

Load, Stress & Performance Testing

With



Taming the Microsoft .NET ViewState with SCL

By Antony Marcano

Abstract: This article details a solution for capturing and replaying the Microsoft .NET ASP ViewState parameter using OpenSTA, a 'Free to use' Open Source, HTTP/S Load, Stress & Performance testing tool. It also discusses practical performance tuning issues associated with the use of the .NET ViewState value.

Introduction

OpenSTA is a commercial strength performance-testing tool based upon CORBA, and was originally developed by Cyrano. Running on Microsoft Windows, OpenSTA can be used to load test virtually any application that uses HTTP/S. It uses a powerful scripting language called the Script Control Language (SCL) which allows it to manipulate everything from dynamic values to byte-code. It is now Open Source, under the GNU general public license and is therefore FREE to use.

Microsoft .NET is growing in popularity. The OpenSTA user forum and **etest associates'** OpenSTA Technical Support team has been increasingly queried on how to use OpenSTA to test Microsoft .NET web applications. One of the first challenges that stump some OpenSTA engineers is how to dynamically manage the ASP .NET ViewState.

More information about OpenSTA and Microsoft .NET ViewState Tuning can be found through "Links to Additional Resources" at the end of this article.

What is the ASP .NET ViewState

Web servers are 'stateless', and as such so are ASP .NET pages. An HTTP request could be from anyone and about anything at the time it is received by the web-server.

Most applications, however, cannot be stateless and must know who the user is and what they were doing to determine how to deal with their current request. Usually a session ID will track who the user is. It is the question of 'what they were doing' that is addressed by the ViewState.

Previously in ASP, it was the developer's responsibility to write code that keeps track of a user's previous interaction. **ViewState** now takes this concern away from the developer by automatically tracking the state of controls in a previously instantiated form. Developer effort per page is reduced, thus productivity increased. This, however, comes at a cost, which is addressed later in this article.

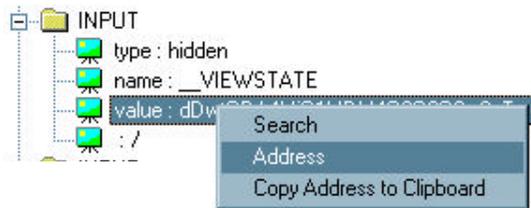
Any more detail on stateless interaction and how ASP .NET ViewState is used, is outside of the scope of this paper. It is obvious, however, that if the developers are taking advantage of it then OpenSTA needs to be told how to use it. This article explains how to achieve this, enabling the use of OpenSTA for load testing .NET applications.

Script Modelling and capturing the ViewState

After a script is recorded, it must be edited to capture the ViewState value dynamically. The ViewState value may be found in many pages but should only be captured just before a request that needs to use it. This is performed by simply “Addressing a DOM element” which is explained in detail by the Modeller help. How that process relates to ViewState is described here, in summary only.

After recording the script, no changes should be made to the URLs. The  icon in the tool-bar of the Script Modeller will populate the Query Pane with content of the server’s response for the selected URL. The “HTML Tree” tab in the query pane displays a tree representation of the Document Object Model (DOM).

Right click on the HTML Node and choose “Search”. In the textbox that appears, enter “VIEWSTATE” (without the quotes) and hit [Return]. Then right click on the VIEWSTATE value:



From the shortcut menu that appears, select “Address” and assign a variable name, for example “TEMP_VSTATE”.

This will automatically insert the necessary code into the script:

```
LOAD RESPONSE_INFO BODY ON 1 &
INTO TEMP_VSTATE &
,WITH "HTML(0)/BODY(1)/DIV(0)/FORM(0)/INPUT(0):ATTRIBUTE:value(2)"
```

Unfortunately, after capturing the ViewState, simply substituting the string with the new variable in the next request will result in an HTTP 500 error when played back. There is a little more to using the captured ViewState than this, hence the “TEMP” in the chosen variable-name, “TEMP_VSTATE”.

Using the ViewState

The format of the ViewState is issued to the browser in literal characters. There are numerous encoded values in the ViewState, each delimited by “+”. The server, however, expects to receive the ViewState back with special characters replaced with their ASCII equivalent. The only special character that is observed to be dynamic in the ViewState string is the delimiter “+”, represented in a post body or URL query string as “%2B”.

Thus, before the captured ViewState can be successfully used, each of these “+” symbols that are interspersed in the string must be substituted with “%2B”. This can be reformatted by passing the value obtained by the “LOAD RESPONSE_INFO” command to a subroutine held in an “Include” file.

The subroutine that performs this task is available to **etest associates’** OpenSTA clients.

