



# How to integration

## Testlink - Robot framework - Jenkins



<a href="#">Overview</a>	<a href="#">1</a>
<a href="#">Target</a>	<a href="#">2</a>
<a href="#">Work flow</a>	<a href="#">2</a>

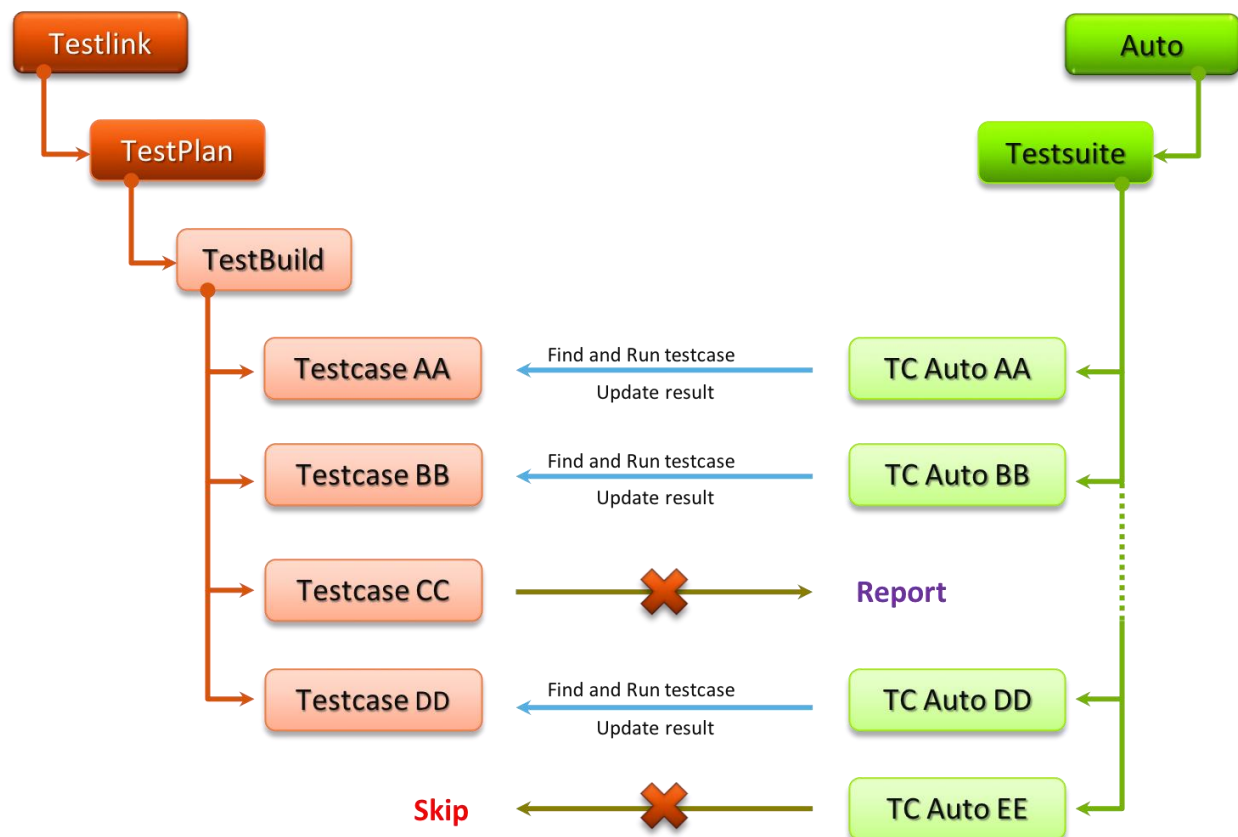
Jane Dinh – Iris

Sep 10, 2018

## Overview

With the company's current model, managers can check the report of automation and manual testing by two independent locations: Jenkins for the automation model, and Test link for the manual model. With increasing demand in the review reports, to manage members' performance and to test how the system works quickly and efficiently, the requirement to report on one site is necessary. To solve that problem, the integration support report of both automation and manual on test link site is the subject that we need to discuss.

