



삼성 오픈소스 컨퍼런스
SAMSUNG OPEN SOURCE CONFERENCE

OPEN YOUR UNIVERSE WITH SOSCON

Robot Framework

문성훈

2015. 10. 28.

ROBOT FRAMEWORK



Generic test automation framework for acceptance testing and ATDD

[Introduction](#) [Examples](#) [Test libraries](#) [Tools](#) [Documentation](#) [Support & Contact](#)

Robot Framework is a generic test automation framework for acceptance testing and acceptance test-driven development (ATDD). It has easy-to-use tabular test data syntax and it utilizes the keyword-driven testing approach. Its testing capabilities can be extended by test libraries implemented either with Python or Java, and users can create new higher-level keywords from existing ones using the same syntax that is used for creating test cases.

Robot Framework project is hosted on [GitHub](#) where you can find further documentation, source code, and issue tracker. Downloads are hosted at [PyPI](#). The framework has a rich ecosystem around it consisting of various generic [test libraries](#) and [tools](#) that are developed as separate projects.

Robot Framework is operating system and application independent. The core framework is implemented using [Python](#) and runs also on [Jython](#) (JVM) and [IronPython](#) (.NET).

[GET LIBRARIES](#)

[READ DOCS](#)

[USE TOOLS](#)

Robot Framework itself is open source software released under [Apache License 2.0](#), and most of the libraries and tools in the ecosystem are also open source. The development of the core framework is supported by [Nokia Networks](#).

Tweets

Follow



PyPI Recent Updates

@pypi_updates

robotframework-appiumlibrary 1.3.5: app testing library for Robot Framework [bit.ly/1M7jGNa](#)

Retweeted by Robot Framework

Expand



DevOpsFinland

@DevOpsFinland

Next #DevOpsFinland meetup coming up at 2nd of Nov! @ApiOps #testing #robotframework and testing with real robots! [meetup.com/devops-finland...](#)

Retweeted by Robot Framework

Show Summary



BigBinary

@BigBinary

We wrote a blog on "Configuring ..."

Tweet to @robotframework

SOSCON

robotframework / robotframework

Unwatch

159

★ Unstar

606

Fork

224

Contributors

Commits

Code frequency

Punch card

Network

Members

Apr 28, 2008 – Oct 24, 2015

Contributions to master, excluding merge commits

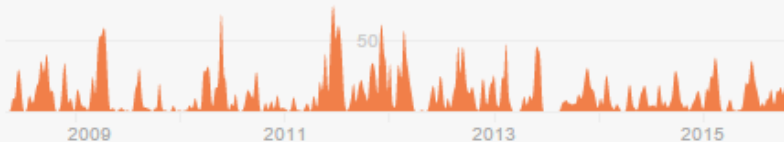
Contributions: Commits



pekkaklarck

#1

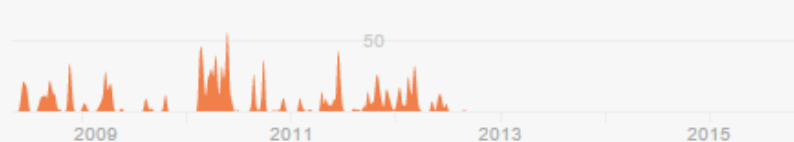
4,969 commits / 1,316,882 ++ / 839,708 --



