

Automated Testing - Web Services

“WSATS”

Contents

1.0	Introduction	3
2.0	Intended Audience	3
3.0	Problems in Testing Web Services	3
3.1	General Manual Approach	3
3.2	Automated Approach	3
4.0	Case Study – WSATS.....	4
4.1	Design	4
4.1.1	Configuration and Input Files.....	4
4.1.2	Runtime Engine	5
4.1.3	Report Generator.....	5
4.2	Future Enhancement for WSATS	5
4.3	Benefits & Features of WSATS.....	6
5.0	ROI on WSATS	6
6.0	Closing Note.....	7

1.0 Introduction

Web services are more and more used for the realization of distributed applications crossing domain borders. The below topic explains need for automation testing of Web Services and Case Study of Automation testing Tool for Web Services.

2.0 Intended Audience

Readers should have basic Knowledge of Web Services, awareness about automation testing and some knowledge of object oriented testing. Ideally suitable for:

- QA Engineers working on Web Services Regression Testing
- Managers handling development and Testing of Web Services projects
- Developers working on Test driven Development projects

3.0 Problems in Testing Web Services

Consumers of Web Services are a wide variety of applications built using different technologies like c#, Vb.net, Java, Perl, VC++ so on so forth. As Web services are UI less and it becomes challenging for QA Engineers to test them.

3.1 General Manual Approach

Manual testing of Web Services is mostly done by small open source/free utility software like .Net Web Service Studio, SOAP UI etc. used just to invoke a web method. These tools do not provide any mechanism for reporting the complete status of testing. Challenges faced in manual testing for frequent builds/ releases can be:

- Resource dependency
- Time Consumption
- Monotonous
- Inaccurate response time report
- Choice of Unit testing tool
- Reporting mechanism

3.2 Automated Approach

The major challenge faced in automation is the lack of robust and dependable testing tools that are easy to use by any QA Engineer. Automation testing would certainly solve the manual testing issues but if the choice of automation tool is in-appropriate it would lead to a new set of issues. Once the Automation setup is ready, the advantages it has to offer are endless, but a few are listed below:

- Significant reduction of Testing Time
- Consistent Reporting
- Accurate response times
- Resource Independent
- Increased Productivity and Turn Around Time
- Load/ Stress Testing features

4.0 Case Study – WSATS

A consistent need was observed for having a generic tool to test web services across projects. Thus, a tool was conceptualized and developed to overcome manual testing challenges and to increase productivity of QA Engineers across the projects. This paper further discusses the design pro's and con's of the tool, Web Service Automation Testing Studio (abbreviated as WSATS).

4.1 Design

The Design focuses on automation of almost all the Xml/SOAP Based Web Services. The WSATS consist of three major components:

