

Correlate Business Transactions for Browser RUM

On this page:

- [Benefits of Correlating Business Transactions](#)
- [Requirements](#)
- [App Server Agents Supporting Business Transaction Correlation](#)
- [Avoid Tagging Cookies with the HttpOnly Flag](#)
- [How It Works](#)
- [Enable Business Transaction Correlation](#)
- [View Business Transaction Correlation](#)
- [Get Complete Timing Data for Correlated Business Transactions](#)

You can correlate page and Ajax requests with business transactions. In actuality, the correlation is made between instances of page/Ajax requests (browser snapshots) and instances of business transactions (transaction snapshots).

The correlation enables you to map end-user requests with an underlying backend application. Business-transaction correlation is disabled by default.

Benefits of Correlating Business Transactions

By correlating business transactions with browser snapshots, you can identify potential issues with the backend application that are causing bad user experiences. For example, you might find that server errors or a database query is causing a slow or very slow user experience.

Requirements

To correlate business transactions, you are required to do the following:

- purchase licenses for Application Performance Monitoring (APM) and End-User Monitoring (EUM)
- instrument an application with an [app server agent that supports business transaction correlation](#)
- inject a browser application with the JavaScript Agent
- [enable business transaction correlation](#) in the Controller UI

App Server Agents Supporting Business Transaction Correlation

To correlate business transactions, your business application must have one of the following app server agents installed:

- [Java Agent](#)
- [.NET Agent](#)
- [Node.js Agent](#)
- [PHP Agent](#)
- [Python Agent](#)
- [Apache Agent](#)

Avoid Tagging Cookies with the HttpOnly Flag

`HttpOnly` is a flag that servers can set on cookies to prevent their contents from being accessed by JavaScript. This is often done for session cookies to hide the session identifier as a security measure. The JavaScript Agent, however, needs to be able to read special cookies set by the AppDynamics server-side agent (all prefixed with ADRUM) to collect correlation information. If `HttpOnly` is set on these cookies, no server-side correlation information can be transmitted. Thus, make sure that your server does *not* set the `HttpOnly` flag on any cookies prefixed with ADRUM.

If you want to securely transmit cookies, use HTTPS. The app agent sets the `secure` flag if the originating base page is loaded over HTTPS.

These can be turned off, but that'd prevent correlation from working for base pages unless the customer changes the page source code to do footer injection.

How It Works

When an end user requests a page from your browser app:

1. The app agent does the following:
 - sends HTTP headers identifying the business transaction and the HTML with the injected JavaScript Agent to the end user's browser.
 - aggregates backend metrics and sends them along with the business transaction identifiers to the Controller. This serves as the content for the transaction snapshot.
2. The JavaScript Agent sends browser metrics and the business transaction identifiers (from the HTTP header) to the EUM Server. This serves as the content for the browser snapshot.
3. The Controller fetches the metrics and business transaction identifiers from the EUM server and then uses the business transaction identifiers to correlate the browser snapshot with the transaction snapshots.

Enable Business Transaction Correlation

You need to configure the Controller to correlate business transactions. The Controller will map the business transactions with the browser snapshots based on the process described above in [How It Works](#).

To enable business transaction correlation:

1. From the **Application Dashboard**, click **Configuration**.
2. Click **User Experience App Integration >**.
3. From the **Business Transaction Correlation** tab, check the **Enable Business Transaction Correlation** check box.

Specify Business Transactions to Include Correlation Headers

You can also specify which business transactions will include or exclude correlation headers. If you do not add request match rules or request exclude rules, correlation headers will be added to all requests.

To add a request rule:

1. Click **+** to open the **Create HTTP Request Match Rule** dialog.
2. From the **Create HTTP Request Match Rule** dialog:
 - a. Check the **Method** check box and select an HTTP method that you want to match. If you do not select an HTTP method, the rule will be applied to all HTTP requests.
 - b. Check the **URI** check box and enter your criteria.
 - c. Click **Save** to save the match rule.
3. From the **Business Transaction Correlation** tab, click **Save**.

