Automation workflow of Office Dependent Data (ODD) Verification

January 12, 2015

Table of Contents

January 12, 2015 1

1 Introduction 1

2 Scope of Work 2

2.1 Automation Scope for ODD verification 2

2.2 Automation Coverage of ODD verification 2

2.3 ODD verification Reporting 3

2.4 ODD verification Test Prerequisites 3

2.5 iTest Parameters for ODD verification 4

2.6 Test Case logic to read content from the Excel sheet 6

2.7 Test Case logic for ODD verification 7

1 Introduction

Automated provisioning is the ability to deploy an information technology or telecommunications service by using predefined procedures that are carried out electronically without requiring human intervention.

Data validation is an important part of any application, as it helps to make sure that the device is configured with the required data.

Currently the ODD verification Testing effort requires RJIL engineers to validate data in the device. These checks are performed when vendor provides the device logs and ODD sheets.

Automating the ODD verification will reduce the manual effect and time for validation. Reports will be generated for each execution with the matching and mismatching information. So, it is very easy to identify the mismatching data. Once the script is developed for a specific node, it is easy to reuse the same script to validate latest ODD sheets provided by the vendor.

2 Scope of Work

2.1 Automation Scope for ODD verification

The scope of the automation effort is to create a framework using **iTest Enterprise** that will be able to run automatically (or on demand) and return a report with the validation results.

2.2 Automation Coverage of ODD verification

Automated process for **ODD** verification covers the following Network Nodes:

- **EPC Team:**

  a. ***MME NODAL*** (100% complete).

  b. ***SAE-Gateway*** (100% complete)

  c. ***PCRF*** (100% complete)

  d. ***DRA*** (25% complete)

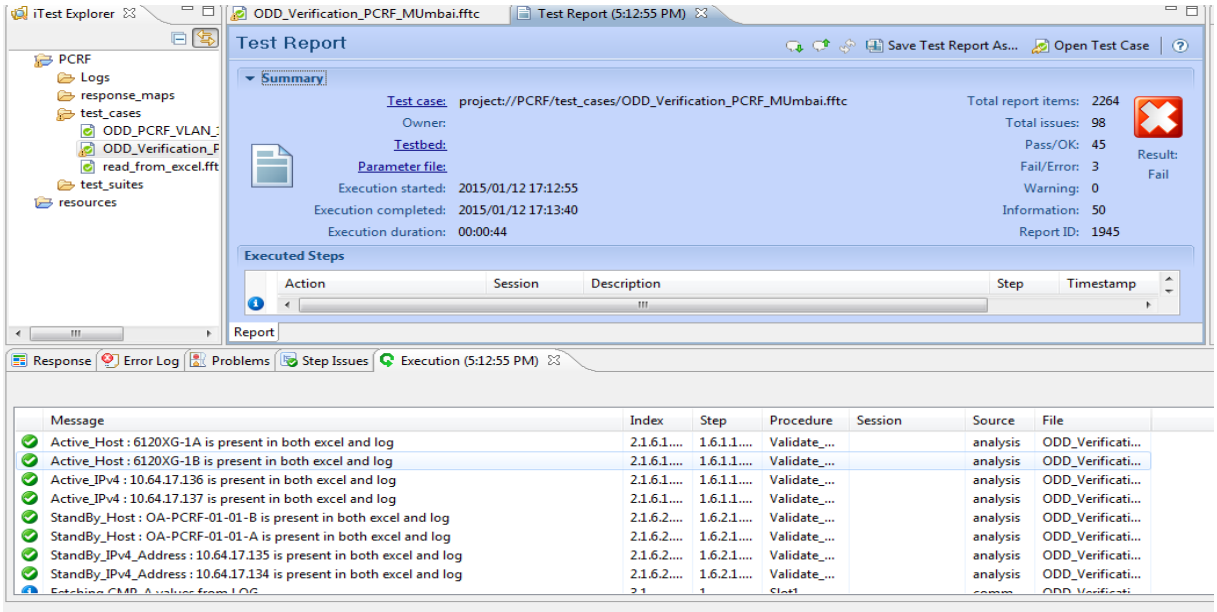
- **IMS Team:**

  e. Radisys Media Resource Function (MRF) - (100% complete)

2.3 ODD verification Reporting

Requirements: Reports should be available in HTML/XML/Text format for each test case run.