1. Configuration and Input Files
2. Runtime Engine
3. Report Generator

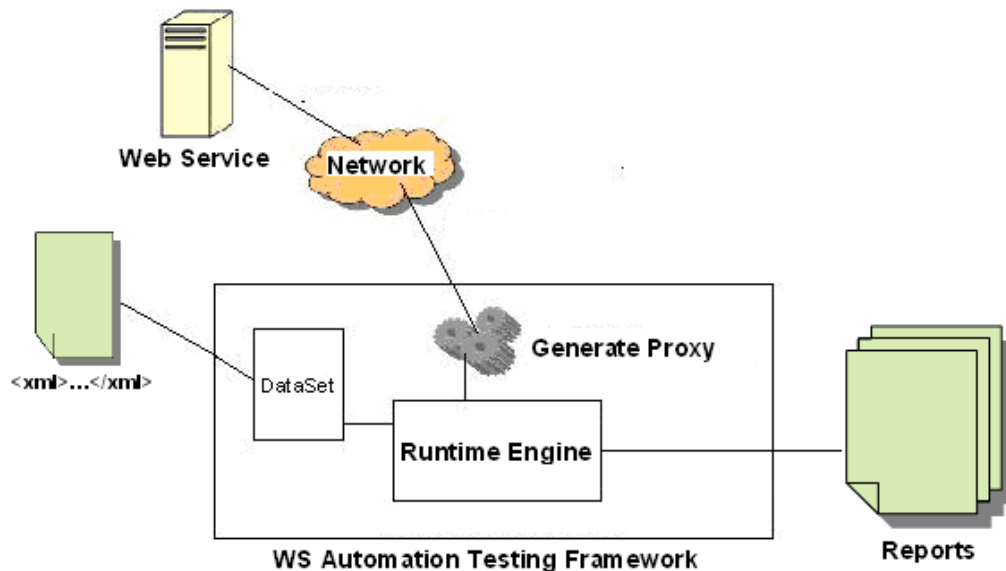


Figure 1: HLD WSATS

4.1.1 Configuration and Input Files

WSATS processes the "configuration XML" file for the initial settings. Then, WSATS feeds on the "test case XML" file and processes it and extracts individual test level information. The test level information in XML is the test name, input/output conditions, comments for the test cases, expected results and all the details pertaining to execution of the test case.

4.1.2 Runtime Engine

This subsystem is the heart of the WSATS. It acts as a *controller* that administers the other subsystems, initializes the whole system, manages the Proxy instances, and co-ordinates interactions necessary among all the subsystems. The Runtime Engine is also responsible for comparing the Actual results with the Expected results and marking the test case with the Pass/Fail Status.

4.1.3 Report Generator

This subsystem processes the test results, extracts the necessary information, and exports it to an MS Excel file. All the details like inputs, methods executed, Exceptions if any, Results (Pass/Fail) and user comments are logged to the Excel file.

The snapshot below details how the WSATS consumes the MSN Web Service that exposes “Search web method”. Once the Test Suite is created Automation testing is a three step process.

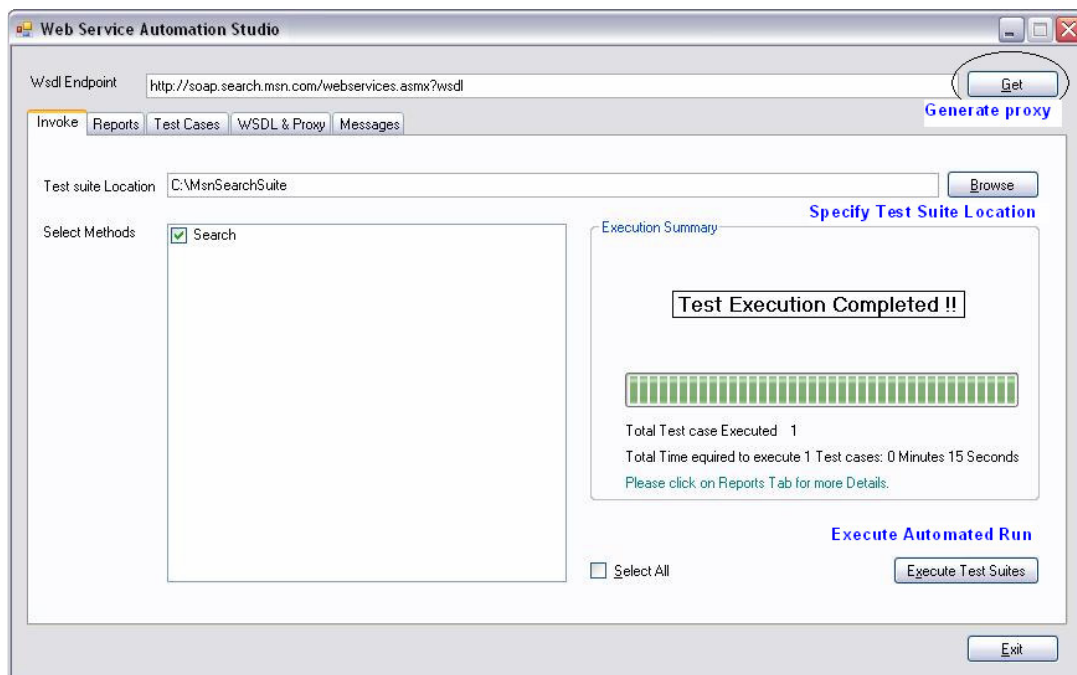


Figure 2: WSATS snapshot

4.2 Future Enhancement for WSATS

As part of future enhancement the following features could be easily integrated in the existing design:

- This design could be extended to carry out Load/ Stress Testing, spawning multiple threads which could be configured via configuration file
- The application could be converted to Web application re-using the WSATS Framework classes
- Reporting mechanism could be enhanced to generate customized reports and documentation formats like Html, Xml etc

4.3 Benefits & Features of WSATS

- Reduced Time Consumption and Resource independent
- Accurate & Consistent Reporting
- Email Support & Command line execution
- Test Suite Generator & Execution Statistics on UI
- Customizable settings & User Friendly Messages
- SSL Compatibility

5.0 ROI on WSATS

Looking at one of the projects data involving an average three build cycles/ month gives us a realistic feel of the returns derived from the WSATS development.

Testing Nature	Testing Hrs/Build	Testing Hrs/Month	Testing Hrs/Quarter	Testing Hrs/Year
Manual	16 hrs	16 * 3(builds) = 48 hrs	48 * 3 (Months) = 144 hrs	144 * 4(Quarters) = 576 hrs
Using WSATS	156(Dev. efforts) + 1 (Execution efforts) = 157 hrs	156+ (1 * 3 builds) = 159 hrs	156 + (3 * 3 Months) = 165 hrs	156 +(9 * 4 Quarters) = 192 hrs

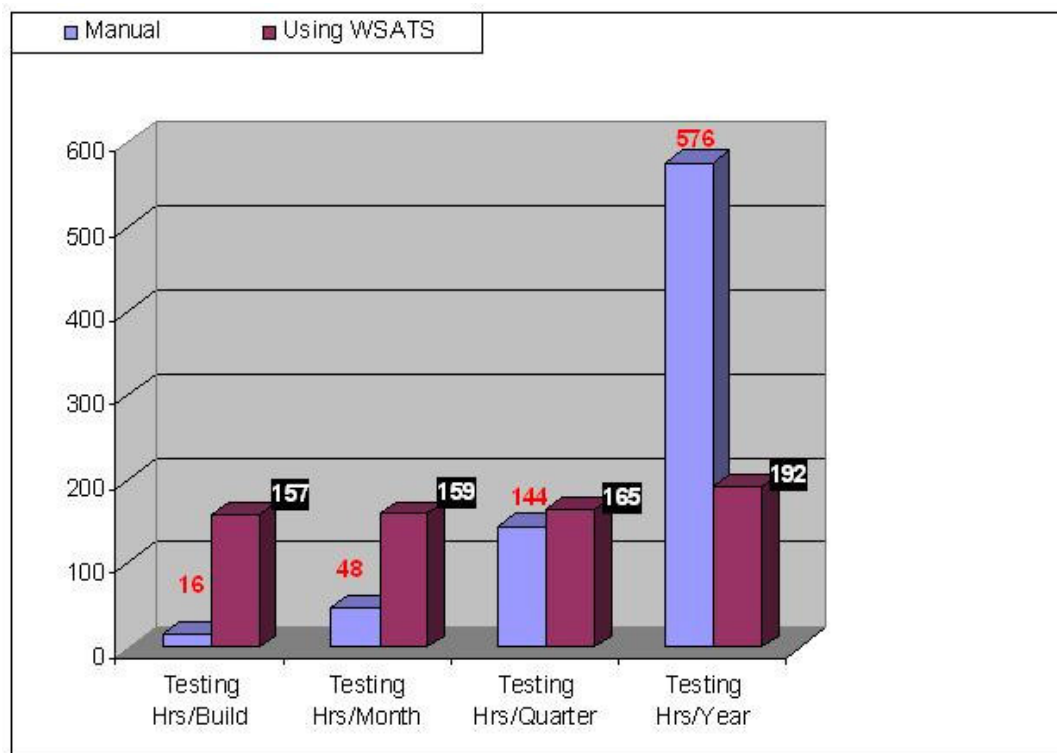


Figure 3: Bar-Chart for testing time saved using WSATS

6.0 Closing Note

Any automation tool would initially involve development efforts as WSATS to reap the benefits with time as seen in the figure three above. The time saved in the one of the projects due to wise conceptualization, design, development of WSATS is almost 384 person-hours or 48 person-days. Of course, there would be lot of features that can be added to the tool to make it more feature rich but at the cost of efforts and the actual requirement.