

# ReporterForUiPath 3.0

## User Manual

Created by : Vivek Ingole, Parmila Lama

Creation Date : 12<sup>th</sup> Jun, 2020

Updated Date : 30<sup>th</sup> April, 2023





## Contents

Introduction :	5
Importance of the Reporter :	5
New features of the Reporter v3.0:	6
1. Features Of the Reporter :	6
1.1 Independency.....	6
1.2 Categorization.....	6
1.3 Filters :	6
1.4 Dashboard :	6
1.5 Screenshot :	7
1.6 Log Message:.....	7
2. Package Activities :	7
2.1 Create Report :	7
2.1.1 Report Name [mandatory]:.....	7
2.1.2 Report Location [mandatory]: .....	7
2.1.3 reportId [optional] .....	8
2.2 Start Suite :	8
2.2.1 Suite Name [mandatory]:.....	8
2.2.2 reportId [optional]: .....	8
2.2.3 suiteId [optional]:.....	8
2.3 Start Test :	9
2.3.1 Test Name [mandatory]:.....	9
2.3.2 Test Description [optional]: .....	9
2.3.3 Test Tag [optional]: .....	9
2.3.3 ReportId [optional]: .....	9
2.3.3 SuiteId [optional]: .....	9
2.3.4 Priority [optional] [Default : Low]:.....	9
2.3.5 testId [optional] : .....	10
2.4 Step Status :	10
2.4.1 Step Name [mandatory]: .....	10
2.4.2 Step Description [mandatory]:.....	10
2.4.3 Status [mandatory] [Default : Compare]: .....	10
2.4.5 ReportId [optional]: .....	11
2.4.6 TestId [optional]:.....	11





---

2.4.7 Actual Result [optional]: .....	11
2.4.8 Expected Result [optional]: .....	11
2.4.9 Code Block [optional]: .....	11
2.4.10 Take Screenshot [optional] [Default : True]: .....	11
2.4.11 Status [optional] : .....	11
2.5 Log Message : .....	11
2.5.1 Message [mandatory]: .....	12
2.5.2 Message Type [mandatory] .....	12
2.5.3 ReportId [optional]: .....	12
2.5.4 TestId [optional]: .....	12
2.5.5 Text [optional] : .....	12
3. Reports : .....	12
3.1 Test Section .....	13
1. Report Title : .....	13
2. Search bar : .....	13
3. Menu : .....	13
4. Filters : .....	13
5. Test Timings : .....	13
6. Screenshot : .....	13
7. Status Types : .....	13
8. Test Item : .....	14
9. Code Block : .....	14
10. Step Item : .....	14
11. Step Filters : .....	14
3.2 Dashboard Section .....	15
1. Test Analysis : .....	15
2. Step Analysis : .....	15
3. Log Events : .....	15
4. Total Tests : .....	15
5. Total Steps : .....	15
6. Start Time : .....	15
7. End Time : .....	15
8. Priority wise Report : .....	15
9. Tag wise Report : .....	16
10. Suite wise Report : .....	16

---





---

3.3 Logs Section .....	16
1. Test List: .....	16
2. Counts: .....	16
3. Filters:.....	16
4. Logs: .....	16





## Introduction :

Robotic Process Automation: An amazing technology to automate every manual process, from a simple routine task of sending emails to a complex business process, RPA allows to do it all smoothly with minimum efforts. Out of many domains where RPA application is gaining popularity, one area where RPA application is taking a good shape is the domain of testing.

Testing is an activity done to deliver quality products to the customers and hence testing becomes an important activity/task for businesses. Test Automation is done efficiently using UiPath tool and with the newest release of UiPath Studio Pro version that grants us the access to Application Testing, RPA Testing, API Testing and Mobile testing projects, Test-Automation becomes effortless.