Sample of validation report:



2.4 ODD verification Test Prerequisites

Requirements: Lab Machine – a windows machine that has access to all Nodes under test.
In order to execute testcase, the following resources need to be configured in the machine:

- **Lab Machine**

1.1 Windows PC with the following tools need to be installed:

a. iTest

b. Mozilla web browser (21 or above version)

c. Microsoft office

d. TCL 8.4 or above version

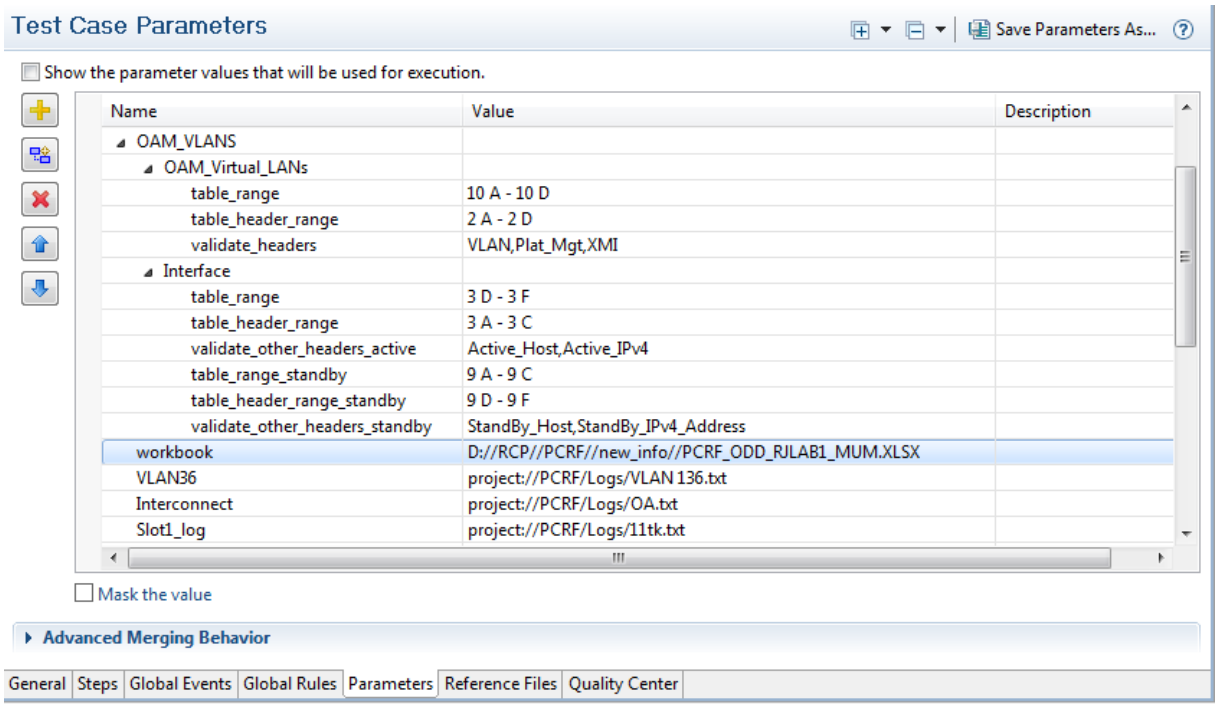
2.5 iTest Parameters for ODD verification

The values in the testcase which need to be modified for each ODD verification can be specified as parameters, so we can directly change the values in the parameter tab instead of modifying in the actual testcase.

Notes: List of parameters need to be used in the testcase.

Note	Details
*	ODD workbook: ODD workbook located path need to speicified as a parameter. Eg: workbook - D://RCP//PCRF//new_info//PCRF_ODD_RJLAB1_MUM.XLSX
**	Device logs: Device logs located path need to speicified as a parameter. Eg: Slot1_log - project://PCRF/Logs/11tk.txt
***	Table Range: Inorder to fetch the data from ODD excel sheet, table header and table values range need to specified. Eg: Table_range: 3 D - 3 F Table_header_range: 3 A - 3 C
****	Validate Headers: The values which need to be validate can be speicified as parameters.

