SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

Load Testing Web Services in ESA with Custom Examples

Applies To:

This tutorial is applicable to automated load testing of custom-developed web services in SAP Enterprise Services Architecture (ESA) based on the NW04 release.

Summary

This paper describes how to use a third-party tool to load test web services developed within ESA based on NW04, and presents examples that include basic authentication, Single Sign-On (SSO), verification of SOAP response, and parameterization of input arguments. The script can either be created based on a client simulation and recording the client, or by using the WSDL file to create a script with valid input arguments and authentication either by username and password or by SSO cookie.

By: Swapan Saha

Company: SAP Labs, LLC

Date: June 6, 2005

Table of Contents

Introduction

An ESA landscape based on SAP NetWeaver provides web services (WS) that are self-contained and self-describing application functions that can be processed through open Internet standards. Web service publishes web services in Web Service Description Language (WSDL), and a web service client sends a request using the SOAP (XML over HTTP/HTTPS).

The basic concept of ESA and its relevance in today's constantly changing business environment is presented well in ESA Introduction on SDN, and hence skipped here. However, load testing of web services is very important for stability, scalability, and hardware sizing. This paper extends a concept paper based on the idea of load testing of web services in ESA by a third-party tool. The idea here is how the basic concept can be extended with more realistic web services examples developed with the NW04 stack, which needs handling of SSO cookies, input argument parameterization, verification of output, and sometimes to automate a client simulation.

Target Audience & Scope

The target audience of this document is all technical persons (development and consulting) who will run load testing both at product development and project implementation stages that have an understanding of both load testing with automatic scripts and architecture of web services in ESA. The requirements of load testing of business scenarios are well understood and not discussed here. The general methodology of load testing and requirement gathering are not covered here, but they are available on SDN in a set of documents created for SAP Enterprise Portal (SAP EP) load testing.

The objective is to present a procedure for how to perform custom-developed or newly-developed WS load testing using Mercury LoadRunner (LR). Hence we do not present any performance results from any installation or any particular project or release. Once a script is created, it can be used within LR Controller to run load tests in the same way as other protocols (for instance HTTP/HTTPS). The objective here is how to automate WS testing scripting with a third-party tool with realistic examples in an ESA.

Prerequisites

Web Service Testing

It is recommended to review the introductory paper written on out-of-box web service testing using Mercury LR that covers basic concepts and procedure. This paper extends the former with more realistic examples.

LoadRunner

This document is based on LR version 8.0 and focuses on scripting and runtime settings of WS scripting (VUGen). When the VUGen script works, it can be used to run load tests following the known procedure under LR Controller. For any questions on how to run loading from a VUGen script using LR Controller, please follow the "How to Paper" on load testing SAP EP published on SDN, or similar papers available within http://service.sap.com. For LR software and license, please contact Mercury directly.

Test Landscape

The testing of the WS by LR depends on the availability of WSDL files for working WS.

To begin testing your own WS with LR, the following is required:

- An accessible WS endpoint (SOAPAddress).

- WSDL file(s) that describe the service contract and any complex types such as XML Schema.
- Any user IDs or authorizations needed to interact with the service (either basic authentication or SSO).
- Input arguments to be used in the request.

Procedure

General Check List

- ✓ Make sure you have LR Controller with WS VUser Type license.
- ✓ Make sure you have a test system with working web services. Many of them can be tested with either a client or from test web page.
- ✓ Download WSDL files for the web services to be tested. For out-of-the-box examples in NW04, unzip the download files in different directories as all examples have the same file name, main.wsdl, which calls other WSDL files. Stop here if you have not downloaded the WSDL files.
- ✓ In case you need to send parameters as input arguments, collect the information. For the example presented in this tutorial, we do not need an input parameter.

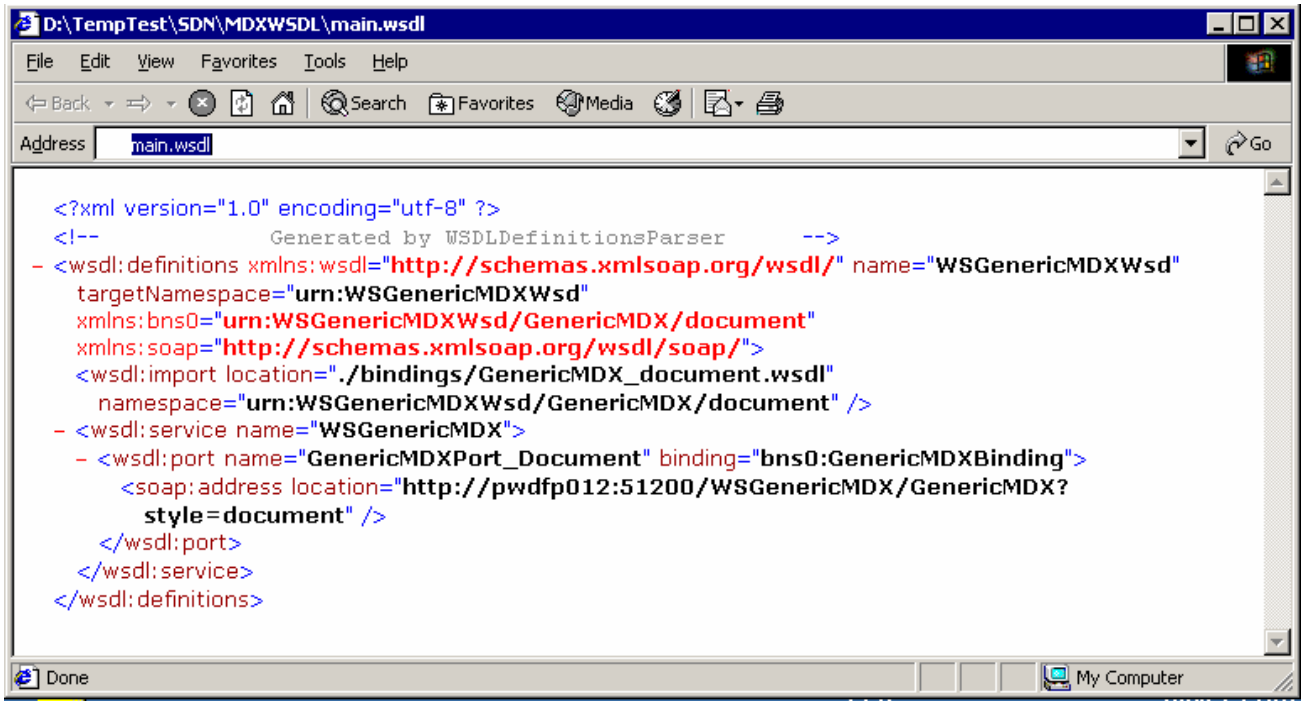
Scripting Procedure & Examples

Example 1 – Basic Authentication and Input Parameters

This example is based on a basic Multidimensional Expressions (MDX) query structured in a fashion similar to the following example.

```
SELECT
[Measures].MEMBERS ON COLUMNS, NON EMPTY [NKSEASON].[LEVEL01].MEMBERS
DIMENSION PROPERTIES [NKSEASON].[2NKSEASON] ON ROWS
FROM [NKSEASON/ZQT_GPO_SEASON_LOV]
```

