

RENAT

Library version: RENAT 1.7.1
Library scope: global
Named arguments: supported

Introduction

Document for RENAT framework

All in one pdf [renat.pdf](#)

Libraries

RENAT includes following libraries:

Common:

Common library of RENAT

VChannel:

Library controls connection to targets (servers, routers, ...)

Logger:

Library provides enhanced logging keywords

Optical:

Library provides keywords to control L1 switches, includes [mod_calient](#) mod, [mod_ntm](#) mod

Router:

Library provides keywords to control routers, includes [mod_juniper](#) mod , [mod_cisco](#) mod and [mod_gr](#) mod

Tester:

Library provides keywords to control testers, includes [mod_ixnet](#) and [mod_ixload](#)

WebApp:

Common library for web application, includes 2 child libraries: [Samurai](#) and [Arbor](#)

Others

Changes:

Changes information

Choose each libraries for detail information and samples about keywords.

Shortcuts

Keywords

Keyword	Arguments	Documentation
---------	-----------	---------------

Altogether 0 keywords.
Generated by [Libdoc](#) on 2018-04-12 19:20:29.



Common

Library version: RENAT 1.7.1
Library scope: global
Named arguments: supported

Introduction

Common library for RENAT

It loads config files and create necessary variables. The file should be the 1st library included from any test case.

Table of Contents

- [Configuration file](#)
- [Variables](#)
- [Shortcuts](#)
- [Keywords](#)

Configuration file

Global configuration

There are 2 kinds of configuration files. The global configuration files (aka master files) include device information, authentication etc that are used for all the test cases in the suite. The local configuration file `local.yaml` includes information about nodes, tester ports etc. that are used in a specific test case.

At the beginning, the module makes a local copy the master files and initialize necessary variables.

The master files folder is defined by `renat-master-folder` in `$RENAT_PATH/config/config.yaml`. Usually, users do not need to modify the master files. The most common case is when new device is deployed, the `device.yaml` need to be update so that device could be used in the test cases.

- `device.yaml`: contains global device information

```
device:
  apollo:
    type: ssh-host
    description: main server
    ip: 10.128.3.101
  artemis:
    type: ssh-host
    description: second server
    ip: 10.128.3.91
  vmx11:
    type: juniper
    description: r1
    ip: 10.128.64.11
  vmx12:
    type: juniper
    description: r2
    ip: 10.128.64.12
```

- `auth.yaml`: contains authentication information

```
auth:
  plain-text:
    default:
      user: user
      pass: nttXXX
  flets:
    user: user
    pass: lpcoXXXX
  arbor:
    user: admin
    pass: nttXXX

  public-key: # for Public Key authentication
  default:
    user: robot
    key: /home/user/.ssh/robot_id_rsa
  test:
    user: jenkins
    key: /var/lib/jenkins/.ssh/id_rsa
```

- `template.yaml`: contains devvice template information

```
access-template:
  ssh-host:
    access: ssh
    auth: public-key
    profile: default
    prompt: \$
```

```

  append:
  init: unalias -a
juniper:
  access: telnet
  auth: plain-text
  profile: default
  prompt: "(#|>)"
  append: ' | no-more'
  init:
cisco:
  access: ssh
  auth: plain-text
  profile: default
  prompt: "~@.*(#|>)"
  append:
  init:
snmp-template:
  juniper:
    mib: ./mib-Juniper.json
    community: public
    poller: renat
  cisco:
    mib: ./mib-Cisco.json
    community: public
    poller: renat

```

Local Configuration

Local configuration (aka `local.yaml`) was used by a test case of its sub test cases. Test cases could includes several test cases (the sub level is not limited). The local configuration is defined by `local.yaml` in the `config` folder of each test case. If a test case does not has the `local.yaml` in its `config` folder, it will use the `local.yaml` file in its parent test case and so on. This will help users to share the test information for related test case without having the same `local.yaml` for each test case (**Note:** this feature is enabled from RENAT 0.1.4). The `local.yaml` that is really used for the test is called `active local.yaml`.

When user used the wizard `case.sh` to create a new test case, they have the ability to crete new `local.yaml` or not. `local.yaml` could be edited and inserted new information later to hold more informations for the test case.

When a test is run, it will display its current active `local.yaml`

- `<testcase>/config/local.yaml`: contains local data for a test case

```

node:
  vmx11:
    device: vmx1 1
    snmp_polling: yes
  vmx12:
    device: vmx1 1
    snmp_polling: yes
  apollo:
    device: vmx11
    snmp_polling: yes

tester:
  tester01:
    type: ixnet
    ip: 10.128.32.70
    config: vmx_20161129.ixncfg

port-mapping:
  uplink01:
    device: vmx1 1
    port: ge-0/0/0
  downlink01:
    device: vmx12
    port: ge-0/0/2

default:
  ignore_dead_node: yes
  terminal:
    width: 80
    height: 32
  result_folder: result

```

Variables

The module automatically create `GLOBAL` & `LOCAL` variable for other libraries. It also creates global list variables `GLOBAL`, `LOCAL` and `NODE` that could be accessed from Robot Framework` test cases.

The `GLOBAL` variable holds all information defined by the master files and `LOCAL` variable holds all variables defined by active `local.yaml`. And `NODE` is a list that hold all active nodes defined in the `local.yaml`.

Users could access to the information of a key in `local.yaml` by ``${LOCAL['key']}``, information of a node by ``${LOCAL['node']['vmx11']}`` or simply ``${NODE['vmx']}``. When a keyword need a list of current node, `@{NODE}` could be used.

Notes: By default, RENAT will stop and raise an exception if connection to a node is failed. But if `ignore_dead_node` is defined as `yes` (default) is the current active `local.yaml`, RENAT will omit an warning but keep running the test and remove the node from its active node list.

Shortcuts

Change Mod · Cleanup Result · Convert Html To Pdf · Count Keyword · Count Keyword Line · Count Match Regexp · Create Sequence · Csv Concat · Csv Merge · Csv Select · Diff File · Err · Error Line Should Not Be Bigger Than · Error Should Not Be Bigger Than · File Md5 · Fold Str · Follow Syslog And Trap · Get Config Path · Get File Without Error · Get Item Config Path · Get Item Name · Get Renat Path · Get Result Folder · Get Result Path · Get Test Device · Is Stable · Keyword Line Should Not Be Bigger Than · Keyword Should Not Be Bigger Than · Load Plugin · Log · Log To Console · Loop For Node Tag · Md 5 · Merge Files · Mib For Node · Node With Attr · Node With Tag · Node Without Tag · Pause · Ping Until Ok · Random Name · Random Number · Renat Version · Set Multi Item Variable · Set Result Folder · Slack · Str 2 Seq · Version

Keywords

Keyword	Arguments	Documentation										
Change Mod	<i>name, mod, relative=False</i>	<p>Changes file mod, likes Unix chmod</p> <p><code>mod</code> is a string specifying the privilege mode <code>relative</code> is <code>False</code> or <code>True</code></p> <p>Examples:</p> <table><tr><td>Common.</td><td>Change Mod</td><td>tmp</td><td>0775</td></tr></table>	Common.	Change Mod	tmp	0775						
Common.	Change Mod	tmp	0775									
Cleanup Result	<i>ignore=^(log.html output.xml report.html)\$</i>	<p>Cleans up the result folder</p> <p>Deletes all files in current active folder that does not match the <code>ignore</code> expression and are older than the time the test has started.</p> <p>Note: The keyword only removes files but not folders</p>										
Convert Html To Pdf	<i>html_file, pdf_file</i>	Converts html file to pdf file										
Count Keyword	<i>keyword, *pattern_list</i>	Count the keyword in files. Keyword is not case-sensitive										
Count Keyword Line	<i>keyword, *pattern_list</i>	<p>Count the number of lines contains the keyword</p> <p>Notes: Keyword is matched partially. For example, <code>error</code> or <code>errorXXX</code> will be matched by <code>error</code> keyword.</p>										
Count Match Regexp	<i>regex, *pattern_list</i>	<p>Count the number of <code>regex</code> found in <code>pattern_list</code></p> <p>Examples:</p> <table><tr><td>\$(err_num)=</td><td>Count Match RegExp</td><td>.*error.*</td><td>result/*.csv</td><td>result/*.txt</td></tr></table>	\$(err_num)=	Count Match RegExp	.*error.*	result/*.csv	result/*.txt					
\$(err_num)=	Count Match RegExp	.*error.*	result/*.csv	result/*.txt								
Create Sequence	<i>start, end, interval, option=float</i>	<p>Creates a list with number from <code>start</code> to <code>end</code> with <code>interval</code></p> <p>Example:</p> <table><tr><td>@{list}=</td><td>Create Sequence</td><td>10</td><td>15</td><td>0.5</td></tr></table> <p>will create a list of [11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5]</p>	@{list}=	Create Sequence	10	15	0.5					
@{list}=	Create Sequence	10	15	0.5								
Csv Concat	<i>src_pattern, dst_name, has_header=None</i>	<p>Concatinates CSV files vertically If the CSV files has header, set <code>has_header</code> to <code>\$(TRUE)</code></p> <p>Examples:</p> <table><tr><td>Common.</td><td>CSV Merge</td><td>config/data0[3,4].csv</td><td>result/result2.csv</td><td></td></tr><tr><td>Common.</td><td>CSV Merge</td><td>config/data0[3,4].csv</td><td>result/result2.csv</td><td>has_header=\$(TRUE)</td></tr></table>	Common.	CSV Merge	config/data0[3,4].csv	result/result2.csv		Common.	CSV Merge	config/data0[3,4].csv	result/result2.csv	has_header=\$(TRUE)
Common.	CSV Merge	config/data0[3,4].csv	result/result2.csv									
Common.	CSV Merge	config/data0[3,4].csv	result/result2.csv	has_header=\$(TRUE)								
Csv Merge	<i>src_pattern, dst_name, on_key=0, has_header=None</i>	<p>Merges all CSV files horizontally by <code>on_key</code> key from <code>src_pattern</code></p> <p><code>on_key</code> is the order of key column that is used as key when merging the files. Default is zero.</p> <p>When <code>has_header</code> is not <code>None</code> (default value), it is the order of the row used to make the column name. Returns <code>False</code> if only one file was found, no merging happend</p> <p>Examples:</p> <table><tr><td>Common.</td><td>CSV Merge</td><td>config/data0[3,4].csv</td><td>result/result2.csv</td><td></td></tr><tr><td>Common.</td><td>CSV Merge</td><td>config/data0[3,4].csv</td><td>result/result2.csv</td><td>has_header=\$(TRUE)</td></tr></table>	Common.	CSV Merge	config/data0[3,4].csv	result/result2.csv		Common.	CSV Merge	config/data0[3,4].csv	result/result2.csv	has_header=\$(TRUE)
Common.	CSV Merge	config/data0[3,4].csv	result/result2.csv									
Common.	CSV Merge	config/data0[3,4].csv	result/result2.csv	has_header=\$(TRUE)								
Csv Select	<i>src_file, dst_file, str_row=:, str_col=:, has_header=None</i>	<p>Select part of the CSV file and write it to other file <code>str_row</code> and <code>str_col</code> are used to specify necessary rows and columns. They are using the same format with slice for Python list.</p> <ul style="list-style-type: none">▪ <code>:</code> and <code>:</code> means all rows and columns▪ <code>:2</code> and <code>:</code> means first 2 rows and all columns▪ <code>:</code> and <code>1,2</code> means all rows and 2nd and 3rd columns▪ <code>0:3</code> and <code>1</code> means 3 rows from the 1st one(0,1,2) and second column▪ <code>0:5:2</code> and <code>1</code> means 3 rows(0,3,5) and second column										