	Eg: validate_headers: Virtual_IP,Active_IP,Hostname

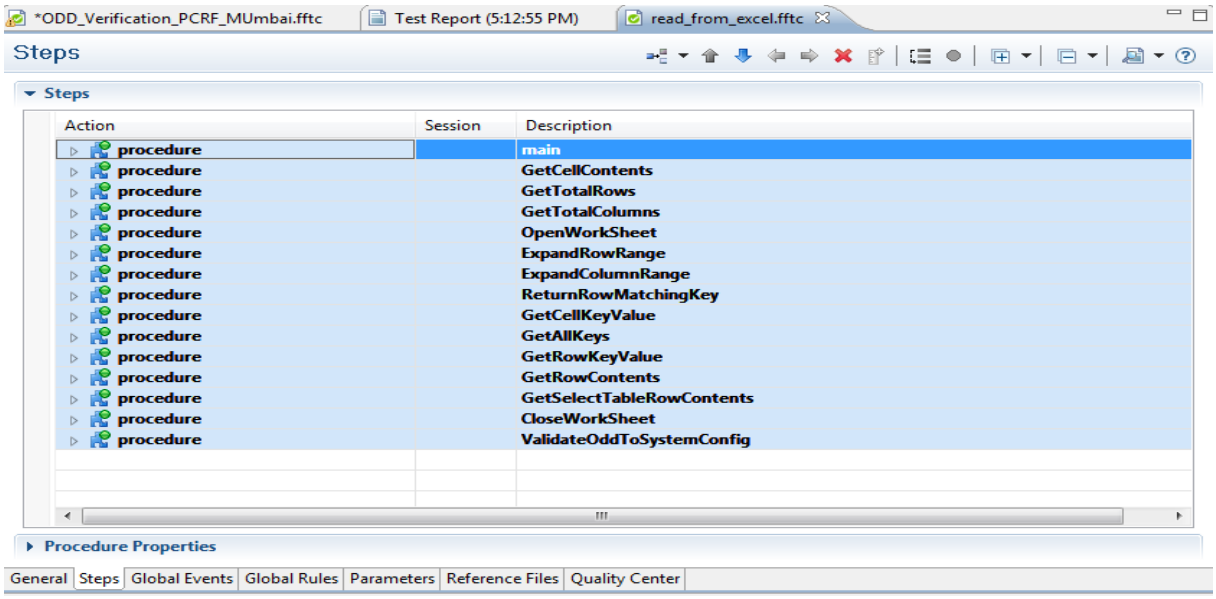


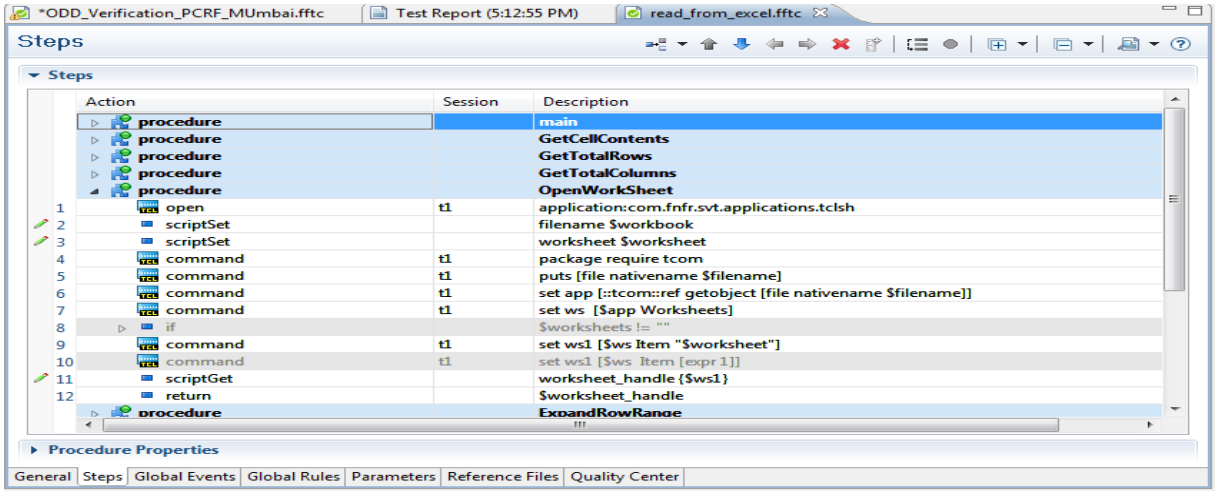
2.6 Test Case logic to read content from the Excel sheet

We can fetch the information from the excel sheet by using TCL package 'TCOM'.

Steps:

1. Speicfy all needed parameters
2. Open tclsh session profile.
3. Invoke the tcom packages
    - Package require tcom
4. By using tcom API commands create separate procedures to read content from the excel file.





2.7 Test Case logic for ODD verification

We have to create seperate procedures for each module to get the infromation and validate the values.

Steps:

1. Read the content from the device log file by using command ”readfile”

   Ex: readFile [param VLAN36]

   In the parameter the path of the VLAN36 log is specified as shown in chapter **2.5**

