Capture Resource Timing Data Without Loss

Related pages:

- Modify Resource Sampling Options

Most browsers stop capturing resource timing data in a page when the number of resources reaches 150 because of the limit set by the Resource Timing API. You can configure the JavaScript Agent for conventional web pages (non-single page applications) and single-page applications (SPAs), however, to overcome this limitation.

> For Internet Explorer, you currently *cannot* use the JavaScript Agent configuration above to override the Resource Timing API's default limit of resources returned by the browser.

Because the JavaScript Agent configuration is different for non-SPA and SPA, see the following sections for details and instructions:

- Set and Clear the Resource Timing Buffer for Non-SPAs
- Disable the Clearing of the Resource Timing Buffer for SPAs

Set and Clear the Resource Timing Buffer for Non-SPAs

For non-SPAs, you can configure the JavaScript Agent to use a buffer to capture resource timing data and then set a flag to clear the buffer once the beacon has transmitted the resource timing data to the EUM Server. Clearing the buffer clears the browser's resource buffer array and ensures that new resources will be sent in the next beacon.

> Overriding the Resource Timing API and clearing the buffer may affect JavaScript code on your page that is using the Resource Timing API. Thus, the JavaScript Agent uses the default set by the Resource Timing API unless you override the default by configuring the JavaScript Agent to set and clear the resource timing buffer.

JavaScript Configuration Example

The `resTiming` object is used for configuring the buffer size (maximum number of resources to return) and whether the buffer is reset once the beacon is transmitted to the EUM Server. The code snippet below shows you how to specify the buffer size and set the flag for clearing the buffer with the properties `bufSize` and `clearResTimingOnBeaconSend`.

```
<head>
    <script type='text/javascript' charset='UTF-8'>
        window['adrum-config'] = {
            resTiming: {
                bufSize: 200,
                clearResTimingOnBeaconSend: true
            }
        };
    </script>
    <script src='//cdn.appdynamics.com/adrum/adrum-latest.js' type='text/javascript' charset='UTF-
8'></script>
</head>
```

Disable the Clearing of the Resource Timing Buffer for SPAs

For SPAs, the JavaScript Agent by default clears the resource timing buffer after it's full and saves the data in a local buffer. You can configure the JavaScript Agent so that the resource timing buffer is not cleared to capture resource timing data.

ⓘ If the resource timing buffer is not cleared, the resource timing correlation may miss some resources due to the buffer limit of the browser.

JavaScript Configuration Example

The default for `clearResTiming` is true. Setting `clearResTiming` to `false`, as shown below, disables the automatic clearing of the resource timing buffer. The resource timing buffer for virtual pages will also not be cleared.

```
 window['adrum-config'] = {
      ...
      "spa": {
           "spa2": {"clearResTiming": false}
      }
}
```