**SoapUI Automation Approach**

**Overview**

This is mainly focused on use of Soap UI Pro and integration using Maven. SVN is using for Configuration Management tool. The test automation framework described as a data driven, and the design is robust and maintainable.

The prerequisite for execution of automation suites would be Maven installed in the system. The automated test cases can be executed directly from Soap UI Pro from a machine that have Soap UI Pro installed in it. Alternatively, one can also opt to run tests directly from Maven with/without having Soap UI Pro installed by choosing the test.

**Brief about Soap UI Pro tool**

Soap UI Pro is a free and open source desktop application for inspecting, invoking, developing, simulating/mocking, and functional/load/compliance testing of web services over HTTP. With an easy-to-use graphical interface, and enterprise-class features, in a single test environment, Soap UI Pro provides complete test coverage and supports all the standard protocols and technologies. Soap UI Pro tests shall also be integrated into Jenkins and run automatically. Testers and Developers can run Soap UI Pro tests locally.

**Soap UI Pro minimum system requirements**

* 1GHz or higher 32-bit or 64-bit processor
* 512MB of RAM
* 220MB of hard disk space for installation (Soap UI Pro)
* Windows XP or later

**Prerequisites**

* Soap UI Pro (4.5.1 or above)
* SoapUI Pro License
* Working WSDL/XSD
* Excel Sheet for DataSource (Web Service Input Test Data)
* Excel Sheet for Summary Report
* S/V environment and RUMBA access should be there

**NOTE:**

EXCEL used for external data store and organizing of the data along with (Pass/Fail) test status.

**Soap UI Pro Project Structure**

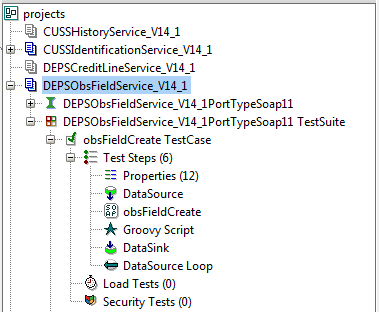
**NOTE:**

Soap UI project contain one Suite with all the test cases (Merging all the operations with one test suite).

**Test Suite**

Test Case - (Use Case/Operation Name)

* DataSource – Input Data
* TestRequest – Web Service Request
* GroovyScript – Validation for Response
* DataSink – Report Generation
* DataSourceLoop – Looping Datasource value to Test Request



**Run existing tests**

1. Before running the soapui test, **"*common*"** folder should check-out from[*/trunk/soapui/src/resources/projects*](https://svn.edb.com/repos/java/enterprise/aan/server/trunk/aan-soapui/src/resources/projects)**,"*common*"**foldercontain"commonutills.groovy" re-usable groovy script for removing space and null value from soapui response xml node.
2. **Maven Command:** *mvn test –D<enter pom.xml profileid>=true*
3. Configure the Soap UI POM.xml:   
    Define the endpoint:   
   **<serviceendpoint>IP:PORT</serviceendpoint>**

http://wiki.edb.com/download/thumbnails/26808207/POMEndpointConfig.png?version=3&modificationDate=1365415273000&api=v2

**POM File Contains:**

* Service Endpoint.
* Insert scripts for Database.
* Insert Test Data (Mock Data) into DB Table.
* Execute the Soap UI Pro Test.
* Export the Result.

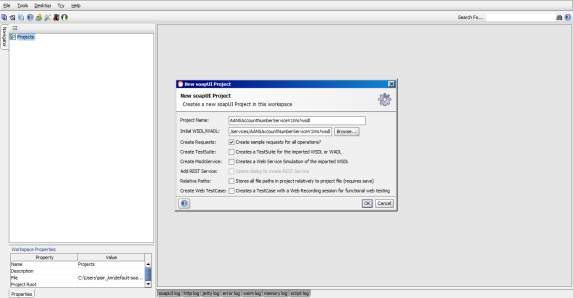
**NOTE:** Verify the project location:

http://wiki.edb.com/download/thumbnails/26808207/POM.png?version=4&modificationDate=1365411030000&api=v2 