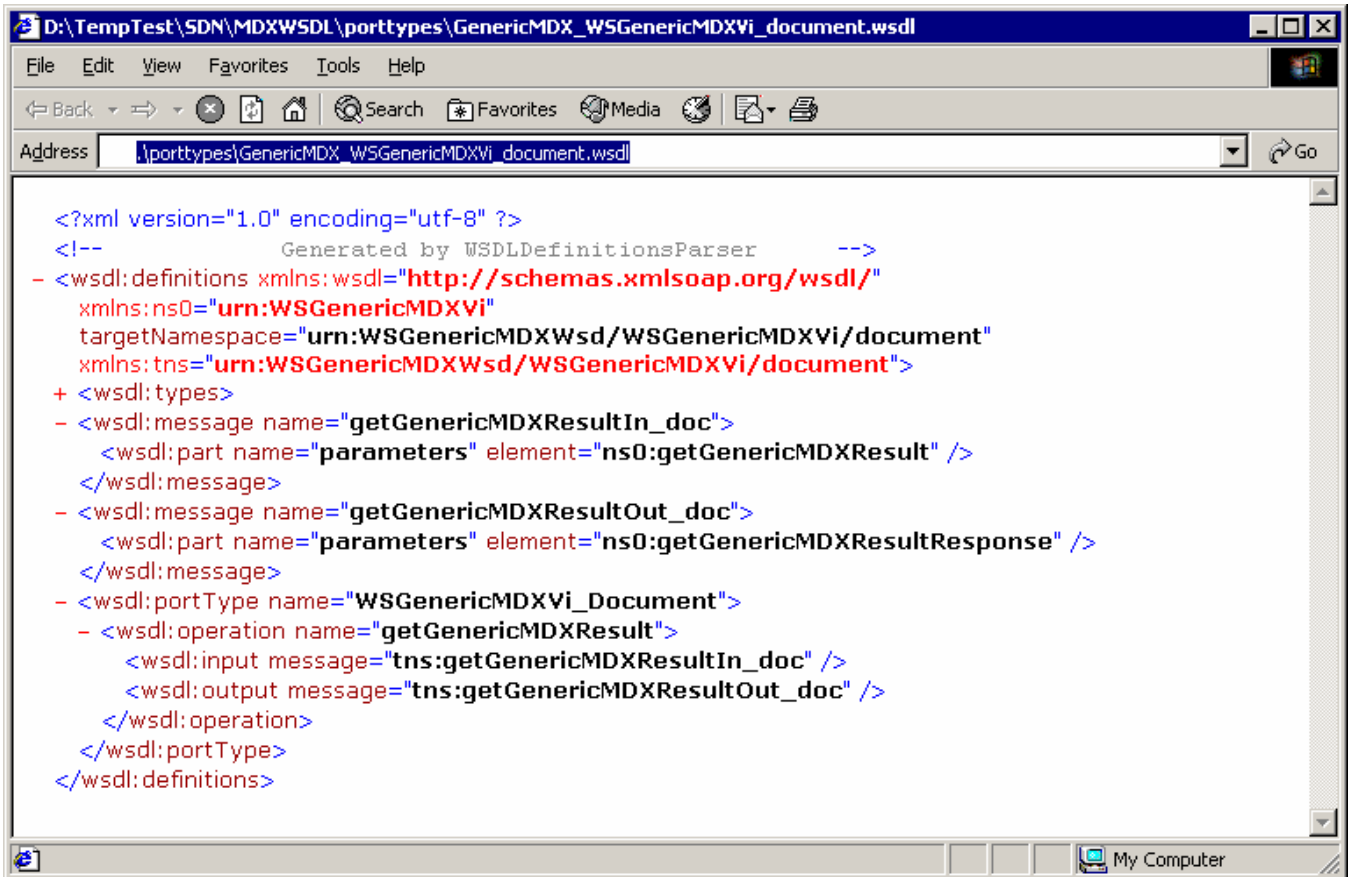
The WSDLs used in this example are presented below. It has one method for the query.



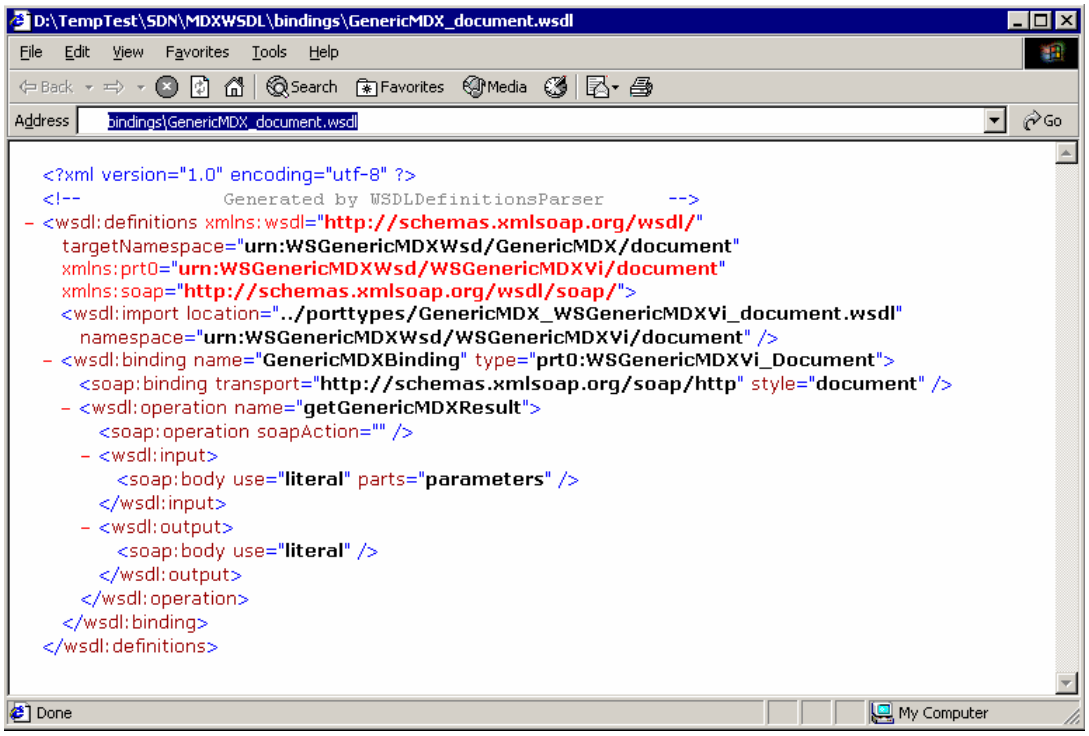
LR reads the main.wsdl file, which reads the following two WSDL files:

./bindings/GenericMDX_document.wsdl

./porttypes/GenericMDX_WSGenericMDXVi_document.wsdl



In this WSDL, we do not expand “types” to save space.