yanne

#2

1,590 commits / 165,745 ++ / 345,162 --



jussimalinen

#3

839 commits / 42,776 ++ / 69,726 --



mkorpela

#4

615 commits / 28,143 ++ / 20,325 --

Robot Framework

Flow

1. Robot Framework

2. Example

3. Why Robot?

4. RIDE

5. CI

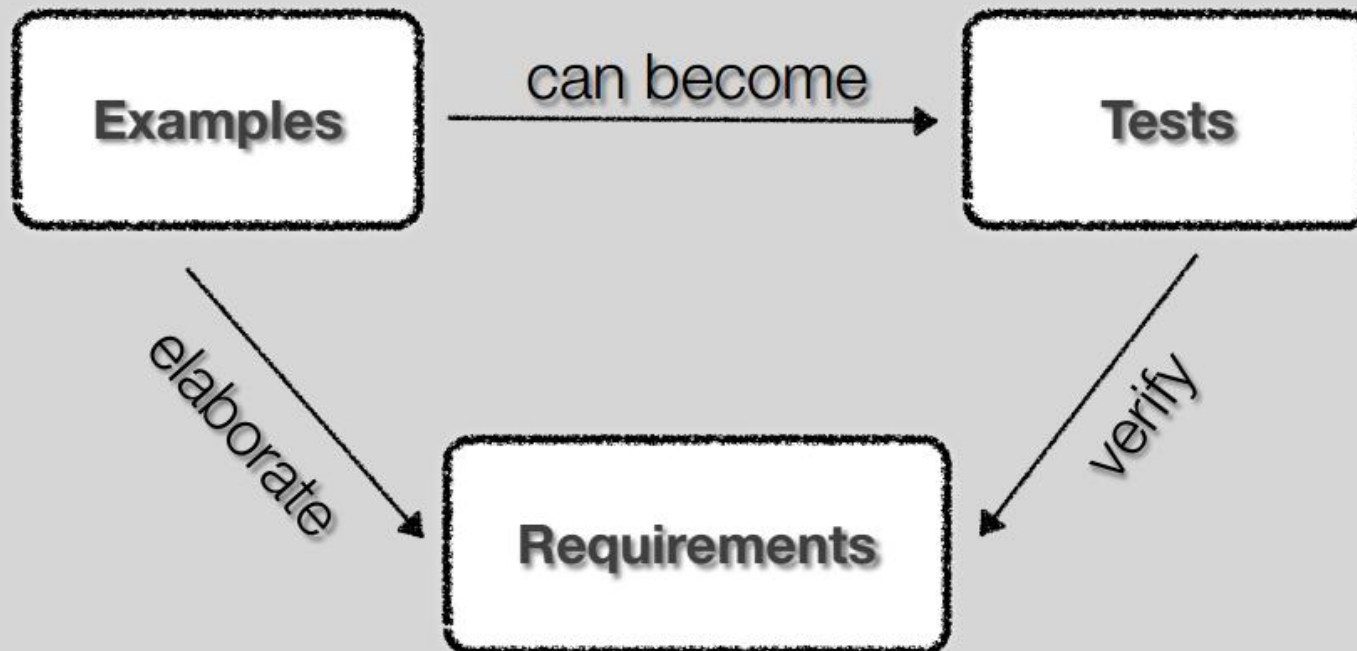
6. Future

Generic test automation framework for acceptance testing and **ATDD**

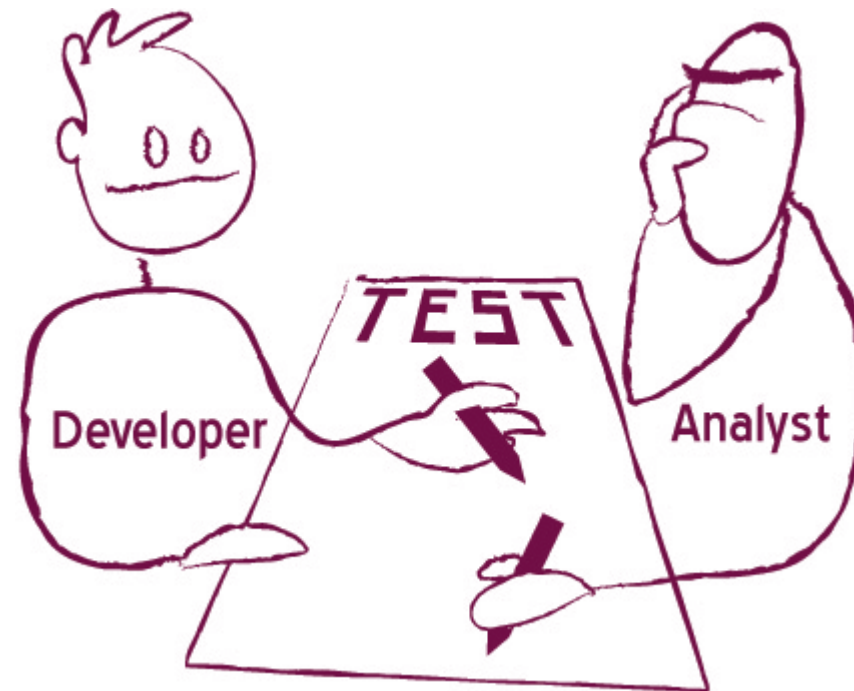




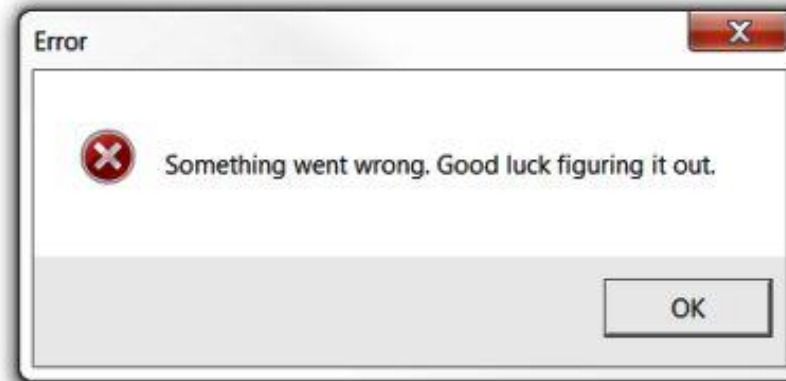
Examples, Tests, and Spec



Living documentation



Executable requirements/examples

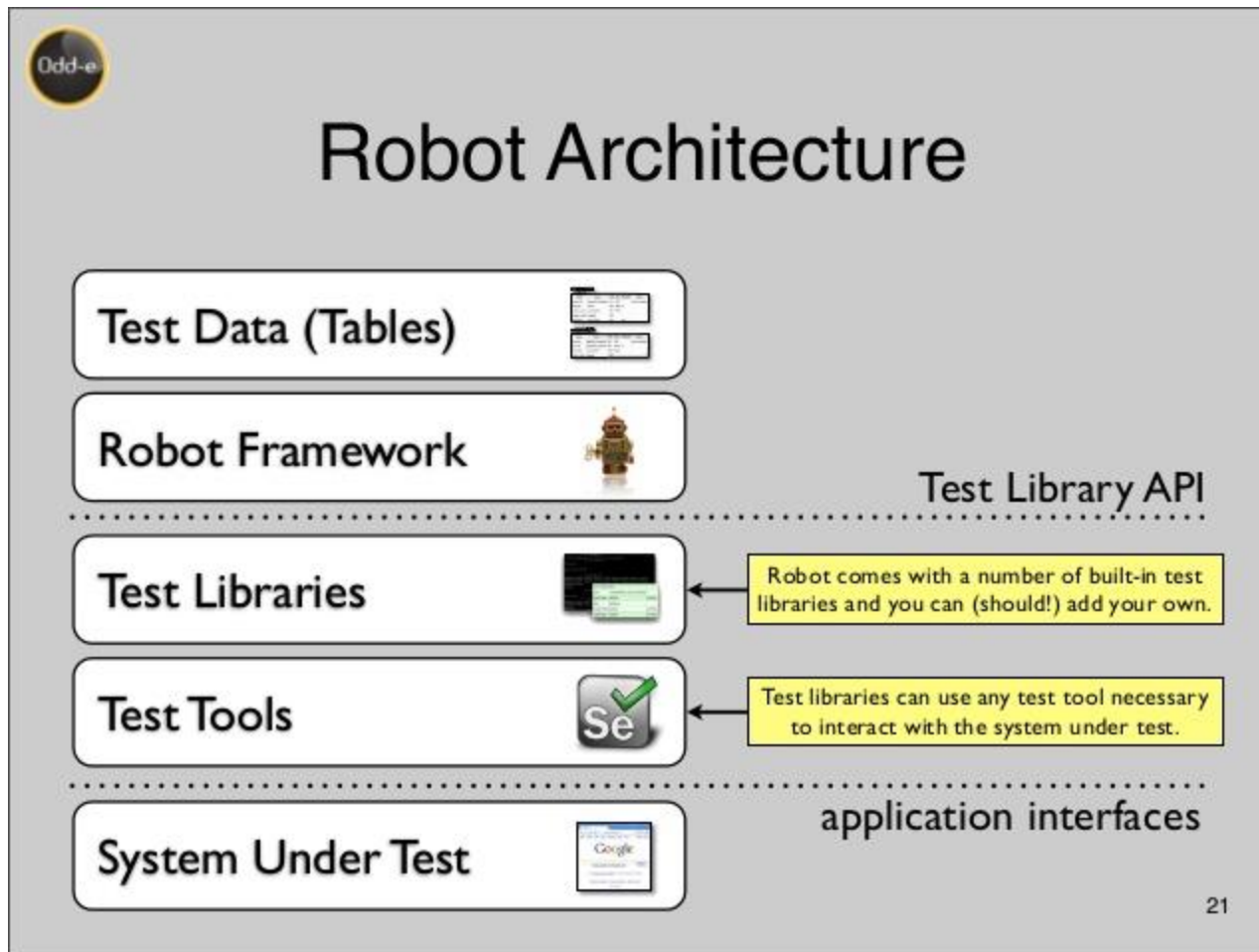


1-4. All test cases are in the tables



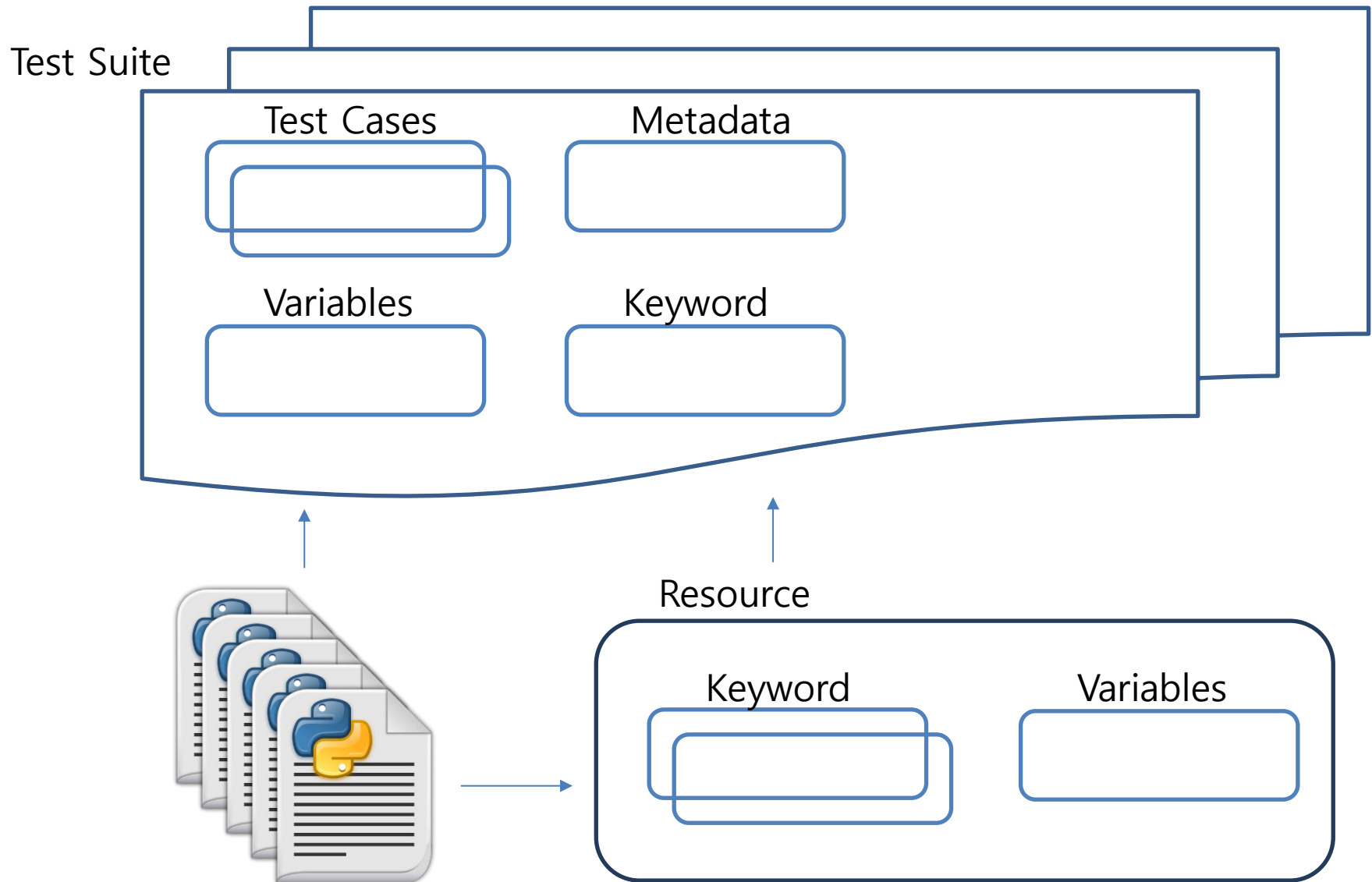
It's all in the tables

Test Case	Step
Valid Login	Given login page is open
	When valid username and password are inserted
	and credentials are submitted
	Then welcome page should be open
	Then welcome page should be open
	and credentials are submitted