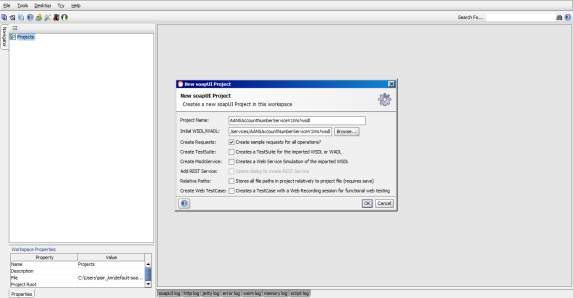
1. To run SoapUI test, go to checked out location of soapui folder,open command prompt and execute mvn from soapui location.  
   **Maven Command:** *mvn test –D<enter pom.xml profileid>=true*
2. After successfully running of soapui test, go to the[*/trunk/soapui*](https://svn.edb.com/repos/java/enterprise/aan/server/trunk/aan-soapui)***"target"*** folder; go to ***"surefire-reports"***folder inside of ***"target"*** folder to verify the test reports whether pass/fail status of test cases.

Add new tests

**a.**Right click on ***“Project”***or***“File Menu”*** and select New Soap UI Pro Project.

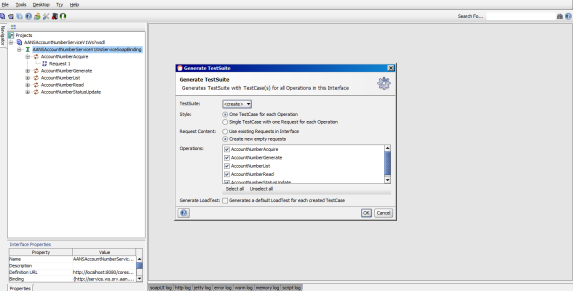


**b.**InNew Soap UI Pro Project dialog box, enter project name, enter the **WSDL** URL or browse the **WSDL** and click on **OK** button.



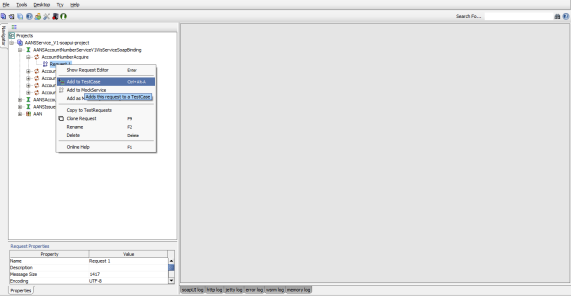
**c.***Adding Test Suite*:

* Right click on the service interface and choose **“*Generate Testsuite*”** in order to create the container for the Test cases and select the ***“Operations”*** for which user needs to automate.



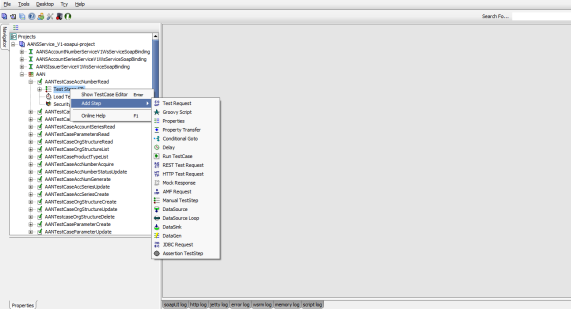
**d.***Adding Test Case:*

* Right click on the newly created Testsuite and choose***“New TestCase”***. Add Project name and the new test case will be created.



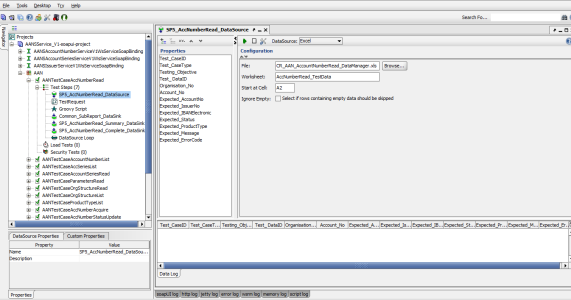
**e.**Adding *Data Source*:

* Right click on the Test Case and select***“Data source”***.