This library is designed to get all test cases on the test link written to test this feature compare with the test case written automation. Otherwise if matched with automation test case, they will be executed and recorded the result after that test case is executed on testlink. And the library will record one more report overview of results, both result manual and automation to this feature.



## Target

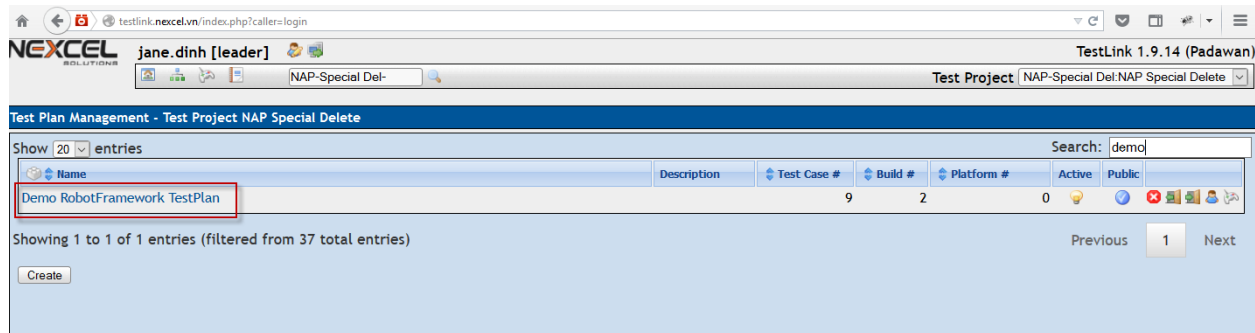
1. Create a full test cases on the test link for each feature.
2. Write automation on the robot framework and record the results on test link site.
3. Create a job on Jenkins to run the feature permanently before release on production environment.
4. Config send mail report to managers & leaders

## Work flow

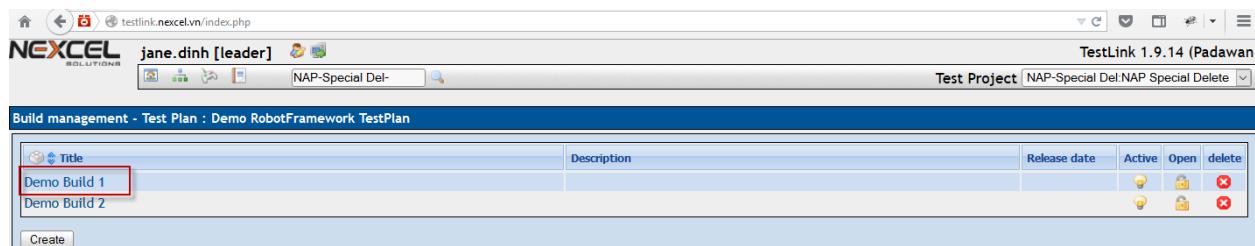
*(Work flow from receiving a feature to run a job on jenkins with integrated logging of results on the testlink and send mail report)*

