IXCHARIOT – USER MANUAL

Bell Canada; ATL Lab

Contact:

Angelo Virgilio

Angelo.virgilio@ecin.ca

Mobile: 416.605.1296

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Arijit Saha	First Draft version	

Review & Approval

Requirements Document Approval History

Approving Party	Version Approved	Signature	Date
Bell Canada: **Intissar Harrabi**			

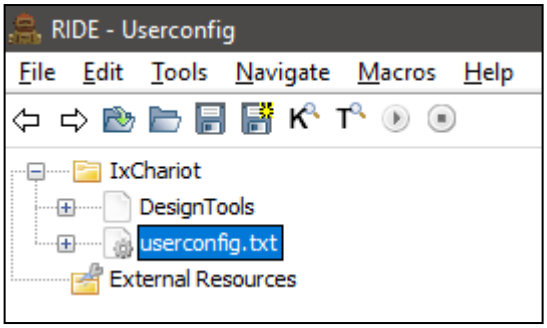
Requirements Document Review History

Reviewer	Version Reviewed	Signature	Date
Angelo Virgilio			

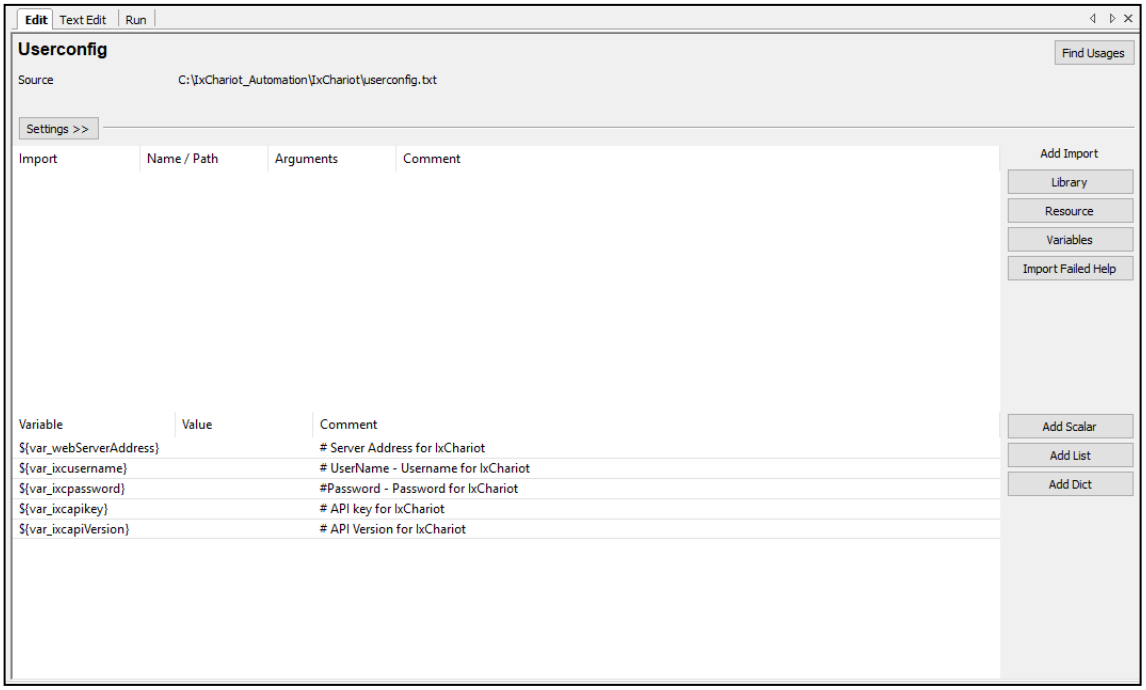
Contents

1. Configuring Common Parameters

1. From the left panel, click on the **userconfig.txt**.



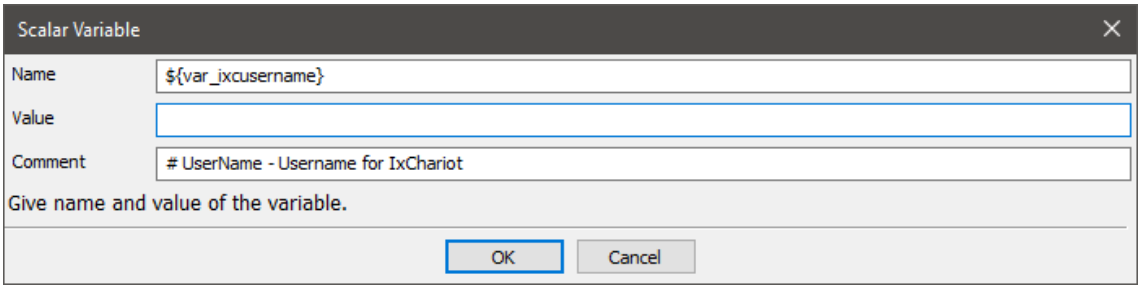
2. On the right, it will open the edit tab of the userconfig.txt.
   Here the user will find the common parameters and the values for each of them.



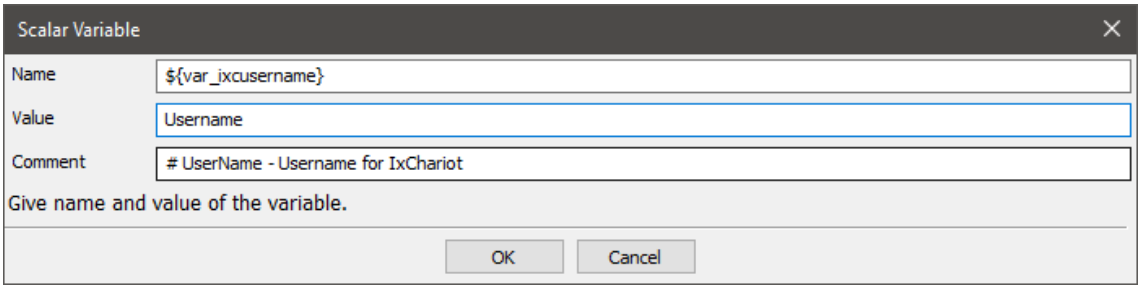
3. The user has to provide the values for username, password, API key, API version and the IP address for IxChariot web version.