Sunday, 8 September, 13

1-6. Test Data Structure



*** Keywords ***

Open Browser To Login Page

Open Browser \${LOGIN URL} \${BROWSER}

Maximize Browser Window

Set Selenium Speed \${DELAY}

Login Page Should Be Open





Variables

Variable	Value
<code>\${USERNAME}</code>	janedoe
<code>\${PASSWORD}</code>	J4n3D0e
<code>\${NEW PASSWORD}</code>	e0D3n4J
<code>\${DATABASE FILE}</code>	<code>\${TEMPDIR}\${/}robotframework-quickstart-db.txt</code>
<code>\${PWD INVALID LENGTH}</code>	Password must be 7-12 characters long
<code>\${PWD INVALID CONTENT}</code>	Password must be a combination of lowercase and uppercase letters and numbers

*** Settings ***

Documentation 로그인이 잘 되는지 확인하는 테스트.

...

... 이 테스트에 사용된 Keyword는
... resource.txt에 명시되어 있어요.

Resource resource.txt

*** Test Cases ***

로그인

웹 브라우저를 열어 로그인 페이지로 이동

사용자이름 입력 demo

암호 입력 mode

확인버튼 클릭

로그인 후 페이지가 정상적으로 열려야 함

[Teardown] 브라우저를 닫는다

*** Settings ***

Documentation A test suite with a single test for valid login.

...

... This test has a workflow that is created using keywords in
... the imported resource file.

Resource resource.txt

*** Test Cases ***

Valid Login

Open Browser To Login Page

Input Username demo

Input Password mode

Submit Credentials

Welcome Page Should Be Open

[Teardown] Close Browser

*** Settings ***

Library Selenium2Library

*** Variables ***

\${SERVER} localhost:7272
\${BROWSER} Firefox
\${DELAY} 0
\${LOGIN URL} http://\${SERVER}/

*** Keywords ***

Open Browser To Login Page

Open Browser \${LOGIN URL} \${BROWSER}

Maximize Browser Window

Set Selenium Speed \${DELAY}

Login Page Should Be Open

...

2-3. Result Report & Log

LOG

Generated

20151020 02:07:12 GMT +10:00

1 minute 36 seconds ago

Soscon Test Report

Summary Information

Status:

Start Time:

End Time:

Elapsed Time:

Log File:

Soscon Test Log

Generated

20151020 02:07:12 GMT +10:00

19 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:00	
All Tests	1	1	0	00:00:00	

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
Soscon	1	1	0	00:00:00	
Soscon. Soscon	1	1	0	00:00:00	

Test Execution Log

☒ SUITE Soscon

Full Name: Soscon
Source: C:\Users\moon\soscon
Start / End / Elapsed: 20151020 02:07:11.915 / 20151020 02:07:12.107 / 00:00:00.192
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

☒ SUITE Soscon

Full Name: Soscon.Soscon

Documentation:

Documentation: C:\Users\moon\soscon\docs

Test Statistics

Critical Tests

All Tests

No Tags

Soscon

Soscon. Soscon

Test Details

Totals

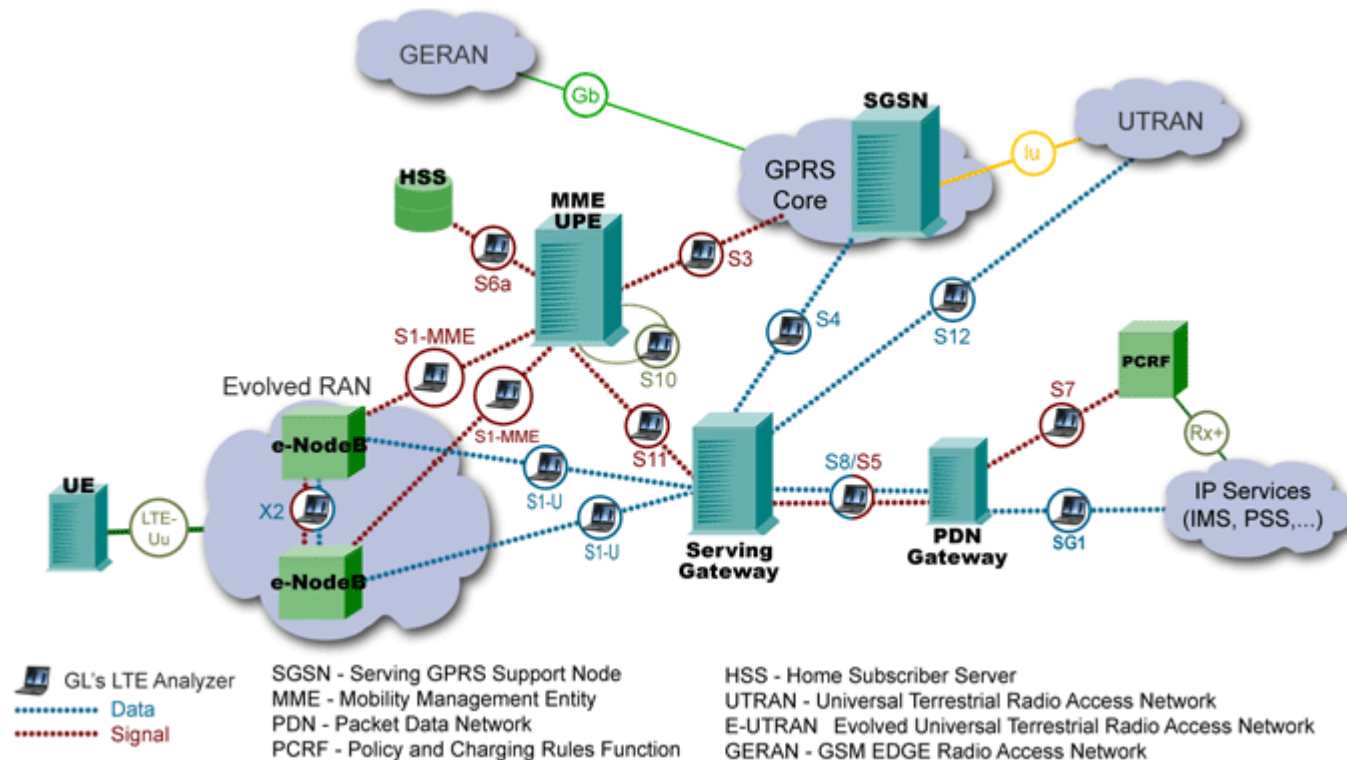
Tags

Type:

3. Why Robot?



3-1. LTE Overview



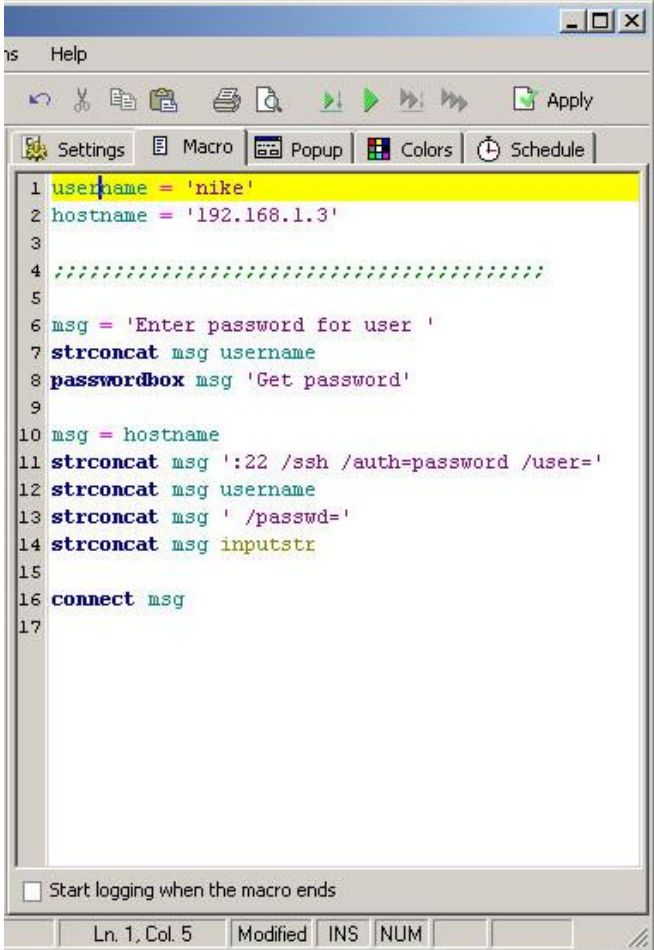
3-2. Many LTE Vendor



3-3.Before using Robot Framework

Terminal Script 사용

- Terminal Client에서 제공하는 Macro 방식
- 간단한 반복 작업을 빠른 시간 안에 작성 가능
- Client별 Script문법이 달라 학습 시간이 필요
- Terminal 이외의 기능을 수행하기 어려움
- 다양한 환경에서 재사용성이 떨어짐
- 복잡한 기능을 작성하기엔 무리가 있음
- 수행 후 log수집과 report작성이 어려움



```
1 username = 'nike'
2 hostname = '192.168.1.3'
3
4 //////////////////////////////////
5
6 msg = 'Enter password for user '
7 strconcat msg username
8 passwordbox msg 'Get password'
9
10 msg = hostname
11 strconcat msg ':22 /ssh /auth=password /user='
12 strconcat msg username
13 strconcat msg '/passwd='
14 strconcat msg inputstr
15
16 connect msg
17
```

Start logging when the macro ends

Ln. 1, Col. 5 Modified INS NUM

3-4. Before using Robot Framework

oo사의 Automation Tool 사용

- 상용 버전의 자동화 도구
- 다양한 기능을 제공하지만 그만큼 고성능을 요구함.
- 라이선스 제약이 있어 많은 사람들이 사용하기엔 부적합.
- 처음 사용하는 사람은 도구 사용법을 먼저 익혀야 함.
- 고유의 문법을 가지고 있어서 Syntax를 학습해야 Script 해독/작성이 가능.
- 확장성이 떨어져 원하는 기능을 추가로 구현하기 힘든 상황.

3-5. RIDE - Suite Settings

The screenshot shows the RIDE - Soscon application window. The left sidebar displays a project tree for 'Soscon' with the following items:

- Soscon (selected)
- Valid Login
- resource.txt
 - ③ \${SERVER}
 - ③ \${BROWSER}
 - ③ \${DELAY}
 - ③ \${VALID USER}
 - ③ \${VALID PASSWORD}
 - ③ \${LOGIN URL}
 - ③ \${WELCOME URL}
 - ③ \${ERROR URL}
- Open Browser To Login Page
- Login Page Should Be Open
- Go To Login Page
- Input Username
- Input Password
- Submit Credentials
- Welcome Page Should Be Open
- External Resources

The main panel shows the 'Edit' tab for the 'Soscon' suite. The 'Source' is set to 'C:\Users\Wmoon\Wsoscon\Wsoscon.robot'. Below this is a 'Settings >>' button. The 'Import' table is as follows:

Import	Name / Path	Arguments	Comment
Resource	resource.txt		

Below the 'Import' table is a horizontal scrollbar. The 'Variable' table is empty. The 'Metadata' table is also empty. On the right side, there are several buttons: 'Add Import', 'Library', 'Resource', 'Variables', 'Import Failed Help', 'Add Scalar', 'Add List', 'Add Dict', and 'Add Metadata'.

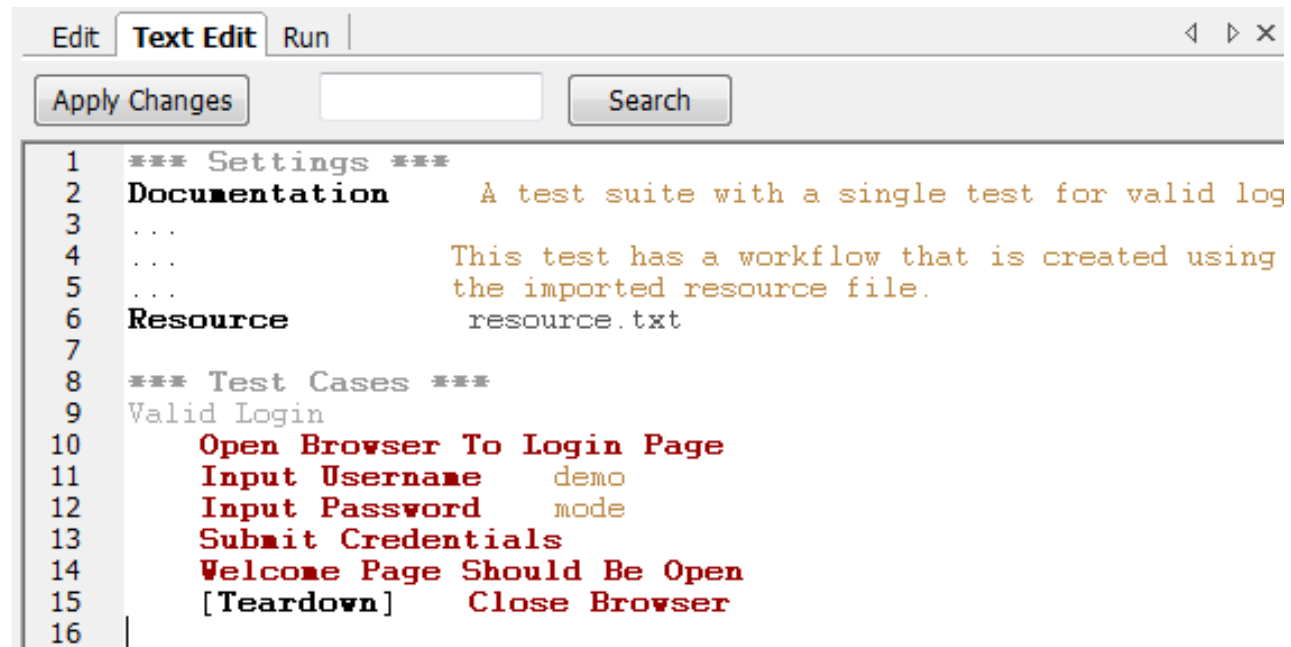
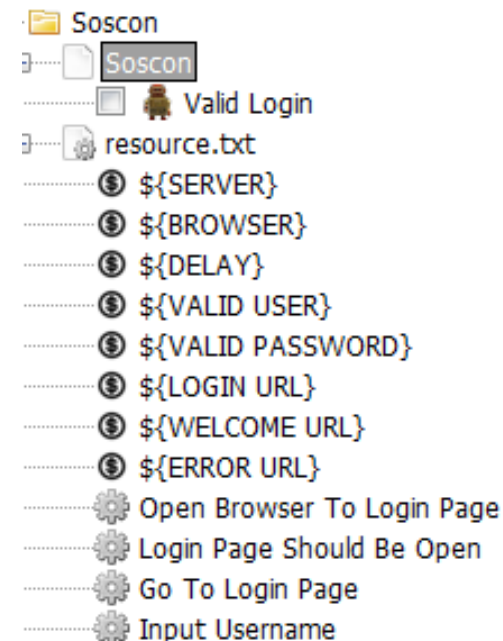
3-6. RIDE - resource.txt Settings

The screenshot shows the RIDE - Resource window. On the left is a tree view of the project structure, with 'resource.txt' selected under the 'Soscon' folder. The main area is titled 'Resource' and shows the source file path: 'C:\Users\moon\soscon\resource.txt'. Below this is a 'Settings >>' button. The main table lists imports, with 'Selenium2Library' selected. To the right of the table are buttons for 'Add Import', 'Library', 'Resource', 'Variables', 'Import Failed Help', 'Add Scalar', 'Add List', and 'Add Dict'. At the bottom, a table lists variables and their values.

