



E.C.I.NETWORKS

IXCHARIOT – USER MANUAL

Bell Canada; ATL Lab

Bell

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

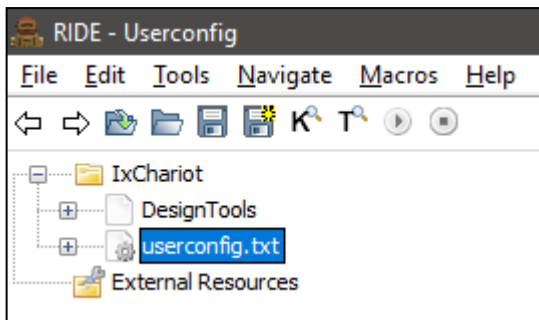
| Reviewer | Version Reviewed | Signature | Date |
|-----------------|------------------|-----------|------|
| Angelo Virgilio | | | |
| | | | |
| | | | |

Contents

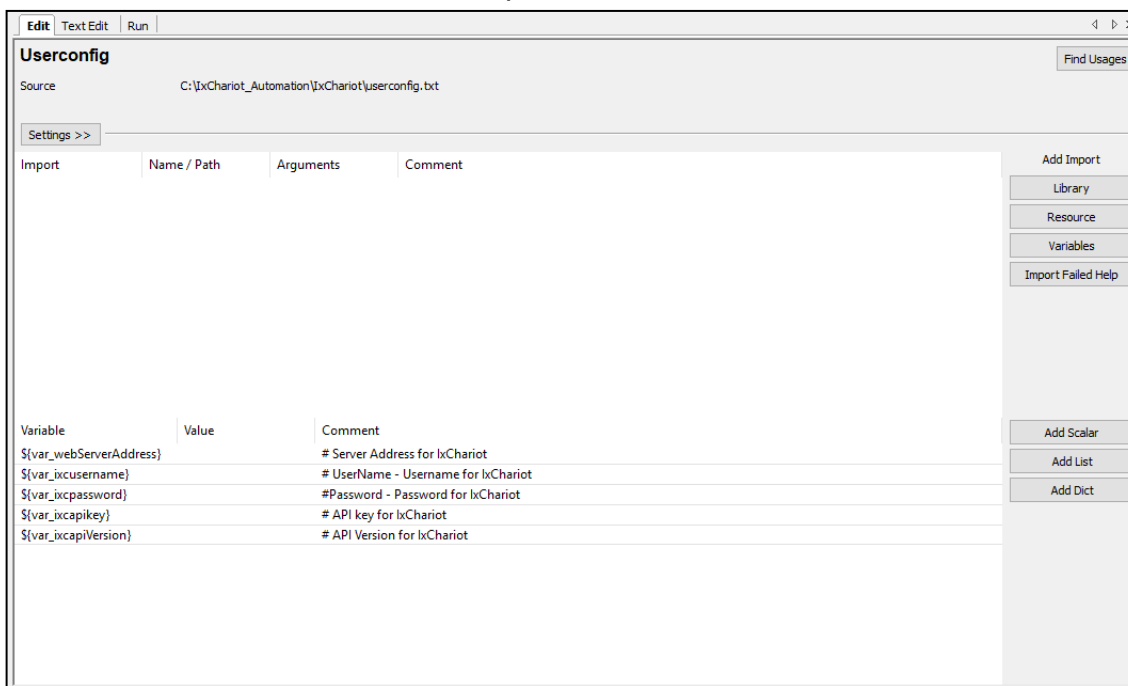
| | |
|---------------------------------------|----|
| 1. Configuring Common Parameters..... | 3 |
| 2. Configuring Test Parameters | 7 |
| 3. Configuring Result Arguments | 12 |
| 4. Executing Test Case..... | 14 |