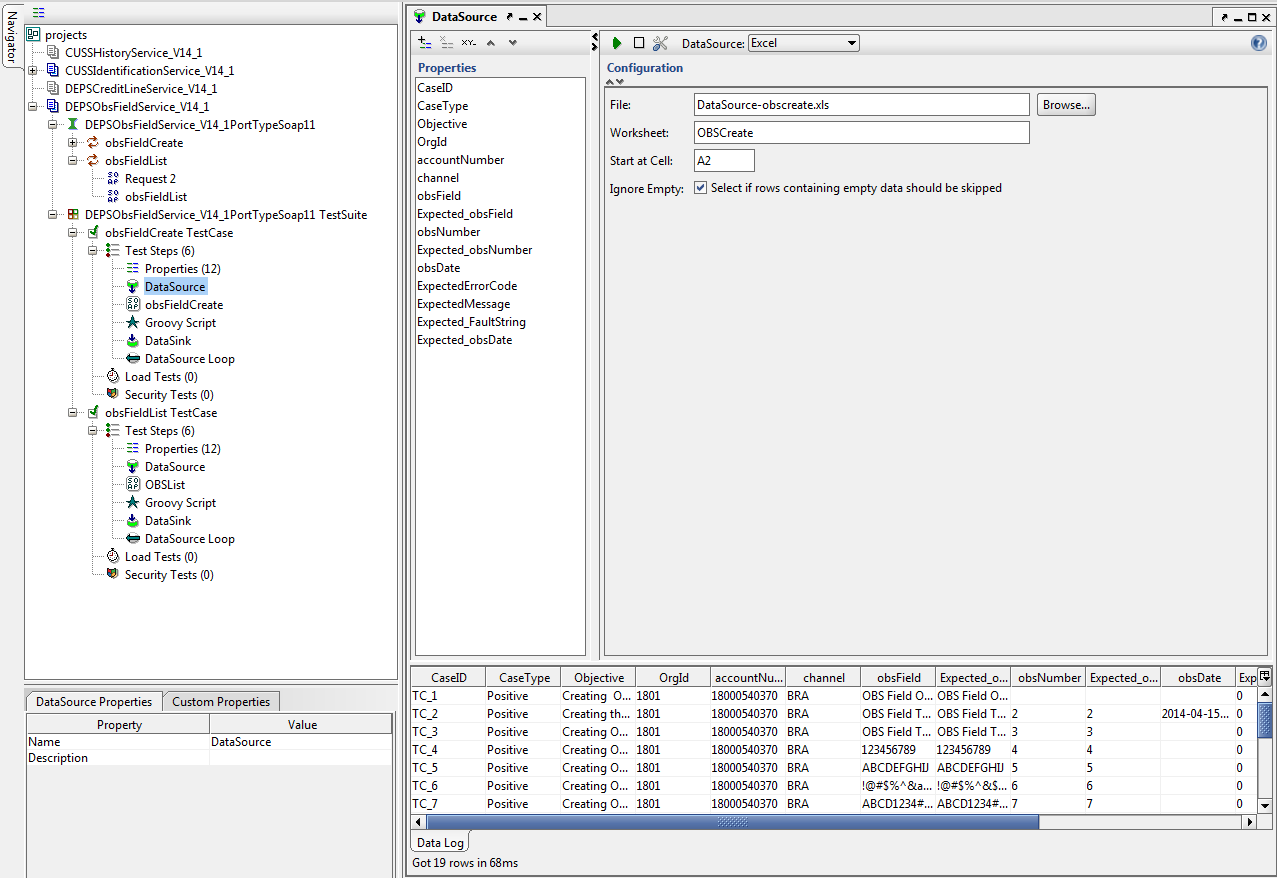
**f.**Configuration of External Excel sheet:

* Add the external excel (.xls) file clicking on ***“Browse”*** button.
* ***“Worksheet”*** name should be same as excel file.
* ***“Start at cell”*** for reading the data from excel sheet, it should start cell from**“A2”**. A1 contains headers which has been used properties in data source.