Import	Name / Path	Arguments	Comment
Library	Selenium2Library		

Variable	Value	Comment
\${SERVER}	localhost:7272	
\${BROWSER}	Firefox	
\${DELAY}	0	
\${VALID USER}	demo	
\${VALID PASSWORD}	mode	
\${LOGIN URL}	http://\${SERVER}/	
\${WELCOME URL}	http://\${SERVER}/welcome.html	
\${ERROR URL}	http://\${SERVER}/error.html	

3-7. RIDE - Text Edit



3-8. No Silver Bullet



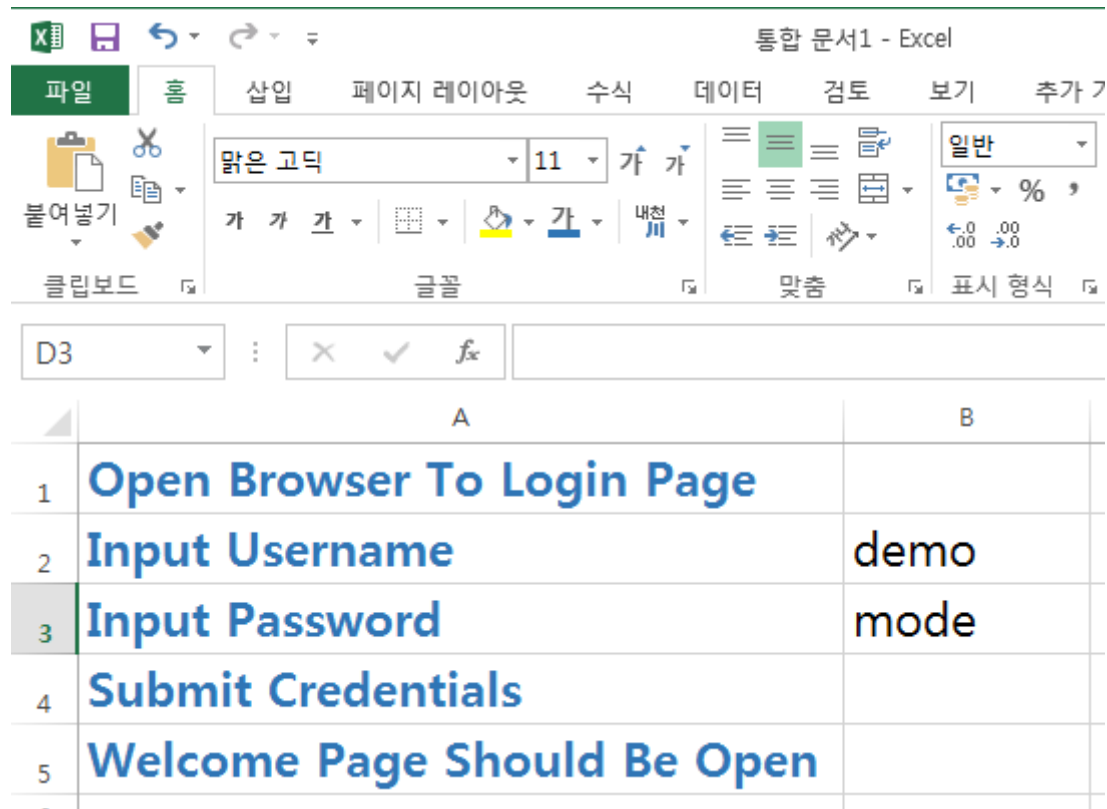
<http://www.silverbulletinc.com/whysilverbullet.htm>

3-9. RIDE - Test Case

The screenshot shows the RIDE - Soscon application interface. On the left is a project tree for 'Soscon' containing a 'Valid Login' test case and a 'resource.txt' file. The 'Valid Login' test case is expanded, showing a list of steps: 'Open Browser To Login Page', 'Input Username', 'Input Password', 'Submit Credentials', and 'Welcome Page Should Be Open'. The main area displays the 'Valid Login' test case details, including a 'Settings' button, 'Documentation', 'Setup', 'Teardown', 'Tags', 'Timeout', and 'Template' sections. The 'Setup' section contains a text box with 'demo' and a 'mode' button. The 'Teardown' section contains a text box with 'Close Browser'. The 'Tags' section contains a button '<Add New>'. The 'Timeout' and 'Template' sections each have a text box and 'Edit' and 'Clear' buttons. At the bottom, a table lists the test case steps:

Step	Test Case Step	Value
1	Open Browser To Login Page	
2	Input Username	demo
3	Input Password	mode
4	Submit Credentials	
5	Welcome Page Should Be Open	
6		

3-10. RIDE - Excel like



The screenshot shows the RIDE application interface, which mimics the Microsoft Excel environment. The title bar reads '통합 문서1 - Excel'. The ribbon includes tabs for '파일' (File), '홈' (Home), '삽입' (Insert), '페이지 레이아웃' (Page Layout), '수식' (Formulas), '데이터' (Data), '검토' (Review), '보기' (View), and '추가' (Add). The '홈' tab is active, showing options for font (맑은 고딕, size 11), paragraph (alignment, bullet points), styles (background color, borders), and numbers (percentage, decimal places). The formula bar shows 'D3'. The main grid has columns A and B, and rows 1 through 5. Row 3 is selected.

	A	B
1	Open Browser To Login Page	
2	Input Username	demo
3	Input Password	mode
4	Submit Credentials	
5	Welcome Page Should Be Open	

3-11. RIDE - Excel like

The screenshot displays the RIDE (Robot Framework IDE) interface, which mimics the Excel application. The title bar reads '통합 문서1 - Excel'. The ribbon includes tabs for '파일' (File), '홈' (Home), '삽입' (Insert), '페이지 레이아웃' (Page Layout), '수식' (Formulas), '데이터' (Data), '검토' (Review), '보기' (View), and '추가 기' (Add-ons). The '홈' tab is active, showing options for font (맑은 고딕, size 18), paragraph (alignment, bullet points), styles (background color, borders), and numbers (percentage, decimal places). The formula bar shows 'B2' as the active cell with the formula '=user'. The spreadsheet grid has columns A and B, and rows 1 through 5. The content in the grid is as follows:

	A	B
1	Open Browser To Login Page	
2	Input Username	user
3	Input Password	mode
4	Submit Credentials	
5	Welcome Page Should Be Open	

3-12. RIDE - Excel like

The screenshot shows the RIDE - Soscon application window. The left sidebar displays a project tree for 'Soscon' containing a test suite 'Valid Login' and a 'resource.txt' file. The 'Valid Login' suite lists several variables and steps: `${SERVER}`, `${BROWSER}`, `${DELAY}`, `${VALID USER}`, `${VALID PASSWORD}`, `${LOGIN URL}`, `${WELCOME URL}`, `${ERROR URL}`, and a series of steps from 'Open Browser To Login Page' to 'Welcome Page Should Be Open'. The 'External Resources' folder is also visible.

The main window shows the 'Valid Login' test suite in 'Edit' mode. It features a 'Settings >>' button and a table of steps. The 'Input Username' step (step 2) is highlighted with a yellow background and contains the text 'user'.