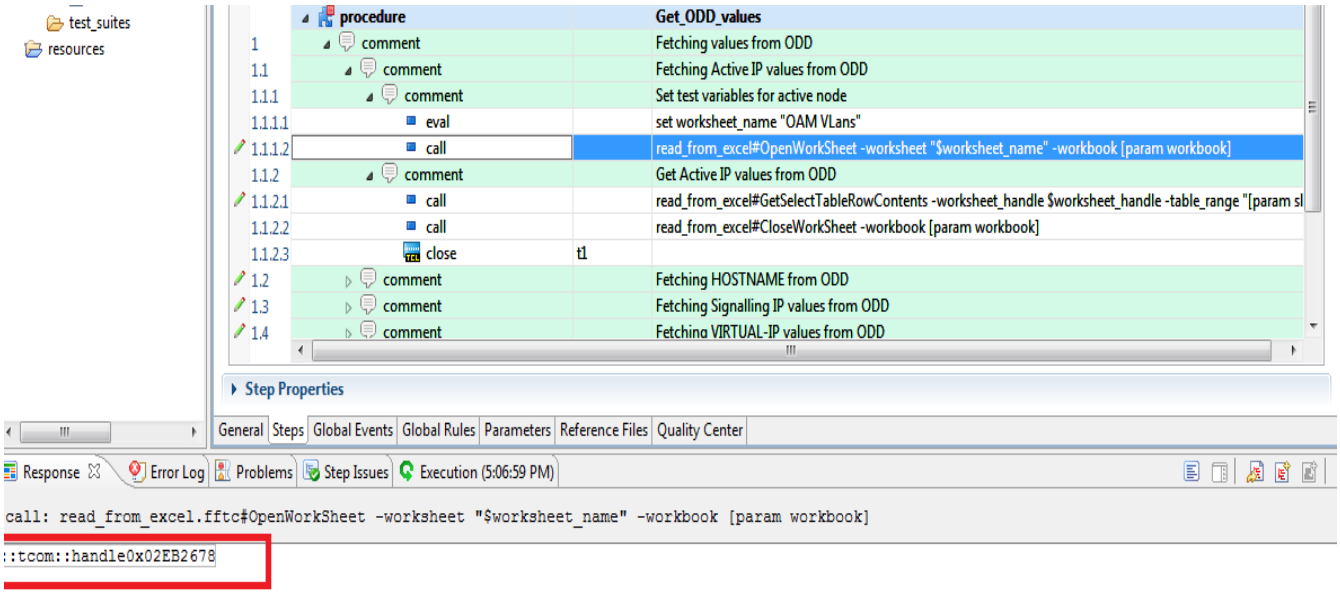
The following process need to follow to fetch the data from excel sheet:

2. We have to specify the worksheet name located in the excel workbook.

   Eg: set worksheet_name "OAM VLans"

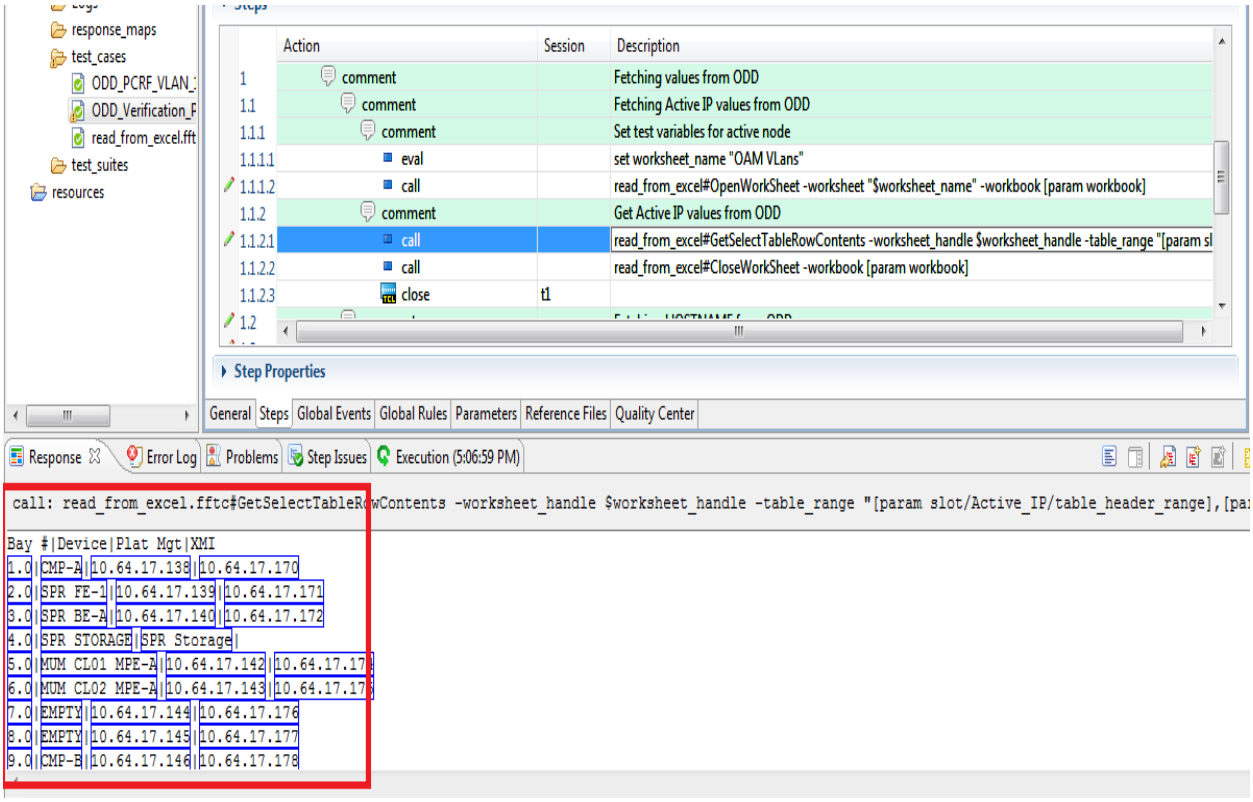
3. Call the read_from_excel.fftc testcase procedure ”**OpenWOrkSheet**” by speicifying the arguments (workbook path and worksheet name) to get the worksheet handle. And store the step response into a variable worksheet_handle.