### I. Create testcases on testlink

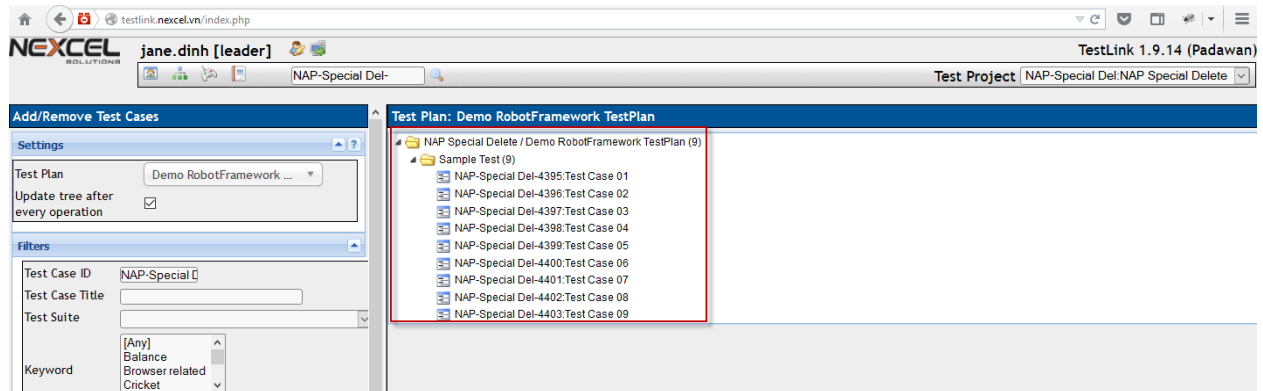
- Create test plan in project by feature (ignore prefix #)



- Create test build in the new test plan



- Create test case for this feature, and add test case to the test plan



TestLink 1.9.14 (Padawan)

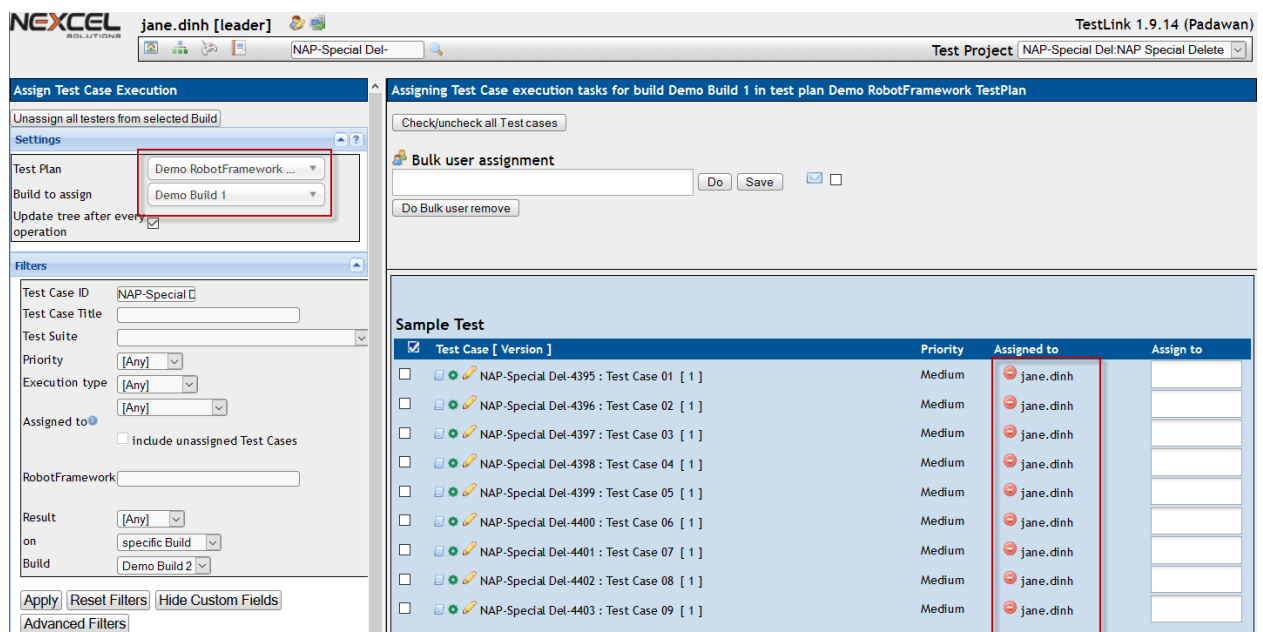
Test Project: NAP-Special Del-NAP Special Delete

Test Plan: Demo RobotFramework TestPlan

Sample Test (9)