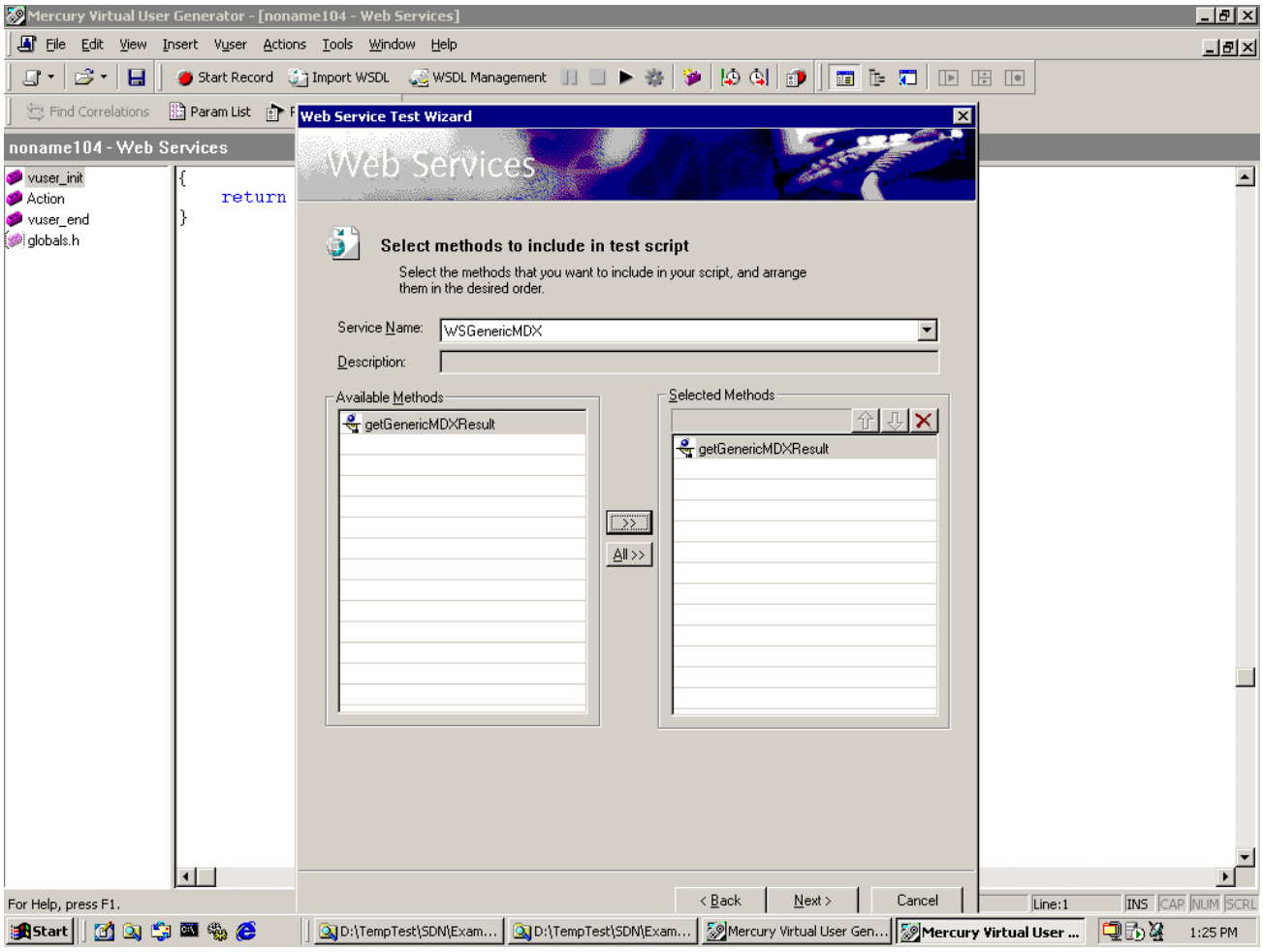


A web service request (SOAP) containing the above MDX query is sent to SAP BW with a valid username and password based on the above WSDL files. The following steps are followed to create a LR script and run the script.

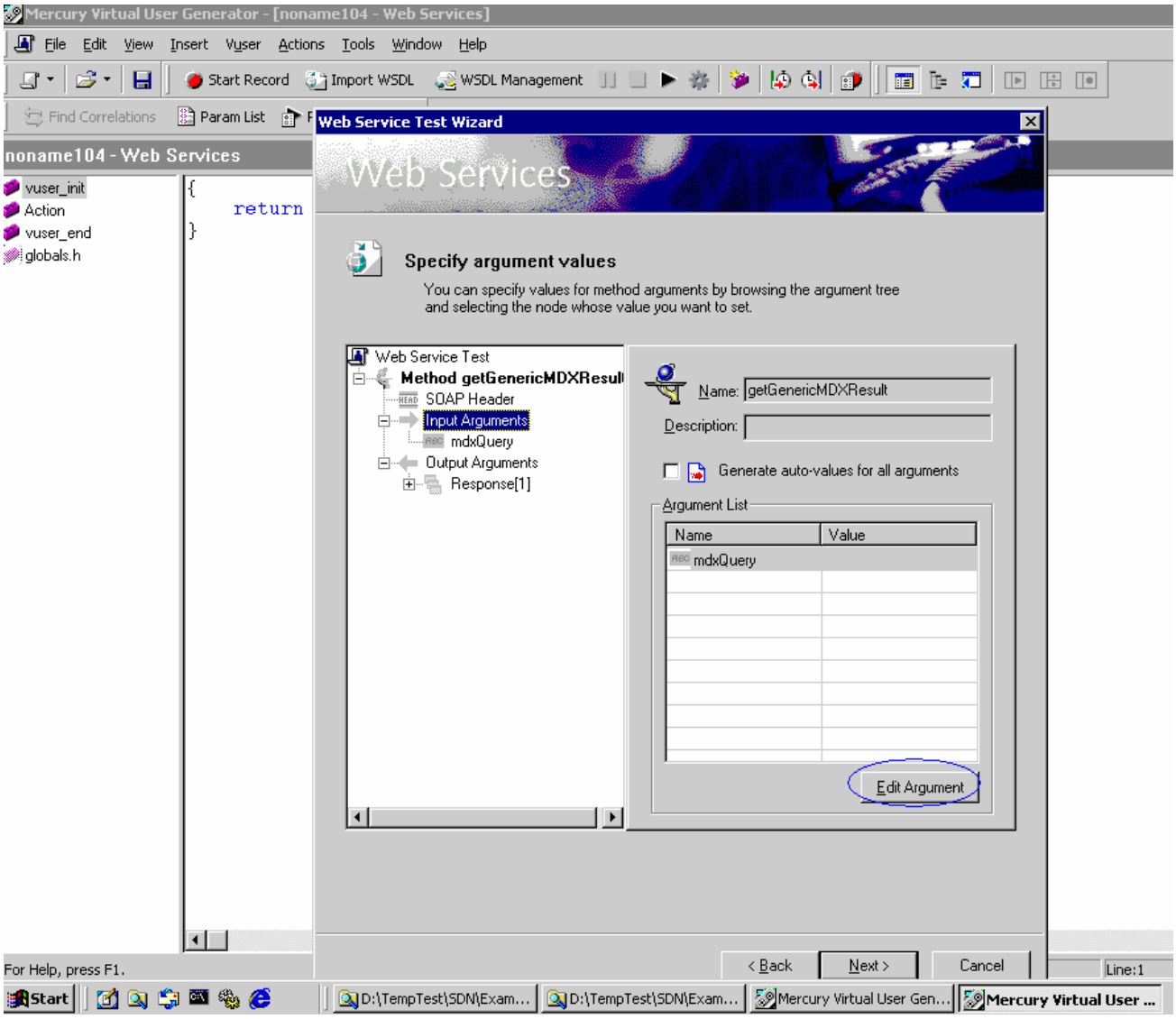
Step 1: From File Menu of Mercury Virtual User, choose New and choose Web Service

Step 2: Choose Scan WSDL and open the WSDL file as shown below

Step 3: Choose your method to be used in sending the SOAP. Here *getGenericMDXResult* method is selected from the WSDL file.