Variable	Value	Comment
${var_webServerAddress}	https://192.168.2.30	# Server Address for IxChariot
${var_ixcusername}	cage.noire@gmail.com	# UserName - Username for IxChariot
${var_ixcpassword}		# Password - Password for IxChariot
${var_ixcapikey}	25596217-423e-4072-8d0a-ff6300d37d5f	# API key for IxChariot
${var_ixcapiVersion}	v1	# API Version for IxChariot
${MAC1}	192.168.2.100	# IP Address of MacBook 1

4. Double click on the variable name, pop up will appear.

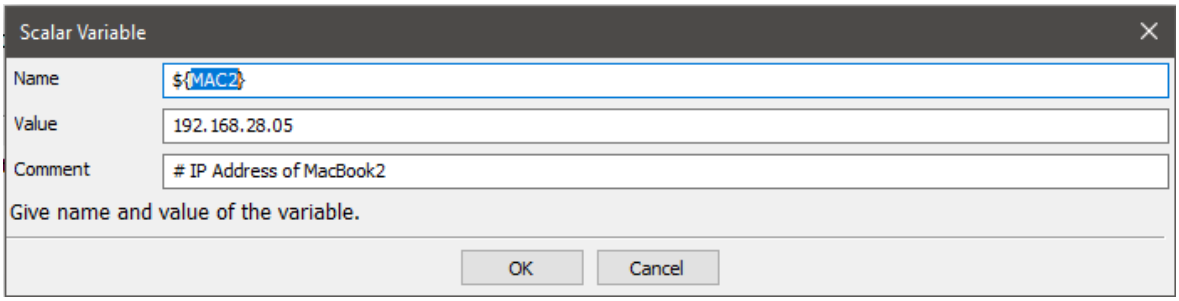


5. Here the user can enter the value for each variable. <u>Do not change the variable name</u>.



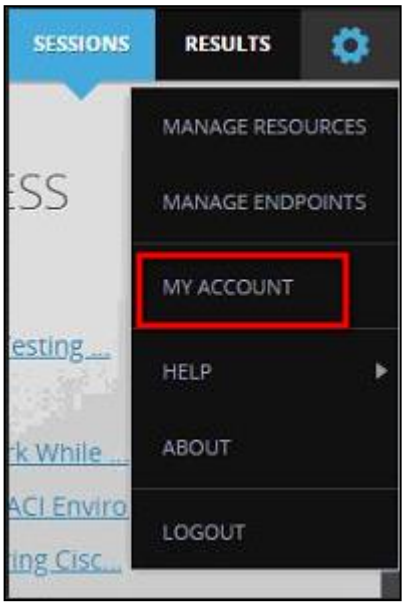
6. The user can give the value of the other variables repeating step 4 and 5.
7. The user can add new variables by clicking on the **Add Scalar** button on the right side of the window.

8. The user can enter name of the variable, its corresponding value and a comment about the variable name.

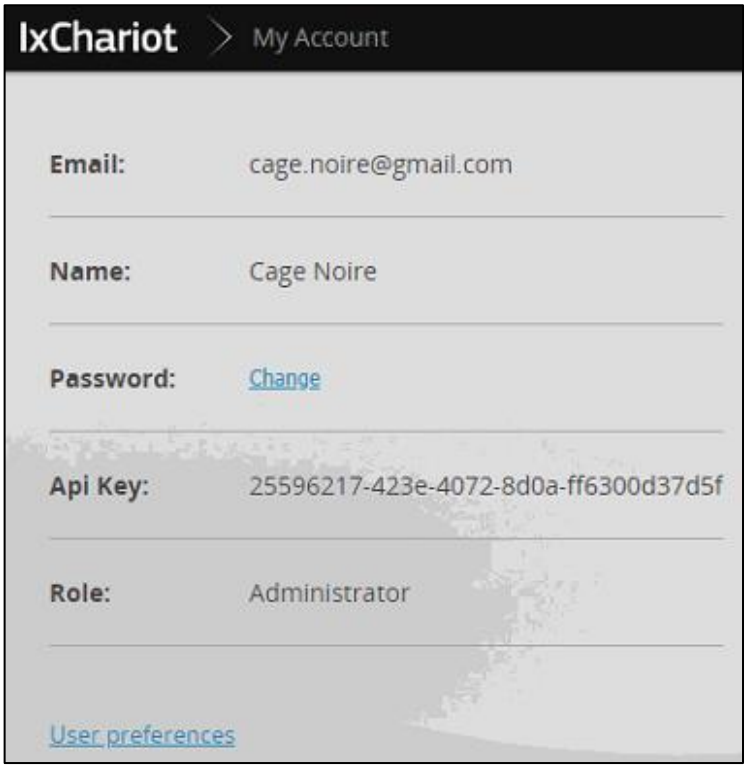


9. The user can create new variables by repeating the steps 7 & 8.
10. The user can find the values of username, password and API Key in the settings of IxChariot WebAPI.
11. To find the above values, follow the steps below.
12. After logging into the user account, on the top right side, the user will find the **settings** button.

13. On clicking of that, a drop down will come. Choose **My Account** from the dropdown.

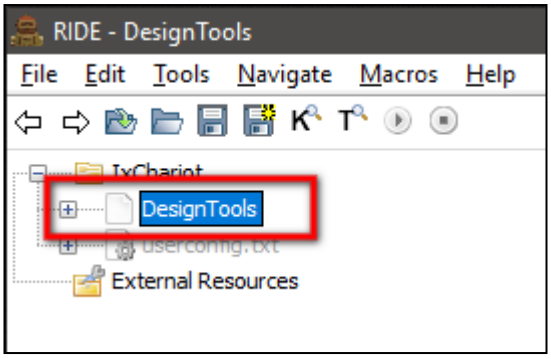


14. From the My Account page, the user can find the values for the username, password, and API key (Click on the Show button to see).

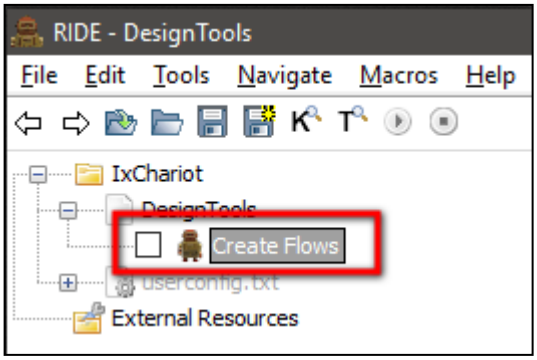


2. Configuring Test Parameters

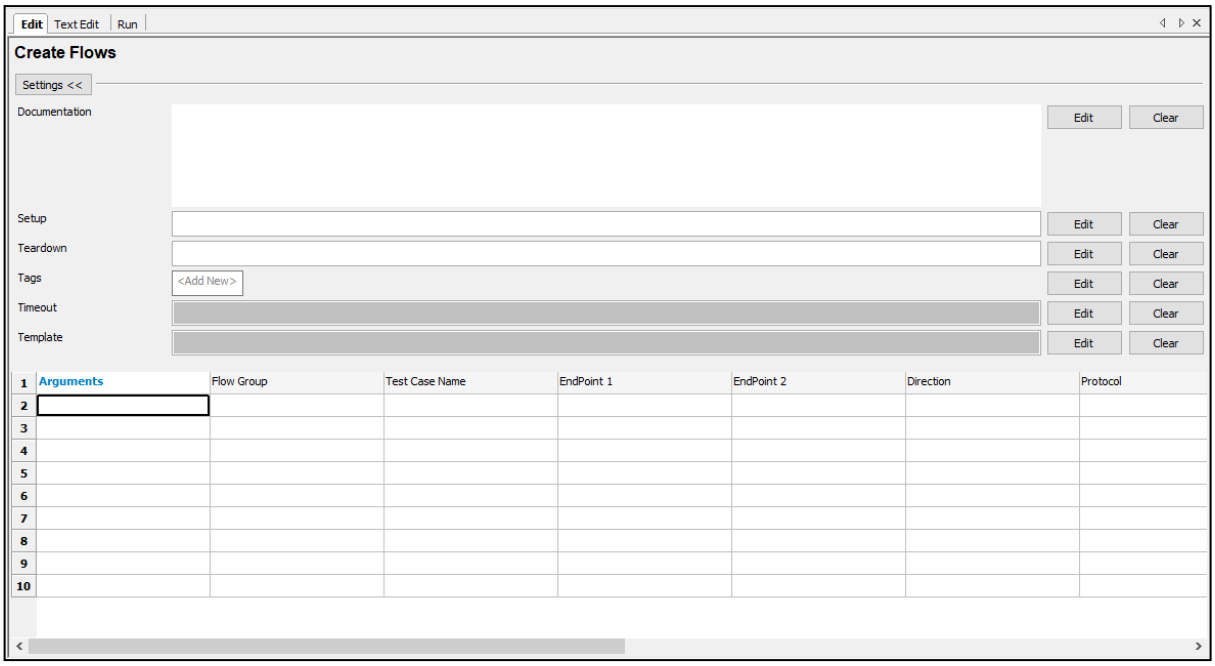
1. Now on the left panel, click on the plus sign next to **DesignTools** to expand.