- NAP-Special Del-4395: Test Case 01
- NAP-Special Del-4396: Test Case 02
- NAP-Special Del-4397: Test Case 03
- NAP-Special Del-4398: Test Case 04
- NAP-Special Del-4399: Test Case 05
- NAP-Special Del-4400: Test Case 06
- NAP-Special Del-4401: Test Case 07
- NAP-Special Del-4402: Test Case 08
- NAP-Special Del-4403: Test Case 09

- Assign test case to members



TestLink 1.9.14 (Padawan)

Test Project: NAP-Special Del-NAP Special Delete

Assigning Test Case execution tasks for build Demo Build 1 in test plan Demo RobotFramework TestPlan

Check/uncheck all Test cases

Bulk user assignment

Do Bulk user remove

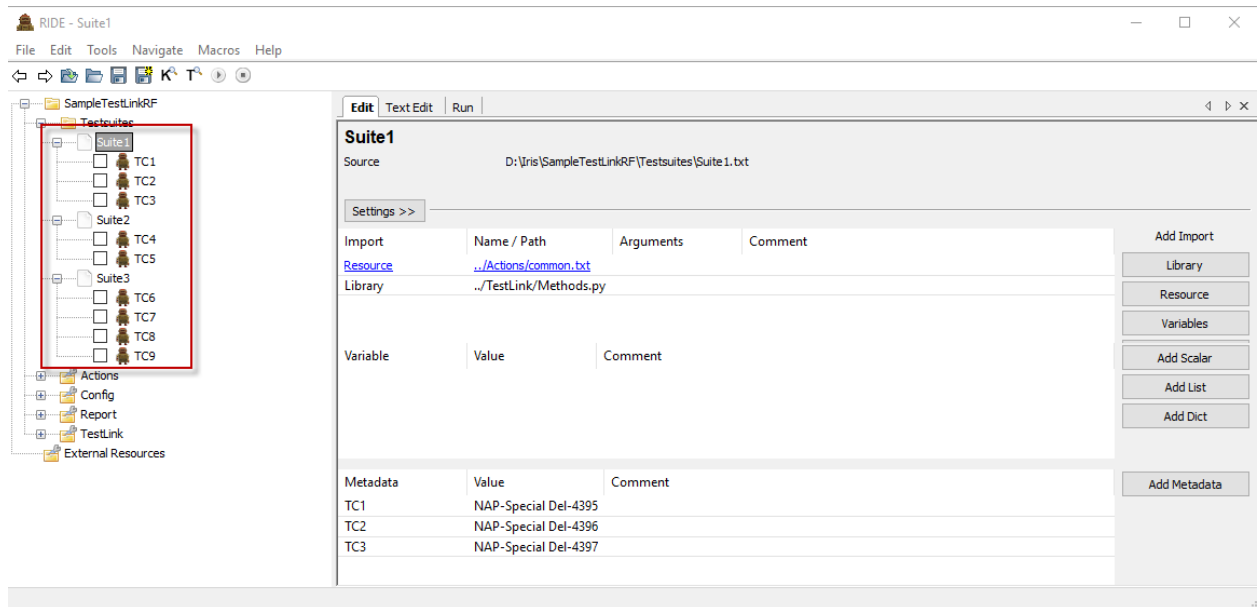
Sample Test

Test Case [ Version ]	Priority	Assigned to	Assign to
<input type="checkbox"/> NAP-Special Del-4395 : Test Case 01 [ 1 ]	Medium	jane.dinh	
<input type="checkbox"/> NAP-Special Del-4396 : Test Case 02 [ 1 ]	Medium	jane.dinh	
<input type="checkbox"/> NAP-Special Del-4397 : Test Case 03 [ 1 ]	Medium	jane.dinh	
<input type="checkbox"/> NAP-Special Del-4398 : Test Case 04 [ 1 ]	Medium	jane.dinh	
<input type="checkbox"/> NAP-Special Del-4399 : Test Case 05 [ 1 ]	Medium	jane.dinh	
<input type="checkbox"/> NAP-Special Del-4400 : Test Case 06 [ 1 ]	Medium	jane.dinh	
<input type="checkbox"/> NAP-Special Del-4401 : Test Case 07 [ 1 ]	Medium	jane.dinh	
<input type="checkbox"/> NAP-Special Del-4402 : Test Case 08 [ 1 ]	Medium	jane.dinh	
<input type="checkbox"/> NAP-Special Del-4403 : Test Case 09 [ 1 ]	Medium	jane.dinh	