<http://opensta.etest-associates.com>

This subroutine must use the SCL commands “~LOCATE” to find the position of the next “+”:

```
SET PortionLen = ~LOCATE ("+", TEMP_VSTATE)
```

It is the text prior to that position that is of interest. Using “~EXTRACT”, that section of the ViewState can be copied to another variable, appending “%2B”:

```
SET VSTATE = VSTATE+~EXTRACT(0,PortionLen,TEMP_VSTATE)+"%2B"
```

That portion and the “+” is then removed from the temporary ViewState variable:

```
SET OFFSET = PortionLen+1
SET TEMP_VSTATE = ~EXTRACT(OFFSET,5120,TEMP_VSTATE)
```

This must all be repeated until the temporary ViewState variable is empty. It is occasionally necessary to substitute trailing “=” symbols.

The VSTATE variable can then be used in the original script, instead of the ViewState string originally recorded:

```
,BODY "__VIEWSTATE="+VSTATE+"%3D%3D&hdnIncom:
```

Limitations

In a case reported to the OpenSTA user-forum, a performance test engineer was unable to test a .NET application because the ViewState had exceeded 64KB.

OpenSTA has an artificial limitation of 64KB for character variables. If the HTML alone was 64KB, it would be considered a ‘fairly large’ web page, let alone one hidden form-field of that size. Microsoft emphasise that ViewState should be used for ‘small’ amounts of data, otherwise, ‘Session State’ should be used.

There is, however, at least one common situation when this 64KB limit might be exceeded and that is if a prototype is being built. ViewState may have been used arbitrarily to allow the prototype to be built with minimal development effort. If this is the case then it probably isn’t the appropriate time to be load testing anyway, as it is the common wisdom that ‘performance’ must be designed into the application.

If, however, this becomes an issue with a design destined for production then it is quite possible that the developers don’t realise that there is a price to pay for the benefits of ViewState.

In Conclusion

With some simple scripting in SCL, OpenSTA will have no problem testing pages that take advantage of the ASP .NET ViewState functionality. This is assuming that ViewState is used appropriately.

The OpenSTA script-error caused by a ViewState string that is too large should be interpreted as an early warning of some potentially serious performance issues. These issues should probably be addressed before taking load testing any further.

The ViewState value, if used inappropriately, brings with it a cost to performance. The most obvious impact is to users with limited bandwidth, affected by the increased page size resulting from an excessively large ViewState.

Perhaps more significantly, the application server is likely to suffer from increased ‘managed memory’ usage, not to mention the additional CPU overhead required to instantiate a ‘ViewState heavy’ page. For a system under high-loads, this alone could result in the alienation of users, or worse, be the catalyst to the server’s ultimate demise.

Links to Additional Resources...

OpenSTA

“Addressing a DOM Element” in OpenSTA:

<http://www.opensta.org/docs/ug/os-model.htm#49405>

OpenSTA Homepage: <http://www.opensta.org/>

OpenSTA Forums and Articles: <http://portal.opensta.org/>

Commercial Technical Support: <http://opensta.etest-associates.com>

Microsoft .NET Application Tuning

Microsoft article on getting the best from ViewState:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspnet/html/asp11222001.asp>

Excessive ViewState as a cause of ‘Managed Memory’ Problems:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspnet/html/monitor_perf.asp

‘Stateless’ vs. ‘Stateful’ Interaction

http://whatis.techtarget.com/definition/0,,sid9_gci213051,00.html



etest associates is a Software Testing consultancy that excels in Load, Stress & Performance Testing. They provide comprehensive Software Testing Services and is the leading provider of OpenSTA consultancy and technical support services in the UK and Europe.

Software Testing Consulting:

<http://www.etest-associates.com/services>

Software Tester & Consultant Recruitment Services:

http://www.etest-associates.com/recruit_home.htm

The Author, Antony Marcano has over 8 years of experience in Software Testing, working on some of BT's first e-business applications. He moved on to play a significant part in projects such as the initial launch of Vodafone.net, the re-launch of FT.com in 2000 and the first release of Barclay's Corporate Internet Banking.

An experienced Test Manager and testing practitioner, Antony contributes regularly to the OpenSTA user's forum. He has been recognised as "one of the few experts in web testing" by Steven Splaine (author of the Web Testing Handbook) and as a "credible expert on e-business testing methods, techniques, tools and commercial realities" by Neil Thomson (co-author of Risk Based E-Business Testing).

COPYWRITE NOTICE: THE COPYRIGHT HOLDER GRANTS PERMISSION FOR THIS DOCUMENT TO BE FREELY DISTRIBUTED UNDER THE PROVISION THAT THE WHOLE DOCUMENT, INCLUDING THIS NOTICE, REMAINS UNALTERED. ANY PARTIAL OR EDITED REPRODUCTION OF THIS DOCUMENT IS NOT PERMITTED WITHOUT THE SPECIFIC WRITTEN PERMISSION OF ETEST ASSOCIATES (UK) LTD.