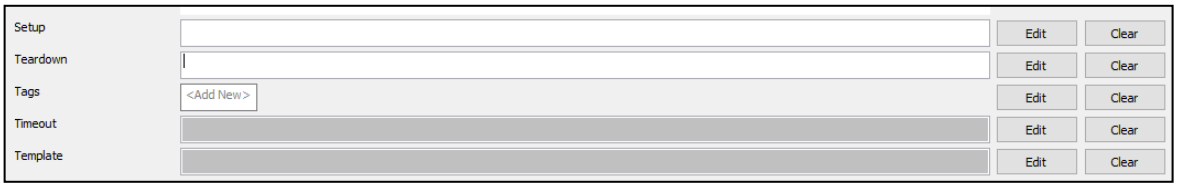
2. Under the DesignTools, the user will find **Create Flows**.
3. Click on the **Create Flows**.



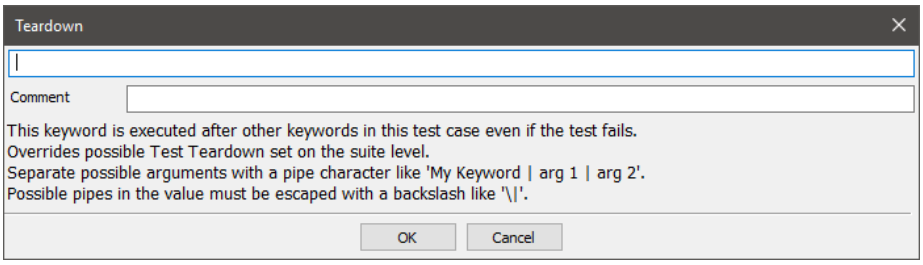
4. It will open the edit tab.



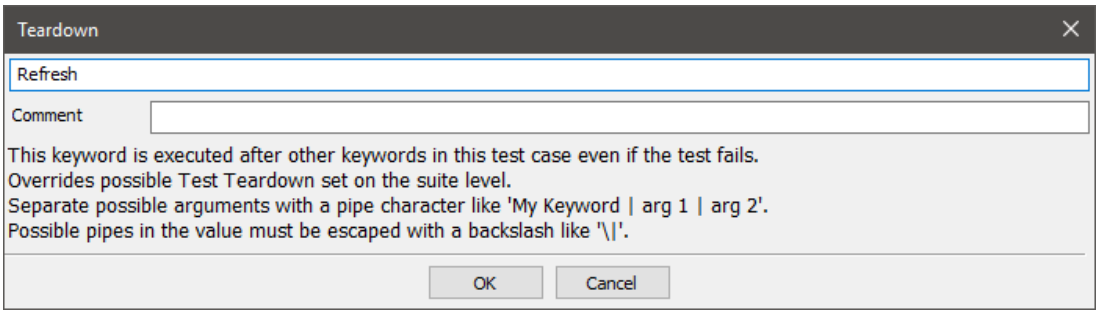
5. Click on the **Edit** button next to the **Teardown** text box.



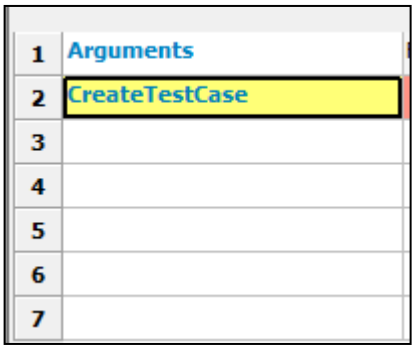
6. A pop up will come up.



7. From the popup, enter **Refresh** in the first textbox and click OK button.



8. Now the user has to configure the parameters. In the first cell of the row, the user has to enter the command name **CreateTestCase**.



9. The parameters that the user has to configure for creating the test case files are mentioned below.

Parameters	Comments
Flow Group	User can enter the value for group name. Example MAC.
Test Case Name	It can be given as the user wants to name the test case. Example: Bell_MAC1
End Point 1	This will be the name of the variable name mentioned in userconfig.txt file for the endpoint IP. T Example: If user has used MAC1 for 192.168.2.100 in userconfig.txt file, then enter MAC1 for EndPoint1.
End Point 2	Similar to End Point 1, this value must match the value that is given in the userconfig.txt. Note: It must not be the same as the value of the End Point 1.
Direction	Must be either DS / US.
Protocol	Must be either TCP / UDP.
Script	Throughput Tests • TCP High Performance • TCP Low Performance • TCP Small Packets Performance • TCP Baseline Performance • TCP Small Packets Performance • UDP Low Performance • UDP Baseline Performance • UDP High Performance
Duration (Seconds)	The user can enter an integer value.

Create	Must be either Yes / No. Default value is Yes if left blank.
Number of Users	The maximum number of user can be 10 as per the current license at Bell.
Radio1, Radio2, Radio3, Radio4, Radio5	The user can give an integer value which will denote the radio of the HomeHub device.
Channel1, Channel2, Channel3, Channel4, Channel5	This is set of commas separated integer values for the corresponding radio values.