		<p>Notes:</p> <ul style="list-style-type: none">Rows and columns are indexed from zeroWhen ':' is used, the string has format: <start>:<stop> or <start>:<stop>:<step> For convenience, ':' means all the data, 'x' means first 'x' data <p>Examples:</p> <table><tr><td>CSV Select</td><td>result/data05.csv</td><td>result/result3.csv</td><td>0,1,2</td><td>0,1</td></tr><tr><td>CSV Select</td><td>result/data05.csv</td><td>result/result4.csv</td><td>:</td><td>0,1</td></tr><tr><td>CSV Select</td><td>result/data05.csv</td><td>result/result5.csv</td><td>:2</td><td>:</td></tr><tr><td>CSV Select</td><td>result/data05.csv</td><td>result/result6.csv</td><td>0:3</td><td>:</td></tr><tr><td>CSV Select</td><td>result/data05.csv</td><td>result/result7.csv</td><td>0:5:2</td><td>:</td></tr></table>	CSV Select	result/data05.csv	result/result3.csv	0,1,2	0,1	CSV Select	result/data05.csv	result/result4.csv	:	0,1	CSV Select	result/data05.csv	result/result5.csv	:2	:	CSV Select	result/data05.csv	result/result6.csv	0:3	:	CSV Select	result/data05.csv	result/result7.csv	0:5:2	:
CSV Select	result/data05.csv	result/result3.csv	0,1,2	0,1																							
CSV Select	result/data05.csv	result/result4.csv	:	0,1																							
CSV Select	result/data05.csv	result/result5.csv	:2	:																							
CSV Select	result/data05.csv	result/result6.csv	0:3	:																							
CSV Select	result/data05.csv	result/result7.csv	0:5:2	:																							
Diff File	path1, path2, newline=True	Shows difference between files Returns the diff result (multi lines) path1 , path2 are absolute paths.																									
Err	msg	Prints error msg to console																									
Error Line Should Not Be Bigger Than	num, *pattern_list	Checks whether the number of lines that contains error be less than a number																									
Error Should Not Be Bigger Than	num, *pattern_list	Checks whether the number of error be less than a number																									
File Md5	path	Returns MD5 hash of a file path is an absolute path																									
Fold Str	str	Folds a string by adding Non-Width-Space char (0x200b) at 6th char																									
Follow Syslog And Trap	pattern, log_file_name=syslog-trap.log, delay_str=1s	Pauses the execution and wait for the pattern is matched if the file log_file_name located in the current result folder. By default the log_file_name is ./result/syslog-trap.log which is created by Follow Syslog and Trap keyword. The keyword should be in tests between Follow Syslog adn Trap Start and Follow Syslog and Trap Stop keywords.																									
Get Config Path		Returns absolute path of RENAT config folder path																									
Get File Without Error	file_path	Get content of the file and return null string if the file does not exist																									
Get Item Config Path		Returns absolute path of current item config folder																									
Get Item Name		Returns the name of the running item																									
Get Renat Path		Returns the absolute path of RENAT folder																									
Get Result Folder		Returns current result folder name. Default is result in current test case. Note: the keyword only returns the name of the result folder not its absolute path.																									
Get Result Path		Returns absolute path of the current result folder																									
Get Test Device		Return a list of all test device that is used in this test Notes: Device number could less than node number																									
Is Stable	seq, threshold, percentile=90	Checks if the value sequence is stable or not																									
Keyword Line Should Not Be Bigger Than	num, keyword, *pattern_list	Checks whether the number of line containing the keyword be less than a number																									
Keyword Should Not Be Bigger Than	num, keyword, *pattern_list	Checks whether the number of keyword be less than a number																									
Load Plugin		Load plugin in renat/plugin folder																									
Log	msg, level=1	Logs msg to the current log file (not console) The msg will logged only if the level is bigger than the global level \${DEBUG} which could be defined at runtime. If \${DEBUG} is not defined, it will be considered as the default level as 1. Examples: <table><tr><td>Common.Log</td><td>XXX</td><td># this always be logged</td><td></td></tr><tr><td>Common.Log</td><td>AAA</td><td>level=2</td><td># this will not be logged with common run.sh</td></tr><tr><td>Common.Log</td><td>BBB</td><td>level=2</td><td># ./run.sh -v DEBUG:2 will log the message</td></tr></table> Notes: For common use <ul style="list-style-type: none">level 1: is defaultlevel 2: is debug modelevel 3: is very informative mode	Common. Log	XXX	# this always be logged		Common. Log	AAA	level=2	# this will not be logged with common run.sh	Common. Log	BBB	level=2	# ./run.sh -v DEBUG:2 will log the message													
Common. Log	XXX	# this always be logged																									
Common. Log	AAA	level=2	# this will not be logged with common run.sh																								
Common. Log	BBB	level=2	# ./run.sh -v DEBUG:2 will log the message																								

Log To Console	msg, level=1	Logs a message to console																
Loop For Node Tag	var, tags, *keywords	<p>See Common.<i>Print</i> for more details about debug level</p> <p>Repeatedly executes RF keyword for nodes that has tag tags</p> <p>multi tags are separated by : keywords has same meaning with keywords used by <i>Run Keywords</i> of RobotFramework (keyword and its arguments are separated by AND with the others.</p> <p>Example:</p> <table><tr><td><i>Loop For Node Tag</i></td><td>\\${node}</td><td>tag1</td><td></td></tr><tr><td>...</td><td>Switch</td><td>\\${node}</td><td>AND</td></tr><tr><td>...</td><td>Cmd</td><td>show system user</td><td>AND</td></tr><tr><td>...</td><td>Cmd</td><td>show system uptime</td><td></td></tr></table> <p>Note: \$ in variable name must be escaped</p>	<i>Loop For Node Tag</i>	\\${node}	tag1		...	Switch	\\${node}	AND	...	Cmd	show system user	AND	...	Cmd	show system uptime	
<i>Loop For Node Tag</i>	\\${node}	tag1																
...	Switch	\\${node}	AND															
...	Cmd	show system user	AND															
...	Cmd	show system uptime																
Md 5	str	Returns MD5 hash of a string																
Merge Files	path_name, file_name	<p>Merges all the text files defined by path_name to file_name</p> <p>Example:</p> <table><tr><td><i>Merge Files</i></td><td>./result/*.csv</td><td>./result/test.csv</td></tr></table>	<i>Merge Files</i>	./result/*.csv	./result/test.csv													
<i>Merge Files</i>	./result/*.csv	./result/test.csv																
Mib For Node	node	<p>Returns the mib file name for this node mib file is define by mib keyword under the node in local.yaml</p> <pre>... node: vmx11: device: vmx11 snmp_polling: yes mib: mib11.txt ...</pre> <p>Default value is defined by mib keyword from global config/snmp-template.yaml for the type of the node</p> <p>Example:</p> <table><tr><td>\\${mib}=</td><td>Common.<i>MIB For Node</i></td><td>vmx11</td></tr></table>	\\${mib}=	Common. <i>MIB For Node</i>	vmx11													
\\${mib}=	Common. <i>MIB For Node</i>	vmx11																
Node With Attr	attr_name, value	Returns a list of nodes which have attribute attr_name with value value																
Node With Tag	*tag_list	<p>Returns list of node from local.yaml that has ALL tags defined by tag_list</p> <p>Tag was defined like this in local.yaml</p> <pre>vmx11: device: vmx11 snmp_polling: yes tag: - tag1 - tag2</pre> <p>Examples:</p> <table><tr><td>\\${test3}=</td><td>Common.<i>Node With Tag</i></td><td>tag1</td><td>tag3</td></tr></table>	\\${test3}=	Common. <i>Node With Tag</i>	tag1	tag3												
\\${test3}=	Common. <i>Node With Tag</i>	tag1	tag3															
Node Without Tag	*tag_list	<p>Returns list of node from local.yaml that does not has ANY tags defined by tag_list</p> <p>Tag was defined like this in local.yaml</p> <pre>vmx11: device: vmx11 snmp_polling: yes tag: - tag1 - tag2</pre> <p>Examples:</p> <table><tr><td>\\${test3}=</td><td>Common.<i>Node Without Tag</i></td><td>tag1</td><td>tag3</td></tr></table>	\\${test3}=	Common. <i>Node Without Tag</i>	tag1	tag3												
\\${test3}=	Common. <i>Node Without Tag</i>	tag1	tag3															
Pause	msg=, time_out=3h, error_on_timeout=True, default_input=	<p>Displays the message msg and pauses the test execution and wait for user input</p> <p>In case of error_on_timeout is True(default), the keyword will raise an error when timeout occurs. Otherwise, it will continue the test.</p> <p>Notes: If the variable \${RENAT_BATCH} was defined, the keyword will print out the message and keeps running without pausing.</p> <p>Examples:</p> <table><tr><td>Common.<i>Pause</i></td><td>Waiting...</td><td>10s</td><td>error_on_timeout=\${TRUE}</td><td>default input</td></tr><tr><td>Common.<i>Pause</i></td><td>Waiting...</td><td>10s</td><td></td><td></td></tr></table>	Common. <i>Pause</i>	Waiting...	10s	error_on_timeout=\${TRUE}	default input	Common. <i>Pause</i>	Waiting...	10s								
Common. <i>Pause</i>	Waiting...	10s	error_on_timeout=\${TRUE}	default input														
Common. <i>Pause</i>	Waiting...	10s																
Ping Until Ok	node, wait_str=5s, extra=-c 3	Ping a node until it gets response. Then wait for more wait_str Default extra option is -c 3																

Random Name	base, a=0, b=99	Returns a random name by a <i>base</i> and a random number between [a,b] Example: <div> \${FOLDER}= <i>Random Name</i> capture_%05d 0 99 </div>																
Random Number	a=0, b=99	Returns a random number between [a,b]																
Renat Version		Returns RENAT version string																
Set Multi Item Variable	*vars	Set multiple variables to be <i>suite variable</i> at the same time Suite variables (or item variable) could be access anywhere in all the item scenario.																
Set Result Folder	folder	Sets the result folder to <i>folder</i> and return the old result folder. The result folder contains all output files from the test likes tester ouput, config file ... <i>folder</i> is a folder name that under current test case folder The system will create a new folder if it does not exist and set its mode to 0775 Note: Result folder should be set at the begining of the test. Changing result folder only has effect on up comming connection																
Slack	msg, channel=#automation_dev, user=renat, host=127.0.0.1	Post a message to Slack																
Str 2 Seq	str_index, size	Returns a sequence from string format Samples: <div><table><tr><td><i>Str2Seq</i></td><td>::</td><td>5</td><td># (0,1,2,3,4)</td></tr><tr><td><i>Str2Seq</i></td><td>:2</td><td>5</td><td># (0,1)</td></tr><tr><td><i>Str2Seq</i></td><td>1:3</td><td>5</td><td># (1,2)</td></tr><tr><td><i>Str2Seq</i></td><td>0:5:2</td><td>5</td><td># (0,2,4)</td></tr></table></div>	<i>Str2Seq</i>	::	5	# (0,1,2,3,4)	<i>Str2Seq</i>	:2	5	# (0,1)	<i>Str2Seq</i>	1:3	5	# (1,2)	<i>Str2Seq</i>	0:5:2	5	# (0,2,4)
<i>Str2Seq</i>	::	5	# (0,1,2,3,4)															
<i>Str2Seq</i>	:2	5	# (0,1)															
<i>Str2Seq</i>	1:3	5	# (1,2)															
<i>Str2Seq</i>	0:5:2	5	# (0,2,4)															
Version		Returns the current version of RENAT																

Altogether 48 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:22.



VChannel

Library version: RENAT 1.7.1
Library scope: test suite
Named arguments: supported

Introduction

A basic library that provides Terminal connection to routers/hosts

VChannel is a core RENAT library that maintains input/output to nodes with an attached virtual terminal. It encapsulates the SSH/Telnet connections behind and provides common usage of access and execute commands to the nodes. Each channel instance has its own log file and a virtual terminal.

Table of Contents

- [Device, Node and Channel](#)
- [Connections](#)
- [Shortcuts](#)
- [Keywords](#)

Device, Node and Channel

RENAT has 3 types of connection target. Device, Node and Channel.

Device

Each device stands for a real physical box that has its own IP address and is defined in the master file `device.yaml`. Users do not directly use device in keywords.

Node

Node is a logical instance of a device. It could stand for a logical instance of a router or just a virtual terminal to the router. Nodes were defined in `local.yaml` of the test case. Several nodes could point to a same device.

Channel

Each channel holds a session to a node. Each channel has its own log file and a virtual terminal. Any command used by `Cmd`, `Write` or `Read` will be logged to the log file. Each channel is identified by a name when it is created with `Connect` keyword and is released with `Close` keyword.

Notes: multi sessions to a same device could be done with predefined multi nodes to same device in the `local.yaml` file or by using multi `Connect` with different `name`.

Connections

The library provides a channel to a target node. Each channel is attached with a virtual terminal. Input and output to the node are made through this virtual terminal. This will help to provide the output looks like the output when operator is using the real terminal.

When keywords `Read`, `Write`, `Cmd` are used, if the connection is not available anymore, the system will try to reconnect to the host with the information provided in the 1st connect. It will try `max_retry_for_connect` times and wait for `interval_between_retry` seconds between retries. The values of `max_retry_for_connect` and `interval_between_retry` are defined in `./config/config.yaml`

Usually when RENAT could not make the connections to the target, the system will raise an exception. But if the `ignore_dead_node` is defined as `yes` in the current active `local.yaml`, the system will ignore the dead node, remove it from the global variable `LOCAL[node]` and `NODE` and keep running the test.