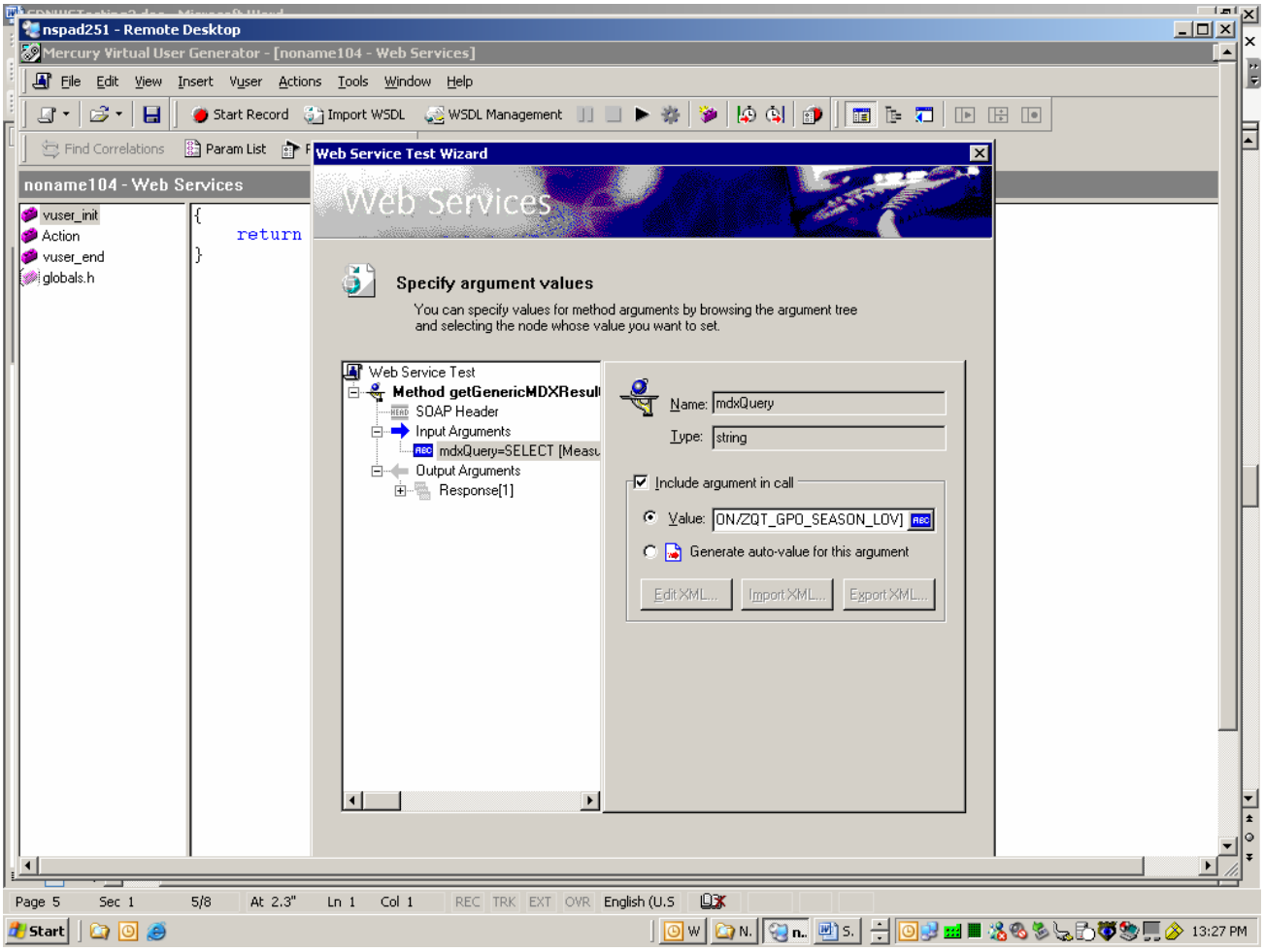


Step 4: As shown below, enter the Input Arguments.



For the argument *mdzQuery*, the following parameter is entered as follows:

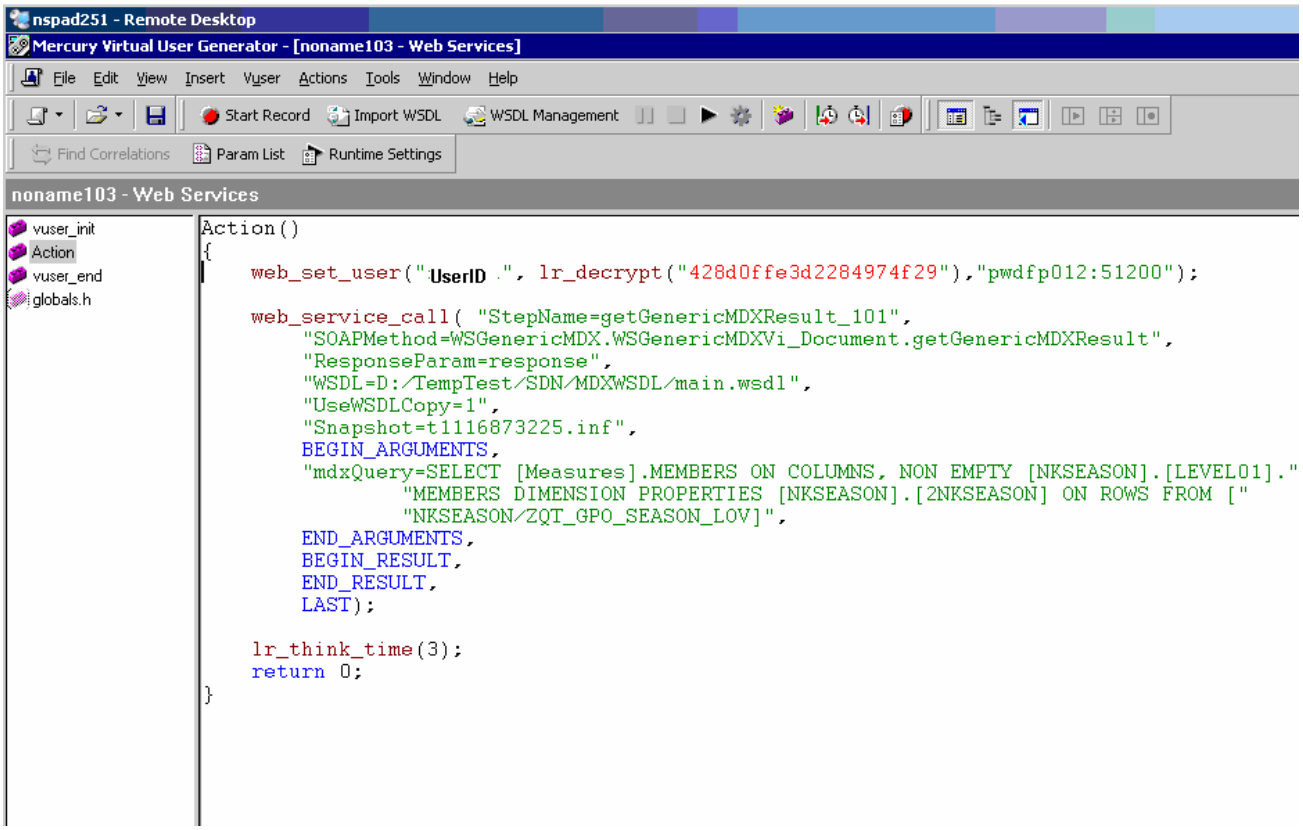
```
SELECT [Measures].MEMBERS ON COLUMNS, NON EMPTY [NKSEASON].[LEVEL01].MEMBERS
DIMENSION PROPERTIES [NKSEASON].[2NKSEASON] ON ROWS FROM
[NKSEASON/ZQT_GPO_SEASON_LOV]
```



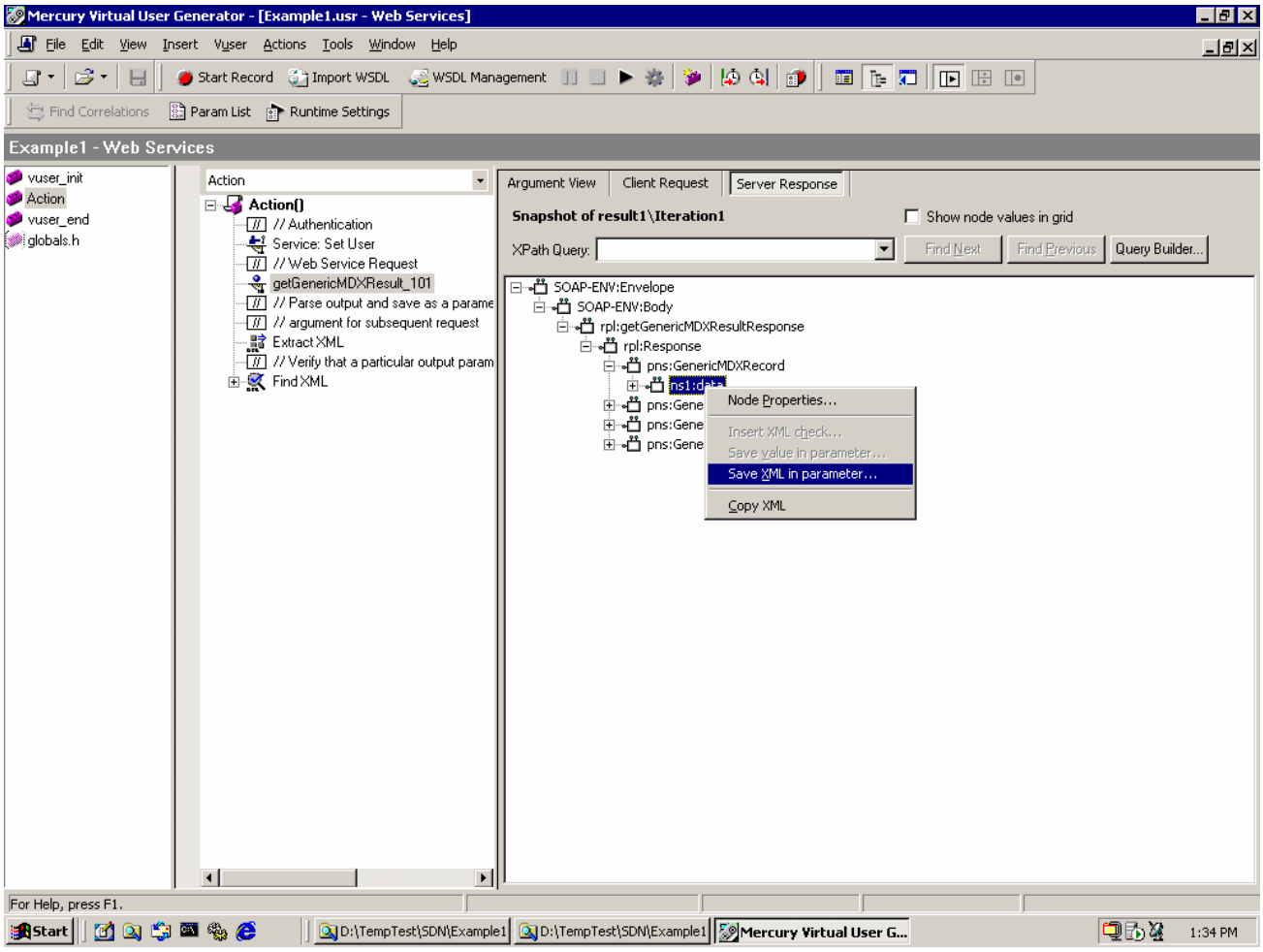
Step 5: Save the script.

