next step.

- **starts-with**, which means you will specify a string, which will be compared against the beginning of the stored data. This will result in a list of requests/pages/sessions whose **Parameter to compare** values start with the string you will specify in the next step. Note that this comparison is not case sensitive.

- **ends-with**, which means you will specify a string, which will be compared against the final characters of the values in the stored data. This will result in a list of requests/pages/sessions whose **Parameter to compare** values end with the string you will specify in the next step. Note that this comparison is not case sensitive.

- **contains**, which means you will specify a string, which will be compared against the values in the stored data. This will result in a list of requests/pages/sessions whose **Parameter to compare** values include (contain) the string you will specify in the next step. Note that this comparison is not case sensitive.

Note: Comparisons depend on the specific database engine used to store the meta-data. Currently, Pion supports only the embedded SQLite database engine.

6. Using the **Value to compare against** field, specify the search criteria for the session(s) you want to review.

   Note: The **Value to compare against** field is disabled for comparisons for which it is not applicable, such as **is-null** and **is-not-null**.

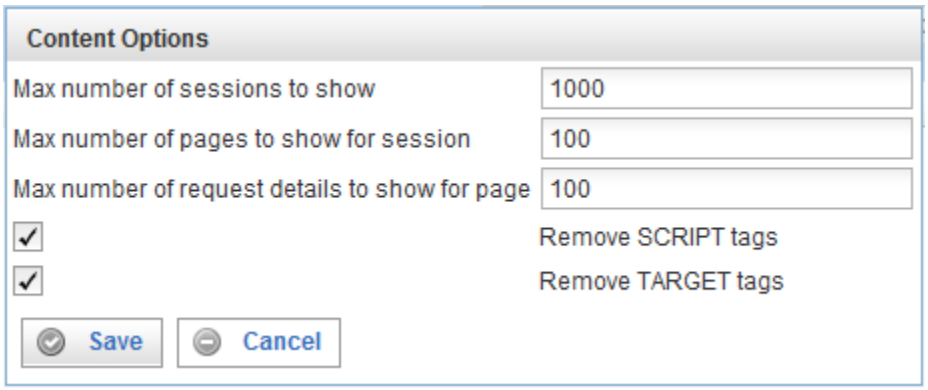
   Depending on the selection you made in the search criteria above, enter the value/regular expression/sequence identifier for comparison.

7. Select the Date and time range for the period of the time in which you want to compare criteria.

   Use the date and time controls to specify the start and end of the period you want to examine. For time values, select the radio button that indicates if the time you entered was the local time (the Pion instance's local time) or the UTC time.

8. Optionally, you can click on the Options button to display the **Content Options** screen. The fields on this screen allow you to specify the maximum number of sessions, pages, and request details to be returned by a search. If the number of pages within a session exceeds the number you specify, the additional pages will be truncated when viewed.

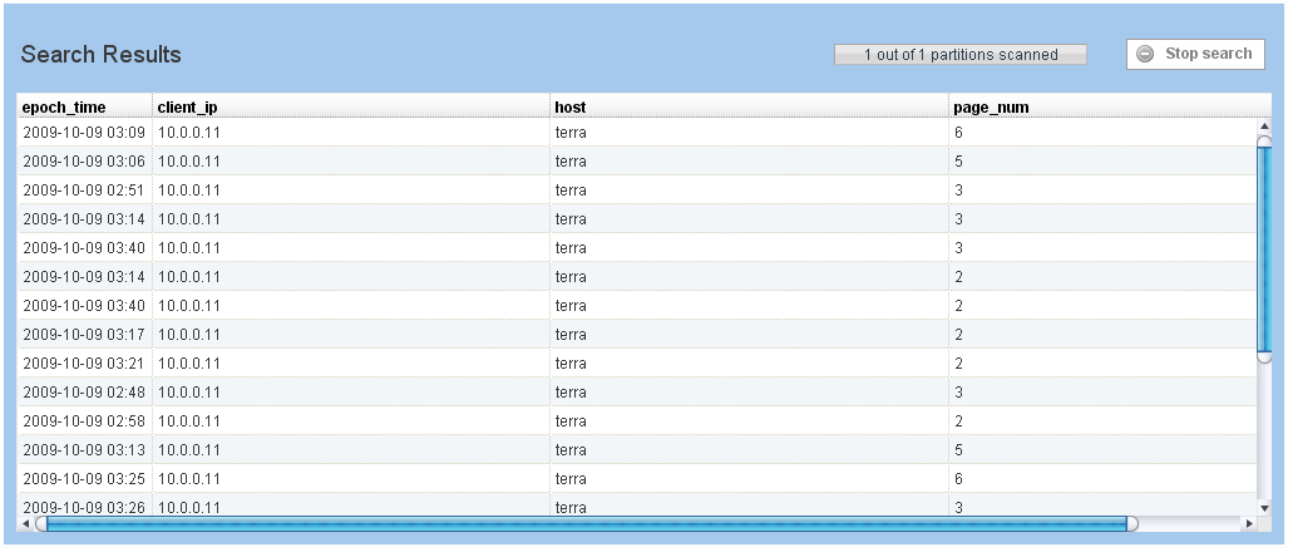
   To remove script and/or target tags from the search results, fill the appropriate check box(es).