View Business Transaction Correlation

There are several ways to navigate from a browser snapshot to its correlated business transaction. The following steps show you one possible way.

1. From the **Browser App Dashboard**, click **Browser Snapshots**.
2. Click **Filters** and check the **Server Snapshot Exists** check box.

3. You should now only see browser snapshots that have transaction snapshots as seen below:

Time	End User Response Time (ms)	URL	Page, AJAX Request, iFrame, or Virtual Page	Browser	Device	OS	Country	State / Region	City
10/27/16 4:48:41 PM	270	http://lbr/appdynamicspilot/	lbr/appdynamicspilot	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:47:42 PM	517	http://lbr/appdynamicspilot/	lbr/appdynamicspilot	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:38:34 PM	415	http://lbr/appdynamicspilot/	lbr/appdynamicspilot	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:37:35 PM	277	http://lbr/appdynamicspilot/	lbr/appdynamicspilot	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:31:12 PM	47	http://lbr/appdynamicspilot/rest/items/all...	lbr/appdynamicspilot/rest	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:29:43 PM	242	http://lbr/appdynamicspilot/ViewItems.a...	lbr/appdynamicspilot/View...	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:28:30 PM	208	http://lbr/appdynamicspilot/	lbr/appdynamicspilot	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:28:12 PM	39	http://lbr/appdynamicspilot/rest/items/all...	lbr/appdynamicspilot/rest	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:27:28 PM	447	http://lbr/appdynamicspilot/	lbr/appdynamicspilot	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:18:23 PM	325	http://lbr/appdynamicspilot/	lbr/appdynamicspilot	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:17:20 PM	439	http://lbr/appdynamicspilot/	lbr/appdynamicspilot	Fr...	Compu...	Linux	United ...	Oregon	Board...
10/27/16 4:12:15 PM	200	http://lbr/appdynamicspilot/ViewItems.a...	lbr/appdynamicspilot/View...	Fr...	Compu...	Linux	United ...	Oregon	Board...

4. Double-click one of the browser snapshots to open the **Browser Snapshot Details** dialog containing a transaction snapshot.

Real User Snapshot

End User Response Time	270 ms
DOM Ready Time	222 ms
First Byte Time	7 ms
Server Connection Time	2 ms
Response Available Time	5 ms
Server Time	3 ms
HTML Download Time	0 ms
DOM Building Time	215 ms
Resource Fetch Time	48 ms
Load Event	
Post Page Load	

Business Transactions

Name	Time (ms)
/appdynamicspilot	3

Transaction Snapshots

Time	Exec Time (ms)	URL	Business Transaction	Tier	Node
10/27/16 4:48:40 PM	3	/appdynamicspilot/	/appdynamicspilot/	ECommerce-Ser...	ECommerce-E2...

5. You can then click links in the transaction snapshot or transaction snapshot itself to view corresponding pages in APM.

Get Complete Timing Data for Correlated Business Transactions

To get the full real execution time for correlated business transactions, your injection method may need to write the `JS_FOOTER` variable to your page. Manual injection gives the server-side agent the ability to write data only to the header of the page as it is being constructed by your web application. It is possible that complete business-transaction timing information is not available at the moment that the header data is written. Using the footer allows the server-side agent to write timing data at the footer of the page, by which time a fuller picture of business transaction timing may be available.

You can write the `JS_FOOTER` data variable into the footer of a web page using the following techniques:

- If you use automatic injection for the injecting into the head section, you automatically get an injection into the footer as well.
- If you use manual injection for the head section, for applications built on Java platforms you can use assisted injection to inject into the footer. Or for applications built on Java servlet or ASP.NET platforms, you can use assisted injection using attribute injection.

If you cannot add the `JS_FOOTER` variable to your page, the timing shown for correlated business transactions may be the average response time for that transaction rather than the real execution time for that specific page.