   Eg: read_from_excel.fftc#OpenWorkSheet -worksheet "$worksheet_name" -workbook [param workbook]



4. To get the values from the worksheet, call the testcase ”read_from_excel.fftc” procedure ”**GetSelectTableRowContents**” by specifying the arguments worksheet handle, table header and table value ranges.

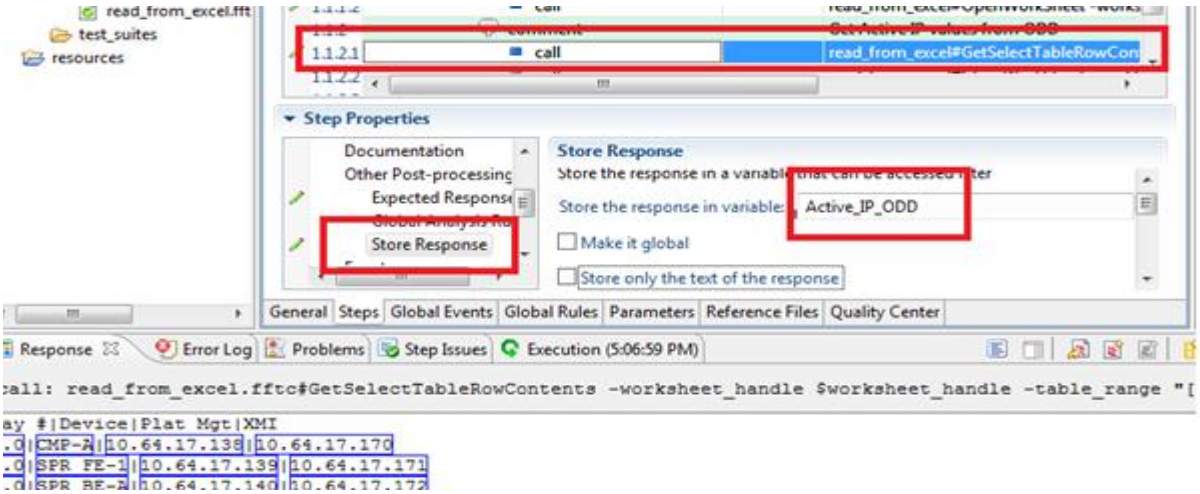
   Ex: read_from_excel.fftc#GetSelectTableRowContents -worksheet_handle $worksheet_handle -table_range "[param slot/Active_IP/table_header_range],[param slot/Active_IP/table_range]"



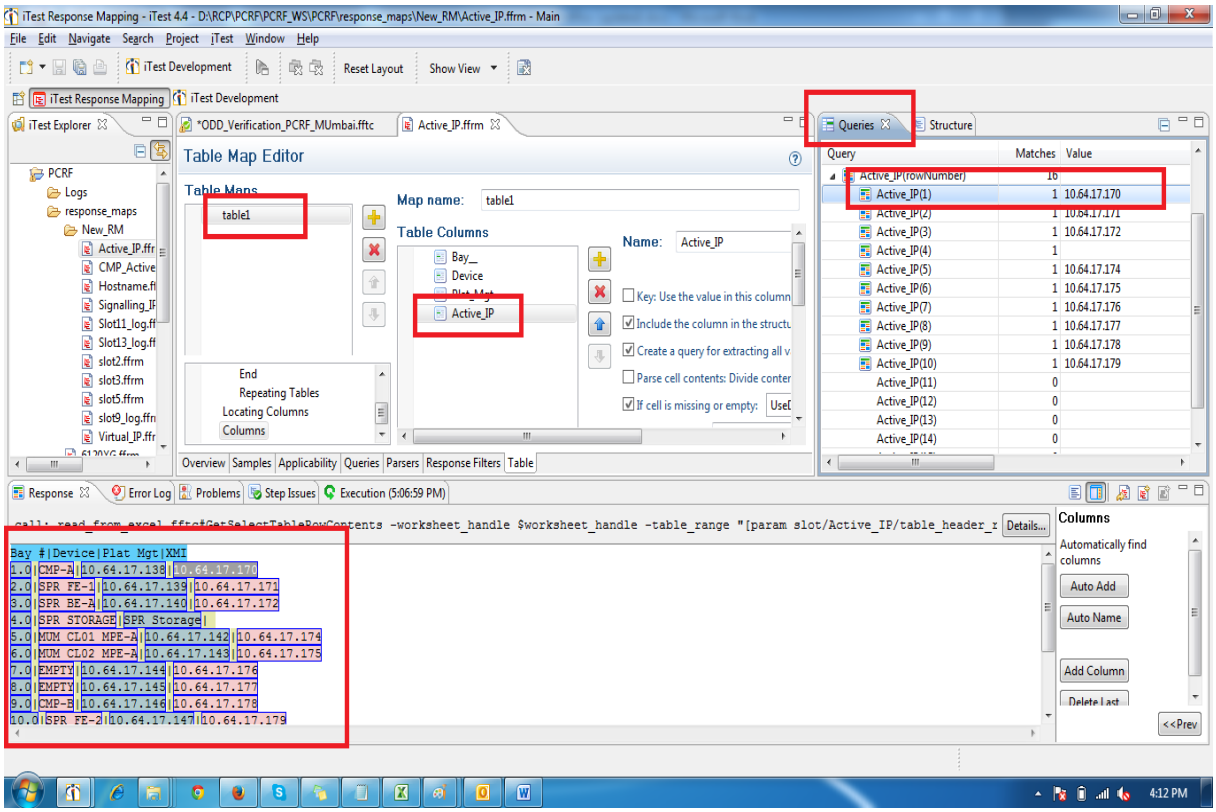
5. After reading the content from the excel file, call the testcase ”read_from_excel.fftc” procedure **”CloseWorkSheet”** to close the worksheet.