9. Click **Search**.

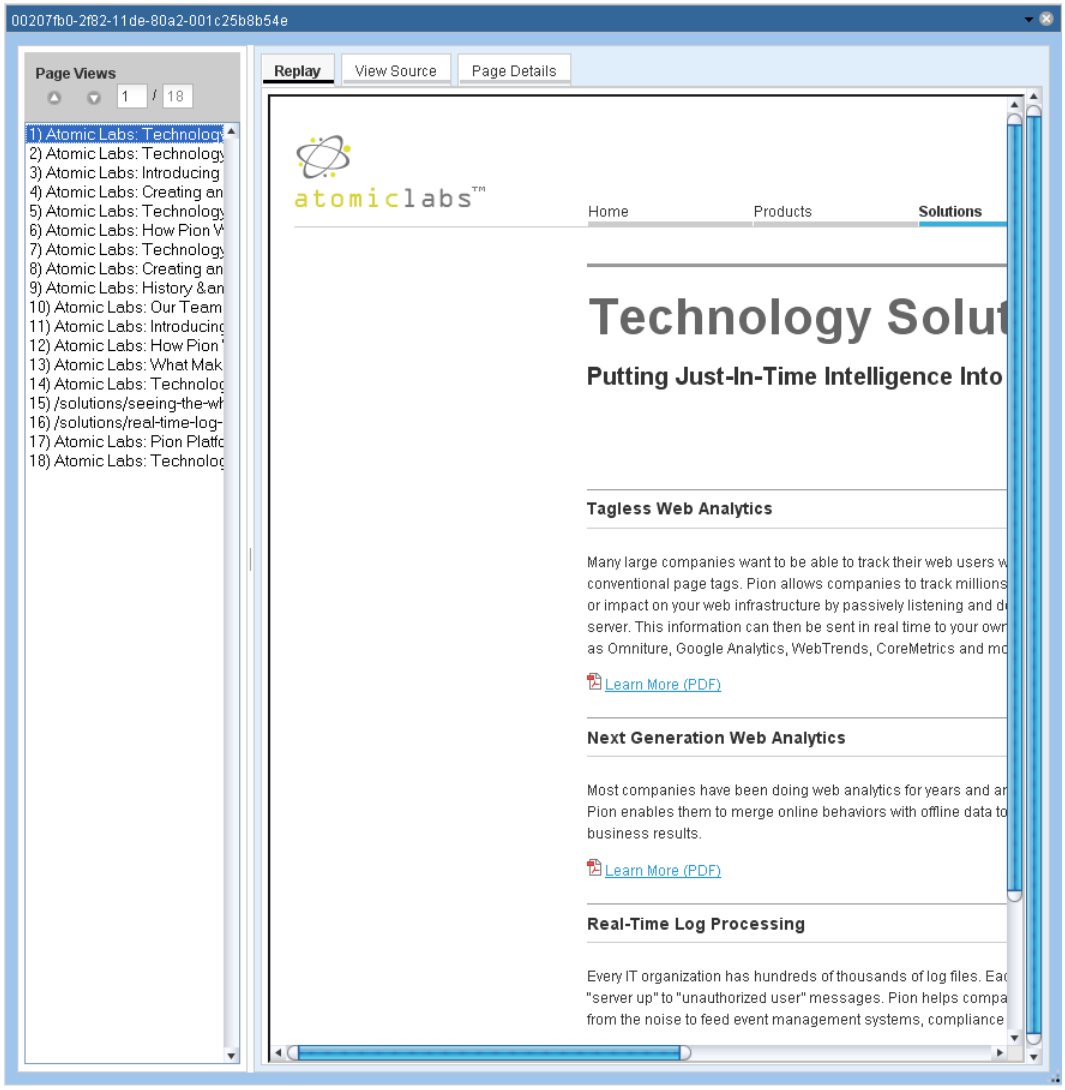
   Note: If the search time exceeds 10 seconds, the search will time out. If you experience this situation, consider adding indexes. For information on optimizing searches and using Replay, see Error: Reference source not found, on page Error: Reference source not found.

   The search results, if any, are displayed below the search criteria area.