Step 6: Add web_set_user("<username>","<password>","<host>:<port>") for authentication.

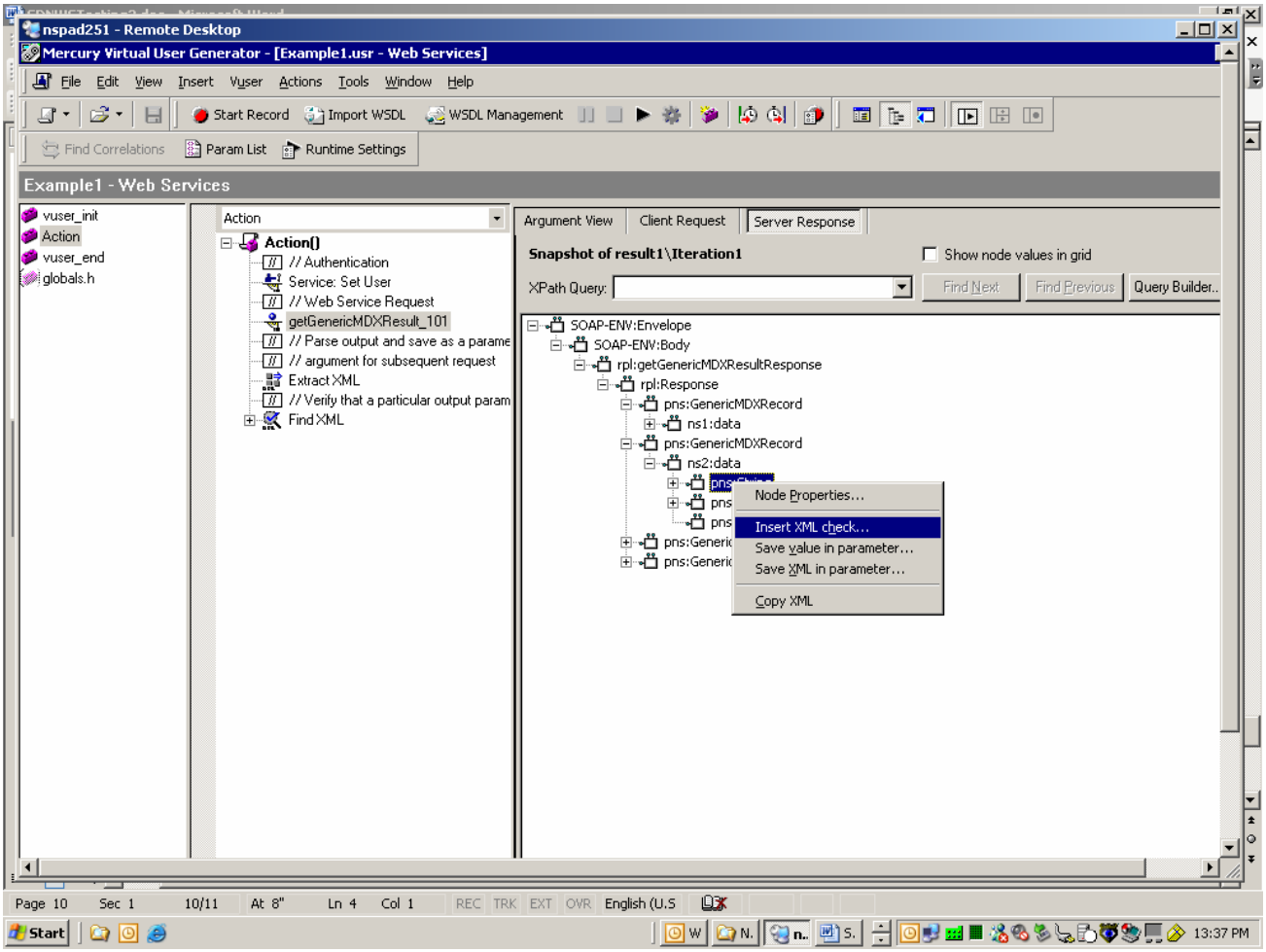
The script will look like the following with username "UserID" and encrypted password.



Step 7: For parsing output and verifying the output, view the script under “tree” view and click on the web service call and click on Server Response tab. Right-click on any parameter as shown below; the return value will be saved as a parameter.