1. Configuring Common Parameters

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



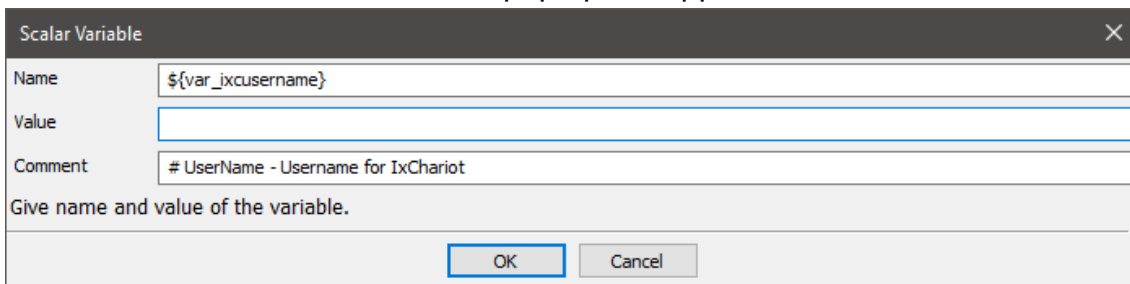
2. On the right, it will open the edit tab of the userconfig.txt.
Here the user will find the common parameters and the values for each of them.



- 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} | 192.168.1.100 | # IP Address of MacBook 1 |

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



Scalar Variable

Name: \${var_ixcusername}

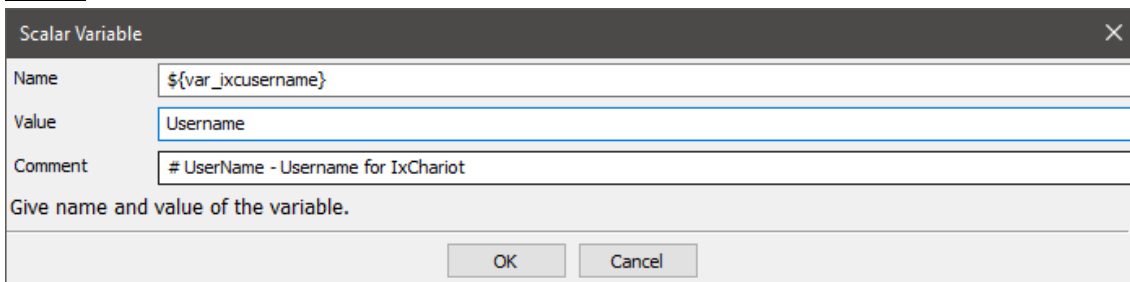
Value:

Comment: # UserName - Username for IxChariot

Give name and value of the variable.

OK Cancel

- Here the user can enter the value for each variable. Do not change the variable name.



Scalar Variable

Name: \${var_ixcusername}

Value: Username

Comment: # UserName - Username for IxChariot

Give name and value of the variable.

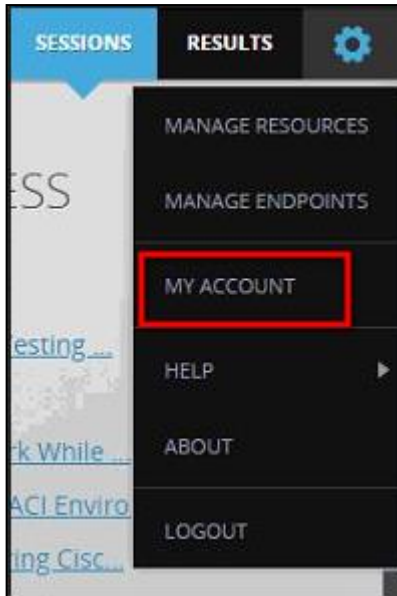
OK Cancel

- The user can give the value of the other variables repeating step 4 and 5.
- The user can find the values of username, password and API Key in the settings of IxChariot WebAPI.
- To find the above values, follow the steps below.

9. After logging into the user account, on the top right side, the user will find the **settings** button.



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





11. 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).