## II. Write automation in robot framework

Refer to Library: [git@gitlab.nexcel.vn:lex.khuat/SampleTestLinkRF.git](https://git@gitlab.nexcel.vn:lex.khuat/SampleTestLinkRF.git)

- Create test suite in the automation of each system



- Run “cmd” to install 2 packages  
*pip install numpy*  
*pip install testlink-api-python-client*
- Add the folder “Testlink” to project

Name	Date modified	Type	Size
Templates	10/19/2018 8:56 AM	File folder	
Methods.py	10/19/2018 8:56 AM	Python File	2 KB
Misc.py	10/19/2018 8:56 AM	Python File	3 KB
ReportPrinter.py	10/19/2018 8:56 AM	Python File	11 KB
TestLinkAPI.py	10/19/2018 8:56 AM	Python File	8 KB
TestLinkListener.py	10/19/2018 8:56 AM	Python File	5 KB
TestReport.py	10/19/2018 8:56 AM	Python File	2 KB
TL_Config.txt	10/19/2018 8:56 AM	Text Document	1 KB
TL_Exec_Info.txt	10/19/2018 8:56 AM	Text Document	1 KB

- Content TL\_Config.txt
  - + isJenkinRun: Run on Jenkins is True, run on local is False
  - + useSummaryAsStep: if True, will update all steps to summary field on Testlink

- + useCSVTestLinkMapper: if True, will use testlink id in csv file in suite if metadata is null
- + syncTestCaseSteps: if True, will always update steps of testcase
- + syncTestCaseResult: if True, will always update status of testcase
  - Create a file that stores the environment variables for each feature that you want to execute TL\_Exec\_Info.txt
- + DEVKEY: Personal API access key, this value is required to connect to Testlink
- + PROJECT: Project name (not include project key)
- + TESTPLAN: Test plan of feature
- + TESTBUILD: Test build of test plan (build want to run)
- + RUNOWNER: Name of member (Only use for report)
  - Create test case and complete the test case by step

**TC1**

Settings <<

Documentation

This test case verify search

Edit

Clear

Setup

Edit

Clear

Tearardown

Edit

Clear

Tags

<Add New>

Edit

Clear

Timeout

Edit

Clear

Template

Edit

Clear

1	Log Step	Login	Done	
2	# Do something			
3	Log Step	Search matchid	Check invalid match	
4	# Do something			
5	Log Step	Show tickets	Check detail in betlist	
6	# Do something			
7				
8				

- Add library 'Testlink/Method.py' to test suite of project, it can support 'Log step' to update in testlink

1	Log Step	Login	Done
2	# Do something		
3	Log Step	Search matchid	Check invalid match
4	# Do something		
5	Log Step	Show tickets	Check detail in betlist
6	# Do something		
7	Log Step		
8			
9			
10			
11			

- Key in metadata on each test suite

RIDE - Suite1

File Edit Tools Navigate Macros Help

SampleTestLinkRF

Test Suites

- TC1
- TC2
- TC3
- Suite2
  - TC4
  - TC5
- Suite3
  - TC6
  - TC7
  - TC8
  - TC9

Actions

Config

Report

TestLink

External Resources

Edit Text Edit Run

**Suite1**

Source D:\Iris\SampleTestLinkRF\Testsuites\Suite1.txt

Settings >>

Import	Name / Path	Arguments	Comment
Resource	../Actions/common.txt		
Library	../TestLink/Methods.py		