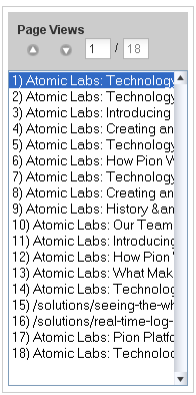


epoch_time	client_ip	host	page_num
2009-10-09 03:09	10.0.0.11	terra	6
2009-10-09 03:06	10.0.0.11	terra	5
2009-10-09 02:51	10.0.0.11	terra	3
2009-10-09 03:14	10.0.0.11	terra	3
2009-10-09 03:40	10.0.0.11	terra	3
2009-10-09 03:14	10.0.0.11	terra	2
2009-10-09 03:40	10.0.0.11	terra	2
2009-10-09 03:17	10.0.0.11	terra	2
2009-10-09 03:21	10.0.0.11	terra	2
2009-10-09 02:48	10.0.0.11	terra	3
2009-10-09 02:58	10.0.0.11	terra	2
2009-10-09 03:13	10.0.0.11	terra	5
2009-10-09 03:25	10.0.0.11	terra	6
2009-10-09 03:26	10.0.0.11	terra	3

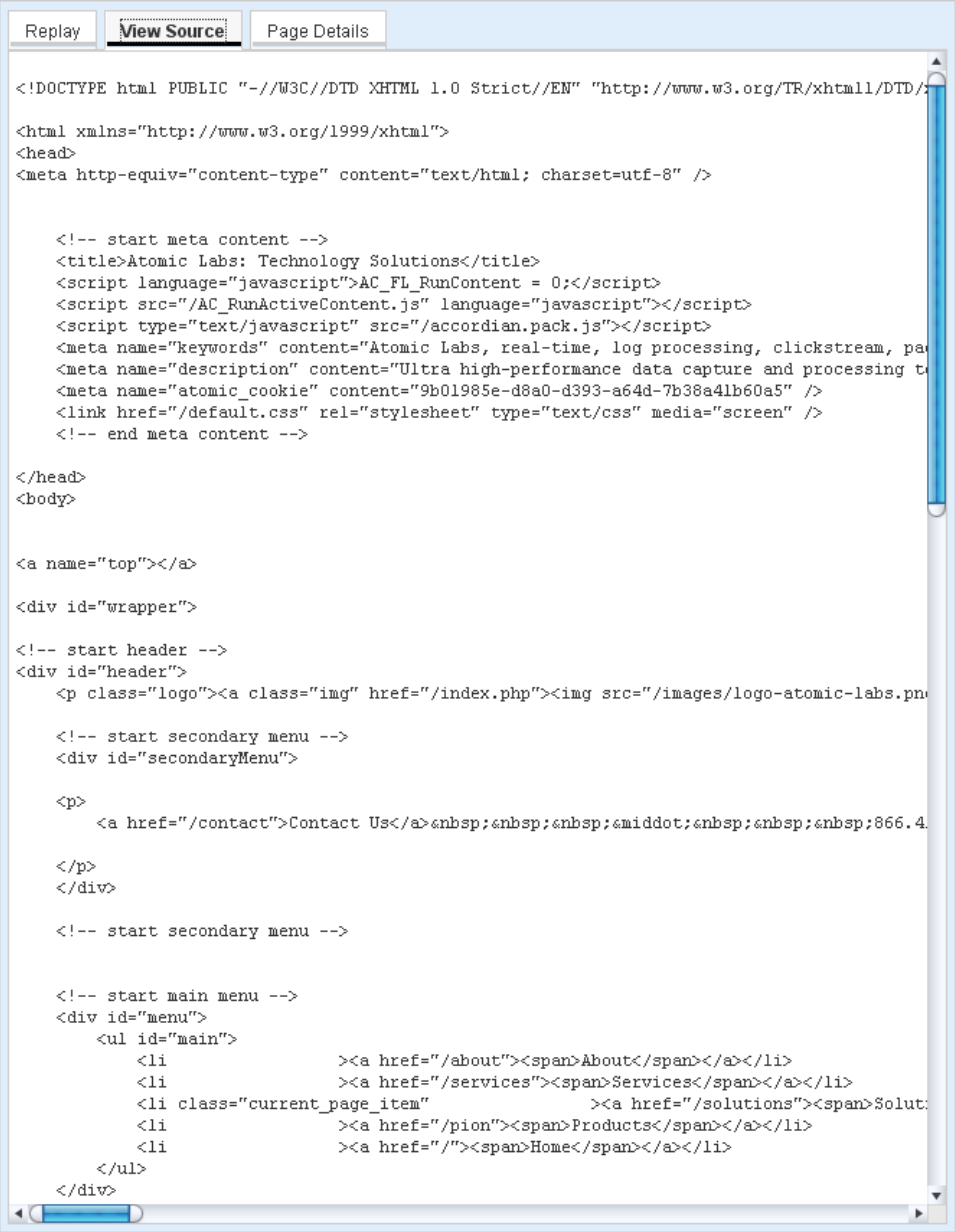
10. Double-click on the session you want to view. The Replay window appears:



- **To see the pages that the web site visitor saw**, go to the **Page Views** list and click on the page you want to view.



- **To see the page source**, select the **View Source** tab.



- **To see the page details for the currently selected page view**, select the **Page Details** tab.