## Importance of the Reporter :

Any test activity would be incomplete if we do not provide a test-report. Test-report is an important document which comprises visual/graphical representation of the tests done on the product and the corresponding test results, and it is shared among the stakeholders of the project. As the test-report holds an important value for the test activities, we felt a strong need of an activity in UiPath that would generate good test-report automatically along with the automated test execution. ReporterForUiPath addresses the above stated need perfectly. ReporterForUiPath is the UiPath activity suite with many features packed in it that helps in generating beautiful test-reports simultaneously along with the execution of the tests.





## New features of the Reporter v3.0:

A new feature has been introduced in the latest version of ReporterForUiPath, which enables parallel report generation. This means that multiple HTML reports can be generated simultaneously during execution. To enable parallel report generation new parameters has been introduced in all activities. Detailed information can be found in individual section of activities.

### Note:-

*If the parallel report generation feature is not needed, the parameters can simply be ignored. In this case, the reporter will behave the same as before. Additionally, for those who have implemented the reporter according to the previous version, there is no need to worry as it supports backward compatibility.*

## 1. Features Of the Reporter :

**1.1 Independency** : The activities in this program are both robust and flexible, allowing them to adapt to any situation. They can be used at any point within the workflow. For example, if a report hasn't been created and a suite is created directly, the reporter will recognize this and initialize the report. Similarly, if nothing has been created beforehand and the step status activity is used, the default report, suite, and test will be generated first, with the step added to the default test afterwards.

**1.2 Categorization** : By default, this reporter categorizes tests by test, suite, and priority. Additionally, it allows you to define tags for your tests to further categorize them.

**1.3 Filters** : Filters can be used to display only categorized test cases in the generated report. The report will have four filters: pass/fail, suite, tag, and priority filter. Multiple filters can be added to display specific tests as well.

**1.4 Dashboard** : The reporter records every logging movement and generates a dashboard. This dashboard provides information on execution timings, the count of all passed/failed tests, status-wise counts of all steps, and contains various sub-reports.





**1.5 Screenshot :** One of the most important and useful features is the ability to add runtime screenshots to test steps. These screenshots can be used as execution proof, or in case of a failed test case, to help identify the cause - whether it is a code issue or application issue.

**1.6 Log Message:** As like test step you can add log messages for the test as well.

## 2. Package Activities :

This package will come with 8 activities.

- Create Report
- Start Suite
- Start Test
- Step Status
- Log Message

**2.1 Create Report :** This Activity creates new Report at the given location. It can be used by multiple times within same workflow to get separate reports for different modules. Both the reports will get stored in same location.

### Steps Followed by Create Report Activity :

- Creates new Folder with Reporter name & current date & time to avoid overriding.
- Inside newly created folder, It creates two sub folders i.e. recordings, screenshots with main .html report.

### Properties :

#### Input :

**2.1.1 Report Name [mandatory]:** It accepts string value, Which will be part of report folder name & Report name.

**2.1.2 Report Location [mandatory]:** It accepts string value representing location for where you want to store report files.

#### Output :





**2.1.3 reportId [optional]** : The reporter returns a unique ID that can be used to link the report with its associated suite or tests. This ID must be passed as input to other activities.

For example, if two separate reports are initially created (report1 and report2), when adding a new suite, test, or step status activity, it is possible to choose which report to update with the details.

**2.2 Start Suite :** This Activity creates new test suite in current report. It is not mandatory to create report before creating any suite. You can directly start with this activity. But as a good practice first initialize report before using this activity.

**Steps Followed by Start Suite Activity :**

- First it will check is any report initialized before this activity, If not then reporter will initialize it by itself.
- If report is Initialized by the reporter, then execution reports will get stored at project location inside Reports folder.
- If report is already initialized by the user then it will simply add new suite into it, but after creation of any suite will not updates the report content. Reports content only updated by Start Test & Step Status Activity.

**Properties :**

**Input :**

**2.2.1 Suite Name [mandatory]:** It accepts string value, which is name of the suite.

**2.2.2 reportId [optional]:** The activity accepts a string value, which is the report ID that we want to link with. If reportId is not null then new suite will be created in given report reference, or else it will use current report.