Shortcuts

Change Log · Change Prompt · Close · Close All · Cmd · Cmd Yesno · Connect · Connect All · Flush All · Get Channel · Get Channels · Get Current Channel · Log · Read · Reconnect · Set Log Separator · Start Screen Mode · Stop Screen Mode · Switch · Write

Keywords

Keyword	Arguments	Documentation															
Change Log	<code>log_file</code> , <code>mode=w</code>	Stops current log file and create a new log file. Every log from that point will be saved to the new log file Return old log filename															
Change Prompt	<code>str_prompt</code>	Changes the current prompt of the channel Returns previous prompt. User should change the prompt before execute the new command that expects to see new prompt. Example: <table><tr><td>Router.<code>Switch</code></td><td>vmx11</td><td></td></tr><tr><td>\${prompt}=</td><td>VChannel.<code>Change Prompt</code></td><td>%</td></tr><tr><td>VChannel.<code>Cmd</code></td><td>start shell</td><td></td></tr><tr><td>VChannel.<code>Cmd</code></td><td>ls</td><td></td></tr><tr><td>VChannel.<code>Change Prompt</code></td><td>\${prompt}</td><td></td></tr></table>	Router. <code>Switch</code>	vmx11		\${prompt}=	VChannel. <code>Change Prompt</code>	%	VChannel. <code>Cmd</code>	start shell		VChannel. <code>Cmd</code>	ls		VChannel. <code>Change Prompt</code>	\${prompt}	
Router. <code>Switch</code>	vmx11																
\${prompt}=	VChannel. <code>Change Prompt</code>	%															
VChannel. <code>Cmd</code>	start shell																
VChannel. <code>Cmd</code>	ls																
VChannel. <code>Change Prompt</code>	\${prompt}																

		Vchannel. Cmd	exit													
Close		Closes current connection and reset the channel name														
Close All		Closes all current sessions and flush out all log files. Current node name was reset to <code>None</code>														
Cmd	<i>command, prompt=, match_err=(unknown command./syntax error, expecting <command>.)</i>	Executes a <code>command</code> and wait until for the prompt. This is a blocking keyword. Execution of the test case will be postponed until the prompt appears. If <code>prompt</code> is a null string (default), its value is defined in the <code>./config/template.yaml</code> Output will be automatically logged to the channel current log file. See Common for details about the config files.														
Cmd Yesno	<i>cmd, ans=yes, question=? [yes,no]</i>	Executes a <code>cmd</code> , waits for <code>question</code> and answers that by <code>ans</code>														
Connect	<i>node, name, log_file, timeout=20m, w=80, h=32, mode=w</i>	Connects to the node and create a VChannel instance Login information is automatically extracted from yaml configuration. By default a virtual terminal (vty100) with size 80x64 is attached to this channel. If a login was successful, VChannel will create a log file name <code>log_file</code> for the connection in the current result folder of the test case. This log file will contain any command input/output executed on this channel. Multi sessions to the same node could be open with different names. Use Switch to change the current active session by its name Examples: <table><tr><td>Connect</td><td>vmx11</td><td>vmx11</td><td>vmx11.log</td><td></td><td></td></tr><tr><td>Connect</td><td>vmx11</td><td>vmx11</td><td>vmx11.log</td><td>80</td><td>64</td></tr></table> See <code>Common</code> for more detail about the yaml config files.			Connect	vmx11	vmx11	vmx11.log			Connect	vmx11	vmx11	vmx11.log	80	64
Connect	vmx11	vmx11	vmx11.log													
Connect	vmx11	vmx11	vmx11.log	80	64											
Connect All	<i>prefix=</i>	Connects to all nodes that are defined in active <code>local.yaml</code> . A prefix <code>prefix</code> was appended to the alias name of the connection. A new log file by <code><alias>.log</code> was automatically created. See <code>Common</code> for more detail about active <code>local.yaml</code>														
Flush All																
Get Channel	<i>name</i>	Returns a channel by its <code>name</code>														
Get Channels		Returns all current vchannel instances														
Get Current Channel		Returns the current active channel														
Log	<i>msg</i>	Writes the log message <code>msg</code> to current log file of the channel														
Read	<i>silence=False</i>	Returns the current output of the virtual terminal and automatically logs to file. In <code>normal mode</code> this will return the unread output only, not all the content of the screen.														
Reconnect	<i>name</i>	Reconnects to the <code>name</code> node using existed information The only difference is that the mode of the log file is set to <code>`a+`</code> by default														
Set Log Separator	<i>sep=</i>	Set a separator between the log of <code>read</code> , <code>write</code> or <code>cmd</code> keywords														
Start Screen Mode		Starts the <code>screen mode</code> . In the <code>screen mode</code> , the output is just the same with the real terminal. It means that any real-time application likes <code>top</code> will be captured as-is. Consecutive read from this VChannel instance may produce redundancy output.														
Stop Screen Mode		Stops the <code>screen mode</code> and returns to <code>normal mode</code> In <code>screen mode</code> , Write does not return any thing and no output is logged. In <code>normal mode</code> , escape sequences are not processed by the virtual terminal.														
Switch	<i>name</i>	Switches the current active channel to <code>name</code> . There only one active channel at any time Examples: <table><tr><td>VChannel.Switch</td><td>vmx12</td></tr></table>			VChannel. Switch	vmx12										
VChannel. Switch	vmx12															
Write	<i>str_cmd, str_wait=1s, start_screen_mode=False</i>	Sends <code>str_cmd</code> to the target node and return after <code>str_wait</code> time. If <code>start_screen_mode</code> is <code>True</code> , the channel will be shifted to <code>Screen Mode</code> . Default value of <code>screen_mode</code> is <code>False</code> . In <code>normal mode</code> , a <code>new line</code> char will be added automatically to the <code>str_cmd</code> and the command return the output it could get at that time from the terminal and also logs that to the log file. In <code>screen Mode</code> , if it is necessary you need to add the <code>new line</code> char by your own and the output is not be logged or returned from the keyword. Parameters:														

- `str_cmd`: the command
- `str_wait`: time to wait after apply the command
- `start_screen_mode`: whether start the screen mode right after writes the command

Special input likes Ctrl-C etc. could be used with global variable `${CTRL-<char>}`

Returns the output after writing the command the the channel.

Notes: This is a non-blocking command.

Examples:

VChannel. <i>Write</i>	monitor interface traffic	start_screen_mode=\${TRUE}
VChannel. <i>Write</i>	<code>\${CTRL_C}</code>	# simulates Ctrl-C

Altogether 20 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:25.



Logger

Library version: RENAT 1.7.1
Library scope: test suite
Named arguments: supported

Introduction

Provides advanced logging functions. Every [Logger](#) instance has one [VChannel](#) object and the is synchronized with the current active [VChannel](#).

Notes: log file is not updated pararelly. Anytime a terminal is switched to, it will update its log file.

Shortcuts

Log · Log All · Switch

Keywords

Keyword	Arguments	Documentation								
Log	<i>msg,</i> <i>with_time=False,</i> <i>mark=***</i>	Inserts a message <code>msg</code> to the current <i>VChannel</i> log file. A default mark of <code>***</code> will be added at the beginning ant the end of this message. Example: <table><tr><td>Logger.Log</td><td>START TRAFFIC FROM HERE</td><td>\${TRUE}</td><td></td></tr><tr><td>Logger.Log</td><td>START TRAFFIC FROM HERE</td><td>\${False}</td><td>===</td></tr></table>	Logger. Log	START TRAFFIC FROM HERE	\${TRUE}		Logger. Log	START TRAFFIC FROM HERE	\${False}	===
Logger. Log	START TRAFFIC FROM HERE	\${TRUE}								
Logger. Log	START TRAFFIC FROM HERE	\${False}	===							
Log All	<i>msg,</i> <i>with_time=False,</i> <i>mark=***</i>	inserts a message <code>msg</code> to current all VChannel log files. A default <code>mark</code> of <code>***</code> and newline will be added at the beggining and the end of this message. Example: <table><tr><td>Logger.Log All</td><td>START TRAFFIC FROM HERE</td><td>\${TRUE}</td><td></td></tr><tr><td>Logger.Log All</td><td>START TRAFFIC FROM HERE</td><td>\${TRUE}</td><td>===</td></tr></table> The log file will look likes this: <pre>user@vmx12> *** 06:01PM on August 13, 2017: START TRAFFIC FROM HERE *** === 06:01PM on August 13, 2017: START TRAFFIC FROM HERE === configure</pre>	Logger. Log All	START TRAFFIC FROM HERE	\${TRUE}		Logger. Log All	START TRAFFIC FROM HERE	\${TRUE}	===
Logger. Log All	START TRAFFIC FROM HERE	\${TRUE}								
Logger. Log All	START TRAFFIC FROM HERE	\${TRUE}	===							
Switch	<i>name</i>	Switches the current VChannel instance to <code>name</code> . <code>name</code> is the name of the VChannel (usually is the node name defined in the current active <code>local.yaml</code>). Example: <table><tr><td>Logger.Switch</td><td>vmx11</td></tr></table>	Logger. Switch	vmx11						
Logger. Switch	vmx11									

Altogether 3 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:23.



OpticalSwitch

Library version: RENAT 1.7.1
Library scope: test suite
Named arguments: supported

Introduction

A library provides control for L1 Optical Switch

Unlike other device, there is no *Switch* keywork with optical switch. Usually user only need to care about the interfaces not the ports of the switches.

Shortcuts

Add · **Clear By File** · **Close All** · **Connect All** · **Delete** · **Get Connection Info** · **Load From File** · **Save To File**

Keywords

Keyword	Arguments	Documentation				
Add	dev1, intf1, dev2, intf2, direction=bi, force=False	Adds a connection. See details in each module help				
Clear By File	file_name=, comment=#	Clears all x-connections defined in the connection file Default connection file is defined in optic/connection of config/local.yaml				
Close All		Close all connections				
Connect All		Connect to all L1 switch and read all necessary information				
Delete	dev1, intf1, dev2, intf2, direction=bi, force=False	Deletes a connection. See details in each module help				
Get Connection Info	dev, intf	Returns connection information. See details in each module help.				
Load From File	file_name=, force=True, comment=#	Loads the connection file and set the connections filename is the name of the connection file under the current config folder. If filename is empty, the value of optic/connection from config/local.yaml will be used. The connection file supports jinja2 template language. Besides, # is the default comment char which could be changed The format of connection file follows: <ul style="list-style-type: none">■ each connection is described by 1 line■ source and destination are separated by ` - or > , which mean `bidirection or unidirection (unidirection connects source tx to dest rx Connection file sample: <pre>device1:port1 - device2:port2 device1:port3 > device2:port</pre> Examples: <table><tr><td>OpticalSwitch.Load From File</td><td></td></tr><tr><td>OpticalSwitch.Load From File</td><td>save1.conn</td></tr></table>	OpticalSwitch. Load From File		OpticalSwitch. Load From File	save1.conn
OpticalSwitch. Load From File						
OpticalSwitch. Load From File	save1.conn					
Save To File	file_name	Saves the current connection of all devices in this test. By default, all interfaces of the devices are save. If a connection file is given, only interfaces specified in the connection file are saved Examples: <table><tr><td>OpticalSwitch.Save To File</td><td>save1.conn</td></tr></table>	OpticalSwitch. Save To File	save1.conn		
OpticalSwitch. Save To File	save1.conn					

Altogether 8 keywords.
Generated by [Libdoc](#) on 2018-04-12 19:20:24.



calient

Library scope: global
Named arguments: supported

Introduction

A library provides control for Calient Optical Switch

Table of Contents

- Master file
- Connection file Format
- Shortcuts
- Keywords

Master file

The L1 switch provides a mechanism to remotely connect device interface. Each device interface has been wired to L1 switch already. The connection was described in the master file located specific by *calient-master-path* in the configuration file *renat/config/config.yaml*.
The master file includes several Calients in each tab. The column meaning and order is trivial.

Connection file Format

Keywords *Load From File*, *Clear By File* and *Save To File* use the x-connection file. X-connection files are text files and have the following format:

```
# this is the comment
device1,interface1,-,device2,interface2
device1,interface1,>,device2,interface2
```

The separator `-` means a bidirection connection and `>` means a unidirection connection. For a unidirection connection, `device1/interface1` TX will be connected to `device2/interface2` RX.
Note: The separator character must be surrounded by spaces or commas.
The connection file also support jinja2 template format. After the template is evaluated, comment could be used by `comment char`
There is no need to specify which L1 switch for the x-connection. The system will automatically find the appropriate switch.