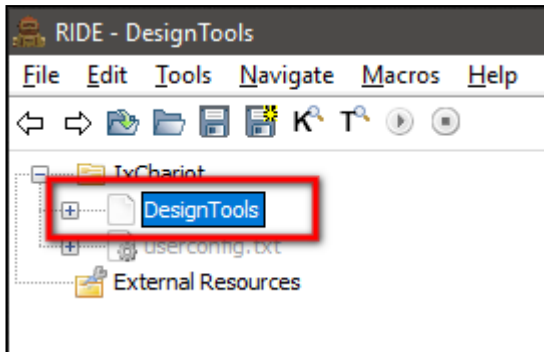
IxChariot > My Account

| | |
|------------------|--------------------------------------|
| Email: | cage.noire@gmail.com |
| Name: | Cage Noire |
| Password: | Change |
| Api Key: | 25596217-423e-4072-8d0a-ff6300d37d5f |
| Role: | Administrator |

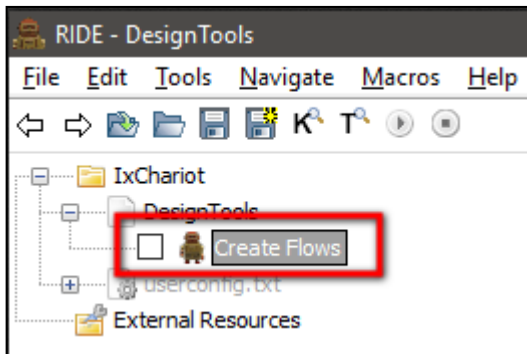
[User preferences](#)

2. Configuring Test Parameters

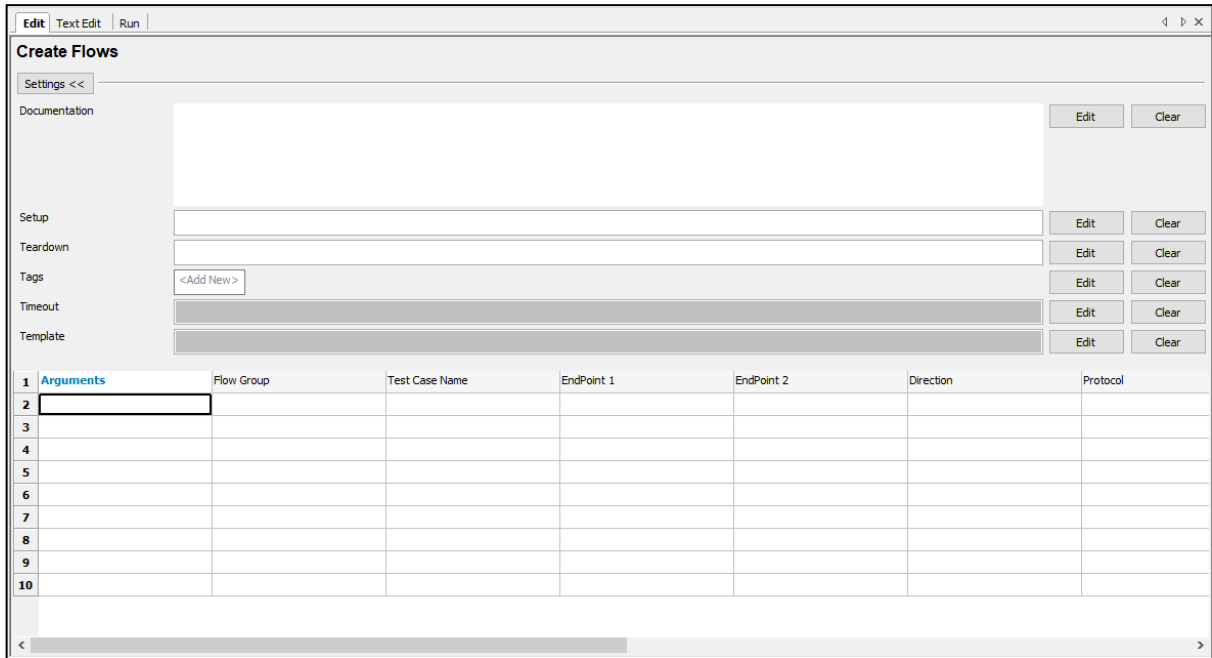
1. Now on the left panel, click on the plus sign next to **DesignTools** to expand.