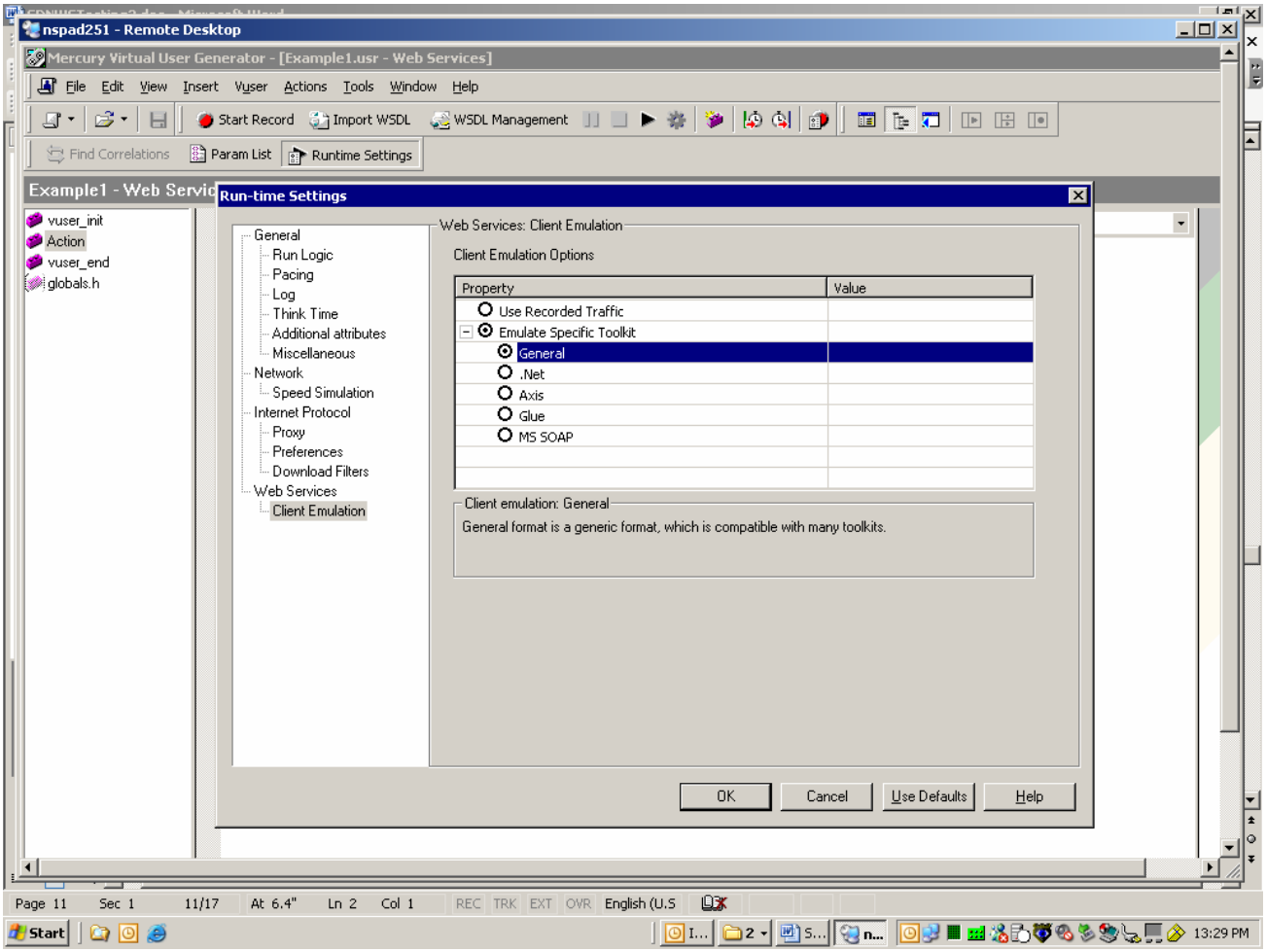


To verify whether the output parameter is correctly returned, right-click on "Insert XML check" as shown below.



The following screen shot presents the complete script with all the steps.

Step 8: After saving the script, set the runtime settings. The additional setting for a web service is only client simulation and the rest of the settings are identical for web (HTTP/HTML) as they are in the document “How to Load Testing EP” or a similar document. For an example, the “General” format is chosen.

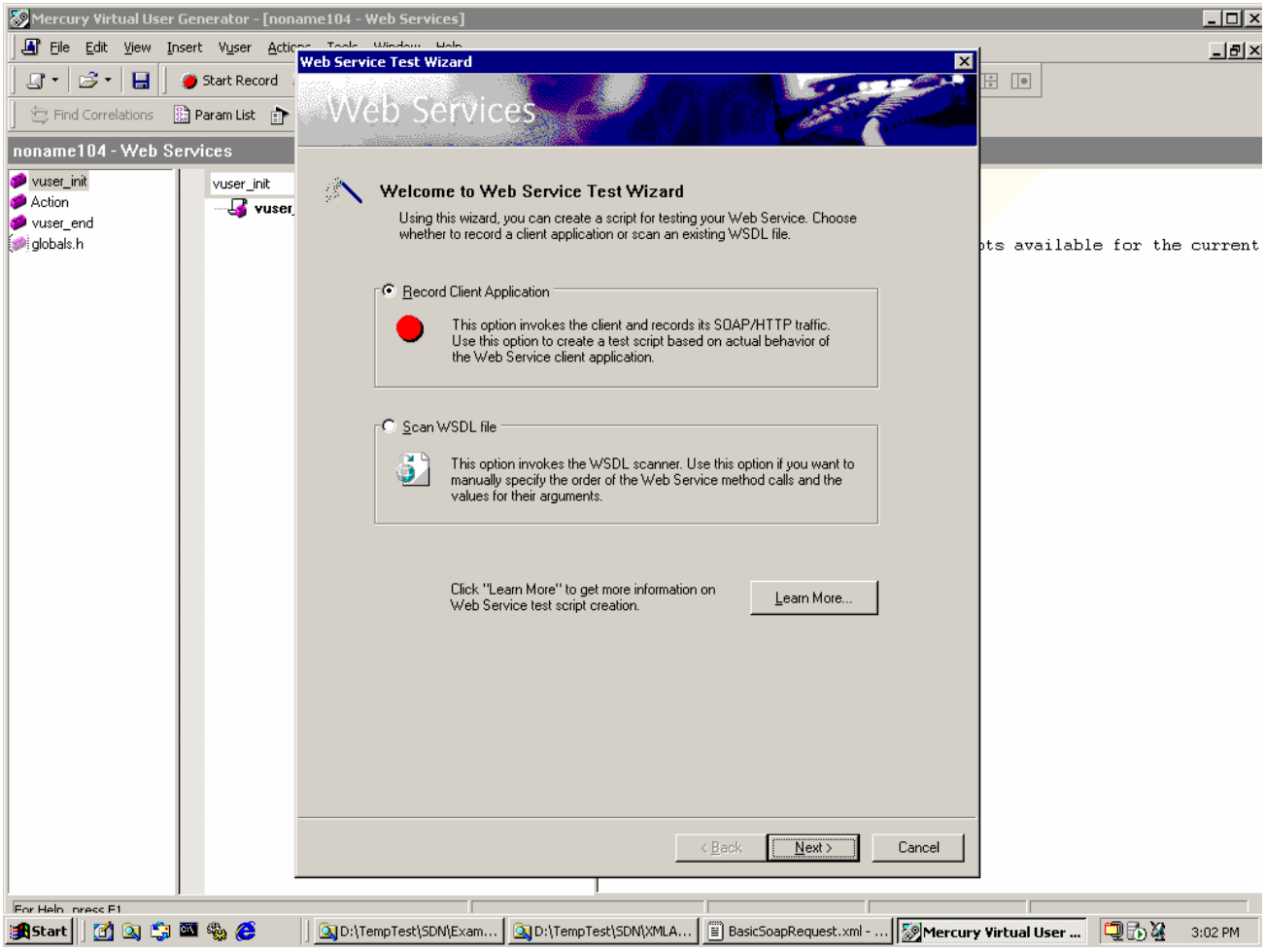


Example 2 – Simulating a Client and Handling the SSO Cookie

This example shows how to automate a client simulation of a web service using LR tool. It also shows how to send a SOAP object with an SSO cookie obtained after login. Here, the first part is get an SSO ticket from the portal server using a valid username and password. Once the SSO ticket is obtained that is valid for the specified domain, it is used to send a HTTP post.

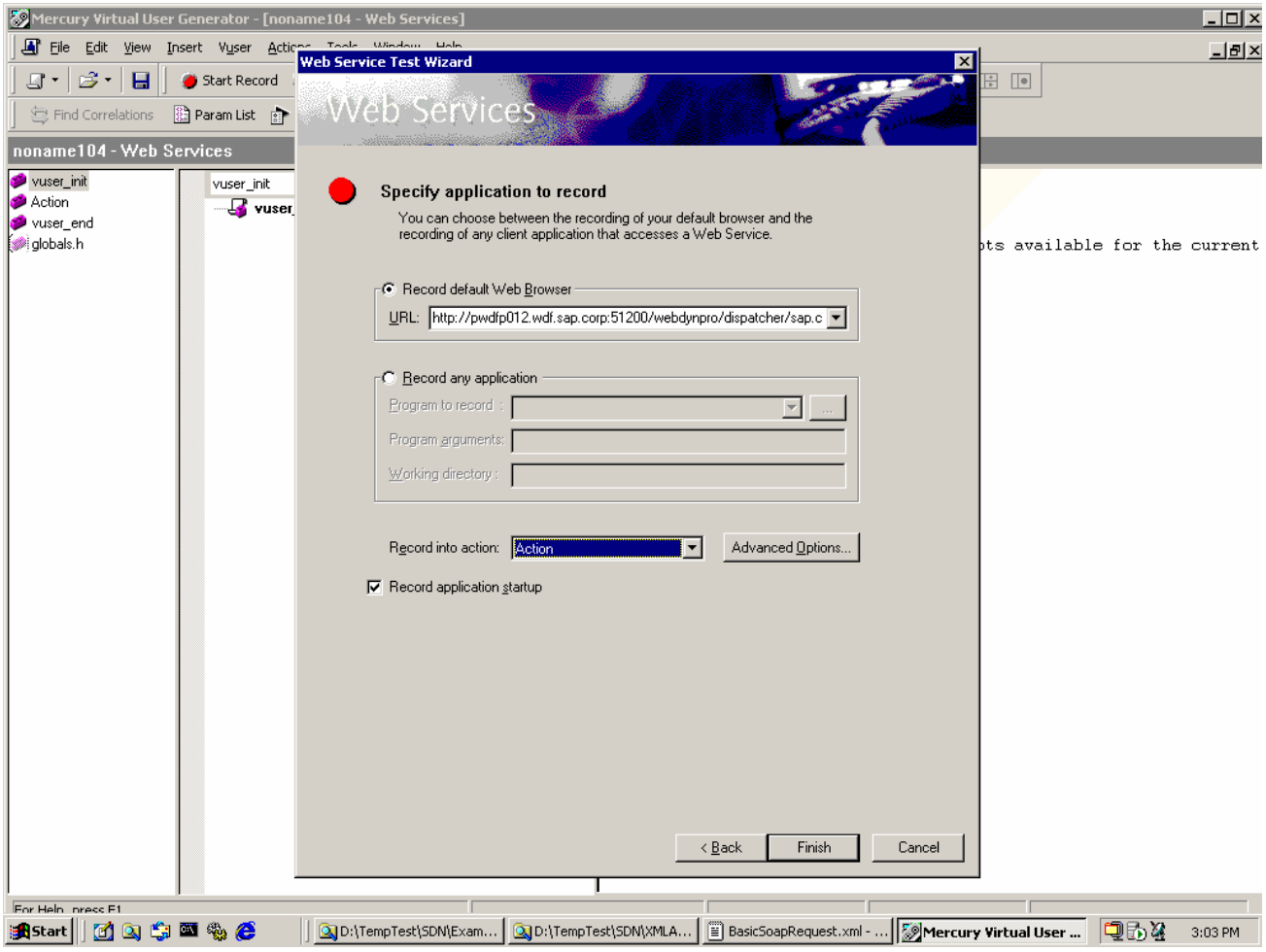
Step 1: From File, choose New Web Service in UVser tool.