Step	Description	Value
1	Open Browser To Login Page	
2	Input Username	user
3	Input Password	mode
4	Submit Credentials	
5	Welcome Page Should Be Open	
6		
7		
8		
9		
10		

3-13. RIDE - Excel like

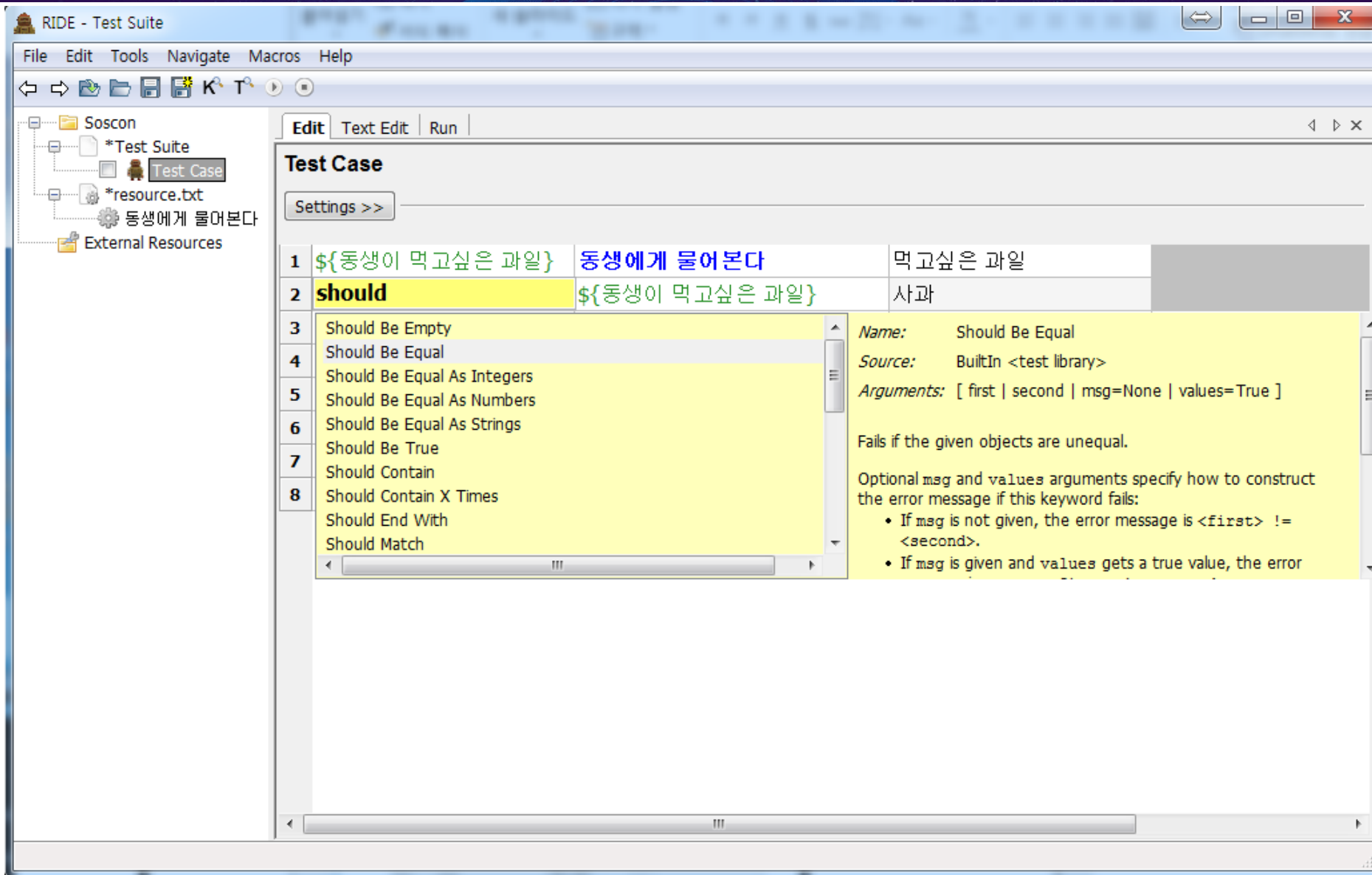


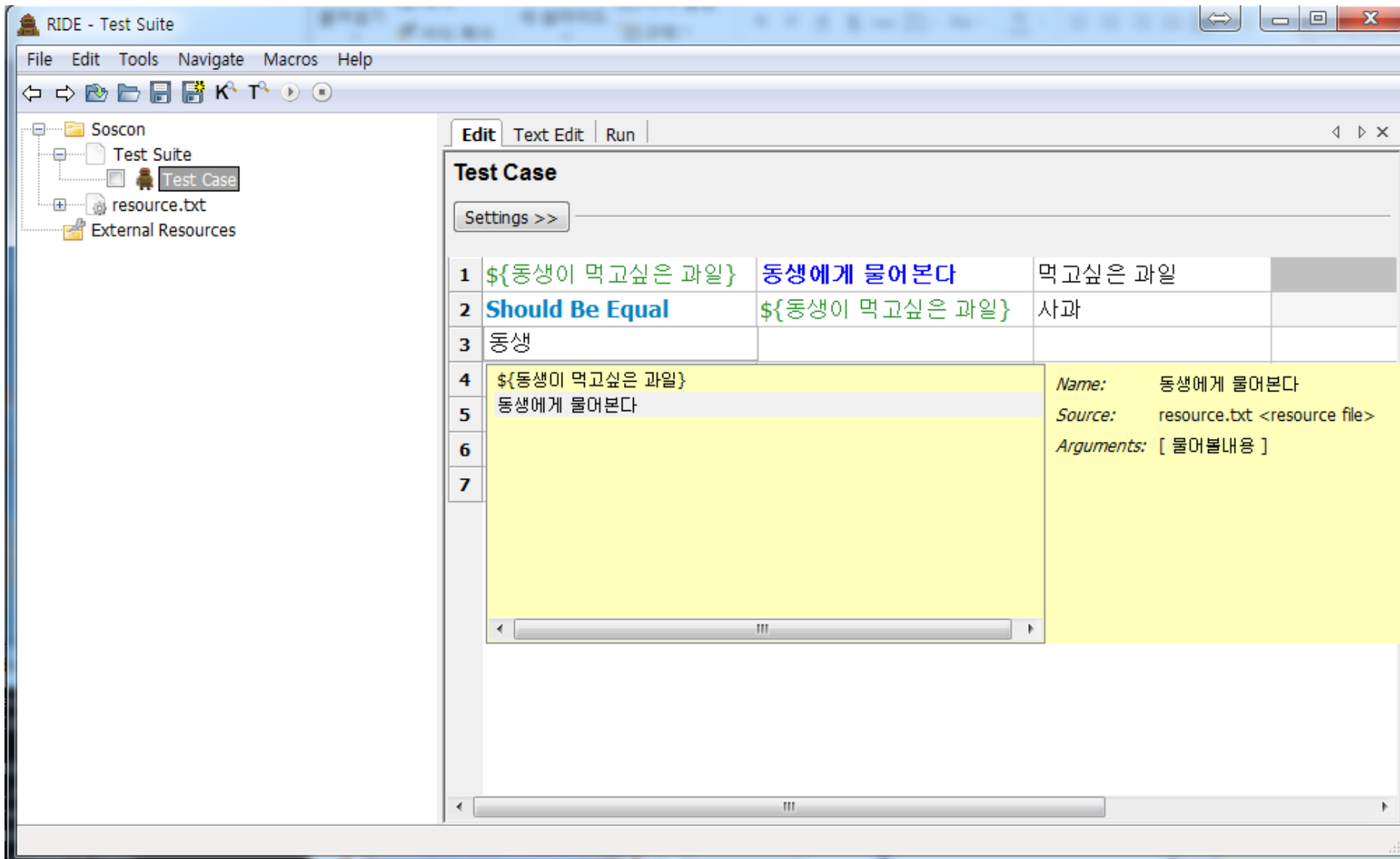
Slam dunk comics

Test Case

Settings >>

1	<code>\${동생이 먹고싶은 과일}=</code>	동생에게 물어본다	먹고싶은 과일
2	Should Be Equal	<code>\${동생이 먹고싶은 과일}</code>	사과





간단한 예제로 알아보는 RIDE 사용법 - Cont'd

RIDE - Resource

File Edit Tools Navigate Macros Help

Soscon

- Test Suite
- Test Case
- resource.txt
 - 동생에게 물어본다
 - 형에게 물어본다
- External Resources