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

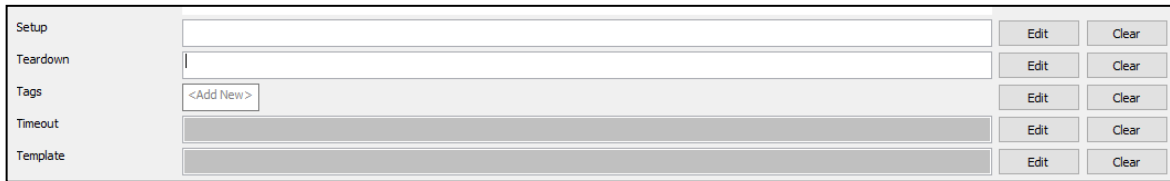


4. It will open the edit tab.

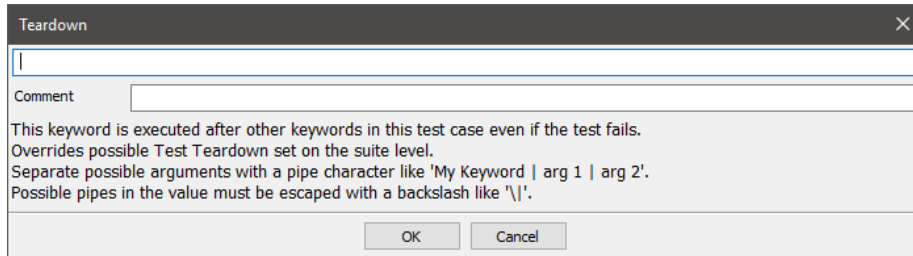


| | Arguments | Flow Group | Test Case Name | EndPoint 1 | EndPoint 2 | Direction | Protocol |
|----|-----------|------------|----------------|------------|------------|-----------|----------|
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |

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

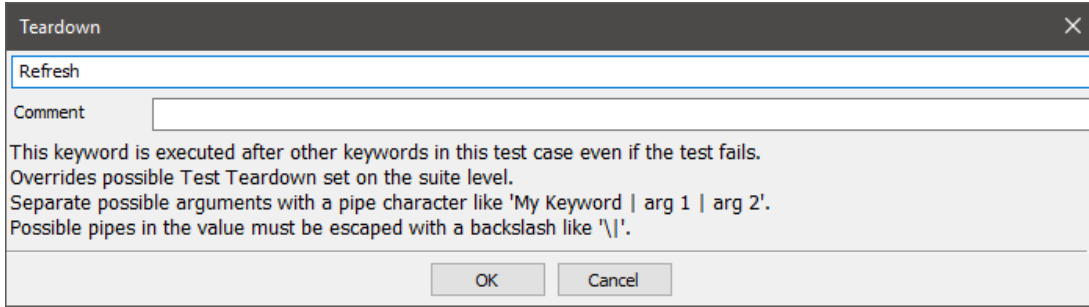


6. A pop up will come up.



This keyword is executed after other keywords in this test case even if the test fails.
 Overrides possible Test Teardown set on the suite level.
 Separate possible arguments with a pipe character like 'My Keyword | arg 1 | arg 2'.
 Possible pipes in the value must be escaped with a backslash like '\\'.
 OK Cancel

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



The image shows a 'Teardown' dialog box with a title bar containing a close button. It features a text input field with 'Refresh' entered, a 'Comment' label, and a larger text area containing instructions: 'This keyword is executed after other keywords in this test case even if the test fails. Overrides possible Test Teardown set on the suite level. Separate possible arguments with a pipe character like 'My Keyword | arg 1 | arg 2'. Possible pipes in the value must be escaped with a backslash like '\\'. At the bottom are 'OK' and 'Cancel' buttons.