**Output :**

**2.2.3 suiteId [optional]:** It returns the string value, which is unique id for the suite. We can pass suiteid as a input to Start Test activity to link together.







**2.3 Start Test :** This Activity creates new test in current suite. It is not mandatory to create report/suite before creating any test. You can directly start with this activity. But as a good practice first initialize report then create new suite then use this activity.

**Steps Followed by Start Test Activity :**

- First it will check is any report initialized, If not then reporter will initialize default report.
- Then it will check is any suite created, If not then reporter will creates default suite in current report.
- After confirmation of Report & Suite creation, it will add new test in current suite.
- Finally updates report content.

**Properties :**

**Input :**

**2.3.1 Test Name [mandatory]:** It accepts string value, which is name of the test.

**2.3.2 Test Description [optional]:** It accepts string value, which describes what the test all about.

**2.3.3 Test Tag [optional]:** It accepts string value, which can be used to filter out tests. Tag name could be module name or any text that represents test.

**2.3.3 ReportId [optional]:** The activity accepts a string value, which is the report ID that we want to link with. If reportId is not null then new test will be created in given report reference, or else it will use current report.

**2.3.3 SuiteId [optional]:** It accepts string value, which is suite id we want to link with.

If suiteId is not null then new test will be created in given suite reference, or else it will use current suite.

**Options :**

**2.3.4 Priority [optional] [Default : Low]:** It accepts Enum value of type priority. Priority can be Low, Medium, High, Critical.

**Output :**





**2.3.5 testId [optional]** : It returns the string value, which is unique id of the test. This id can be passed as input to Step Status or Log Message activity to link then together.

**2.4 Step Status :** This Activity logs result of the step into the current test. It is not mandatory to create report/suite/test before logging step status. You can directly start with this activity.

But as a good practice first initialize report then create new suite then create test & after test creation use this activity.

**Steps Followed by Step Status Activity :**

- First it will check is any report initialized, If not then reporter will initialize default report.
- Then it will check is any suite created, If not then reporter will creates default suite in current report.
- Then it will check is any test created, If not then reporter will creates default test in current suite.
- After confirmation of Report, Suite & Test creation, it will add step status in current test.
- Finally updates report content.

**Properties :**

**Input :**

**2.4.1 Step Name [mandatory]:** It accepts string value, which is name of the Step.

**2.4.2 Step Description [mandatory]:** It accepts string value, which describes what the step all about.

**2.4.3 Status [mandatory] [Default : Compare]:** There are two ways to log status: manually or programmatically.

If the user wants to directly set the step status, they can use the Pass, Fail, Info, Skip, Exception, and Warning status options. Alternatively, if they want to set the step status by comparing actual and expected results, they should select the compare mode (one of the step status values) and provide the actual and expected values in the input field. The reporter will then compare the results and set the status accordingly.





**2.4.5 ReportId [optional]:** The activity accepts a string value, which is the report ID.

**2.4.6 TestId [optional]:** The activity accepts a string value, which is the test ID. The combination of the report ID and test ID is used to identify which report the step status should be updated in.

**Result :**

**2.4.7 Actual Result [optional]:** It accepts string value, which is the Actual value of the step that we get at runtime.

**2.4.8 Expected Result [optional]:** It accepts string value, which is Expected value of the step that we have to compare with actual result.

**Exception Message :**

**2.4.9 Code Block [optional]:** It accepts string value, we can use this property to display coding information like exception messages. This can be used to display any information but as a recommendation use it for exception messages.

**Options :**

**2.4.10 Take Screenshot [optional] [Default : True]:** It accepts Boolean value, if value is true then runtime screenshot gets attached to the step or else it will skip this action.

**Output :**

**2.4.11 Status [optional] :** It returns String value, which is nothing but status of the step. This property will helpful in case if step status sets to compare mode and you want to know what exactly the status is i.e. Pass / Fail.