Shortcuts

Add · Delete · Get Connection Info

Keywords

Keyword	Arguments	Documentation
Add	<i>self, dev1, intf1, dev2, intf2, direction=bi, force=False</i>	Adds x-connection between <code>dev1:intf1</code> and <code>dev2:intf2</code> <code>direction</code> is <code>bi</code> for bi-direction or <code>uni</code> for uni-direction. If <code>direction</code> is <code>uni</code> , the tx of <code>dev 1:port 1</code> will be connected to <code>dev 2:port 2</code> . With <code>force</code> mode, existed connection that use those ports will be deleted. Without <code>force</code> mode, an existed connection will make the keyword fails Examples: <div>OpticalSwitch.<i>Add</i> mx2008-31-33 xe-3/0/0 mx2008-31-33 xe-3/0/1 bi \${TRUE}</div> Note: when <code>force</code> is <code>False</code> but the current ports is owned by the same connection endpoints, keyword will succeed. For a bidirection connection, 2 single uni-direction connection will be made instead of 1 bi-direction connection. This will make the link could be simulated tx/rx failure later.
Delete	<i>self, dev1, intf1, dev2, intf2, direction=bi, force=False</i>	Deletes the connection between <code>dev1:intf1</code> - <code>dev2:intf2</code> Examples: <div>OpticalSwitch.<i>Delete</i> mx2008-31-33 xe-3/0/1 mx2008-31-33 xe-3/0/1 uni</div>
Get Connection Info	<i>self, dev, intf</i>	Returns information of the optic switch port that connected to <code>dev:intf</code> . The information is in jason format. Examples: <div>OpticalSwitch.<i>Get Connection Info</i> mx2008-31-33 xe-3/0/1</div> return information looks like below: <div>result = {u'outoc': u'NOHW', u'outopwdh': u'-20.0', u'inos': u'OOS', u'outalias': u'', u'inowner': u'TRANSIT', u'outopwct': u'-23.0', u'inppower': u'-3.4', u'inas': u'IS', u'outpower': u'-4.8', u'outas': u'OOS-NP', u'inopt': u'-17.0', u'inoph': u'13.0', u'incircuit': u'3.3.1>3.3.2', u'inalias': u''</div>

	<code>u'inoc': u'NOHW', u'inopte': u'-20.0', u'outos': u'OOS', u'port': u'3.3.1', u'outowner': u'NONE', u'outcircuit': u''}</code>
--	--

Altogether 3 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:27.



g4ntm

Library scope: global
Named arguments: supported

Introduction

A library provides control for Telescent Network Topology Management (NTM) robot patch.

Shortcuts

Add · **Delete** · **Get Connection Info**

Keywords

Keyword	Arguments	Documentation
Add	<i>self, dev1, intf1, dev2, intf2, direction=bi, force=False</i>	
Delete	<i>self, dev1, intf1, dev2, intf2, direction=bi, force=False</i>	Deletes the connection between dev1:intf1 - dev2:intf2
Get Connection Info	<i>self, dev, intf</i>	Returns information about the connection by router/interface

Altogether 3 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:28.



Router

Library version: RENAT 1.7.1
Library scope: test suite
Named arguments: supported

Introduction

A class provides keywords for router controll. Usual command could be executed via [VChannel](#). This class provides the vendor independent commands

Device's `type` is defined in master `device.yaml`. The system will load appropriate modules for each device.

Details about keywords provided by modules could be found in document of each module likes:

- [Juniper module](#)
- [Cisco module](#)
- [GR module](#)

Keywords provides by above module could be executed through [Xrun](#) keyword or directly called from `Router`. Examples:

Router. Xrun	Load Config
Router.Load Config	

Shortcuts

Cmd · **C**md And Wait For · **E**xec File · **F**ollow Mib · **G**et Ip · **R**ead · **S**nap · **S**nap Diff · **S**top Screen Mode · **S**witch · **W**rite · **X**run

Keywords

Keyword	Arguments	Documentation									
Cmd	<i>command=, prompt=</i>	<p>Runs the command <code>command</code> and waits until the prompt defined for this router. This keyword is identical to <code>VChannel.Cmd</code></p> <p>Examples:</p> <table><tr><td>Router.Cmd</td><td>set system login user testtest authentication plain-text-password</td><td>password: # wait for <i>password:</i></td></tr><tr><td>Router.Cmd</td><td>Renat2017</td><td>password: # wait for <i>password:</i></td></tr><tr><td>Router.Cmd</td><td>Renat2017</td><td># wait for default prompt</td></tr></table> <p>The above sample creates an output likes this:</p> <pre>user@vmx11# set system login user testtest authentication plain-text-password New password:Renat2017 Retype new password:Renat2017 [edit]</pre>	Router. Cmd	set system login user testtest authentication plain-text-password	password: # wait for <i>password:</i>	Router. Cmd	Renat2017	password: # wait for <i>password:</i>	Router. Cmd	Renat2017	# wait for default prompt
Router. Cmd	set system login user testtest authentication plain-text-password	password: # wait for <i>password:</i>									
Router. Cmd	Renat2017	password: # wait for <i>password:</i>									
Router. Cmd	Renat2017	# wait for default prompt									
Cmd And Wait For	<i>command, keyword, interval=30s, max_num=10, error_with_max_num=True</i>	<p>Execute a command and expect <code>keyword</code> occurs in the output. If not wait for <code>interval</code> and repeat the process again</p> <p>After <code>max_num</code>, if <code>error_with_max_num</code> is <code>True</code> then the keyword will fail. Otherwise the test continues.</p>									
Exec File	<i>file_name, vars=, comment=#, step=False, str_error=syntax,error</i>	<p>Executes commands listed in <code>file_name</code>. Lines started with <code>comment</code> character is considered as comments</p> <p><code>file_name</code> is a file located inside the <code>config</code> folder of the test case.</p> <p>This command file could be written in Jinja2 format. Default usable variables are <code>LOCAL</code> and <code>GLOBAL</code> which are identical to <code>Common.LOCAL</code> and <code>Common.GLOBAL</code>. More variables could be supplied to the template by <code>vars</code>.</p> <p><code>vars</code> has the format: <code>var1=value1,var2=value2</code></p> <p>If <code>step</code> is <code>True</code>, after very command the output is check agains an error list. And if a match is found, execution will be stopped. Error list is define by <code>str_err</code>, that contains multi regular expression separated by a comma. Default value of <code>str_err</code> is <code>error</code></p> <p>A sample for command list with Jinja2 template:</p> <pre>show interface {{ LOCAL['extra']['line1'] }} show interface {{ LOCAL['extra']['line2'] }} {% for i in range(2) %} show interface et-0/0/{{ i }} {% endfor %}</pre> <p>Examples:</p>									

		<div>Router.Exec File cmd.lst</div> <div>Router.Exec File step=\${TRUE} str_error=syntax,error</div> <p>Note: Comment in the middle of the line is not supported For example if <code>comment</code> is <code>"# "</code></p> <pre># this is comment line <-- this line will be ignored ## this is not an comment line, and will be entered to the router cli, but the router might ignore this</pre>
Follow Mib	<i>node_list, wait_time=10s, interval_time=5s, len=12, percentile=80, threshold=75, max_len=300, factor=1</i>	<p>Waits until all the nodes defined in <code>node_list</code> become <code>stable</code>.</p> <p>Stableness is checked by SNMP polling result. The MIB list is define by <code>mib</code> in <code>node</code> section Parameter:</p> <ul style="list-style-type: none"> <code>wait_time(1)</code>: the time before the evaluation starting <code>interval_time(2)</code>: interval between SNMP polling time <code>threshold</code>: below this value is evaluated as <code>stable</code> <code>len(3)</code>: the size of the evaluation window (number of values that are used in each valuation) <code>percentile</code>: real useful percentage of data (ignore top 100-percentile percent) <code>max_len(4)</code>: maximum waiting <code>lend</code> for this checking <p>time sequence: --(1)--(2)----- ----- ----- ----- </p> <pre><-----> poll poll <-----> <-----> <-----></pre>
Get Ip		<p>Returns the IP address of current node Examples:</p> <div> <div> <div></div> <div> <div></div> <div></div> </div> </div> <div> <div></div> <div>Router.Get IP</div> </div> </div>
Read		Executes command <code>read</code> for the current vchannel coressponded to this router
Snap	<i>name, *cmd_list</i>	<p>Remembers the result of a list of command defined by <code>cmd_list</code></p> <p>Use this keyword with Snap Diff to get the difference between the command's result. The a new snapshot will override the previous result.</p> <p>Each snap is identified by its <code>name</code></p>
Snap Diff	<i>name</i>	<p>Executes the comman that have been executed before by <code>name</code> snapshot and return the difference.</p> <p>Difference is in <code>context diff</code> format</p>
Stop Screen Mode		Stop the screen mode
Switch	<i>name</i>	<p>Changes the current channel of this router to <code>name</code></p> <p>Rerturns old node name</p> <p>Note: This is identical to <code>VChannel.Switch</code></p> <p>Examples:</p> <div> <div> <div></div> <div>Router.Switch vmx11</div> </div> <div> <div></div> <div>Router.Cmd show version</div> </div> </div>
Write	<i>cmd_str, wait_str=1s, start_screen_mode=False</i>	Executes command <code>write</code> for the current vchannel coressponded to this router
Xrun	<i>cmd, *args, **kwargs</i>	<p>Runs the vendor independent keywords.</p> <p>Parametes:</p> <ul style="list-style-type: none"> <code>cmd</code>: a keyword <code>args</code>: other argumemts <p>Examples:</p> <div> <div> <div></div> <div>Router.Xrun Flap Interface ge-0/0/0</div> </div> </div> <p>This keyword will then actually calling the correspond keyword for the device type.</p>



cisco

Library scope: global
Named arguments: supported

Introduction

Provides keywords for Cisco platform

Shortcuts

Get Version

Keywords

Keyword	Arguments	Documentation
Get Version	<i>self</i>	return router version information

Altogether 1 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:26.



gr

Library scope: global
Named arguments: supported

Introduction

Provides keywords for Hitachi GR platform

Shortcuts

Get Chassis Serial · Get Version

Keywords

Keyword	Arguments	Documentation
Get Chassis Serial	<i>self</i>	Returns the serial number of the chassis
Get Version	<i>self</i>	return router version information

Altogether 2 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:26.



juniper

Library scope: global
Named arguments: supported

Introduction

Provides keywords for Juniper platform

Notes: Ignore the *self* parameters when using those keywords.

Shortcuts

Create Best Path Select Data · Disable Interface · Enable Interface · Flap Interface · Get Chassis Serial · Get Cli Mode · Get Config · Get Current Datetime · Get File · Get Intf Addr · Get Route Number · Get Version · Link Status · Load Config · Number Of Bgp Neighbor · Number Of Ospf Neighbor · Number Of Ospf3 Neighbor

Keywords

Keyword	Arguments	Documentation
Create Best Path Select Data	<i>self</i> , <i>route_content</i> , <i>output_excel=best.xlsx</i>	Creates the matrix of best path selection Provides the test described in smb://10.128.3.91/SharePoint01/31_VerificationRoom/31_13_/BGP-Best-Path-Selection All-in-One _20161118 / The test uses predefined Ixia config and follows predefined steps
Disable Interface	<i>self</i> , <i>intf</i>	Disables an interface <i>intf</i>
Enable Interface	<i>self</i> , <i>intf</i>	Enables an interface <i>intf</i>
Flap Interface	<i>self</i> , <i>intf</i> , <i>time_str=10s</i>	Simulates an interface flap for interface <i>intf</i> Disables the interface and wait for a while before turning it up again
Get Chassis Serial	<i>self</i>	Returns the serial number of the chassis
Get Cli Mode	<i>self</i>	Returns current mode of the CLI. Return value is <code>config</code> for configuration mode or <code>command</code> for command mode
Get Config	<i>self</i> , <i>dst_name=</i>	Gets the current configuration file of the router to current <code>result</code> folder. Wait for <i>str_timeout</i> to finish the download, default <i>str_timeout</i> is 10 seconds. Increases this value if the config file is large. Default <i>dst_name</i> is <code>juniper.conf.gz</code>
Get Current Datetime	<i>self</i> , <i>time_format=%H:%M:%S</i> , <i>delta_time=0s</i> , <i>dir=+</i> , <i>**kwargs</i>	Returns the current date time with vendor format <i>delta_time</i> will be added or subtracted to current time, default is <code>0s</code> <i>time_format</i> decides the time part of the output. Example result are : <div>May 24 04:14:25 May 4 04:14:25</div> Note: The date part is padded by space, and the result is allways 15 characters
Get File	<i>self</i> , <i>src_file</i> , <i>dst_file=</i>	Gets a file from router <ul style="list-style-type: none"><i>src_file</i> is a absolute path insides the router<i>dst_file</i> is a file name under <code>result</code> folder
Get Intf Addr	<i>self</i> , <i>intf_name</i> , <i>family=inet</i>	Returns the tuple of address and netmask of an interface <i>family</i> should be <code>inet</code> or <code>inet6</code> If the address is not set, <code>(",")</code> will be returned.
Get Route Number	<i>self</i> , <i>table=inet.0</i>	Returns number of active route in the <i>table</i> <i>table</i> could be <code>inet.0</code> or <code>inet.6</code>
Get Version	<i>self</i>	Returns router version information
Link Status	<i>self</i> , <i>if_name</i>	Returns link physical status as string (aka: "up down", "up up")
Load Config	<i>self</i> , <i>mode=set</i> , <i>config_file=</i> , <i>confirm=0s</i> , <i>vars=</i> , <i>err_match=</i> (<i>syntax</i> <i>error</i>)	Loads configuration to a router. Usable <i>mode</i> is <code>set</code> , <code>override</code> , <code>merge</code> and <code>replace</code> <code>set</code> mode uses configuration that contains <code>set</code> command. Mode <code>override</code> , <code>merge</code> and <code>replace</code> use ordinary JunOS configuration file with appropriate mode. <i>config_file</i> is a configuration file inside the <code>config</code> folder of the current test case. Config file could includes jinja2 template. The template will be evaluated with <i>LOCAL</i> , <i>GLOBAL</i> and variables defined by <i>vars</i> . The <i>vars</i> has the format: <i>var1=value1</i> , <i>var2=value2</i> ... If the loading has no error that match the <i>error_match</i> , the configuration will be committed. The keywordl waits for <i>confirm</i> seconds before rollback the committed configuration. A zero value indicates an immediatly commit

Number Of Bgp Neighbor	<i>self, state=Established</i>	Returns number of BGP neighbor in <code>state</code> <code>state</code>
Number Of Ospf Neighbor	<i>self, state=Full</i>	Returns number of OPSF neighbors with status <code>state</code>
Number Of Ospf3 Neighbor	<i>self, state=Full</i>	Returns number of OPSFv3 neighbors with status <code>state</code>

Altogether 17 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:25.