10. Here is a sample after the user enters the values for the parameters (given in the above table).

Edit | Text Edit | Run

Create Flows

Settings >>

	Arguments	Flow Group	Test Case Name	EndPoint 1	EndPoint 2	Direction
1	Arguments	Flow Group	Test Case Name	EndPoint 1	EndPoint 2	Direction
2	CreateTestCase	MAC	Bell_MAC1	DESKTOP	MAC1	DS
3	CreateTestCase	MAC	Bell_MAC1	MAC1	DESKTOP	US
4						
5						

Edit | Text Edit | Run

Create Flows

Settings >>

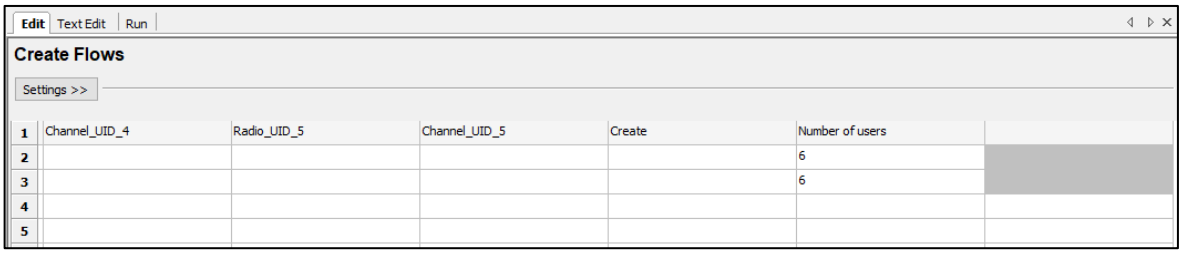
	Script	Duration (secs)	Radio_UID_1	Channel_UID_1	Radio_UID_2	Channel_UID_2
1	Script	Duration (secs)	Radio_UID_1	Channel_UID_1	Radio_UID_2	Channel_UID_2
2	TCP High Performance	100	1	1,2,3	2	36,40,44
3	TCP High Performance	10	1	1,2,3	2	36,40,44
4						
5						

Edit | Text Edit | Run

Create Flows

Settings >>

	Radio_UID_3	Channel_UID_3	Radio_UID_4	Channel_UID_4	Radio_UID_5	Channel_UID_5
1	Radio_UID_3	Channel_UID_3	Radio_UID_4	Channel_UID_4	Radio_UID_5	Channel_UID_5
2						
3						
4						
5						



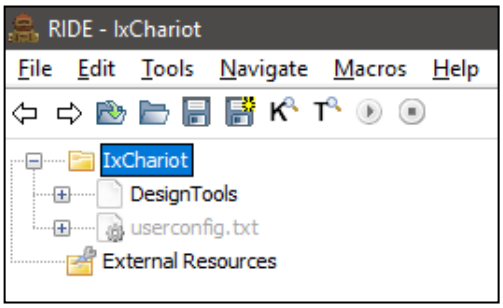
	Channel_UID_4	Radio_UID_5	Channel_UID_5	Create	Number of users
1					
2					6
3					6
4					
5					

11. The user can repeat the steps 8 & 9 for creating multiple test cases.

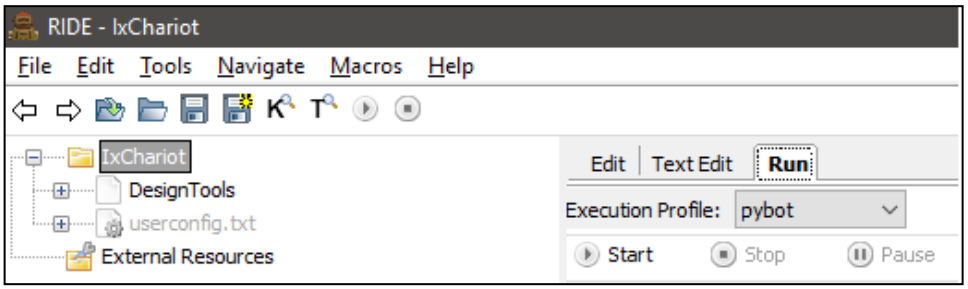
3. Configuring Result Arguments

The section aims to customize the result files name, location and the background color. The above command needs to be inserted in the arguments section, following the steps:

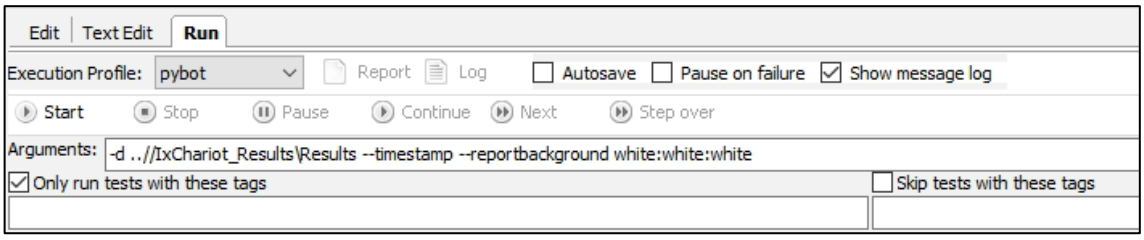
1. Click on project name.



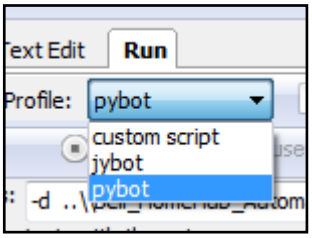
2. Click on **Run** tab.



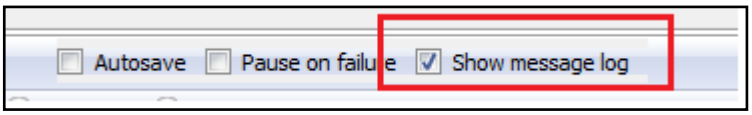
3. Copy the command **"-d ..\\IxChariot_Results\Results --timestamp --reportbackground white:white:white"** and paste it in arguments text area as shown.



4. Select Execution Profile as **pybot** (if not already selected).

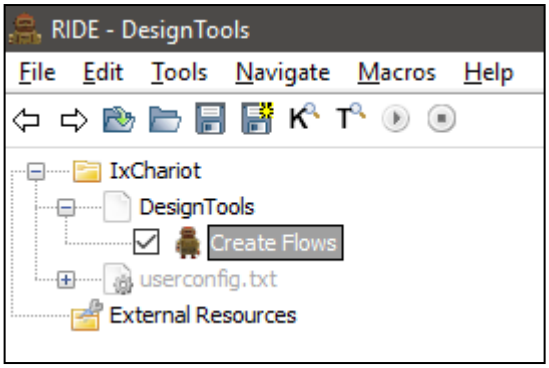


5. Click to checkbox to mark as selected for Show message log (if not already selected).

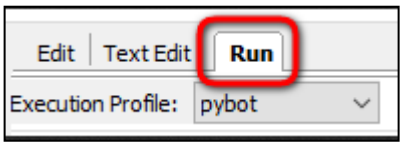


4. Creating Test Case

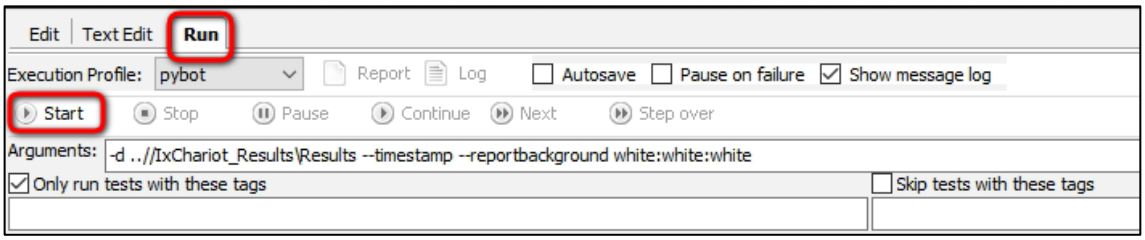
1. To start the execution, click on the checkbox to the left of "**Create Flows**".



