Arthur Zavalkovsky
Director Product Management
Radware, Inc ([www.radware.com](http://www.radware.com))
arthurz@radware.com

Automatic Web performance measurement

# Abstract

This document presents a proposal for extending the HTTP and HTML standards for enable automatic collection of the web page performance data. These data could be made visible to the application delivery infrastructure (Web servers, Application delivery controllers, other networking devices) allowing different types of dynamic functionality.

# Motivation

Performance of Web applications have direct impact on customer satisfaction and user conversion and, therefore are critical for business’s revenue. Today, the visibility on the user experience can be achieved by utilizing the Navigation APIs defined by W3C and implemented by the leading web browser’s vendors. This requires augmenting the application web pages with a performance-data-fetching Java Script code and development of proprietary communication algorithms for reporting the data to a reporting server or the originating web server.

This information can be used by the application delivery infrastructure (Web servers, Application delivery controllers, networking devices, etc…) for several purposes including:

* Early identification of the problems and taking preventive actions. For example, if it takes long time to access an application, the capacity of a web server infrastructure should be increased (more servers deployed).
* Adjusting the resource content to the current user session. For example, if network latency is high, the Web server can serve an image of lower resolution (and smaller size).
* Analysis of the application page and application page element performance in different circumstances and redesign of the application to achieve higher customer satisfaction.

The APM vendors focusing on the RUM (Real User Monitoring), including Radware, are answering on that customer need.

# Challenges

Form the operational perspective, development and deployment of business applications is, in many cases, separated both in time and organizationally from the development of user experience monitoring procedures.

First, the business application (on-line shop, informational portal, etc…) is being developed by an internal dev team or an external vendor and deployed. Later, the operation team implements the monitoring processes, which include deployment of an APM solution.

Today, RUM oriented APM solutions inject performance-data-fetching javascripts into application web pages. This is done either by an agent installed on the web server or by a networking device (e.g. Radware’s ADC). This approach has several serious challenges:

* Insertion of a Javascript can change the web page functionality, e.g. slow down the page or change the visualization.
* The affect of these Javascripts on page performance and functionality of the page might be different on different agents (slow vs. fast machine, available memory, browser configuration (e.g. JS enabled/disabled), laptop vs. mobile, different OS, etc…)
* Malformed resources (e.g. badly formatted web pages) might be still presented by browser but cannot be extended with the Java Script.

 Those challenges make APM deployment complex and requiring customization.

# Proposed solution

A general approach to the solution would be to allow modification of the meta-data of the web page rather than addition of active content/resources (like javascripts).

The meta-data could specify the list of resources and / or page elements for which the performance data is required and the target URL for sending the reports. The browser would automatically collect the required performance data and post the results to the remote server.

### Conceptual proposal #1 HTTP header

The HTTP request can be augmented by adding a new HTTP header:

Performance\_Reporting\_Target: <URL for reporting>\r\n

The header indicates where the report should be sent to. The browser is expected to automatically send the performance information to the provided URL

### Conceptual proposal #2 Element’s attribute

Some elements, like <body>, <a>, <div>, <iframe>, <img>, <object> (others?) will include a new attribute perfcollectwith two values trueandfalse

<IMG src="http://www.somecompany.com/People/Ian/vacation/family.png"

 perfcollect=true>

The browser is expected to automatically collect the performance information about the elements with perfcollect attribute set to true.

### Conceptual proposal #3 Performance information

The performance information collected per elements should include values of the attributes of [PerformanceResourceTiming](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#performanceresourcetiming) : [PerformanceEntry](http://www.w3.org/TR/performance-timeline/#performanceentry) defined in the Navigation APIs

interface [PerformanceResourceTiming](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#performanceresourcetiming) : [PerformanceEntry](http://www.w3.org/TR/performance-timeline/#performanceentry) {

 readonly attribute DOMString [initiatorType](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-initiatortype);

 // "css", "embed", "img", "link", "object", "script", "subdocument", "svg", "xmlhttprequest", "other"

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [redirectStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-redirectstart);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [redirectEnd](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-redirectend);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [fetchStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-fetchstart);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [domainLookupStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-domainlookupstart);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [domainLookupEnd](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-domainlookupend);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [connectStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-connectstart);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [connectEnd](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-connectend);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [secureConnectionStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-secureconnectionstart);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [requestStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-requeststart);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [responseStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-responsestart);

 readonly attribute [DOMHighResTimeStamp](http://www.w3.org/TR/hr-time/#domhighrestimestamp) [responseEnd](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-responseend);

};

### Conceptual proposal #4 Communication protocol

The reporting of the performance results should be done by a POST request to the reporting server URL:

POST /enlighten/calais.asmx/Enlighten HTTP/1.1
Host: api.opencalais.com
Content-Type: application/x-www-form-urlencoded
Content-Length: **length**

resource=**name\_of\_resource1**

resourceURI=**URI\_of\_resource1**

[redirectStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-redirectstart)=**timestamp**

[redirectEnd](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-redirectend)=**timestamp**

[fetchStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-fetchstart)=**timestamp**

[domainLookupStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-domainlookupstart)=**timestamp**

[domainLookupEnd](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-domainlookupend)=**timestamp**

[connectStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-connectstart)=**timestamp**

[connectEnd](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-connectend)=**timestamp**

[secureConnectionStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-secureconnectionstart)=**timestamp**

[requestStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-requeststart)=**timestamp**

[responseStart](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-responsestart)=**timestamp**

[responseEnd](file:///C%3A%5CUsers%5Carthurz%5CDocuments%5CAPM%5CResource%20Timing.htm#dom-performanceresourcetiming-responseend)=**timestamp**

resource=**name\_of\_resource2**

resourceURI=**URI\_of\_resource2**

**...**