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**.

| | |
|---|----------------|
| 1 | Arguments |
| 2 | CreateTestCase |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

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 a 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 <ul style="list-style-type: none"> • 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 users can be 10 as per the current license at Bell. |

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 >> | | | | | | | |
| 1 | Arguments | Flow Group | Test Case Name | EndPoint 1 | EndPoint 2 | Direction | Protocol |
| 2 | CreateTestCase | MAC | Bell_MAC1 | Desktop | MAC1 | DS | TCP |
| 3 | CreateTestCase | MAC | Bell_MAC1 | Desktop | MAC1 | US | UDP |
| 4 | CreateTestCase | OTHER | Bell_MAC2 | Desktop | Desktop | DS | TCP |
| 5 | CreateTestCase | OTHER | Bell_MAC2 | MAC2 | Desktop | US | UDP |
| 6 | | | | | | | |
| 7 | | | | | | | |

| | | | | | | | |
|---------------------|-----------|----------|----------------------|-----------------|--------|-----------------|--|
| Edit Text Edit Run | | | | | | | |
| Create Flows | | | | | | | |
| Settings >> | | | | | | | |
| 1 | Direction | Protocol | Script | Duration (secs) | Create | Number of users | |
| 2 | DS | TCP | TCP High Performance | 9 | | 1 | |
| 3 | US | UDP | UDP High Performance | 60 | | 1 | |
| 4 | DS | TCP | TCP High Performance | 60 | | 5 | |
| 5 | US | UDP | UDP High Performance | 9 | | 5 | |
| 6 | | | | | | | |
| 7 | | | | | | | |

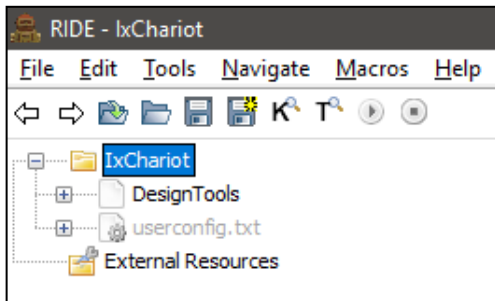
11. The user can repeat the steps 8 & 9 for creating multiple test cases.

3. Configuring Result Arguments

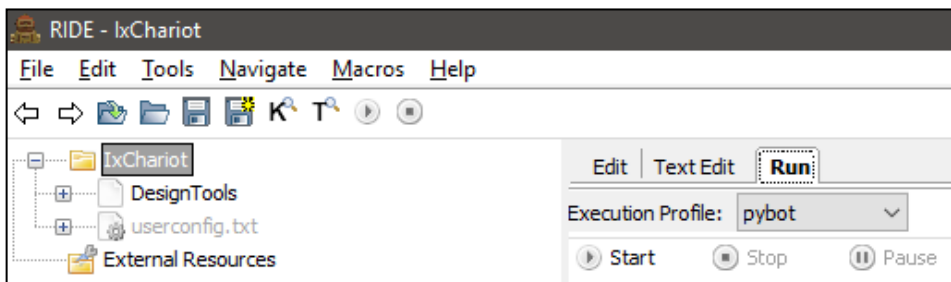
The section aims to customize the result files name, location and the background color.

The above command needs to be inserted in the arguments section, following the steps:

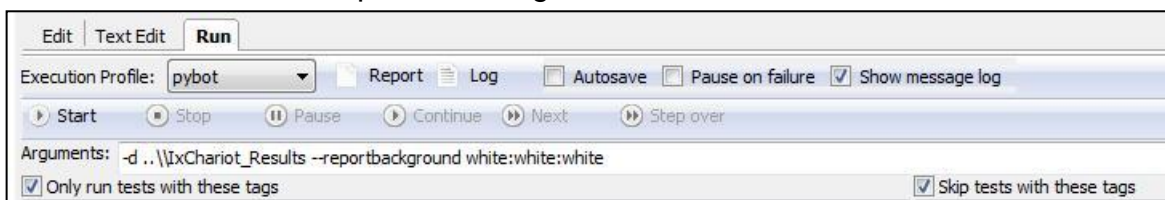
1. Click on project name.