WebApp

Library version: RENAT 1.7.1
Library scope: test suite
Named arguments: supported

Introduction

A library provides common keywords for web applications (aka Samurai, Arbor TMS)

The library utilize *Selenium2Library* and adds more functions to control Samurai application easily.

The *WebApp* uses the configuration in `local.yaml` in `webapp` section:

```
...
webapp:
  samurai-1:
    device: samurai-b
    profile: samurai.profile
  arbor-1:
    device: arbor-sp-a
    profile: samurai.profile
...
```

Selenium2Library keywords still could be used along with this library. See [Selenium2Library](#) for more details.

See [Arbor](#), [Samurai](#) for details about keywords of each application.

Shortcuts

Capture Screenshot · Connect All · Reset Capture Counter · Set Capture Counter · Set Capture Format

Keywords

Keyword	Arguments	Documentation															
Capture Screenshot	<i>filename=None</i> , <i>extra=</i>	<p>Captures the current screen to file</p> <p>Using the internal counter for filename if <code>filename</code> is not specified. In this case, the filename is defined by a pre-set format. Set Capture Format could be used to change the current format.</p> <p>An extra information will be add to the filename if <code>extra</code> is defined</p> <p>Examples:</p> <table><tr><td>Samurai.Capture Screenshot</td><td></td><td># samurai_0000000001.png</td></tr><tr><td>Samurai.Capture Screenshot</td><td>extra=_list</td><td># samurai_0000000002_list.png</td></tr><tr><td>Arbor.Capture Screenshot</td><td></td><td># arbor_0000000001.png</td></tr><tr><td>Arbor.Capture Screenshot</td><td>extra=_xxx</td><td># arbor_0000000001_xxx.png</td></tr><tr><td>Samurai.Capture Screenshot</td><td>file_name=1111.png</td><td># 1111.png</td></tr></table>	Samurai. Capture Screenshot		# samurai_0000000001.png	Samurai. Capture Screenshot	extra=_list	# samurai_0000000002_list.png	Arbor. Capture Screenshot		# arbor_0000000001.png	Arbor. Capture Screenshot	extra=_xxx	# arbor_0000000001_xxx.png	Samurai. Capture Screenshot	file_name=1111.png	# 1111.png
Samurai. Capture Screenshot		# samurai_0000000001.png															
Samurai. Capture Screenshot	extra=_list	# samurai_0000000002_list.png															
Arbor. Capture Screenshot		# arbor_0000000001.png															
Arbor. Capture Screenshot	extra=_xxx	# arbor_0000000001_xxx.png															
Samurai. Capture Screenshot	file_name=1111.png	# 1111.png															
Connect All		<p>Connects to all applications defined in <code>local.yaml</code></p> <p>The name of the connection will be the same of the <i>webapp</i> name</p>															
Reset Capture Counter		<p>Resets the counter of the screen capture</p>															
Set Capture Counter	<i>value=0</i>	<p>Sets the counter of the screen capture to <code>value</code></p>															
Set Capture Format	<i>format</i>	<p>Sets the format for the screen capture file</p> <p>The format does not include the default prefix <code>.png</code> The default format is <code><mod>_%010d</code>. <code>mod</code> could be <code>samurai</code> or <code>arbor</code></p> <p>See https://docs.python.org/2/library/string.html#format-specification-mini-language for more details about the format string.</p> <p>Examples:</p> <table><tr><td>Samurai.Set Capture Format</td><td>\${case}_%010d</td><td># \${case} is a predefined variable</td></tr></table>	Samurai. Set Capture Format	\${case}_%010d	# \${case} is a predefined variable												
Samurai. Set Capture Format	\${case}_%010d	# \${case} is a predefined variable															

Altogether 5 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:28.



Samurai

Library version: RENAT 1.7.1
Library scope: test suite
Named arguments: supported

Introduction

A library provides functions to control Samurai application

The library utilize *Selenium2Library* and adds more functions to control Samurai application easily. Without other further mentions, all of the concepts of `user`, `user group` are Samurai concepts.

By default, RENAT will try to connect to all Samurai nodes defined in active `local.yaml` at the beginning of the test and disconnect from them at the end of the test automatically. Usually user does not need to use `Connect All` and `Close` explicitly.

Currently, this module supposed that Samurai is used in Japanese locale. When Samurai module has error, it tried to make the last snapshot in `result/selenium-screenshot-x.png`. Checking this capture will help to understand the reason of the error.

Some keywords of [Samurai](#) is using `xpath` to identify elements. See *Selenium2Library* for more details about xpath.

See [WebApp](#) for common keywords of web applications.

Selenium2Library keywords still could be used together within this library. See [Selenium2Library](#) for more details.

Shortcuts

Add Policy · **Add Policy Group** · **Add User** · **Capture Screenshot** · **Change Policy View Group** · **Click All Elements** · **Close** · **Close All** · **Connect** · **Connect All** · **Delete Policy** · **Delete Policy Group** · **Delete User** · **Edit Policy** · **Left Menu** · **Login** · **Logout** · **Make Item Map** · **Reset Capture Counter** · **Select Items In Table** · **Set Capture Counter** · **Set Capture Format** · **Show Policy Basic** · **Show Policy Mitigation** · **Show Policy Mo** · **Show Policy Monitor** · **Start Mitigation** · **Stop Mitigation** · **Switch**

Keywords

Keyword	Arguments	Documentation																																																																																																																
Add Policy	**policy	Adds a new Samurai policy																																																																																																																
		policy is a map containing the below information to create the new policy.																																																																																																																
		<table><tr><th>key</th><th>meaning</th><th>mandatory</th><th>sample</th></tr><tr><td>name</td><td>name of the policy</td><td>yes</td><td>test001</td></tr><tr><td>basic_alias</td><td>alias name of the policy</td><td></td><td>test001</td></tr><tr><td>basic_port_id</td><td>another alias</td><td></td><td></td></tr><tr><td>basic_facing</td><td>customer or backbone</td><td></td><td>customer</td></tr><tr><td>basic_intf_list</td><td>list of router and interface pair, separated by comma</td><td>yes</td><td>10.128.18.31:xe-0/0/0.1</td></tr><tr><td>basic_cidr_list</td><td>list of CIDR separate by comma</td><td></td><td></td></tr><tr><td>basic_option_filter</td><td>optinal filter</td><td></td><td></td></tr><tr><td>basic_direction</td><td>direction of the traffic (incoming or outgoing)</td><td></td><td>Incoming</td></tr><tr><td>traffic_enabled</td><td>Enable traffic monitoring or not</td><td>yes</td><td>\${True} or \${False}</td></tr><tr><td>detection_enabled</td><td>Enable detection or not</td><td>yes</td><td>\${True} or \${False}</td></tr><tr><td>mitigation_zone_name</td><td>Name of the zone for mitigation</td><td></td><td>zone001</td></tr><tr><td>mitigation_zone_prefix</td><td>Prefixes that could mitigate</td><td></td><td>1.1.1.1/32</td></tr><tr><td>mitigation_thr_bps</td><td>Upper limit (bps)</td><td></td><td>800,000,000</td></tr><tr><td>mitigation_thr_pps</td><td>Upper limit (pps)</td><td></td><td>54,000,000</td></tr><tr><td>mitigation_mo_enabled</td><td>Using Arbor TMS MO or not</td><td>yes</td><td>\${True} or \${False}</td></tr><tr><td>mitigation_device_list</td><td>Devices used for TMS, separated by comma</td><td></td><td>ArborSP-A</td></tr><tr><td>mitigation_mo_name</td><td>MO name, separated by comma</td><td></td><td>OCN12(ALU)_LOOSE</td></tr><tr><td>mitigation_comm_list</td><td>commna separated peer/community list</td><td>yes</td><td>1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777</td></tr><tr><td>nw_monitor_gre1</td><td>1st GRE address for NW monitor</td><td></td><td>210.0.1.1</td></tr><tr><td>nw_monitor_gre2</td><td>2nd GRE address for NW monitor</td><td></td><td>210.0.1.1</td></tr><tr><td>nw_monitor_ce1</td><td>1st CE address for NW monitor</td><td></td><td>210.0.1.2</td></tr><tr><td>nw_monitor_ce2</td><td>2nd CE address for NW monitor</td><td></td><td>210.0.1.2</td></tr><tr><td>nw_monitor_pe1</td><td>1st PE for NW monitor (list)</td><td></td><td>edge01hige-MX2020-15(118.23.176.244)</td></tr><tr><td>nw_monitor_pe2</td><td>2nd PE for NW monitor (list)</td><td></td><td>edge01hige-MX2020-15(118.23.176.244)</td></tr><tr><td>event_name</td><td>name of the message event to make</td><td></td><td>info1</td></tr><tr><td>event_addr</td><td>address to send the events</td><td></td><td>user@mail.com</td></tr><tr><td>view_group</td><td>user group that could view this policy, separated by comma</td><td>yes</td><td>SuperGroup,test_group_007</td></tr></table>	key	meaning	mandatory	sample	name	name of the policy	yes	test001	basic_alias	alias name of the policy		test001	basic_port_id	another alias			basic_facing	customer or backbone		customer	basic_intf_list	list of router and interface pair, separated by comma	yes	10.128.18.31:xe-0/0/0.1	basic_cidr_list	list of CIDR separate by comma			basic_option_filter	optinal filter			basic_direction	direction of the traffic (incoming or outgoing)		Incoming	traffic_enabled	Enable traffic monitoring or not	yes	\${True} or \${False}	detection_enabled	Enable detection or not	yes	\${True} or \${False}	mitigation_zone_name	Name of the zone for mitigation		zone001	mitigation_zone_prefix	Prefixes that could mitigate		1.1.1.1/32	mitigation_thr_bps	Upper limit (bps)		800,000,000	mitigation_thr_pps	Upper limit (pps)		54,000,000	mitigation_mo_enabled	Using Arbor TMS MO or not	yes	\${True} or \${False}	mitigation_device_list	Devices used for TMS, separated by comma		ArborSP-A	mitigation_mo_name	MO name, separated by comma		OCN12(ALU)_LOOSE	mitigation_comm_list	commna separated peer/community list	yes	1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777	nw_monitor_gre1	1st GRE address for NW monitor		210.0.1.1	nw_monitor_gre2	2nd GRE address for NW monitor		210.0.1.1	nw_monitor_ce1	1st CE address for NW monitor		210.0.1.2	nw_monitor_ce2	2nd CE address for NW monitor		210.0.1.2	nw_monitor_pe1	1st PE for NW monitor (list)		edge01hige-MX2020-15(118.23.176.244)	nw_monitor_pe2	2nd PE for NW monitor (list)		edge01hige-MX2020-15(118.23.176.244)	event_name	name of the message event to make		info1	event_addr	address to send the events		user@mail.com	view_group	user group that could view this policy, separated by comma	yes	SuperGroup,test_group_007
		key	meaning	mandatory	sample																																																																																																													
		name	name of the policy	yes	test001																																																																																																													
		basic_alias	alias name of the policy		test001																																																																																																													
		basic_port_id	another alias																																																																																																															
		basic_facing	customer or backbone		customer																																																																																																													
		basic_intf_list	list of router and interface pair, separated by comma	yes	10.128.18.31:xe-0/0/0.1																																																																																																													
		basic_cidr_list	list of CIDR separate by comma																																																																																																															
		basic_option_filter	optinal filter																																																																																																															
		basic_direction	direction of the traffic (incoming or outgoing)		Incoming																																																																																																													
		traffic_enabled	Enable traffic monitoring or not	yes	\${True} or \${False}																																																																																																													
		detection_enabled	Enable detection or not	yes	\${True} or \${False}																																																																																																													
		mitigation_zone_name	Name of the zone for mitigation		zone001																																																																																																													
		mitigation_zone_prefix	Prefixes that could mitigate		1.1.1.1/32																																																																																																													
		mitigation_thr_bps	Upper limit (bps)		800,000,000																																																																																																													
		mitigation_thr_pps	Upper limit (pps)		54,000,000																																																																																																													
		mitigation_mo_enabled	Using Arbor TMS MO or not	yes	\${True} or \${False}																																																																																																													
		mitigation_device_list	Devices used for TMS, separated by comma		ArborSP-A																																																																																																													
		mitigation_mo_name	MO name, separated by comma		OCN12(ALU)_LOOSE																																																																																																													
		mitigation_comm_list	commna separated peer/community list	yes	1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777																																																																																																													
		nw_monitor_gre1	1st GRE address for NW monitor		210.0.1.1																																																																																																													
		nw_monitor_gre2	2nd GRE address for NW monitor		210.0.1.1																																																																																																													
		nw_monitor_ce1	1st CE address for NW monitor		210.0.1.2																																																																																																													
		nw_monitor_ce2	2nd CE address for NW monitor		210.0.1.2																																																																																																													
		nw_monitor_pe1	1st PE for NW monitor (list)		edge01hige-MX2020-15(118.23.176.244)																																																																																																													
nw_monitor_pe2	2nd PE for NW monitor (list)		edge01hige-MX2020-15(118.23.176.244)																																																																																																															
event_name	name of the message event to make		info1																																																																																																															
event_addr	address to send the events		user@mail.com																																																																																																															
view_group	user group that could view this policy, separated by comma	yes	SuperGroup,test_group_007																																																																																																															
		Example:																																																																																																																
		<table><tr><td>Samurai.Switch</td><td>samurai-1</td><td></td></tr><tr><td>Samurai.Add Policy</td><td>name=\${POLICY_NAME}</td><td>basic_alias=\${POLICY_NAME}</td></tr><tr><td>...</td><td>basic_facing=customer</td><td>basic_intf_list=10.128.18.31:xe-0/0/0.1</td></tr><tr><td>...</td><td>basic_cidr_list=1.1.1.0/24</td><td>basic_direction=incoming</td></tr><tr><td>...</td><td>traffic_enabled=\${TRUE}</td><td></td></tr><tr><td>...</td><td>detection_enabled=\${TRUE}</td><td></td></tr></table>	Samurai. Switch	samurai-1		Samurai. Add Policy	name=\${POLICY_NAME}	basic_alias=\${POLICY_NAME}	...	basic_facing=customer	basic_intf_list=10.128.18.31:xe-0/0/0.1	...	basic_cidr_list=1.1.1.0/24	basic_direction=incoming	...	traffic_enabled=\${TRUE}		...	detection_enabled=\${TRUE}																																																																																															
Samurai. Switch	samurai-1																																																																																																																	
Samurai. Add Policy	name=\${POLICY_NAME}	basic_alias=\${POLICY_NAME}																																																																																																																
...	basic_facing=customer	basic_intf_list=10.128.18.31:xe-0/0/0.1																																																																																																																
...	basic_cidr_list=1.1.1.0/24	basic_direction=incoming																																																																																																																
...	traffic_enabled=\${TRUE}																																																																																																																	
...	detection_enabled=\${TRUE}																																																																																																																	