Edit Text Edit Run

동생에게 물어본다 Find Usages

Settings <<

Documentation Edit Clear

Tags <Add New> Edit Clear

Arguments \${물어볼내용} Edit Clear

Teardown Edit Clear

Return Value \${대답} Edit Clear

Timeout Edit Clear

1	\${대답}	Set Variable If	'\${물어볼내용}' == '먹고싶은 과일'	사과
2				
3				
4				

Saved C:\Users\moon\soscon\resource.txt

Robot Framework Dos and Don'ts



© Copyright Nokia Networks
Creative Commons Attribution 3.0 License

<http://www.slideshare.net/pekkaklarck/robot-framework-dos-and-donts>

Test Case			
Settings >>			
1	\${동생이 먹고싶은 과일}=	동생에게 물어본다	먹고싶은 과일
2	Should Be Equal	\${동생이 먹고싶은 과일}	사과
3			
4			
5			
6			
7			
8			

Create Keyword

Extract Keyword

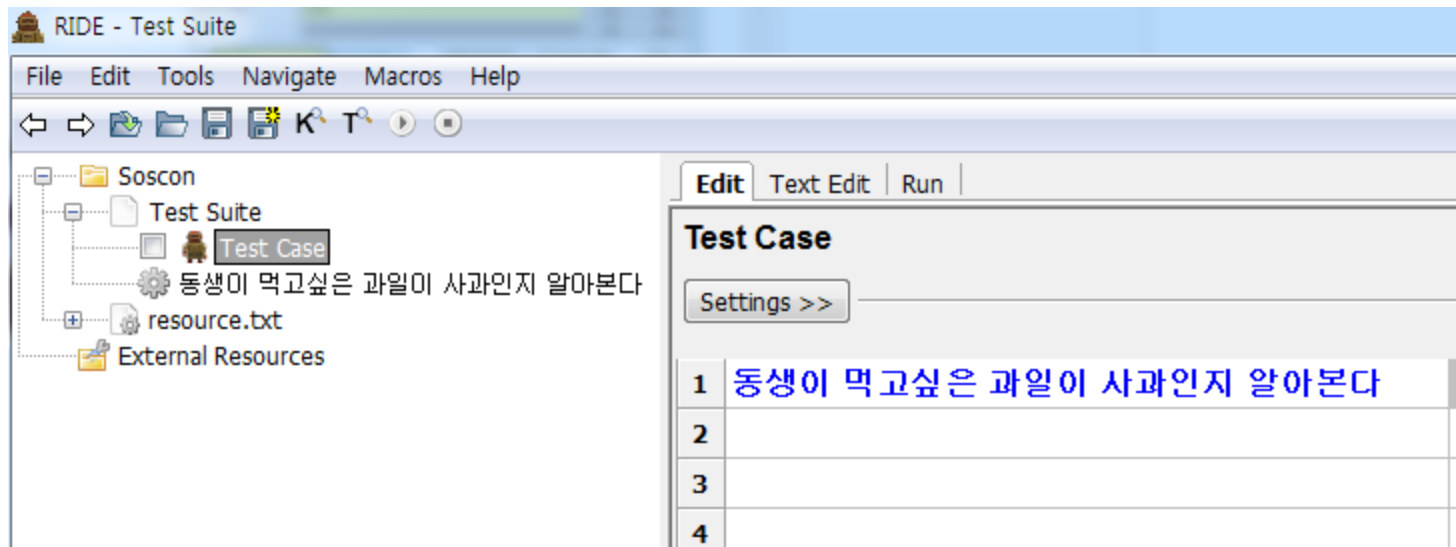
Extract Variable

Rename Keyword

Find Where Used

Make VariableCtrl-1

Make List VariableCtrl-2



간단한 예제로 알아보는 RIDE 사용법 - Cont'd

The screenshot displays the RIDE - Resource application window. The left sidebar shows a project tree with 'Soscon' containing a 'Test Suite' and a 'Test Case' (labeled '동생에게 물어본다'). The main area is divided into 'Edit', 'Text Edit', and 'Run' tabs. The 'Run' tab is active, showing the 'Execution Profile' set to 'pybot'. Below this are buttons for 'Start', 'Stop', 'Pause', 'Continue', 'Next', and 'Step over'. The 'Arguments' field contains '-L DEBUG --variable HOST:localhost'. There are checkboxes for 'Only run tests with these tags' and 'Skip tests with these tags' (which is checked and contains 'skip, delete'). The output pane shows the test results: 'elapsed time: 0:00:00 pass: 1 fail: 0', 'Soscon | PASS |', '1 critical test, 1 passed, 0 failed', and '1 test total, 1 passed, 0 failed'. It also lists the output, log, and report file paths. The bottom status bar indicates 'Saved C:\Users\moon\soscon\resource.txt'.

RIDE - Resource

File Edit Tools Navigate Macros Help

Soscon

- Test Suite
- Test Case
- resource.txt
- 동생에게 물어본다
- 형에게 물어본다
- External Resources

Edit | Text Edit | **Run**

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

Start Stop Pause Continue Next Step over

Arguments: -L DEBUG --variable HOST:localhost

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

skip, delete

elapsed time: 0:00:00 pass: 1 fail: 0

Soscon | PASS |

1 critical test, 1 passed, 0 failed

1 test total, 1 passed, 0 failed

=====

Output: c:\users\moon\AppData\Local\Temp\RIDEy2kpfc.d\output.xml

Log: c:\users\moon\AppData\Local\Temp\RIDEy2kpfc.d\log.html

Report: c:\users\moon\AppData\Local\Temp\RIDEy2kpfc.d\report.html

test finished 20151020 20:01:47

Starting test: Soscon.Test Suite.Test Case

20151020 20:01:47.733 : INFO : \${대답} = 사과

20151020 20:01:47.734 : INFO : \${동생이 먹고싶은 과일} = 사과

20151020 20:01:47.736 : DEBUG :

Argument types are:

<type 'unicode'>

<type 'unicode'>

Ending test: Soscon.Test Suite.Test Case

Saved C:\Users\moon\soscon\resource.txt

SUITE

Test Suite

Full Name: Soscon.Test Suite

Documentation: A test suite with a single test for valid login.

This test has a workflow that is created using keywords in the imported resource file.

Source: C:\Users\moon\soscon\Test Suite.robot

Start / End / Elapsed: 20151020 20:01:47.724 / 20151020 20:01:47.737 / 00:00:00.013

Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

REPORT

Log level: DEBUG

TEST

Test Case

Full Name: Soscon.Test Suite.Test Case

Start / End / Elapsed: 20151020 20:01:47.729 / 20151020 20:01:47.737 / 00:00:00.008

Status: PASS (critical)

KEYWORD

 \${동생이 먹고싶은 과일} = resource. 동생에게 물어본다 먹고싶은 과일

Start / End / Elapsed: 20151020 20:01:47.730 / 20151020 20:01:47.735 / 00:00:00.005

KEYWORD

 \${대답} = BuiltIn. Set Variable If '\${물어볼내용}' == '먹고싶은 과일', 사과

Documentation: Sets variable based on the given condition.

Start / End / Elapsed: 20151020 20:01:47.731 / 20151020 20:01:47.733 / 00:00:00.002

20:01:47.733 INFO \${대답} = 사과

20:01:47.734 INFO \${동생이 먹고싶은 과일} = 사과

KEYWORD

 BuiltIn. Should Be Equal \${동생이 먹고싶은 과일}, 사과

Documentation: Fails if the given objects are unequal.

Start / End / Elapsed: 20151020 20:01:47.735 / 20151020 20:01:47.736 / 00:00:00.001

20:01:47.736 DEBUG Argument types are:
<type 'unicode'>
<type 'unicode'>

STANDARD

EXTERNAL

OTHER

Builtin

Provides a set of often needed generic keywords. Always automatically available without imports.

Dialogs

Provides means for pausing the test execution and getting input from users.

Collections

Provides a set of keywords for handling Python lists and dictionaries.

OperatingSystem

Enables various operating system related tasks to be performed in the system where Robot Framework is running.

Remote

Special library acting as a proxy between Robot Framework and test libraries elsewhere. Actual test libraries can be running on different machines and be implemented using any programming language supporting XML-RPC protocol.