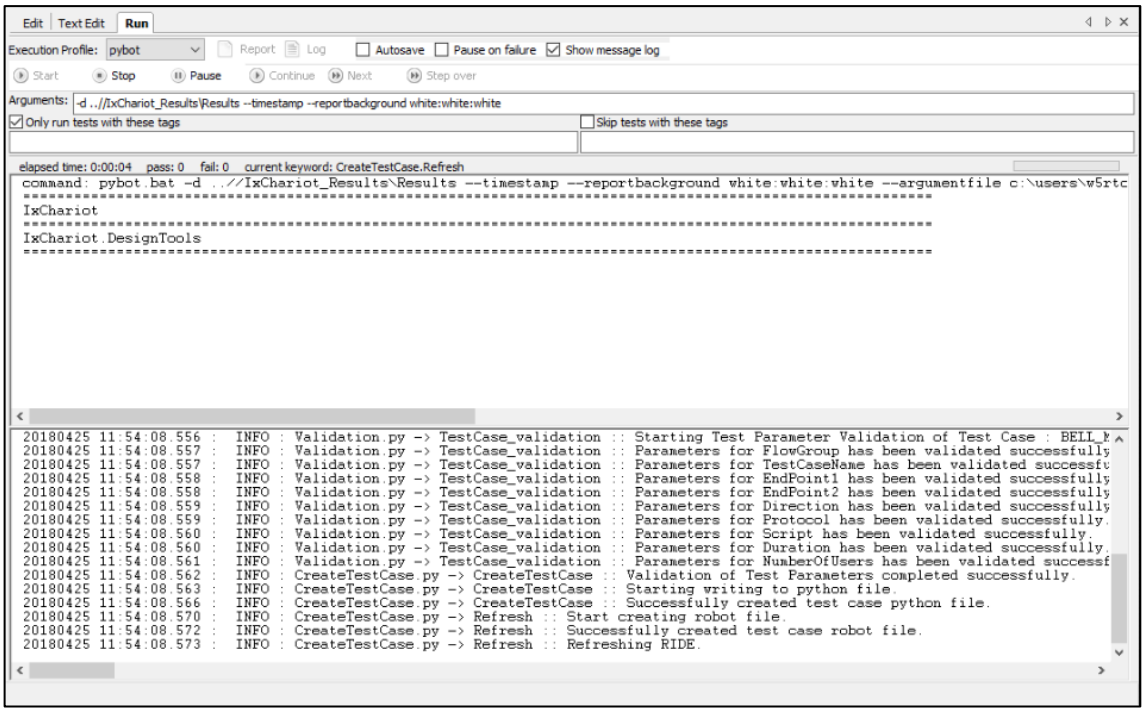
2. Click on the **Run** tab.



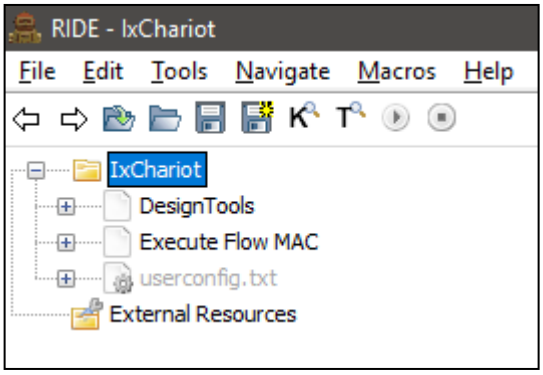
3. Click on the **Start** button to start the execution.



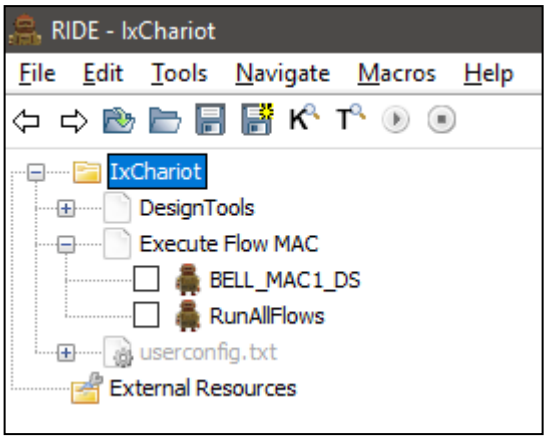
4. On clicking of the button, the execution will start.



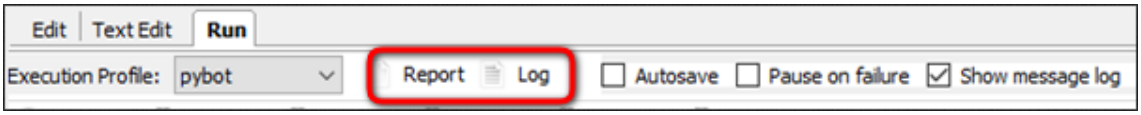
5. On successful execution, RIDE will close and automatically reopen with the Flow group test suites appearing on the left under the Design Tools.



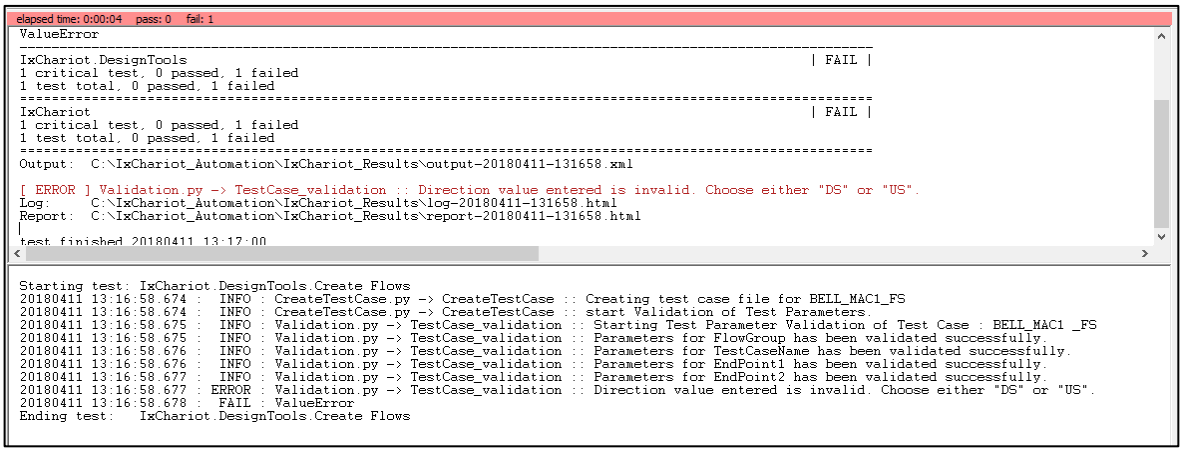
6. It will show all the test case listed under the Flow Group name during creation.



7. Once the test case is executed successfully, then click on the **Log** button to view the log of the executed test case and click on the **Report** button to view the report of the executed test case.