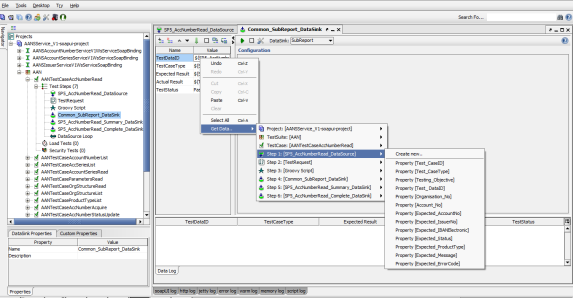


**g.**Adding properties:

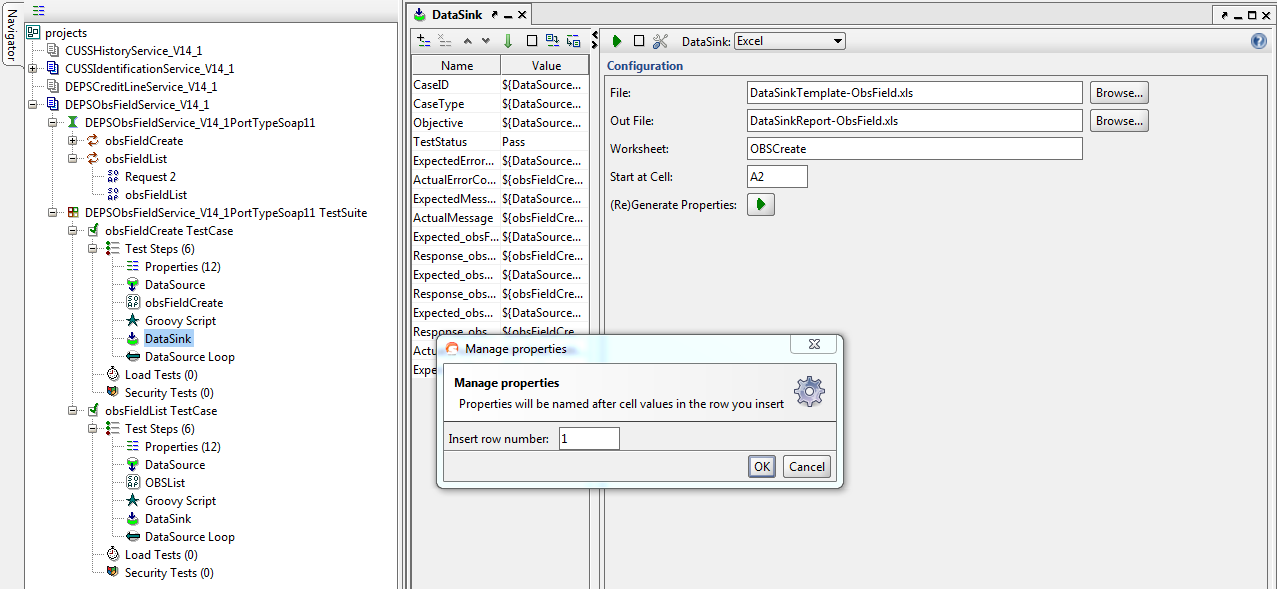
* Add properties based on excel file header for mapping and retrieving the excel input data.



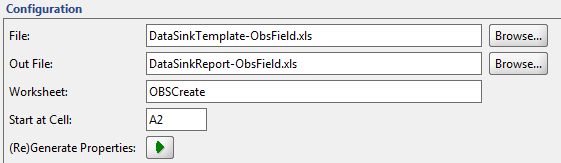
**h.**Adding DataSink for report generation:

Right click on the Test Case and select ***“Data Sink*”**, add properties based on generating the report and Right click on the ***“Value* “field** and add the value.  


**i.**Generate Properties: Click on Generate Properties button and ***“Insert row number”*** *(*Based on import the header and column from excel)



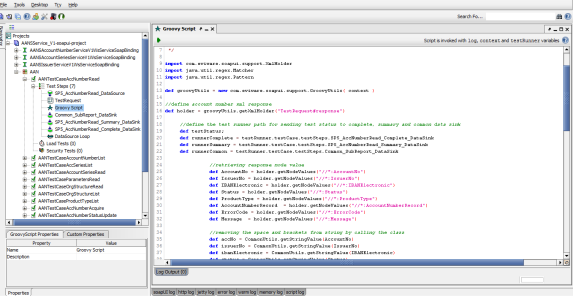
**j.**Configuration of DataSink report generation *"in"* and *"out"* file:



* ***"File"*** the source excel file to use as a template for adding data
* ***"OutFile"*** the target excel file to write.
* ***“Worksheet”*** name should be same as excel file.
* ***“Start at cell”***for printing the report it should start cell from **“A2”**. A1 contains headers which have been used as Names in DataSink.

