XMLVend Protocol Message Validation Suite

25-01-2012

Table of Contents

1. Overview

The XMLVend Message Validator is a flexible validation framework which performs protocol-compliance testing by default, and which can easily be expanded to perform business-logic, or any other kind of, testing by plugging in additional validator components.

The application is configured to reside between an XMLVend Client, and an XMLVend server. It accepts all communication from the client, forwards it to the server, and records it to a log for further processing. The component that does the interception and logging is the standard WS-I (Web Services Interoperability Group) *Monitor* (www.ws-i.org). Thus, standard WS-I testing tools can be applied to this recorded log file, in addition to the XMLVend validation tools.

The XMLVend Report Generator processes this log, producing compliance reports grouped by `ClientID to Server IP Address`. These reports are in the `.xvreport` XML format, which allows for easy access to message sequences, and any error(s) they contain. They are also transformed to a customisable HTML format, which makes it easy to see (at a glance) how well a particular server or client complies to the specification.

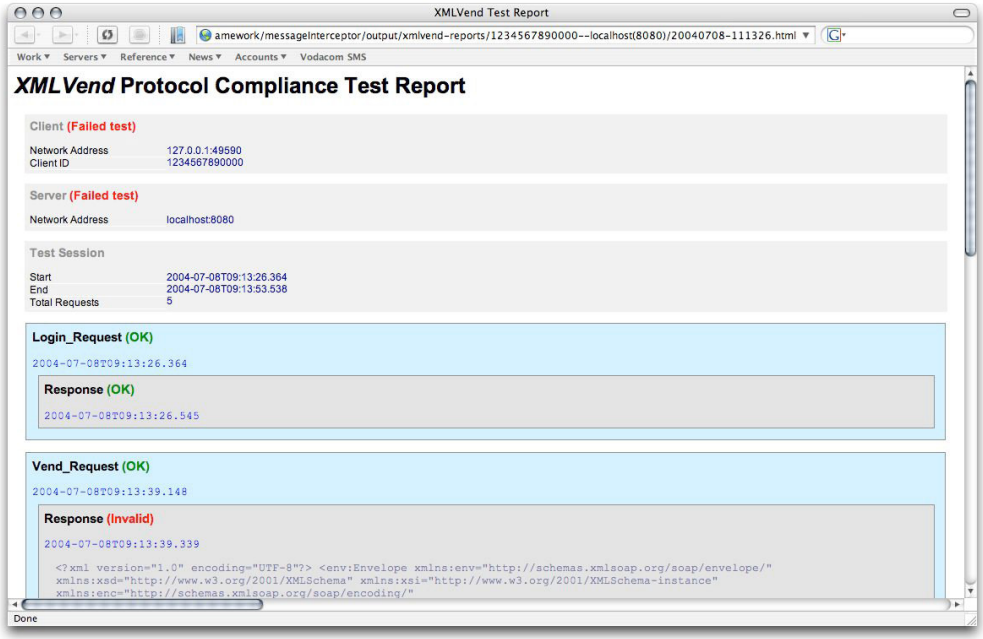

Figure 1. HTML-formatted report

2. Installation and Operational Requirements

The Message Validation Suite is implemented in the Java programming language, using standard tools technology. The following is required on your system before you can make use of the system:

- **Java SDK, version 1.5 or later.** The framework is shipped in source code form, together with all required libraries. The J2SDK is required to compile and run the framework. Due to a reliance on regular expressions, it will not run on earlier versions (e.g. 1.3). The J2SDK can be downloaded from java.sun.com for Windows, Linux and Solaris platforms.

- **Ant (The Apache Building tool), version 1.6 +.** Apache Ant is a Java-based build tool. It is the industry standard for assembling java systems, and is used to both build and run the framework. This enables the avoidance of platform specific batch or shell scripts required to properly configure the java environment to run the application(s). See `http://ant.apache.org` for installation instructions and help. The `ant` executable should be placed on your system path, so that it can be executed from within any directory.

3. Preparing the system

Extract interceptor zip folder into the root directory.

As the system is shipped in source form, it is necessary to prepare it for running by changing into the root directory of the system (`D:\Interceptor\xmlvend-messageInterceptor>)` which contains the `build.xml` file and typing:

```
ant build
```

This only needs to be performed once. The message will look like this after entering the command in the command prompt:

```
Buildfile: D:\Interceptor\xmlvend-messageInterceptor\build.xml

remove-build-dirs:
      [delete] Deleting directory D:\Interceptor\xmlvend-messageInterceptor\build

create-build-dirs:
      [mkdir] Created dir: D:\Interceptor\xmlvend-messageInterceptor\build\classes

compile:
      [javac] D:\Interceptor\xmlvend-messageInterceptor\build.xml:94: warning:
'includeantruntime' was not set, defaulting to build.sysclasspath=last; set to false
for repeatable builds
      [javac] Compiling 145 source files to D:\Interceptor\xmlvend-
messageInterceptor\build\classes
      [javac] Note: Some input files use unchecked or unsafe operations.
      [javac] Note: Recompile with -Xlint:unchecked for details.

move-resources:
      [copy] Copying 4 files to D:\Interceptor\xmlvend-
messageInterceptor\build\classes

build:
```

If the output ends with a message similar to the following, the interceptor is ready for use.

```
BUILD SUCCESSFUL
Total time: 9 seconds
```

Configurations to note:

For the interceptor to recognise the full xml message, remove the compression request [must be false].