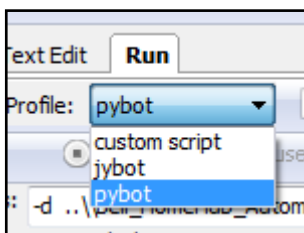
2. Click on **Run** tab.



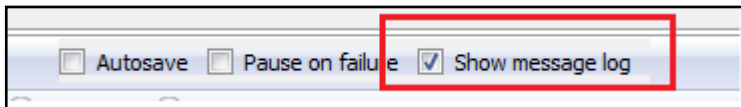
3. Copy the command **“-d ..\IxChariot_Results --timestamp --reportbackground white:white:white”** and paste it in arguments text area as shown.



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

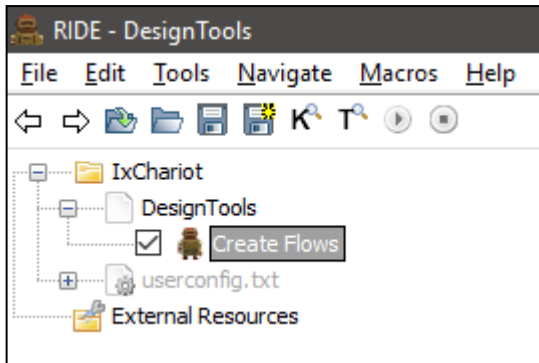


5. Click to checkbox to mark as selected for Show message log (if not already selected).

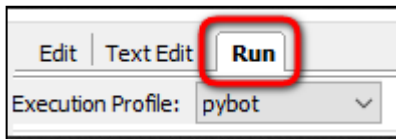


4. Executing Test Case

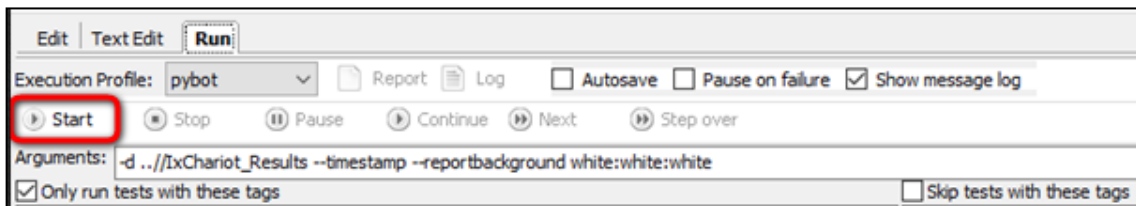
1. To start the execution, click on the checkbox to the left of “**Create Flows**”.



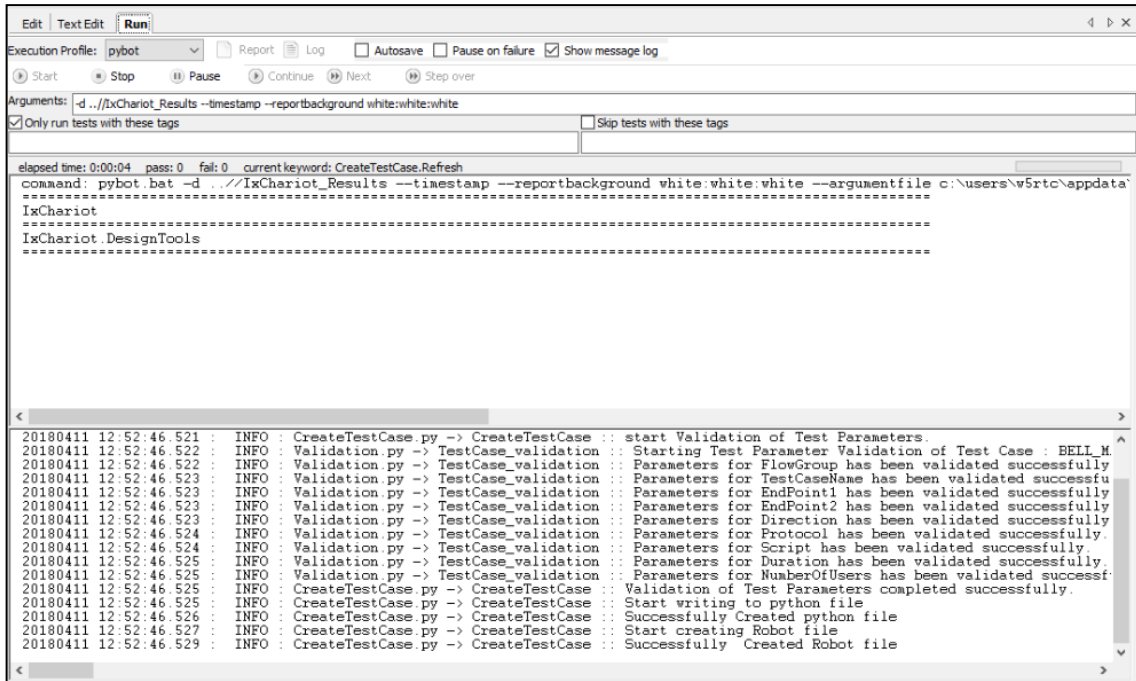
2. Click on the **Run** tab.