		<table><tr><td>...</td><td>mitigation_zone_name=test_zone001</td><td colspan="2">mitigation_zone_prefix=1.1.1.1/32</td></tr><tr><td>...</td><td>mitigation_device_list=ArborSP-A,ArborSP-B</td><td colspan="2"></td></tr><tr><td>...</td><td>mitigation_mo_enabled=\${TRUE}</td><td colspan="2"></td></tr><tr><td>...</td><td>mitigation_mo_name=N000000012_LOOSE</td><td colspan="2"></td></tr><tr><td>...</td><td>mitigation_comm_list=1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777</td><td colspan="2"></td></tr><tr><td>...</td><td>event_name=test</td><td colspan="2">event_addr=user@mail.com</td></tr><tr><td>...</td><td>view_group=SuperGroup</td><td colspan="2"></td></tr></table>	...	mitigation_zone_name=test_zone001	mitigation_zone_prefix=1.1.1.1/32		...	mitigation_device_list=ArborSP-A,ArborSP-B			...	mitigation_mo_enabled=\${TRUE}			...	mitigation_mo_name=N000000012_LOOSE			...	mitigation_comm_list=1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777			...	event_name=test	event_addr=user@mail.com		...	view_group=SuperGroup		
...	mitigation_zone_name=test_zone001	mitigation_zone_prefix=1.1.1.1/32																												
...	mitigation_device_list=ArborSP-A,ArborSP-B																													
...	mitigation_mo_enabled=\${TRUE}																													
...	mitigation_mo_name=N000000012_LOOSE																													
...	mitigation_comm_list=1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777																													
...	event_name=test	event_addr=user@mail.com																												
...	view_group=SuperGroup																													
Add Policy Group	group_name, policy_list=*, limit_bps=4000000000, limit_pps=2700000	Add a new policy group group_name is the name of the new group. policy_list is a comma separated of existed policy that should be bound to this policy. An asterisk for this parameter (*) means all of the existed policy. limit_bps and limit_pps are the mitigation capacity threshold of this group.																												
Add User	group, **user_info	Adds user to the current group user_info is a dictionary contains user information that has following keys: name, password, privilege and policy privilege is existed privilege that has been created (e.g: system_admin. policy could be * for all current policies or a list of policy names that are binded to this user. group is the user group. Dot(.) means current group Examples: <table><tr><td>Samurai.Add User</td><td>OCNDDoS</td><td>name=user000</td><td>password=Test12345678</td></tr><tr><td>...</td><td>privilege=system_admin</td><td>policy=*</td><td></td></tr><tr><td>Samurai.Add User</td><td>OCNDDoS</td><td>username=user001</td><td>password=Test12345678</td></tr><tr><td>...</td><td>privilege=system_admin</td><td>policy=OCN11,OCN12</td><td></td></tr></table>		Samurai.Add User	OCNDDoS	name=user000	password=Test12345678	...	privilege=system_admin	policy=*		Samurai.Add User	OCNDDoS	username=user001	password=Test12345678	...	privilege=system_admin	policy=OCN11,OCN12												
Samurai.Add User	OCNDDoS	name=user000	password=Test12345678																											
...	privilege=system_admin	policy=*																												
Samurai.Add User	OCNDDoS	username=user001	password=Test12345678																											
...	privilege=system_admin	policy=OCN11,OCN12																												
Capture Screenshot	filename=None, extra=	Captures the current screen to file Using the internal counter for filename if filename is not specified. In this case, the filename is defined by a pre-set format. Set Capture Format could be used to change the current format. An extra information will be add to the filename if extra is defined Examples: <table><tr><td>Samurai.Capture Screenshot</td><td></td><td># samurai_0000000001.png</td></tr><tr><td>Samurai.Capture Screenshot</td><td>extra=_list</td><td># samurai_0000000002_list.png</td></tr><tr><td>Arbor.Capture Screenshot</td><td></td><td># arbor_0000000001.png</td></tr><tr><td>Arbor.Capture Screenshot</td><td>extra=_xxx</td><td># arbor_0000000001_xxx.png</td></tr><tr><td>Samurai.Capture Screenshot</td><td>file_name=1111.png</td><td># 1111.png</td></tr></table>		Samurai.Capture Screenshot		# samurai_0000000001.png	Samurai.Capture Screenshot	extra=_list	# samurai_0000000002_list.png	Arbor.Capture Screenshot		# arbor_0000000001.png	Arbor.Capture Screenshot	extra=_xxx	# arbor_0000000001_xxx.png	Samurai.Capture Screenshot	file_name=1111.png	# 1111.png												
Samurai.Capture Screenshot		# samurai_0000000001.png																												
Samurai.Capture Screenshot	extra=_list	# samurai_0000000002_list.png																												
Arbor.Capture Screenshot		# arbor_0000000001.png																												
Arbor.Capture Screenshot	extra=_xxx	# arbor_0000000001_xxx.png																												
Samurai.Capture Screenshot	file_name=1111.png	# 1111.png																												
Change Policy View Group	name, *group_name	Changes the groups that could see this policy name is the policy name. group_name is a list of policies Example: <table><tr><td>Samurai.Change Policy View Group</td><td>super_admin</td><td>test_group001</td></tr></table>		Samurai.Change Policy View Group	super_admin	test_group001																								
Samurai.Change Policy View Group	super_admin	test_group001																												
Click All Elements	xpath	Click all element in current page defined by xpath Returns the number of elements that have been clicked																												
Close		Closes the current active browser																												
Close All		Closes all current opened applications																												
Connect	app, name	Opens a web browser and connects to application and assigns a name. If not defined in local.yaml those following key will have default values: <table><tr><td>browser</td><td>firefox</td><td>optional</td></tr><tr><td>login_url</td><td>/</td><td>optiona</td></tr><tr><td>proxy:</td><td></td><td>optional</td></tr><tr><td>http: 10.128.8.210:8080</td><td>optional</td><td></td></tr><tr><td>ssl: 10.128.8.210:8080</td><td>optional</td><td></td></tr><tr><td>socks: 10.128.8.210:8080</td><td>optional</td><td></td></tr><tr><td>profile_dir</td><td>./config/samurai.profile</td><td>optional</td></tr></table>		browser	firefox	optional	login_url	/	optiona	proxy:		optional	http: 10.128.8.210:8080	optional		ssl: 10.128.8.210:8080	optional		socks: 10.128.8.210:8080	optional		profile_dir	./config/samurai.profile	optional						
browser	firefox	optional																												
login_url	/	optiona																												
proxy:		optional																												
http: 10.128.8.210:8080	optional																													
ssl: 10.128.8.210:8080	optional																													
socks: 10.128.8.210:8080	optional																													
profile_dir	./config/samurai.profile	optional																												
Connect All		Connects to all applications defined in local.yaml The name of the connection will be the same of the webapp name																												
Delete Policy	*policy_names	Deletes poilcies by their names Returned the number of deleted users Notes: If the policy does not exists, the system will not report any error. Examples: <table><tr><td>Samurai.Delete Policy</td><td>test001</td><td>test002</td></tr></table>		Samurai.Delete Policy	test001	test002																								
Samurai.Delete Policy	test001	test002																												
Delete Policy Group	*group_list	Deletes policy groups Returns the number of deleted policy groups Example: <table><tr><td>Samurai.Delete Policy Group</td><td>test_group001</td><td>test_group002</td></tr></table>		Samurai.Delete Policy Group	test_group001	test_group002																								
Samurai.Delete Policy Group	test_group001	test_group002																												
Delete User	group, *user_list	Deletes user from the user group group is the user group. And . means current group Returns the number of deleted users Examples: <table><tr><td>Samurai.Delete User</td><td>SuperGroup</td><td>user001</td><td>user002</td></tr><tr><td>Samurai.Delete User</td><td>.</td><td>user002</td><td></td></tr></table>		Samurai.Delete User	SuperGroup	user001	user002	Samurai.Delete User	.	user002																				
Samurai.Delete User	SuperGroup	user001	user002																											
Samurai.Delete User	.	user002																												