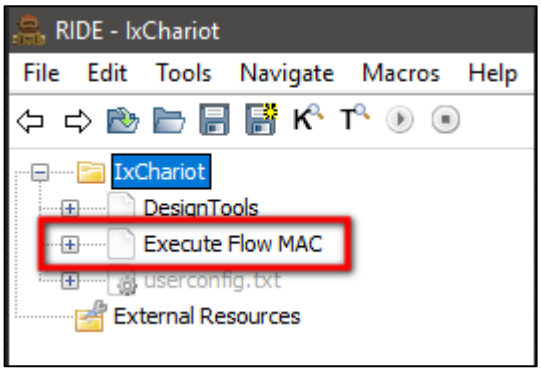
8. RIDE will not close if there is a validation error of the test input parameters and it will show the appropriate error message in the console.



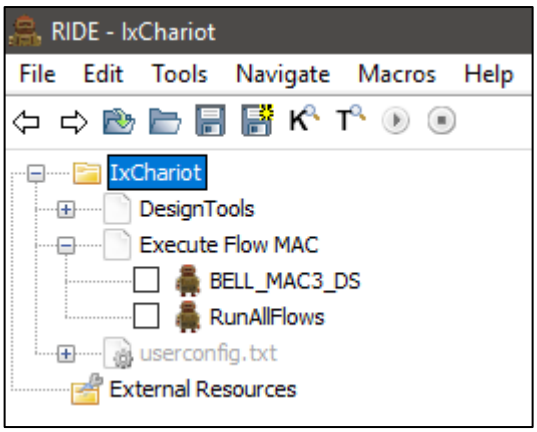
9. After checking and solving the cause of error from the log, the user can re-run the test case.

5. Executing Test Case

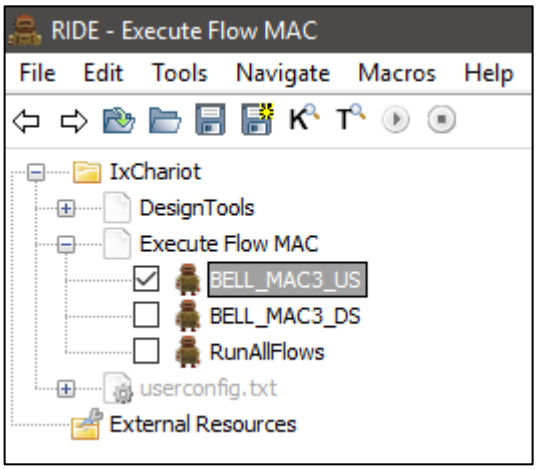
1. After the RIDE gets refreshed, the user can see the Flow group test suites appearing on the left under the Design Tools.



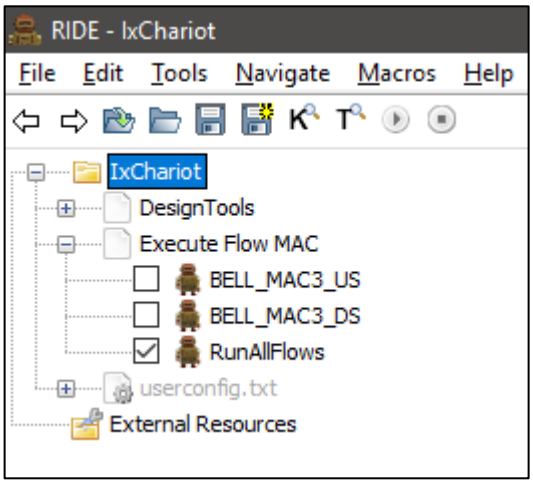
2. It will show all the test case listed under the Flow Group name during creation.



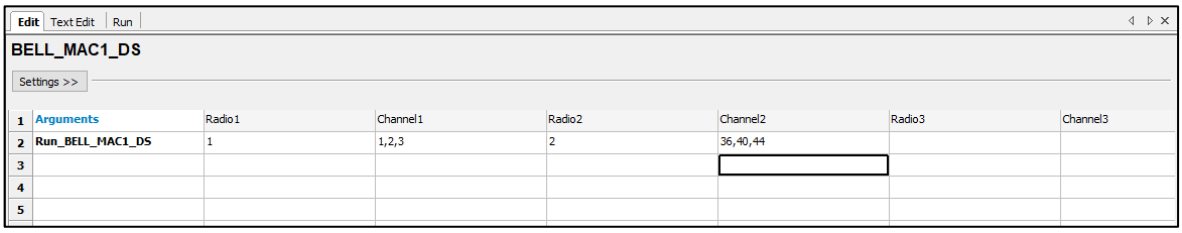
3. Select one or more test cases by checking the check box on the left of test case name.



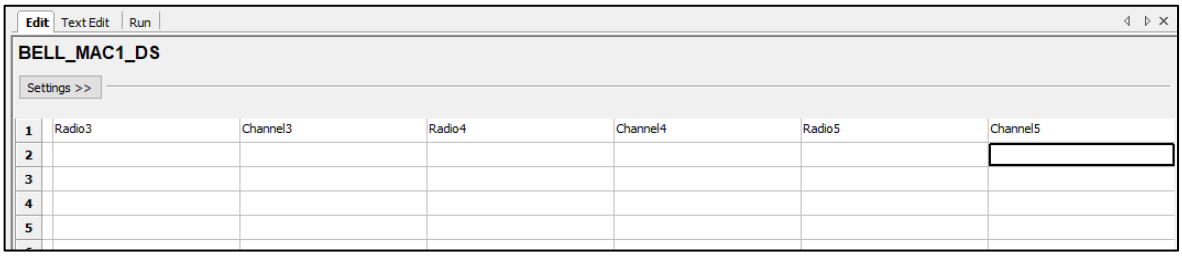
4. If the user wants to run all the test cases, then the user must ONLY check the check box next to the RunAllFlows.



5. The user can see the radio and channel values in test case parameters which were entered in the Design Tools parameters.

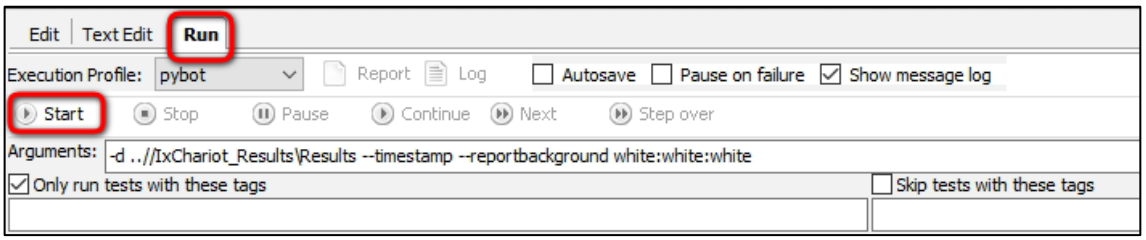


	Arguments	Radio1	Channel1	Radio2	Channel2	Radio3	Channel3
1							
2	Run_BELL_MAC1_DS	1	1,2,3	2	36,40,44		
3							
4							
5							



	Radio3	Channel3	Radio4	Channel4	Radio5	Channel5
1						
2						
3						
4						
5						

6. After selecting the test cases which has to be executed, go to the Run and click on the Start button.



7. It will take traverse the list of entered values for channel and extract each value and check against the current channel. If both are same then it will start the execution. If the current channel and mentioned channel does not match then it changes the current channel to the mentioned channel, verifies it and starts the execution.

   Automation solution will try three times to change channel value if unsuccessful before raising error message.