Screenshot

Provides keywords to capture screenshots of the desktop.

Process

Library for running processes in the system. New in Robot Framework 2.8.

String

Library for generating, modifying and verifying strings.

Telnet

Makes it possible to connect to Telnet servers and execute commands on the opened connections.

XML

Library for generating, modifying and verifying XML files.

DateTime

Library for date and time conversions. New in Robot Framework 2.8.5.

STANDARD

EXTERNAL

OTHER

Android library

Library for all your Android automation needs. It uses Calabash Android internally.

AnywhereLibrary

Library for testing Single-Page Apps (SPA). Uses Selenium Webdriver and Appium internally.

AppiumLibrary

Library for Android- and iOS-testing. It uses Appium internally.

Archive library

Library for handling zip- and tar-archives.

AutoItLibrary

Windows GUI testing library that uses AutoIt freeware tool as a driver.

Database Library (Java)

Eclipse Library

Library for testing Eclipse RCP applications using SWT widgets.

robotframework-faker

Library for [Faker](#), a fake test data generator.

FTP library

Library for testing and using FTP server with Robot Framework.

HTTP library (livetest)

Library for HTTP level testing using livetest tool internally.

HTTP library (Requests)

Library for HTTP level testing using Request internally.

iOS library

Library for all your iOS automation needs. It uses Calabash iOS Server

RemoteSwingLibrary

Library for testing and connecting to a java process and using SwingLibrary, especially Java Web Start applications.

SeleniumLibrary

Web testing library that uses popular Selenium tool internally. Uses deprecated Selenium 1.0 and is also itself deprecated.

Selenium2Library

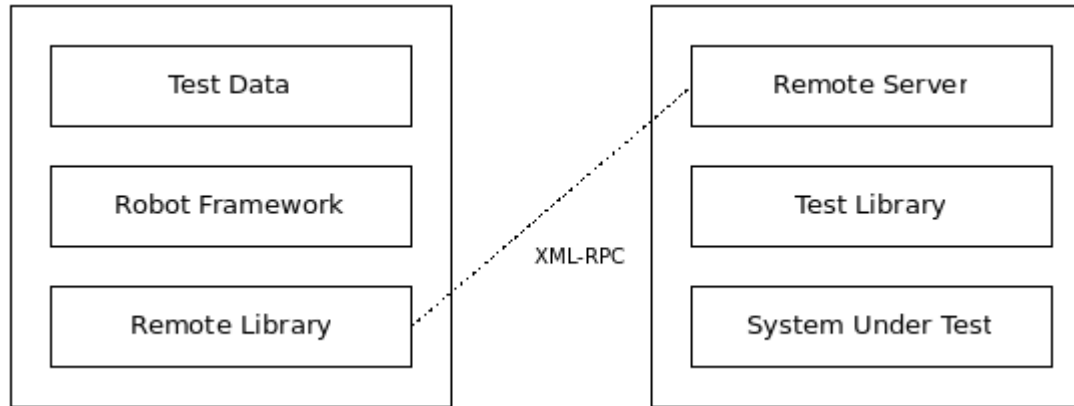
Web testing library that uses Selenium 2. For most parts drop-in-replacement for old SeleniumLibrary.

Selenium2Library for Java

Java port of the Selenium2Library.

SSHLibrary

Enables executing commands on remote machines over an SSH connection. Also supports transferring



<http://sapzapee.tistory.com/>



Jenkins

5-1. Jenkins robot framework plugin

Jenkins

Jenkins

robot

ENABLE AUTO REFRESH

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Robot Results](#)

[Email Template Testing](#)

[Workspace](#)

[Recent Changes](#)

[Latest Test Result](#) (no failures)

Latest Robot Results:

	Total	Failed	Passed	Pass %
Critical tests	10	0	10	100.0
All tests	10	0	10	100.0

[Browse results](#)
[Open smoke_all_report.html](#)
[Open smoke_all_log.html](#)

Build History [\(trend\)](#)

#6 [Aug 20, 2013 2:26:16 PM](#)

#5 [Aug 20, 2013 2:24:49 PM](#)

#3 [Aug 20, 2013 2:14:06 PM](#)

#2 [Aug 15, 2013 4:24:27 PM](#)

#1 [Aug 15, 2013 4:21:47 PM](#)

[RSS for all](#) [RSS for failures](#)

Project robot

[add description](#)

Disable Project

Robot Framework Tests Trend (all tests)

[Zoom graph to changes](#)

Build number	Passed	Failed
#1	4	4
#2	4	4
#3	4	4
#4	10	0
#5	10	0
#6	10	0

Permalinks

- [Last build \(#6\), 20 hr ago](#)
- [Last stable build \(#6\), 20 hr ago](#)
- [Last successful build \(#6\), 20 hr ago](#)

wiki.jenkins-ci.org

SOSCON

5-2. Keywords Online Document

Keyword Documentation - x

localhost:7070/doc/keywords/BuiltIn/

Apps Robot echo ios Gmail Keyword Documenta Trailers groups stackoverflow Programmers Planet Pocket Tool Other Bookmarks

Robot Framework Hub

Search all keywords

BuiltIn

[Call Method](#)
[Catenate](#)
[Comment](#)
[Continue For Loop](#)
[Continue For Loop If](#)
[Convert To Binary](#)
[Convert To Boolean](#)
[Convert To Bytes](#)
[Convert To Hex](#)
[Convert To Integer](#)
[Convert To Number](#)
[Convert To Octal](#)
[Convert To String](#)
[Create List](#)
[Evaluate](#)
[Exit For Loop](#)
[Exit For Loop If](#)
[Fail](#)
[Fatal Error](#)
[Get Count](#)
[Get Length](#)
[Get Library Instance](#)
[Get Time](#)
[Get Variable Value](#)
[Get Variables](#)

BuiltIn

Version: 2.8.4 • Scope: global

Introduction

An always available standard library with often needed keywords.

'BuiltIn' is Robot Framework's standard library that provides a set of generic keywords needed often. It is imported automatically and thus always available. The provided keywords can be used, for example, for verifications (e.g. 'Should Be Equal', 'Should Contain'), conversions (e.g. 'Convert To Integer') and for various other purposes (e.g. 'Log', 'Sleep', 'Run Keyword If', 'Set Global Variable').

Many of the keywords accept an optional error message to use if the keyword fails. Starting from Robot Framework 2.8, it is possible to use HTML in these messages by prefixing them with `**HTML*`. See 'Fail' keyword for a usage example. Notice that using HTML in messages is not limited to BuiltIn library but works with any error message.

Keywords	Arguments	Documentation										
Call Method	object, method_name, *args	Calls the named method of the given object with the provided arguments. The possible return value from the method is returned and can be assigned to a variable. Keyword fails both if the object does not have a method with the given name or if executing the method raises an exception. Examples: <table><tbody><tr><td>Call Method</td><td>\${hashtable}</td><td>put</td><td>myname</td><td>myvalue</td></tr><tr><td>\${isemptv} =</td><td>Call Method</td><td>\${hashtable}</td><td>isEmpty</td><td></td></tr></tbody></table>	Call Method	\${hashtable}	put	myname	myvalue	\${isemptv} =	Call Method	\${hashtable}	isEmpty	
Call Method	\${hashtable}	put	myname	myvalue								
\${isemptv} =	Call Method	\${hashtable}	isEmpty									

- Acceptance Test Driven Development using Robot Framework
 - Pekka Klärck <peke@eliga.fi>
 - Janne Härkönen <jth@reaktor.fi>
- <http://robotframework.org>
- <http://seleniummaster.com>
- <http://sapzapee.tistory.com>
- Thomas Jaspers - <https://blog.codecentric.de/en/2014/10/robot-framework-compact-sheet/>
- Steven Mak - <http://www.slideshare.net/tcmak/quality-comes-free-with-open-source-testing-tools>
- LTE Overview - <http://www.gl.com/lteanalyzer.html>
- Free images - <https://pixabay.com>
- Python Icon image - <http://www.iconarchive.com/show/leaf-mimes-icons-by-untergunter/text-x-python-icon.html>
- **Special Thanks To : Team AUTO, Team MOMO**