Edit Policy	<i>**policy</i>	Edits a Samurai policy policy contains information about the policy. See Add Policy for more details about policy format						
Left Menu	<i>menu</i>	Chooses the left panel menu by its displayed name Return a list of 1st meaningful column Example: <table><tr><td>Samurai.Left Menu</td><td>Traffic</td></tr><tr><td>Samurai.Left Menu</td><td>Detection</td></tr><tr><td>Samurai.Left Menu</td><td></td></tr></table>	Samurai. Left Menu	Traffic	Samurai. Left Menu	Detection	Samurai. Left Menu	
Samurai. Left Menu	Traffic							
Samurai. Left Menu	Detection							
Samurai. Left Menu								
Login		Logs-in into the application User and password is set by the template and authentication methods in the master files						
Logout		Logs-out the current application, the browser remains						
Make Item Map	<i>xpath</i>	Makes a item/webelement defined <i>xpath</i> The map is a dictionary from <i>item</i> to the <i>WebElement</i> Items name found by <i>xpath</i> are used as keys						
Reset Capture Counter		Resets the counter of the screen capture						
Select Items In Table	<i>xpath, xpath2, *item_list</i>	Checks items in Samurai table by <i>xpath</i> <i>xpath</i> points to the column that used as key and <i>xpath2</i> is the relative <i>xpath</i> contains the checkbox column. <i>item_list</i> is a list of item that need to check. Item in the list could be a regular expresion with the format <code>reg=<regular expression</code> . The keyword is called with assuming that the table is already visible. Returns the tuple of all items and selected items Note: Non-width-space (\u200b) will be take care by the keyword. Note: if the first <i>item_list</i> is <code>*</code> then the keyword will try to click a link named <code>.</code>						
Set Capture Counter	<i>value=0</i>	Sets the counter of the screen capture to <i>value</i>						
Set Capture Format	<i>format</i>	Sets the format for the screen capture file The format does not include the default prefix <code>.png</code> The default format is <code><mod>_%010d</code> . <i>mod</i> could be <code>samurai</code> or <code>arbor</code> See https://docs.python.org/2/library/string.html#format-specification-mini-language for more details about the format string. Examples: <table><tr><td>Samurai.Set Capture Format</td><td><code>\${case}_%010d # \${case} is a predefined variable</code></td></tr></table>	Samurai. Set Capture Format	<code>\${case}_%010d # \${case} is a predefined variable</code>				
Samurai. Set Capture Format	<code>\${case}_%010d # \${case} is a predefined variable</code>							
Show Policy Basic	<i>policy_name</i>	Makes the virtual browser show basic setting of the policy <i>name</i> . A following Samurai. Capture Screenshot is necessary to capture the result.						
Show Policy Mitigation	<i>policy_name</i>	Make the virtual browser show the mitigation setting of a policy A following Samurai. Capture Screenshot is necessary to capture the result.						
Show Policy Mo	<i>policy_name</i>	Make the virtual browser show the MO setting of a policy Automatically expand the MO section of other devices if necessary. A following Samurai. Capture Screenshot is necessary to capture the result.						
Show Policy Monitor	<i>policy_name</i>	Make a virtual browser show the mitigation setting of a policy A following Samurai. Capture Screenshot is necessary to capture the result.						
Start Mitigation	<i>policy, prefix, comment=mitigation started by RENAT, device=None, force=False</i>	Starts a mitigation with specific <i>prefix</i> <i>device</i> is used for matching real device name configured by Samurai If <i>force</i> is <code>TRUE</code> then the keyword will fail if selected device does not contain <i>device</i> Returns mitigation <i>id</i> and selected <i>arbor</i> device Example: <table><tr><td><code>\${id} \${device}= Samurai.Start Mitigation 211.1.12.1/32 mitigation by RENAT SP-A \${TRUE}</code></td></tr></table>	<code>\${id} \${device}= Samurai.Start Mitigation 211.1.12.1/32 mitigation by RENAT SP-A \${TRUE}</code>					
<code>\${id} \${device}= Samurai.Start Mitigation 211.1.12.1/32 mitigation by RENAT SP-A \${TRUE}</code>								
Stop Mitigation	<i>id</i>	Stops a mitigation by its ID Example: <table><tr><td>Samurai.Stop Mitigation 700</td></tr></table>	Samurai. Stop Mitigation 700					
Samurai. Stop Mitigation 700								
Switch	<i>name</i>	Switches the current browser to <i>name</i>						

Altogether 29 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:29.



Arbor

Library version: RENAT 1.7.1
Library scope: test suite
Named arguments: supported

Introduction

A library provides functions to control Arbor application

The library utilize *Selenium2Library* and adds more functions to control Arbor application easily.

See [WebApp](#) for common keywords of web applications.

Selenium2Library keywords still could be used along with this library. See [Selenium2Library](#) for more details.

Shortcuts

Capture Screenshot · Close · Close All · Connect · Connect All · Detail First Mitigation · Login · Logout · Reset Capture Counter · Set Capture Counter · Set Capture Format · Set Count · Show All Mitigations · Show Detail Mitigation · Switch

Keywords

Keyword	Arguments	Documentation															
Capture Screenshot	filename=None, extra=	<p>Captures the current screen to file</p> <p>Using the internal counter for filename if filename is not specified. In this case, the filename is defined by a pre-set format. Set Capture Format could be used to change the current format.</p> <p>An extra information will be add to the filename if extra is defined</p> <p>Examples:</p> <table><tr><td>Samurai.Capture Screenshot</td><td></td><td># samurai_0000000001.png</td></tr><tr><td>Samurai.Capture Screenshot</td><td>extra=_list</td><td># samurai_0000000002_list.png</td></tr><tr><td>Arbor.Capture Screenshot</td><td></td><td># arbor_0000000001.png</td></tr><tr><td>Arbor.Capture Screenshot</td><td>extra=_xxx</td><td># arbor_0000000001_xxx.png</td></tr><tr><td>Samurai.Capture Screenshot</td><td>file_name=1111.png</td><td># 1111.png</td></tr></table>	Samurai. Capture Screenshot		# samurai_0000000001.png	Samurai. Capture Screenshot	extra=_list	# samurai_0000000002_list.png	Arbor. Capture Screenshot		# arbor_0000000001.png	Arbor. Capture Screenshot	extra=_xxx	# arbor_0000000001_xxx.png	Samurai. Capture Screenshot	file_name=1111.png	# 1111.png
Samurai. Capture Screenshot		# samurai_0000000001.png															
Samurai. Capture Screenshot	extra=_list	# samurai_0000000002_list.png															
Arbor. Capture Screenshot		# arbor_0000000001.png															
Arbor. Capture Screenshot	extra=_xxx	# arbor_0000000001_xxx.png															
Samurai. Capture Screenshot	file_name=1111.png	# 1111.png															
Close		Closes the current active browser															
Close All		Closes all current opened applications															
Connect	app, name	<p>Opens a web browser and connects to application and assigns a name.</p> <p>Extra information could be added to the webapp sections likes login_url, browser or profile_dir. Default values are:</p> <table><tr><td>browser</td><td>firefox</td></tr><tr><td>login_url</td><td>/</td></tr><tr><td>profile_dir</td><td>./config/samurai.profile</td></tr></table>	browser	firefox	login_url	/	profile_dir	./config/samurai.profile									
browser	firefox																
login_url	/																
profile_dir	./config/samurai.profile																
Connect All		<p>Connects to all applications defined in local.yaml</p> <p>The name of the connection will be the same of the webapp name</p>															
Detail First Mitigation		Shows details about the 1st mitigation on the list															
Login		Logged-into the Arbor application															
Logout		Logs-out the current application, the browser remains															
Reset Capture Counter		Resets the counter of the screen capture															
Set Capture Counter	value=0	Sets the counter of the screen capture to value															
Set Capture Format	format	<p>Sets the format for the screen capture file</p> <p>The format does not include the default prefix .png The default format is <mod>_%010d. mod could be samurai or arbor</p> <p>See https://docs.python.org/2/library/string.html#format-specification-mini-language for more details about the format string.</p> <p>Examples:</p> <table><tr><td>Samurai.Set Capture Format</td><td>\${case}_%010d</td><td># \${case} is a predefined variable</td></tr></table>	Samurai. Set Capture Format	\${case}_%010d	# \${case} is a predefined variable												
Samurai. Set Capture Format	\${case}_%010d	# \${case} is a predefined variable															
Set Count	counter=0	Sets current counter to counter															
Show All Mitigations		Shows all mitigations															
Show Detail	id																

Mitigation		Shows detail information for a mitigation
Switch	<i>name</i>	Switches the current browser to <code>name</code>

Altogether 15 keywords.
Generated by [Libdoc](#) on 2018-04-12 19:20:29.



Tester

Library version: RENAT 1.7.1
Library scope: test suite
Named arguments: supported

Introduction

A class provides keywords for controlling testers and traffic generators.

It could load predefined traffic file, manipulate traffic items, start and stop traffic flows. It also could generate traffic reports and support QuickTest for IxNetwork.

Tester information is stored in the active `local.yaml` likes this:

```
tester:  
  tester01:  
    device: ixnet03_8009  
    config: vmx_20161129.ixncfg  
    real_port:  
      - description: to egde router
```

chassis: 10.128.32.71

```
card: 6  
port: 11  
- description: to backbone router
```

chassis: 10.128.32.71

```
card: 6  
port: 9
```

where `device` is the tester defined in the master `device.yaml` file. If `real_port` does not exist, port remapping will not take place. Otherwise, port remapping will use the `real_port` information to reassign all existed ports and map to Ixia ports.

In this case, the order will be the order when user created the port in Ixia GUI.

Note: User can always confirm the created order by `clear sorting` in Ixia GUI.

Examples:

Tester. Connect All	
Tester. Switch	tester01
Tester. Load And Start Traffic	
Sleep	30s
Tester. Stop Traffic	

Time format used in this module is same with `time string` format of Robot Framework. For more details about this, see [DateTime](#) library of Robot Framework.

Note: See [IxNet module](#), [IxLoad module](#) for details about keyword of each module.

Shortcuts

Close All · Connect · Connect All · Switch

Keywords

Keyword	Arguments	Documentation
Close All		Closes all connections
Connect	<i>name</i>	Connect to the tester <i>name</i>
Connect All		Connects to all testers
Switch	<i>name</i>	Switchs the current tester to <i>name</i>

Altogether 4 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:24.



ixload

Library scope: global
Named arguments: supported

Introduction

provides functions for IxLoad

RENAT runs a virtual IxLoad client locally in the background that connects to a Windows App server. Keywords from test case will send control messages to the client, which in turn will control the test ports.

Different to IxNetwork, an IxLoad test case usually stops within predefined time before `Stop Traffic` was called.

Notes: Ignore the *self* parameters when using those keywords.

Shortcuts

`Close` · `Collect Data` · `Get Test Report` · `Load Config` · `Load Traffic` · `Start Traffic` · `Stop Traffic`

Keywords

Keyword	Arguments	Documentation
Close	<i>self</i>	Disconnects the current tester client
Collect Data	<i>self</i> , <i>prefix=</i> , <i>more_file=</i> , <i>ignore_not_found=True</i>	<p>Collects all result data and save them to the current active <code>result</code> folder</p> <p>A <code>prefix</code> will be automatically added to the file names.</p> <p>Currently the follow data will be downloaded to the local machine</p> <ul style="list-style-type: none">■ HTTP_Server.csv■ HTTP_Client.csv■ HTTP Client - Per URL.csv■ HTTP Server - Per URL.csv■ L2-3 Stats for Client Ports.csv■ L2-3 Stats for Server Ports.csv■ L2-3 Throughput Stats.csv■ Port CPU Statistics.csv <p>Extra files could be add by <code>more_file</code> which is a comma separated filename string</p> <p>When <code>ignore_not_found</code> is True, the keyword will not terminate even when the expected file is not found.</p>
Get Test Report	<i>self</i> , <i>prefix=</i>	Get the test report(PDF) and put it into the active result folder
Load Config	<i>self</i> , <i>config_name=</i>	<p>Loads the test traffic defined by <code>config_name</code></p> <p><code>file_path</code> is the path of the test file on the remote App server A path to a remote network drive could be use to load a config file on Renat server.</p>
Load Traffic	<i>self</i> , <i>file_path</i>	
Start Traffic	<i>self</i>	Starts the test traffic
Stop Traffic	<i>self</i>	<p>Stops the current running test</p> <p>Returns the elapsed time in seconds</p>

Altogether 7 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:27.



ixnet

Library scope: global
Named arguments: supported

Introduction

provides functions for IxNetwork

RENAT will connect to the App server and control the test ports. Test files and result will be inside the RENAT server.

In order to run RENAT test case with *IxLoad*, the *TCLServer* must be activated with *Administrator* privileges on the Ixia App server.

Notes: Ignore the *self* parameters when using those keywords.