**2.5 Log Message :** This activity will adds new log entry with given log type

**Steps Followed by Start Recording Activity :**

- First it will check is any test available, if yes the simply it will add new log entry.
- If no any test found then it will create default test and log the message into it.

**Properties :**



**Input :**

**2.5.1 Message [mandatory]:** It accepts string value, which is log message.

**2.5.2 Message Type [mandatory] [Default : Info]:** It accepts string value, which is log message type (Info, Warning, Error).

**2.5.3 ReportId [optional]:** The activity accepts a string value, which is the report ID.

**2.5.4 TestId [optional]:** The activity accepts a string value, which is the test ID. The combination of the report ID and test ID is used to identify which report the log message should be updated in.

**Output :**

**2.5.5 Text [optional] :** It returns String value (Log message).

### 3. Reports :

Reports can be generated at any of the two locations i.e. user's given location or default location (.\\Reports\\).

If reports are created by default through reporter then it will get stored in default location or else stored at given location.

Inside generated report folder, there are two sub-folders i.e. recordings & screenshots with .html file (Actual Report). Execution recordings will get store in recordings folder only if recording location is set to current report. Screenshots taken from Step Status activity will get stored in screenshots folder. Open .html file in chrome browser to see reports.







**8. Test Item :** Test item will contain test name, starting time, priority, count of total steps, suite name & tag name if exists.

**9. Code Block :** Code block can display very long exception messages using text area.

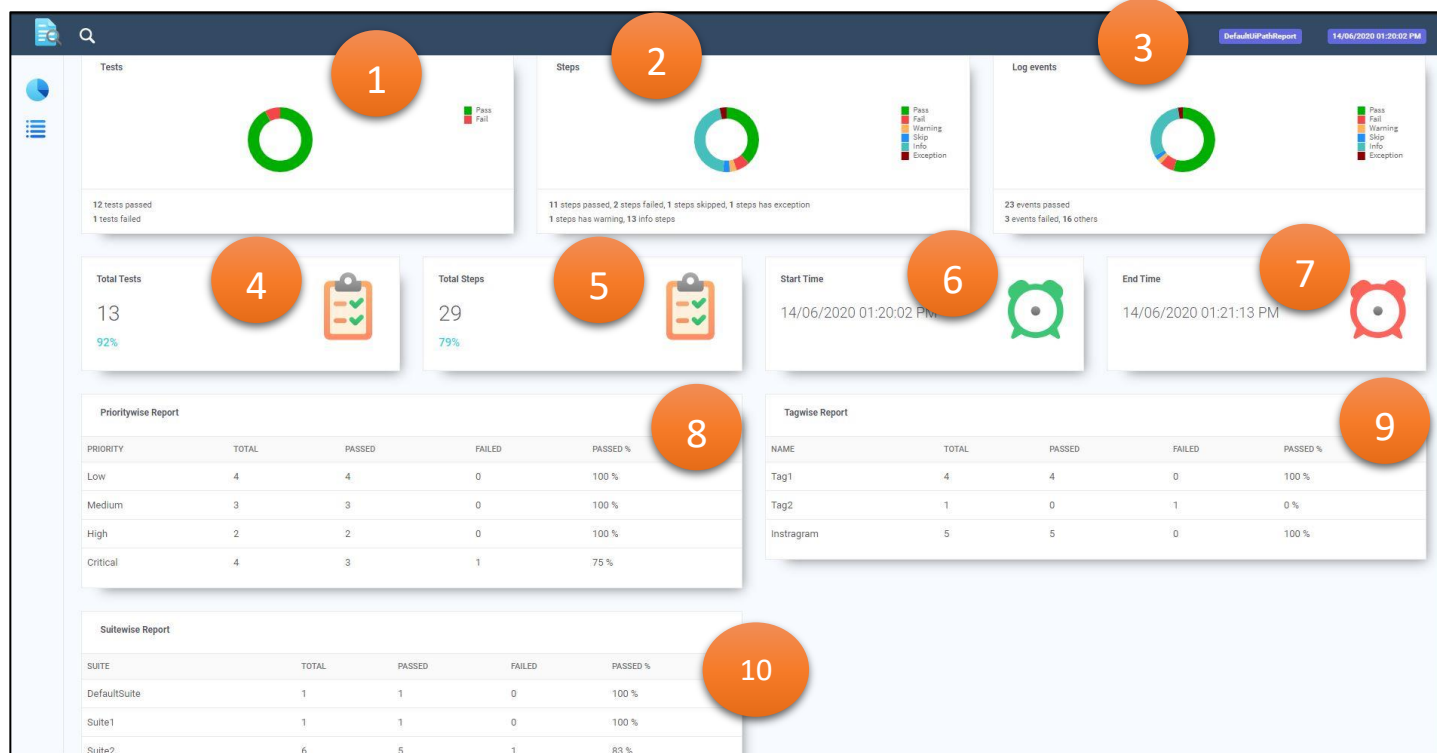
**10. Step Item :** Step item can contains following items,  
Step icon will change accordingly to step status,  
Start time of step,  
Step name along with screenshot button if it has taken,  
Step description,  
Actual & Expected results if it is passed through activity,  
Exception message in separate code block if any.