   Ex: read_from_excel.fftc#CloseWorkSheet -workbook [param workbook]

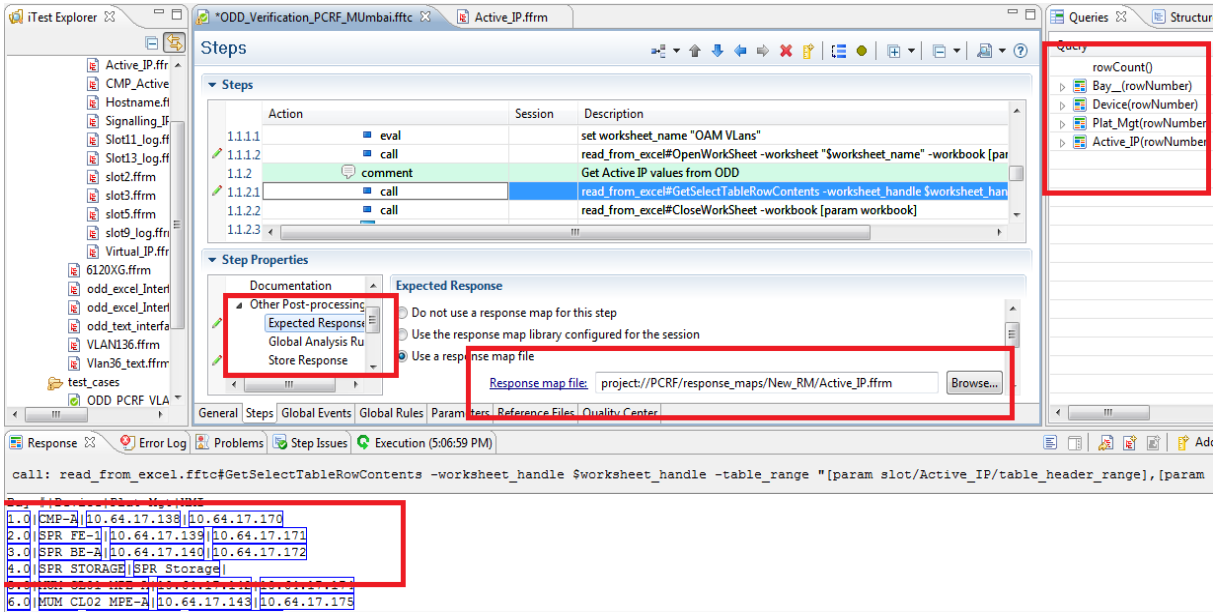
6. After getting the information from both ODD and device logs, store the corresponding step response into a variable.
   - Step -> Step Properties -> Other Post Processing -> Store Response