The XMLVend Server URL : http://localhost:8082/xmlvend-server/service/

4. Intercepting Messages

The local host will always default to **http://localhost:8080** for the server. Configure which **port(s)** to listen on, and which servers to forward to, by editing the standard WS-I configuration file at: **resources/wsi-monitor-config.xml.**

NOTE: if there is a port conflict, that is if the listening port is used by another programme in the system, you can edit the WS-I configuration file by editing it using notepad++ then saving it; this is the part of the file that must be changed with the alternative port to listen on, as long as it is not the same as the host port 8080.

```
<wsi-monConfig:listenPort>8082</wsi-monConfig:listenPort>
```

It is recommended that the recorded log file location be left unchanged at first, as the report generator expects it there. That can, however, be configured.

To start the interception, in the root directory of the system **D:\Interceptor\xmlvend-messageInterceptor>**, type:

```
ant intercept
```

The message will look like this after entering the command in the command prompt:

```
Buildfile: D:\Interceptor\xmlvend-messageInterceptor\build.xml

intercept:
     [java] WS-I Monitor Tool, Version: 1.0.0, Release Date: 2004-01-22
     [java] Copyright (C) 2002-2003 by The Web Services-Interoperability Organization
and Certain of its Members. All Rights Reserved.
     [java] Use of this Material is governed by WS-I licenses included within the
documentation.
     [java]
     [java]    comment ..................... WS-I Monitor, as part of the XMLVend
Testing Framework
     [java]   logURI ..................... output/wsi-log.xml
     [java]   replaceLog ................. true
     [java]   logDuration ................ 1800
     [java]   timeout .................... 3
     [java]             addStyleSheet    ..............    <!--    <?xml-stylesheet
href="../common/xsl/log.xsl" type="text/xsl" ?> -->
     [java]   man-in-the-middle comment ... null
     [java]   redirect [1]
     [java]     comment .................. null
     [java]     listenPort ............... 8082
     [java]     host ..................... http://localhost:8080
```

```
[java]     maxConnections ..... 100
[java]     readTimeoutSeconds ........ 15
[java]
[java] The Monitor tool is ready to intercept and log web service messages.
[java] Type "exit" to stop the Monitor.
```

You may now point your XMLVend client to the address the system is listening on, and communicate with the server (you will see messages in the console indicating that your communication is being logged).

```
[java]
[java] Log message entry [ID, Type, Sender]: 1, request, 127.0.0.1:1895
[java] Log message entry [ID, Type, Sender]: 2, response, localhost:8080
[java] Log message entry [ID, Type, Sender]: 3, request, 127.0.0.1:1898
[java] Log message entry [ID, Type, Sender]: 4, response, localhost:8080
```

An interception session ends when it times out according to the value set in the WS-I configuration file, or when the ant task is cancelled by pressing e.g. **CTRL+C** and the following message will be shown;

```
Terminate batch job (Y/N)? y
```

When cancelling the ant task, any exception messages in the display may be safely ignored, as these are due to the threading interaction between Ant and the WS-I Monitor.

IMPORTANT

The WS-I monitoring software does often not gracefully terminate the log file upon premature ending of a testing session (the duration of which can be configured in the WS-I configuration file). If this is the case, open and edit the **wsi- log.xml** file in the **xmlvend-messageInterceptor\output** folder, edit the file by typing **/<log>** at the end of the file to close the log file, save and close it. This must be done before another intercepting session can be started or before a report can be generated.

5. Generating Reports

To process the log file of the last interception session (described in the previous section) in the root directory of the system (**D:\Interceptor\xmlvend-messageInterceptor>**), type:

```
ant report
```

Reports will now be generated, and by default transformed with the XSLT style sheet in **/resources/stylesheets/xvreport2html.xsl**. The behaviour of the report generator can be fully customised by editing the file: **resources/reportrunner.properties**.

Reports are generated in the **output/xmlvend-reports/** directory by default, and a subdirectory is created for each client (by **clientID**) interacting with each server (by **IP Address**). Report file paths look like the following:

<root>/output/xmlvend-reports/<clientID>--<server>(<port>)/<date>-<time>.xvreport

Report processing produces output similar to the following:

```
$ ant report
Buildfile: build.xml

report:
      [java] Created output directory: output/xmlvend-reports
      [java] Parsing log file...
      [java] Generating Report(s)...
      [java] Validation: Using 10 schema(s) from resources/schemas
      [java] Persisting Report(s)...
      [java]      Wrote      report:      output/xmlvend-reports/ean.1234567890123--
      localhost(8080)/XMLVend
      [java] Transforming Report(s)...
      [java] Wrote report: output/xmlvend-reports/ean.1234567890123--localhost(8080)/
      [java] Done.

BUILD SUCCESSFUL
Total time: 2 seconds
```

Instructions on adding your own report validators to the framework (for customised testing) will be made available in the next release of this document. The report validation framework allows you to easily add validators for your own business logic or implementation-specific features.