**11. Step Filters :** If there are multiple types of step status and we are only interested in the failed steps, we can apply a failed step filter to display only those steps.





## 3.2 Dashboard Section



**1. Test Analysis :** It will represents pass/fail count of all the tests using graphical format.

**2. Step Analysis :** It will represents counts of all types of step status using graphical format.

**3. Log Events :** It will represent counts of all types of logging activity for step as well as test using graphical format.

**4. Total Tests :** Counts of total test with percentage about how much tests are passed from total count.

**5. Total Steps :** Counts of total step with percentage about how much steps are passed from total count (Step which are Passed, Info & Warning considered as pass).

**6. Start Time :** Execution start date & time.

**7. End Time :** Execution end date & time.

**8. Priority wise Report :** Based upon test priority reporter will categorize tests and it will give pass / fail counts for each of them.





**9. Tag wise Report :** Based upon test tag reporter will categorize tests and it will give pass / fail counts for each of them.

**10. Suite wise Report :** Based upon test suite reporter will categorize tests and it will give pass / fail counts for each of them.

### 3.3 Logs Section

The screenshot displays the ReporterForUiPath interface with the following components:

- 1. Test List:** A sidebar on the left containing a list of tests (Test-1 to Test-7). Test-3 is selected and highlighted.
- 2. Counts:** A summary bar for the selected test (Test-3) showing counts for Info (10), Warning (10), and Error (10) messages.
- 3. Filters:** A filter bar above the log table with a 'Clear Filter' button and buttons for 'Info', 'Warning', and 'Error'.
- 4. Logs:** A table displaying individual log messages with columns for Status, Timestamp, and Log Message.

STATUS	TIMESTAMP	LOG MESSAGE
Info	06:02:31 PM	Log Message 1
Warning	06:02:31 PM	Log Message 1
Error	06:02:31 PM	Log Message 1
Info	06:02:33 PM	Log Message 2
Warning	06:02:33 PM	Log Message 2
Error	06:02:33 PM	Log Message 2
Info	06:02:34 PM	Log Message 3
Warning	06:02:34 PM	Log Message 3
Error	06:02:34 PM	Log Message 3
Info	06:02:35 PM	Log Message 4
Warning	06:02:35 PM	Log Message 4
Error	06:02:35 PM	Log Message 4
Info	06:02:36 PM	Log Message 5
Warning	06:02:37 PM	Log Message 5

**1. Test List:** Test list displays all the tests which having log details.

**2. Counts:** It displays separate counts for each log message type (Info, Warning, Error).

**3. Filters:** contains 3 types of filters with clear button. Only single filter can be applied at a time.

**4. Logs:** Lists all the logs added for selected test.







**Thank You**