Step 2: Choose Record Client.

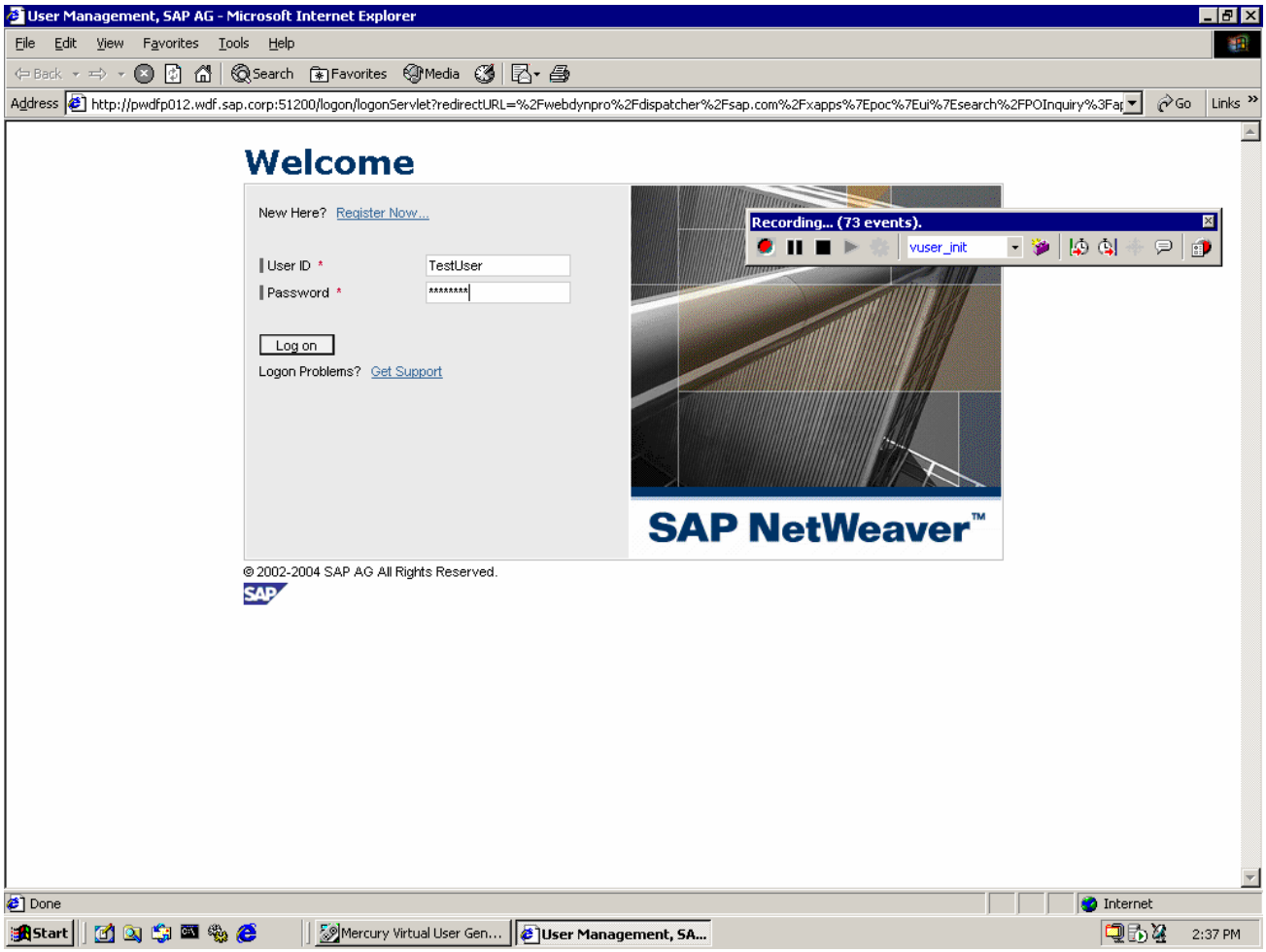


Step 3: Choose "Do not use WSDL file".

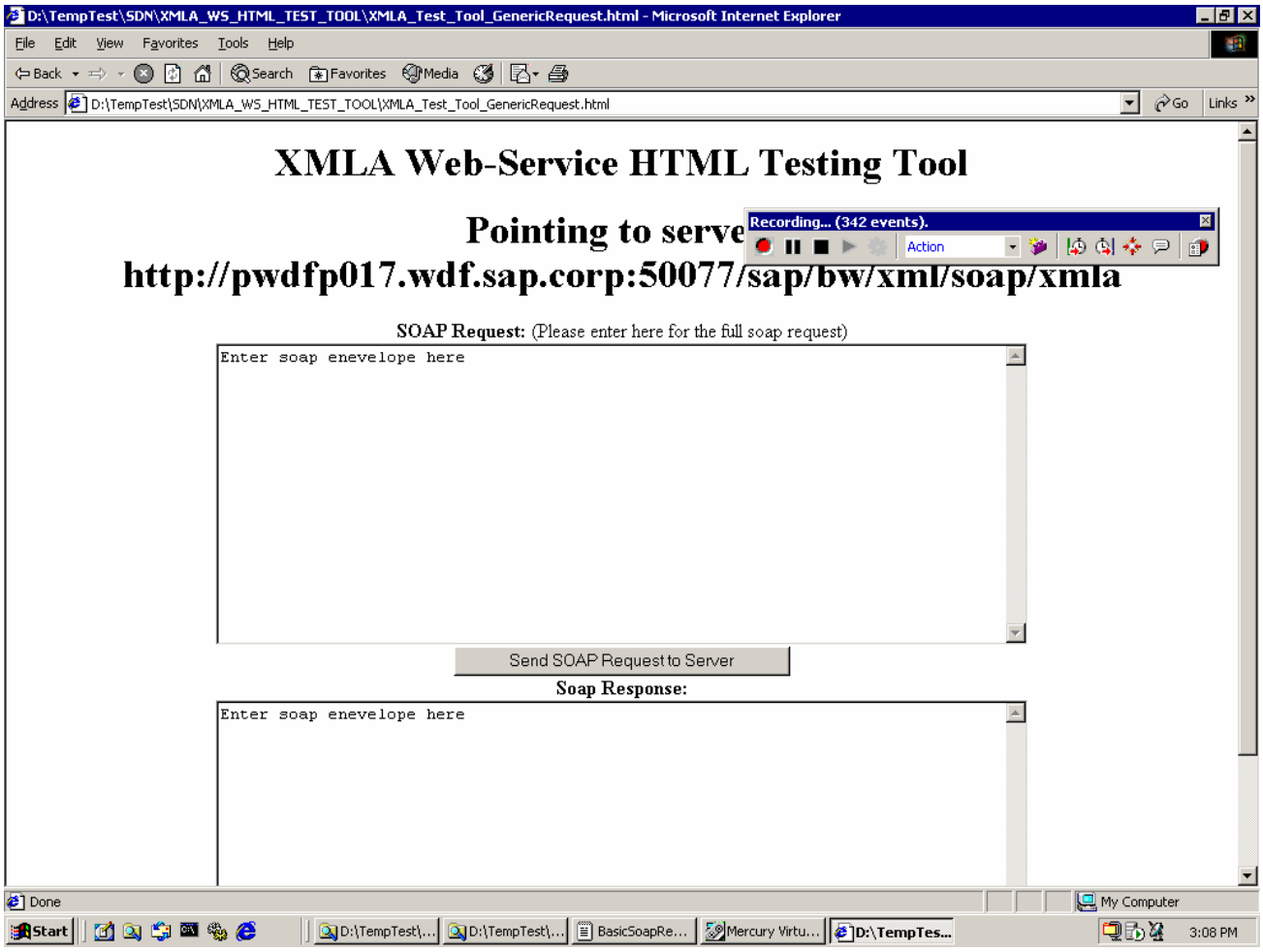
Step 4: Specify the URL as shown below. The client simulation requires access to the application using a browser. Here Microsoft Internet Explorer (IE) 6.0 is used.