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



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

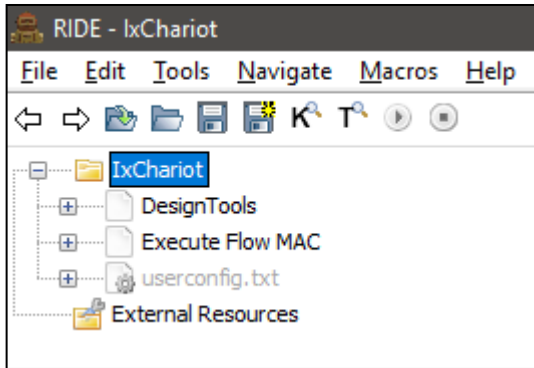


The screenshot shows the 'Run' dialog box in RIDE. The 'Execution Profile' is set to 'pybot'. The 'Arguments' field contains: `-d ../IxChariot_Results --timestamp --reportbackground white:white:white --argumentfile c:\users\v5rtc\appdata\...`. The 'Only run tests with these tags' checkbox is checked. The 'Log' checkbox is also checked. The 'Run' button is highlighted. Below the dialog, a log window displays the execution progress, showing the start of the 'IxChariot' test suite and the 'DesignTools' test case. The log includes timestamps, log levels (INFO), and messages indicating the successful validation of test parameters and the creation of test files.