Variable Value Comment

Metadata	Value	Comment
TC1	NAP-Special Del-4395	
TC2	NAP-Special Del-4396	
TC3	NAP-Special Del-4397	

Add Import

Library

Resource

Variables

Add Scalar

Add List

Add Dict

Add Metadata

- Local: Add parameters to “Argument” area

Execution Profile: **pybot**   ☐ Autosave ☐ Pause on failure ☒ Show message log

Arguments: **--outputdir Report --listener TestLink\TestLinkListener.py**

☐ Only run tests with these tags ☒ Skip tests with these tags

```
--outputdir Report --listener TestLink\TestLinkListener.py
```

- And run the results you just made.

### III. Report

- Record performance on the test link site

Latest execution (any build)

Date : 31/08/2018 16:26:17 - Tested by : jane.dinh - Build : Demo Build 1 - Status : Passed

Latest execution (current build) - Build : Demo Build 1

Date	Build	Tested by	Status	Exec (min)	Version	Run mode
 31/08/2018 16:26:17	Demo Build 1	jane.dinh	Passed	0.00	1	   

- Create report files for reviewing by managers
  - The information of feature
  - The status report of the test case automation (link to view detail)
  - The status report of the test case manual



JOB FAILURE (Passed: 6, Failed: 2, Not run: 0)				
<b>Run owner:</b> Jane Dinh		<b>Automation Results</b>	<b>Manual Results</b>	
• <b>Project:</b> NAP Special Delete		• <a href="#">Detailed Report</a>	• <a href="#">Detailed Report</a>	
• <b>Plan:</b> Demo RobotFramework TestPlan		• <b>Pass Percentage:</b> 56.0%	• <b>Pass Percentage:</b> 100.0%	
• <b>Build:</b> Demo Build 1		• <b>Passed:</b> 5, <b>Failed:</b> 2, Skipped: 2	• <b>Passed:</b> 1, <b>Failed:</b> 0, <b>Not run:</b> 0	
TestLink ID	TestLink Name	Automation Name	Execute Time	Status
Testsuites >> Suite1				
NAP-Special Del-4395	▶ Test Case 01 (Not included in TestPlan)	TC1	0.03 m	SKIP
NAP-Special Del-4396	▶ Test Case 02	TC2	0.13 m	PASS
NAP-Special Del-4397	Test Case 03	TC3	0.02 m	PASS
Testsuites >> Suite2				
NAP-Special Del-4398	Test Case 04	TC4	0.25 m	PASS
NAP-Special Del-4399	Test Case 05	TC5	0.17 m	FAIL
Testsuites >> Suite3				
NAP-Special Del-4400	Test Case 06	TC6	0.15 m	PASS
NAP-Special Del-4401	Test Case 07	TC7	0.35 m	FAIL
NAP-Special Del-4402	Test Case 08	TC8	0.23 m	PASS
None	Unidetified ID from TestLink	TC9	0.1 m	SKIP
Manual TestCases:				
TestLink ID	TestLink Name	Status		
NAP-Special Del-4403	▶ Test Case 09	PASS		

**Note:** If already exist test case automation, but missing mapping in metadata, report keep track is ‘**SKIP**’ and info of testlink is ‘Unidetified ID from Testlink’

If already exist test case automation, and have Testlink ID mapping in metadata, but TestlinkID not include in this TestPlan, report keep track is ‘**SKIP**’ and info of testlink is ‘Not included in TestPlan’

## IV. Create job on jenkins

Refer to config in job Test CI Method

(<http://ci.nexdev.net/view/Iris/job/Test%20CI%20method/>)