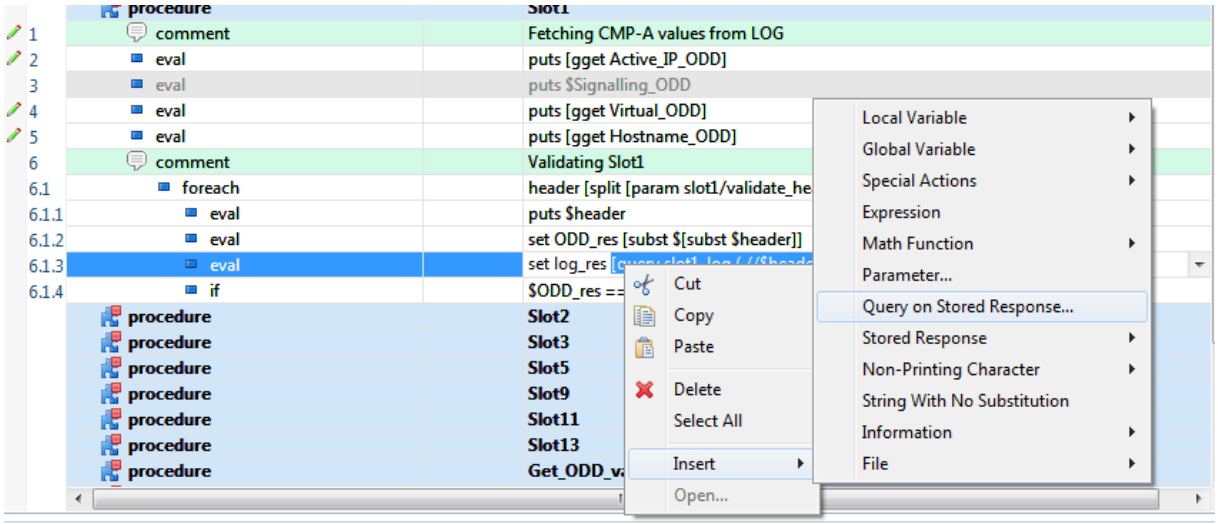
7. Create the response map for each step response to define the queries which need to validate.



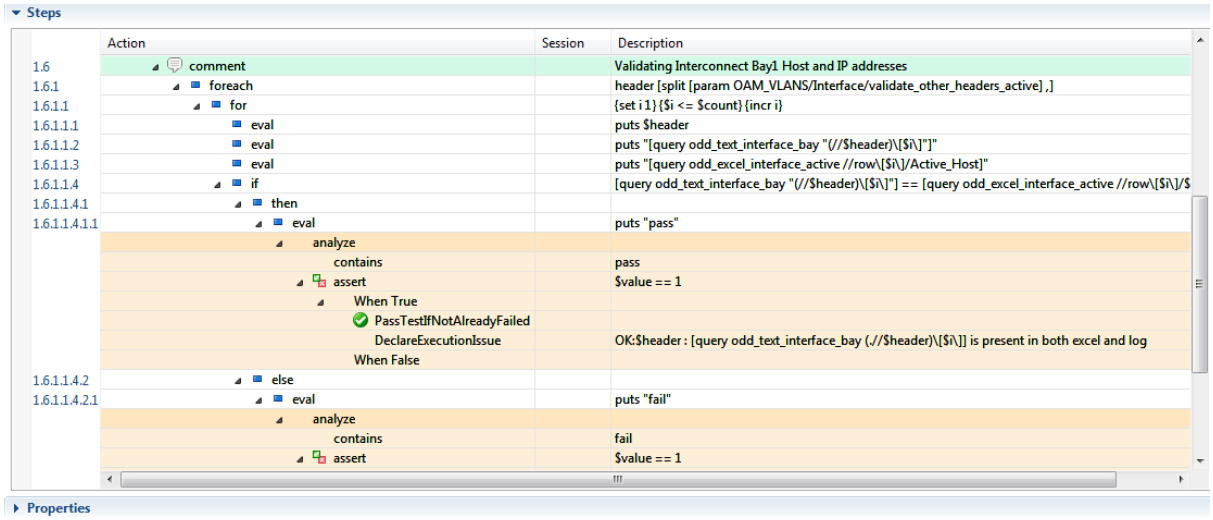
8. Apply the response map to the corresponding step in the testcase. When a response map is applied to a response, we will see blue boxes around each item of data (in the Response view) that matches the queries defined in the response map.