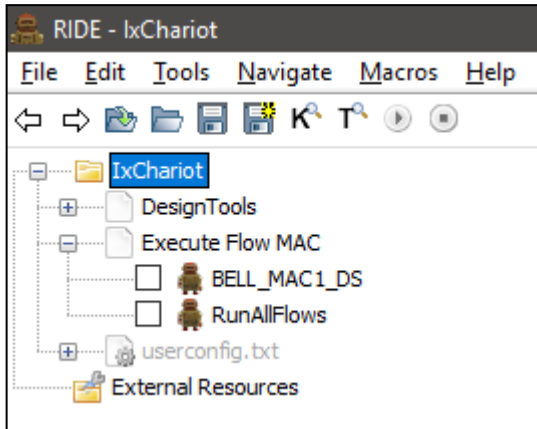
```

elapsed time: 0:00:04 pass: 0 fail: 0 current keyword: CreateTestCase.Refresh
command: pybot.bat -d ../IxChariot_Results --timestamp --reportbackground white:white:white --argumentfile c:\users\v5rtc\appdata\...
IxChariot
IxChariot.DesignTools
20180411 12:52:46.521 : INFO : CreateTestCase.py -> CreateTestCase : start Validation of Test Parameters.
20180411 12:52:46.522 : INFO : Validation.py -> TestCase_validation : Starting Test Parameter Validation of Test Case : BELL_M
20180411 12:52:46.522 : INFO : Validation.py -> TestCase_validation : Parameters for FlowGroup has been validated successfully
20180411 12:52:46.523 : INFO : Validation.py -> TestCase_validation : Parameters for TestCaseName has been validated successfully
20180411 12:52:46.523 : INFO : Validation.py -> TestCase_validation : Parameters for EndPoint1 has been validated successfully
20180411 12:52:46.523 : INFO : Validation.py -> TestCase_validation : Parameters for EndPoint2 has been validated successfully
20180411 12:52:46.523 : INFO : Validation.py -> TestCase_validation : Parameters for Direction has been validated successfully
20180411 12:52:46.524 : INFO : Validation.py -> TestCase_validation : Parameters for Protocol has been validated successfully
20180411 12:52:46.524 : INFO : Validation.py -> TestCase_validation : Parameters for Script has been validated successfully
20180411 12:52:46.525 : INFO : Validation.py -> TestCase_validation : Parameters for Duration has been validated successfully
20180411 12:52:46.525 : INFO : Validation.py -> TestCase_validation : Parameters for NumberOfUsers has been validated successfully
20180411 12:52:46.525 : INFO : CreateTestCase.py -> CreateTestCase : Validation of Test Parameters completed successfully.
20180411 12:52:46.525 : INFO : CreateTestCase.py -> CreateTestCase : Start writing to python file
20180411 12:52:46.526 : INFO : CreateTestCase.py -> CreateTestCase : Successfully Created python file
20180411 12:52:46.527 : INFO : CreateTestCase.py -> CreateTestCase : Start creating Robot file
20180411 12:52:46.529 : INFO : CreateTestCase.py -> CreateTestCase : Successfully Created Robot file
  
```

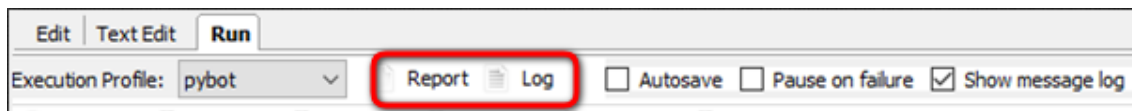
- On successful execution, RIDE will close and automatically reopen with the Flow group test suites appearing on the left under the Design Tools.



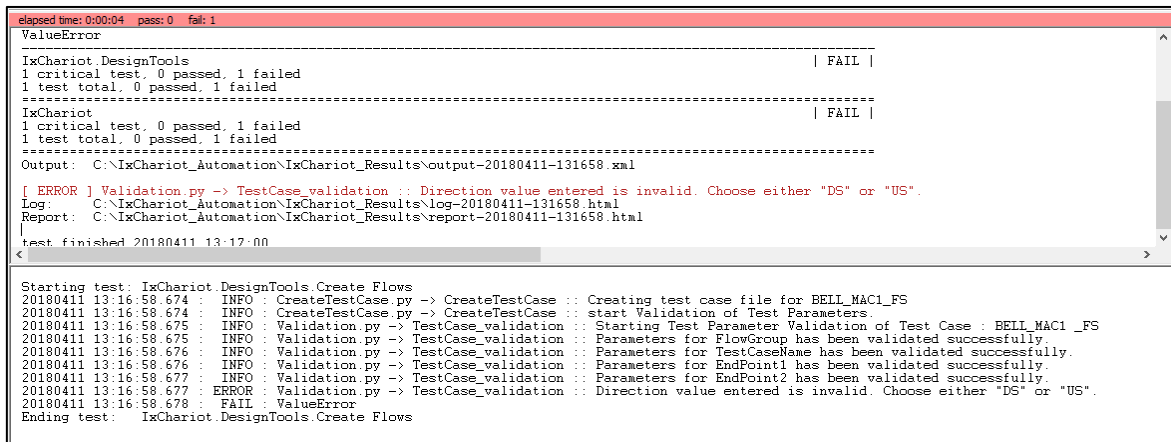
6. It will show all the test case listed under the Flow Group name during creation.



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.