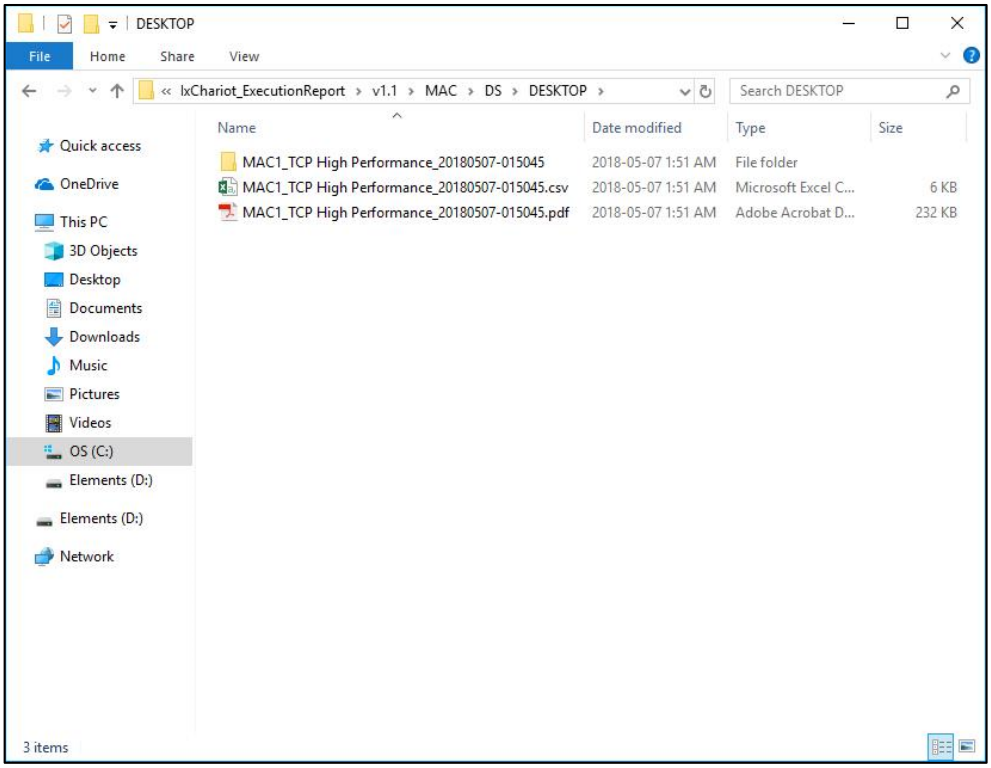
8. This will happen for all the channel values of each radio value.

9. It will start the execution. Wait till it is complete.

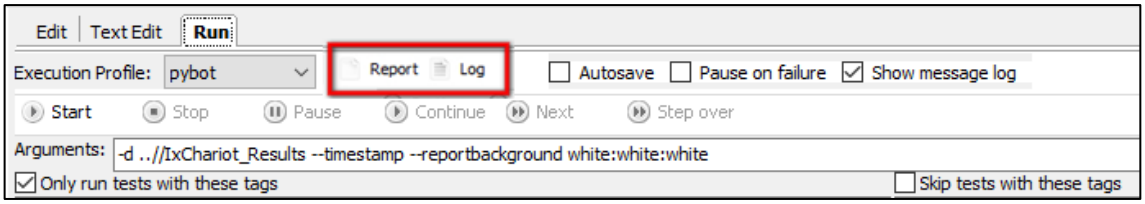
```
elapsed time: 0:00:20   pass: 0   fail: 0   current keyword: BELL_MAC1_DS.Run BELL MAC1 DS
command: pybot.bat -d ..\\IxChariot_Results\Results --timestamp --reportbackground white:white:white --argumentfile c:
==============================================================================================================
IxChariot
==============================================================================================================
IxChariot.Execute Flow MAC
==============================================================================================================
<

Starting test: IxChariot.Execute Flow MAC.BELL_MAC1_DS
20180516 03:20:43.673 :  INFO : ChannelChange.py -> checkChannel :: Starting to check current channel.
20180516 03:20:47.676 :  INFO : ChannelChange.py -> checkChannel :: Current Channel is : 3 for Radio:1
20180516 03:20:47.677 :  INFO : ChannelChange.py -> checkChannel :: Need to change the current channel value to 1
20180516 03:20:53.680 :  INFO : ChannelChange.py -> changeNCheck :: Current Channel is : 1 for Radio: 1
20180516 03:20:53.681 :  INFO : ChannelChange.py -> changeNCheck :: Channel value successfully changed.
20180516 03:20:53.686 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Connecting to https://192.168.2.30
20180516 03:20:53.707 :  INFO : Starting new HTTPS connection (1): 192.168.2.30
20180516 03:20:54.273 :  INFO : Starting new HTTPS connection (1): 192.168.2.30
20180516 03:20:54.308 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Created session 48
20180516 03:20:54.309 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Starting the session...
20180516 03:20:54.541 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Session has Started.
20180516 03:20:54.542 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Configuring the test...
20180516 03:20:55.157 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Starting the test...
```

6. Execution Reports

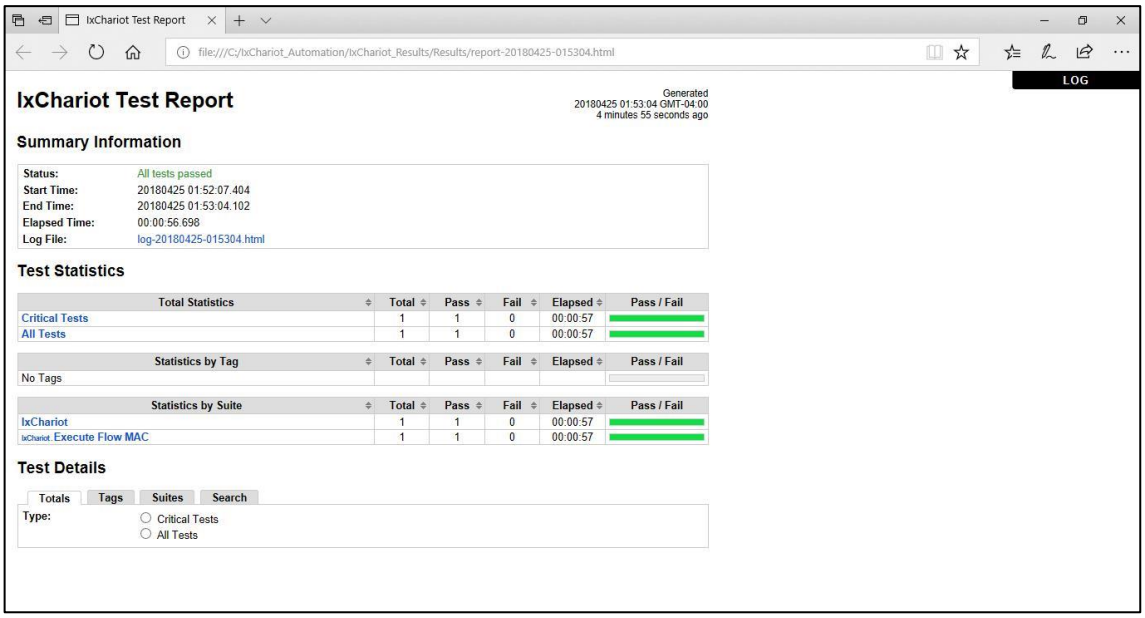
1. The user will be able to view the results of the executions in a specific folder as shown below. The result file will have a timestamp and will be named as
   - **"C:/ IxChariot_Automation/ IxChariot_ExecutionResults/<HomeHub Version>/ <FlowGroup>/<Direction>/<EndPoint1>/<EndPoint2>_<Script>_<Timestamp>.pdf**".
   - **"C:/ IxChariot_Automation/ IxChariot_ExecutionResults/<HomeHub Version>/ <FlowGroup>/<Direction>/<EndPoint1>/<EndPoint2>_<Script>_<Timestamp>.csv**".
   - **"C:/ IxChariot_Automation/ IxChariot_ExecutionResults/<HomeHub Version>/ <FlowGroup>/<Direction>/<EndPoint1>/<EndPoint2>_<Script>_<Timestamp>**".