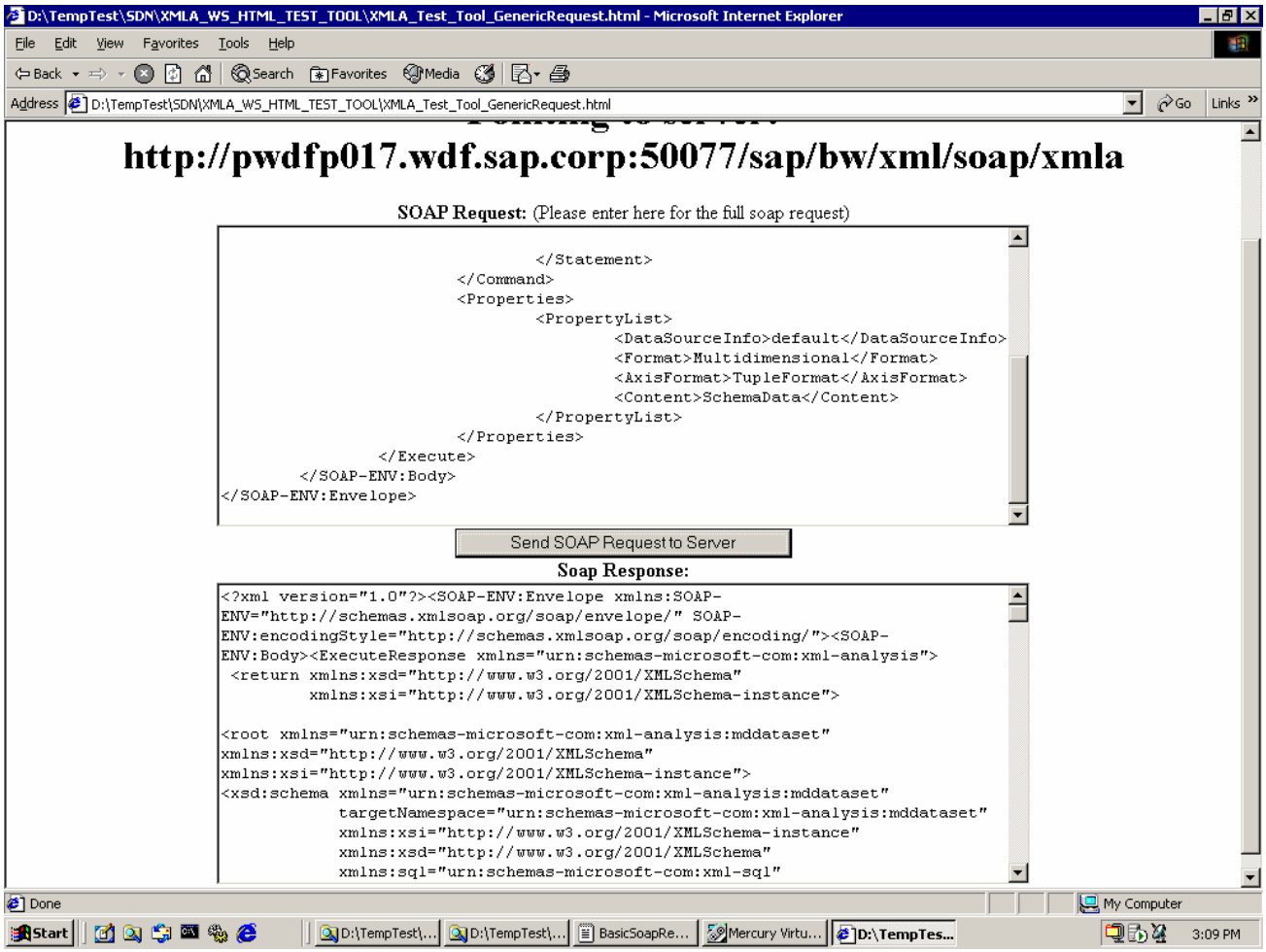


Step 4: Click on finish and wait for the browser and continue the operation with the browser. Here it will bring a URL and an SOAP object will be entered as shown below. After hitting the submit button, it will fetch a SOAP response. The following portal login is done to get an SSO cookie valid for a particular domain.



After successful login, a URL is invoked that brings the following form to enter a valid SOAP object. After entering the SOAP object, LR simulates the “send” using the SSO cookie it received during the login process and to receive a SOAP response.


D:\TempTest\SDN\XMLA_WS_HTML_TEST_TOOL\XMLA_Test_Tool_GenericRequest.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address D:\TempTest\SDN\XMLA_WS_HTML_TEST_TOOL\XMLA_Test_Tool_GenericRequest.html
Go Links
XMLA Web-Service HTML Testing Tool
Pointing to serve
Recording... (342 events).
Action
http://pwdfp017.wdf.sap.corp:50077/sap/bw/xml/soap/xmla
SOAP Request: (Please enter here for the full soap request)
Enter soap enevelope here
Send SOAP Request to Server
Soap Response:
Enter soap enevelope here
Done
My Computer
Start
D:\TempTest\...
D:\TempTest\...
BasicSoapRe...
Mercury Virtu...
D:\TempTes...
3:08 PM



This example shows how to use both a web (HTML/HTTP) scripting process along with a web service client simulation to automate the sending of web server requests and create a script that can be run to simulate the web service request. In this example, the SSO cookie obtained from the portal server is used to send an HTTP POST message. A similar concept can be extended to use SSO tickets to send a SOAP message using a WSDL file, as explained in the first example.

In both examples, default runtime settings are used.

Conclusion

This is a follow-up paper on how to automate load testing of web services developed with the NW04 stack in SAP ESA. The general process is presented and examples are picked up in such a way that it covers the NW04-supported web service features – basic authentication, authentication with an SSO cookie, and also general requirements of parameterization of input arguments and verification of output. This paper also discusses how we can use a third-party tool to automate load testing of web services developed with NW04. The author acknowledges Mercury for providing LoadRunner license and support for this investigation and comments and collaboration of colleagues in the SAP NetWeaver Solution Office for the examples used in this article.

Author Bio

Swapan Saha has been with SAP for nearly four years. He received his Ph.D. from Iowa State University. He has been working in software development for the last ten years, including the last five years in performance engineering and management. His current focus is on performance and monitoring in SAP NetWeaver platform.