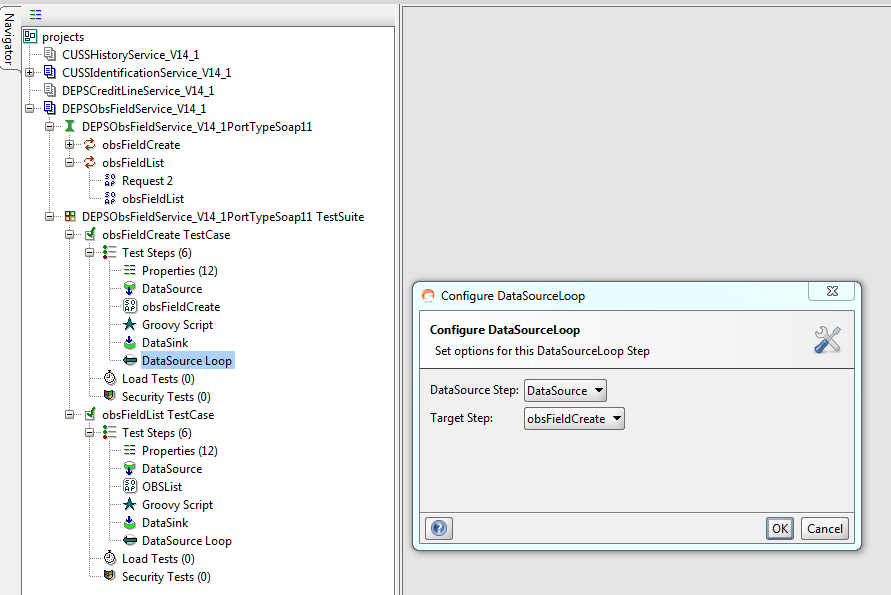
**k.**Adding Groovy Script:

* Validation of response.
* Groovy script is used in Soap UI Pro for getting results from other steps and passing them to other steps.
* Response code/message and expected result that are dynamically handled by Groovy scripts.



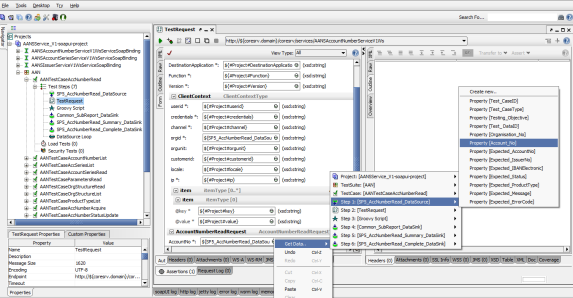
**l.**Adding DataSource Loop:

* Right click on the Test Case and select ***“DataSource Loop*”**, choose the ***“DataSource Step-DataSource”*** and ***“Target Step-TestRequest”***



**m.**Adding the Test Request Input Value:

* Right click on the *Input Field* and add value from ***“DataSource (Property)*”**.



Analysis result and reporting bugs

**How does Reporting Work?**

The Soap UI Pro execution reports status ***"Pass*** *or* ***Fail"*** against each of the test case validation checks. The tests reports **FAIL** when the expected results provided in the ***"Data Source"*** does not match with the actual results on the response. The tests report **PASS** when the expected results provided in the ***"Data Source"*** matches with the actual results on the response.

**When to analyze the results?**

When there is a **FAIL** on reported in the executed tests.

**How to analyze the test results?**

Go to the [*/trunk/soapui*](https://svn.edb.com/repos/java/enterprise/aan/server/trunk/aan-soapui)***"target"*** folder, go to ***"surefire-reports"*** folder(system-generated after execution of soapui test using maven) to check complete report that shows PASS/FAIL to analyze the response message/code.

* In case if the response message/code seems incorrect then manually tries to execute the same steps behavior in the service. Report this as a steps/scenario failure to the development team for further investigation.

Soap UI Pro FAQ

Link: <http://www.soapui.org/About-SoapUI/soapui-faq.html>