- Jenkins: Config in file .bat (test.bat in SampleTestLinkRF)

```

echo:
:: CLEAN UP PREVIOUS REPORTS
echo:
echo #### Clean up junk files
del /f /q /s "%~dp0\Report\*.*)"
del /f /q /s "%~dp0\TempData\*.*)"
if exist "%~dp0\geckodriver.log" del /f /q /s "%~dp0\geckodriver.log"

:: FIRST RUN TEST
echo:
echo #### First run tests
cmd /c pybot --listener "%~dp0\TestLink\TestLinkListener.py" -d "%~dp0\Report" -o output.xml "%~dp0\Testsuites"

:: LOGICAL CHECK FOR RERUN FAIL TEST
if errorlevel 1 goto DGTFO
echo:
echo #### All tests passed
exit /b
:DGTFO

:: SECOND RUN FOR FAIL TEST
echo:
echo #### Rerun failed tests
cmd /c pybot --listener "%~dp0\TestLink\TestLinkListener.py" -d "%~dp0\Report" -R "%~dp0\Report\output.xml" -o rerun.xml "%~dp0\Testsuites"

:: MERGE TEST RESULTS
echo:
echo #### Merge tests
cmd /c rebot -d "%~dp0\Report" -o output.xml --merge "%~dp0\Report\output.xml" "%~dp0\Report\rerun.xml"

echo:
echo #### Finished batch execution

```

Call libraries

```

cmd /c pybot --listener "%~dp0\TestLink\TestLinkListener.py" -d
"%~dp0\Report" -o output.xml "%~dp0\Testsuites"

```

- Get build url of this job

General

Source Code Management

Build Triggers

Build Environment

**Build**

Post-build Actions

Create/Update Text File

File Path:

\auto-jane-test\Build.txt

Create at Workspace:

☒

Text File Content:

`\${BUILD\_URL}`

File options:

☐ Insert at start of file

☐ Append to end of file

☒ Overwrite file

?

Conditional steps (multiple)

Run?

Always

Steps to run if condition is met

Execute Windows batch command

Command

"%WORKSPACE%\auto-jane-test\Test.bat"

See [the list of available environment variables](#)

ADD STEP TO CONDITION ▾

- Config send mail report by html file

General Source Code Management Build Triggers Build Environment Build **Post-build Actions**

**Editable Email Notification**

Disable Extended Email Publisher ☐

Allows the user to disable the publisher, while maintaining the settings

Project Recipient List  
jane.dinh@starixsoft.com, lex.khuat@starixsoft.com

Comma-separated list of email address that should receive notifications for this project.

Project Reply-To List  
\$DEFAULT\_REPLYTO

Comma-separated list of email address that should be in the Reply-To header for this project.

Content Type  
HTML (text/html)

Default Subject  
Report - \$DEFAULT\_SUBJECT

Default Content  
\${FILE.path=auto-jane-test/Report/testlink\_report.html}

Attachments

Can use wildcards like 'module/dist/\*\*/\*.\*.zip'. See the [@includes of Ant fileset](#) for the exact format. The base directory is

Attach Build Log  
Do Not Attach Build Log

Content Token Reference

- Publish html report in main page of job

General **Source Code Management** Build Triggers Build Environment Build **Post-build Actions**

**Publish HTML reports**

Reports

HTML directory to archive  
auto-jane-test\Report


Index page[s]  
testlink\_report.html


Index page title[s] (Optional)


Report title  
Testlink Report


ADD


Jenkins > Iris > Test CI method >


 **Status**


 Changes


 Workspace


 Build Now


 Delete Project


 Configure


 Email Template Testing


 Favorite


 **Testlink Report**

 Robot Results

 Move

 Job Config History


 Open Blue Ocean


 **Build History** trend


#443 Sep 5, 2018 4:54 PM

## Project Test CI method

Jane demo RB - TL - JK integration


 **Testlink Report**


 Workspace

 **Last Successful Artifacts**

junit.xml 2.30 KB [view](#)

output.xml 4.04 MB [view](#)

 **Recent Changes**

 **Disk Usage**

Job	19 MB
All builds	19 MB
Locked builds	-
All workspaces	35 MB
Slave workspaces	35 MB
Non-slave workspaces	-