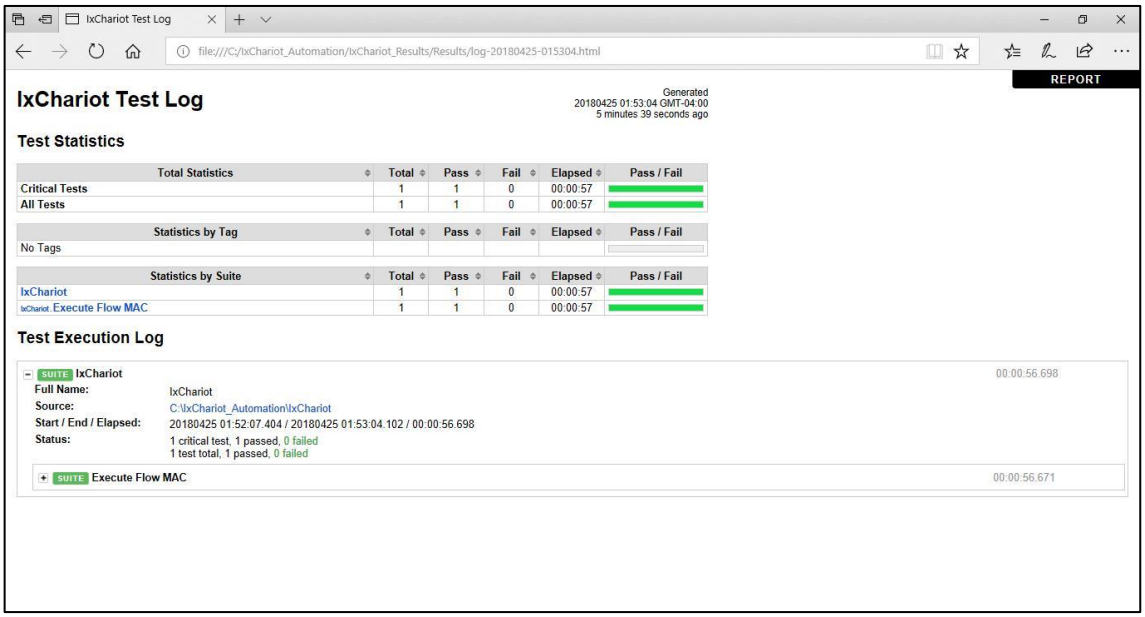
2. Click on the **Log / Report** button to view the result in the RIDE. The button becomes active on executing the test case.



3. On click of the Report button, in the browser, the report appears.



4. In the browser on the top right corner, the log button is there. Clicking on it, will show the log report and vice versa.



5. The user can find the Log files in the file location:

   “C:/ IxChariot_Automation/ IxChariot_Results/ Results/log - <Timestamp>.html”.

6. The user can find the report files in the file location:

   “C:/ IxChariot_Automation/ IxChariot_Results/ Results/report - <Timestamp>.html”.

7. Execution Logs

1. The user can find the log of the executed flow group test case in 2 places – RIDE and Detailed Log file
2. The user can see the log in the RIDE after execution.

```
Starting test: IxChariot.Execute Flow MAC.BELL_MAC1_DS
20180516 03:20:43.673 :  INFO : ChannelChange.py -> checkChannel :: Starting to check current channel.
20180516 03:20:47.676 :  INFO : ChannelChange.py -> checkChannel :: Current Channel is : 3 for Radio:1
20180516 03:20:47.677 :  INFO : ChannelChange.py -> checkChannel :: Need to change the current channel value to 1
20180516 03:20:53.680 :  INFO : ChannelChange.py -> changeNCheck :: Current Channel is : 1 for Radio: 1
20180516 03:20:53.681 :  INFO : ChannelChange.py -> changeNCheck :: Channel value successfully changed.
20180516 03:20:53.686 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Connecting to https://192.168.2.30
20180516 03:20:53.707 :  INFO : Starting new HTTPS connection (1): 192.168.2.30
20180516 03:20:54.273 :  INFO : Starting new HTTPS connection (1): 192.168.2.30
20180516 03:20:54.308 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Created session 48
20180516 03:20:54.309 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Starting the session...
20180516 03:20:54.541 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Session has Started.
20180516 03:20:54.542 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Configuring the test...
20180516 03:20:55.157 :  INFO : BELL_MAC1_DS.py -> BELL_MAC1_DS :: Starting the test...
```

3. The user can find the Detailed Log Report in the folder:

   "C:/ IxChariot_Automation/ IxChariot_Results/DetailedLog<Timestamp>".

   This file will be created every day and is stored in the above location.

```
2018-04-27 05:20:17 AM - INFO - CreateTestCase.py -> CreateTestCase :: Creating test case file for BELL_MAC3_DS
2018-04-27 05:20:17 AM - INFO - CreateTestCase.py -> CreateTestCase :: Starting validation of Test Parameters.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Starting Test Parameter Validation of Test Case : BELL_MAC3 _DS
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for FlowGroup has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for TestCaseName has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for EndPoint1 has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for EndPoint2 has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for Direction has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for Protocol has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for Script has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for Duration has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for NumberOfUsers has been validated successfully.
2018-04-27 05:20:17 AM - INFO - CreateTestCase.py -> CreateTestCase :: Validation of Test Parameters completed successfully.
2018-04-27 05:20:17 AM - INFO - CreateTestCase.py -> CreateTestCase :: Starting writing to python file.
2018-04-27 05:20:17 AM - INFO - CreateTestCase.py -> CreateTestCase :: Successfully created test case python file.
2018-04-27 05:20:17 AM - INFO - CreateTestCase.py -> CreateTestCase :: Creating test case file for BELL_MAC3_US
2018-04-27 05:20:17 AM - INFO - CreateTestCase.py -> CreateTestCase :: Starting validation of Test Parameters.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Starting Test Parameter Validation of Test Case : BELL_MAC3 _US
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for FlowGroup has been validated successfully.
2018-04-27 05:20:17 AM - INFO - Validation.py -> TestCase_validation :: Parameters for TestCaseName has been validated successfully.
```