9. The queries which need to validate should be store into a variable by using the option Query on Stored response.

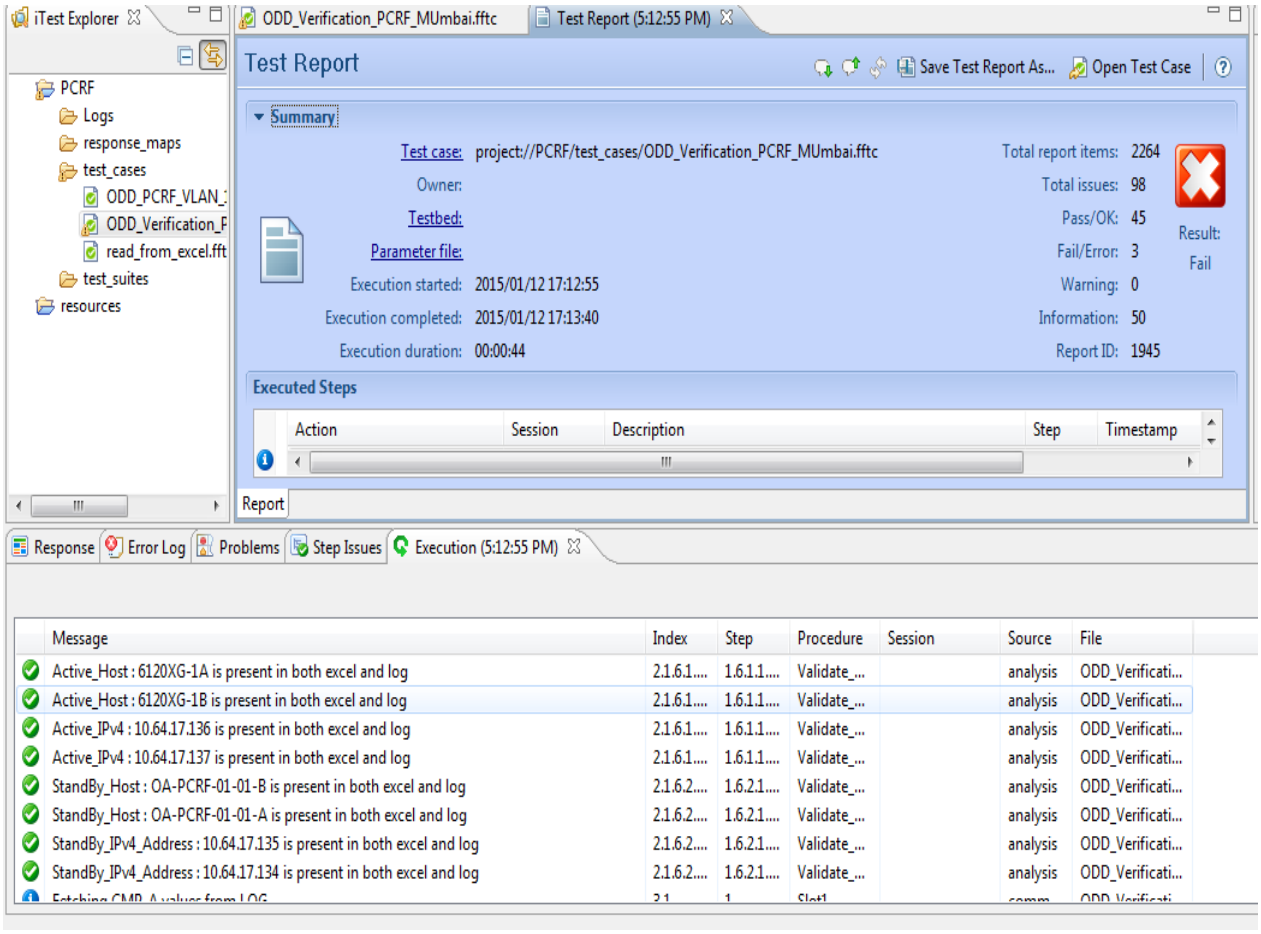


10. After storing the values into a variable, we have to implement logic by using tcl commands to compare values in both ODD and device log.



11. Once the comparsion done, we have to use the iTest analyze rules to perform Pass / Fail validation on the results of a step. We can use the Analysis Rule wizard to create an analysis rule that performs one of the following operations:

    - Validates something about the response to a test case step (and then performs specified actions like passing or failing the test, displaying an execution message, and then charting the extracted value over time).
    - Generate execution message.


iTest Explorer
PCRF
Logs
response_maps
test_cases
ODD_PCRF_VLAN_
ODD_Verification_F
read_from_excel.fft
test_suites
resources
ODD_Verification_PCRF_MUmbai.fftc
Test Report (5:12:55 PM)
Test Report
Save Test Report As...
Open Test Case
Summary
Test case: project://PCRF/test_cases/ODD_Verification_PCRF_MUmbai.fftc
Owner:
Testbed:
Parameter file:
Execution started: 2015/01/12 17:12:55
Execution completed: 2015/01/12 17:13:40
Execution duration: 00:00:44
Total report items: 2264
Total issues: 98
Pass/OK: 45
Fail/Error: 3
Warning: 0
Information: 50
Report ID: 1945
Result: Fail
Executed Steps
Action
Session
Description
Step
Timestamp
Report
Response
Error Log
Problems
Step Issues
Execution (5:12:55 PM)
Message
Index
Step
Procedure
Session
Source
File
Active_Host : 6120XG-1A is present in both excel and log
Active_Host : 6120XG-1B is present in both excel and log
Active_IPv4 : 10.64.17.136 is present in both excel and log
Active_IPv4 : 10.64.17.137 is present in both excel and log
StandBy_Host : OA-PCRF-01-01-B is present in both excel and log
StandBy_Host : OA-PCRF-01-01-A is present in both excel and log
StandBy_IPv4_Address : 10.64.17.135 is present in both excel and log
StandBy_IPv4_Address : 10.64.17.134 is present in both excel and log
analysis
ODD_Verificati...