Shortcuts

Add Port · **Add Quicktest** · **Apply Traffic** · **Change Frame Rate** · **Change Frame Rate Dynamic** · **Change Frame Size** · **Close** · **Collect All Data** · **Collect Data** · **Get All Test Result** · **Get Quicktest List** · **Get Quicktest Result** · **Get Quicktest Result Path** · **Get Test Report** · **Get Test Result** · **Load And Start Traffic** · **Load Config** · **Load Traffic** · **Loss From File** · **Ping** · **Reset Config** · **Run Quicktest** · **Set All Traffic Item** · **Set Bgp Items** · **Set Bgp Neighbor** · **Set Capture Port** · **Set Traffic Item** · **Should Be Pingable** · **Start Capture** · **Start Protocol** · **Start Traffic** · **Stop All Protocols** · **Stop And Save Capture** · **Stop Quicktest** · **Stop Traffic** · **Wait Until Connected**

Keywords

Keyword	Arguments	Documentation
Add Port	<i>self</i> , <i>force=True</i> , <i>time_out=2m</i> , <i>learn_time=2m</i>	<p>Add ports using the information from active local config</p> <ul style="list-style-type: none">■ <i>time_out</i> is the wait time until port is connected (default is 2m)■ <i>learn_time</i> is the time waiting for arp to be learned (default is 2m) <p>Sample of local config tester:</p> <pre>tester: device: ixnet03_8009 config: quicktest.ixncfg real_port: - chassis: 10.128.4.41 card: 4 port: 3 ip: 10.100.11.2 mask: 24 gw: 10.100.11.1 - chassis: 10.128.4.41 card: 4</pre> <p>port: 4</p> <pre>ip: 10.100.14.2 mask: 24 gw: 10.100.14.1</pre>
Add Quicktest	<i>self</i> , <i>name</i> , <i>test_type=rfc2544throughput</i> , <i>tx_mode=interleaved</i> , <i>clear_all=True</i>	<p>Create a new Quicktest with default value</p> <p>Type could be one of following: <i>rfc2544throughput</i>, <i>rfc2544frameLoss</i>, <i>rfc2544back2back</i>. Use Tester.Load Config to load a customized quicktest</p> <p>When <i>clear_all</i> is True, any existed quicktests will be cleared.</p> <p>Transmit mode <i>tx_mode</i> takes following values: <i>interleaved</i> (default) or <i>sequential</i>. The mode should be identical with the transmit mod of the ports.</p> <p>Notes: The keyword does not create necessary ports. It should be used with a existed configuration by Tester.Load Config or Tester.Add Port keyword.</p>
Apply Traffic	<i>self</i>	<p>Applies the current traffic configuration</p> <p>Note: This is a blocking command</p>
Change Frame Rate	<i>self</i> , <i>value</i> , <i>pattern=.*</i>	<p>Changes the frame rate</p> <p>Parameter:</p> <ul style="list-style-type: none">■ <i>value</i>: value to set. Depends on the current configuration, this could be <i>percent line rate</i> or <i>bit per second</i> etc.■ <i>traffic_pattern</i>: a regular expression to identify traffic item name, default is everything <i>.*</i>
Change Frame Rate Dynamic	<i>self</i> , <i>value</i> , <i>pattern=.*</i>	<p>Changes the traffic flow rate on-fly</p> <p>No need to stop the running traffic to change the rate</p> <p>Parameter:</p> <ul style="list-style-type: none">■ <i>value</i>: value to set. Depend on the current configuration, this could be <i>percent line rate</i> or <i>bit per second</i> etc.■ <i>pattern</i>: a regular expression to identify traffic item name, default is everything <i>.*</i>
Change Frame Size	<i>self</i> , <i>type</i> , <i>value</i> , <i>pattern=.*</i>	<p>Changes the frame size</p> <p>Parameter:</p>

		<ul style="list-style-type: none">■ <code>type</code>: could be <code>fixed size</code>, <code>increment_from</code>, <code>increment_step</code> or <code>increment_to</code>■ <code>value</code>: value to set■ <code>traffic_pattern</code>: a regular expression to identify traffic item name, default is everything <code>*</code>										
Close	<code>self</code>	Disconnects the current tester client										
Collect All Data	<code>self, prefix=stat_</code>	Deprecated. Use										
Collect Data	<code>self, view, prefix=stat_</code>	Deprecated. Use Get Test Result										
Get All Test Result	<code>self, prefix=stat_</code>	Collects all Ixia traffic data after traffic is stopped. Results are CSV files that are stored in <code>result</code> folder. The prefix <code>prefix</code> is appended to the original view name										
Get Quicktest List	<code>self</code>	Returns current loaded Quicktest list										
Get Quicktest Result	<code>self, test_index=-1, prefix=, enable_all=True</code>	Get the result.csv file from the latest Quicktests <code>test_index</code> is a index of the current Quicktest. <code>-1</code> means that last one.										
Get Quicktest Result Path	<code>self, test_index=-1</code>	Returns the path of the newest run of a Quicktest <code>test_index</code> is a index of the current Quicktest. <code>-1</code> means that last one.										
Get Test Report	<code>self, local_name=ixnet_report.pdf, enable_all=True</code>	Generates and get report of the current active test in PDF format <code>local_name</code> : name of the report on local machine. Default is <code>ixnet_report.pdf</code>										
Get Test Result	<code>self, view, prefix=stat_</code>	Collects traffic data of a <code>view</code> and export to a CSV file in <code>result</code> folder Currently, supported views are: Port Statistics, Global Protocol Statistics, BGP Aggregated Statistics, BGP Aggregated State Counts, OSPF Aggregated Statistics, OSPF Aggregated State Counts, OSPFv3 Aggregated Statistics, OSPFv3 Aggregated State Counts, L2-L3 Test Summary Statistics, Flow Statistics, Flow Detective, Data Plane Port Statistics, User Defined Statistics, Traffic Item Statistics Result were store as CSV files in <code>result</code> folder. If there is no valid data, view will be silently ignored The prefix <code>prefix</code> is appended to the view name for the CSV file.										
Load And Start Traffic	<code>self, wait_time1=10s, wait_time2=10s</code>	Combines Load Traffic and Start Traffic to one keyword.										
Load Config	<code>self, config_name=, wait_time=2m, wait_time2=2m, apply=True, protocol=True, force=True, tx_mode=interleaved</code>	loads traffic configuration, applies and start protocol if necessary. The config file name was defined in the <code>local.yaml</code> which is a Ixia Network configuration file and located in the <code>config</code> folder of the test. The keyword remap the vports to real port when data is specified in the local configuration file. For some reasons, the txMode is cleared when remapping happens. Use <code>tx_mode</code> to set the TxMode of the remapped ports. Parameters: <ul style="list-style-type: none">■ <code>apply</code>: applies traffic when <code>True</code> otherwise■ <code>protocol</code>: starts all protocols when <code>True</code> otherwise■ <code>force</code>: force to reclaim the ports when <code>True</code> otherwise■ <code>tx_mode</code>: <code>sequential</code> or <code>interleaved</code>(default)■ <code>wait_time</code>: wait time after applying protocols■ <code>wait_time2</code>: maximum wait time befor all ports become available. In common case, this is calculated automatically so user does not need to change this value. See Common for more details about the yaml configuration files.										
Load Traffic	<code>self, wait_time=2m, wait_time2=2m, apply=True, protocol=True, force=True, tx_mode=interleaved</code>											
Loss From File	<code>self, file_name=Flow_Statistics.csv, tx_frame_i=3, frame_delta_i=5, time1_i=23, time2_i=24</code>	Returns <code>packet loss</code> by milliseconds and delta frame. The calculation should be performed when traffic is stopped. The calculation supposed traffic is configured by frame per second										
Ping	<code>self, dst_ip, src_port_index=0, src_intf_index=0</code>	Ping from Ixia to <code>dst_ip</code> The keyword return the output string as it is. The return could be <div><pre>- Port <portName>: ping failed: port not assigned - Response received from <sourceIp>/unknown . Sequence Number <sequenceNumber> - Ping request to <destinationIp>/unknown ip failed: <GenericPingError>/<error>: <genericError>unknown reason - Error: Couldn't find any source interface for Send Ping to <destinationIp> on <portName> Id <id> - Error: Couldn't find any source IP for Send Ping to <destinationIp> on <portName> Id <id></pre></div> Parameters: <ul style="list-style-type: none">■ <code>src_port_index</code>: index of Ixia port (starts from 0)■ <code>src_intf_index</code>: index of interface insides the port (starts from 0) Examples: <table><tr><td>Tester.</td><td>Ping</td><td>1.1.1.1</td><td>0</td><td>0</td></tr><tr><td>Tester.</td><td>Ping</td><td>1.1.1.1</td><td></td><td></td></tr></table>	Tester.	Ping	1.1.1.1	0	0	Tester.	Ping	1.1.1.1		
Tester.	Ping	1.1.1.1	0	0								
Tester.	Ping	1.1.1.1										
Reset Config	<code>self</code>	Clears current config and creates new blank config										
Run Quicktest	<code>self, test_index=0,</code>	Runs the Quicktest and wait until it finishes										

	<code>wait_until_finish=True</code>	Warning: it could take a long time to finish a quicktest										
Set All Traffic Item	<code>self, enabled=True</code>	Enables/Disables all traffic items at once										
Set Bgp Items	<code>self, port_index, neighbor_index, route_range_index, is_enable</code>	Enables/Disables BGP entry by a set of port,neighbor,route_range index Parameters: <ul style="list-style-type: none">▪ <code>port_index</code>: index of the port▪ <code>neighbor_index</code>: index of the neighbor or *▪ <code>route_range_index</code>: index of the route range or *▪ <code>is_enable</code>: \${TRUE} or \${FALSE} Note Examples: <table><tr><td>Tester.Set BGP Items</td><td>0</td><td>*</td><td>*</td><td>\${FALSE}</td></tr><tr><td>Tester.Set BGP Items</td><td>0</td><td>*</td><td>*</td><td>\${TRUE}</td></tr></table>	Tester. Set BGP Items	0	*	*	\${FALSE}	Tester. Set BGP Items	0	*	*	\${TRUE}
Tester. Set BGP Items	0	*	*	\${FALSE}								
Tester. Set BGP Items	0	*	*	\${TRUE}								
Set Bgp Neighbor	<code>self, *indexes, **kwargs</code>	Enables/Disables BGP entry by neighbor index kwargs contains following parameters: <ul style="list-style-type: none">▪ <code>indexes</code>: is a list of index of BGP neighbor (index is started from zero)▪ <code>vport_index</code>: is the target vport index▪ <code>enabled</code>: TRUE or FALSE Examples: <table><tr><td>Tester.Set BGP Item</td><td>0</td><td>1</td><td>vport_index=0</td><td>enabled=\${FALSE}</td></tr><tr><td>Tester.Set BGP Item</td><td>0</td><td>1</td><td>vport_index=1</td><td>enabled=\${TRUE}</td></tr></table>	Tester. Set BGP Item	0	1	vport_index=0	enabled=\${FALSE}	Tester. Set BGP Item	0	1	vport_index=1	enabled=\${TRUE}
Tester. Set BGP Item	0	1	vport_index=0	enabled=\${FALSE}								
Tester. Set BGP Item	0	1	vport_index=1	enabled=\${TRUE}								
Set Capture Port	<code>self, data_mode=True, control_mode=True, port_index=0</code>	Capture packets for follow port <code>port_index</code> : is a index of current test port (start from 0) <code>data_mode</code> : capture data packets and save in <intf>_HW.cap file <code>control_mode</code> : capture controls packets and save in <intf>_SW.cap file Note: <code>control_mode</code> saves all control packets and <code>data_mode</code> only saves data packets. Note: <code>control_mode</code> saves all control packets and <code>data_mode</code> only saves data packet Examples: <table><tr><td>Tester.Set Capture Port</td><td>0</td><td></td><td></td></tr><tr><td>Tester.Set Capture Port</td><td>control_mode=\${TRUE}</td><td>0</td><td>1</td></tr></table>	Tester. Set Capture Port	0			Tester. Set Capture Port	control_mode=\${TRUE}	0	1		
Tester. Set Capture Port	0											
Tester. Set Capture Port	control_mode=\${TRUE}	0	1									
Set Traffic Item	<code>self, *items, **kwargs</code>	Enables/Disables some traffic items <code>items</code> Parameters: <ul style="list-style-type: none">▪ <code>items</code>: a list of Ixia traffic item name▪ <code>enabled</code>: False or True ,the mode to set traffic item to, default is <code>True</code> (enabled) Note: traffic item could be specified by ::<num> format. In this case the <code>num</code> is the order of traffic item count from zero. Returns <code>True</code> if all items are set coordinately or otherwise Examples: <table><tr><td>Set Traffic Item</td><td>Traffic Item 1</td><td>Traffic Item 2</td></tr><tr><td>Set Traffic Item</td><td>@{item_list}</td><td></td></tr><tr><td>Set Traffic Item</td><td>Traffic Item 1</td><td>enabled = \${FALSE}</td></tr></table>	Set Traffic Item	Traffic Item 1	Traffic Item 2	Set Traffic Item	@{item_list}		Set Traffic Item	Traffic Item 1	enabled = \${FALSE}	
Set Traffic Item	Traffic Item 1	Traffic Item 2										
Set Traffic Item	@{item_list}											
Set Traffic Item	Traffic Item 1	enabled = \${FALSE}										
Should Be Pingable	<code>self, dst_ip, src_port_index=0, src_intf_index=0</code>	Ping from Ixia and raise an error if ping fails The keyword return <code>True</code> if succeeds										
Start Capture	<code>self, wait_time=30s</code>	Start packet capture Target ports are set by the configuration file or by [Set Capture] keyword										
Start Protocol	<code>self, wait_time=1m</code>	Starts all protocols and wait for <code>wait_time</code> Default <code>wait_time</code> is 1 minute. Make sure <code>wait_time</code> is big engough to start all protocols.										
Start Traffic	<code>self, wait_time=30s</code>	Starts the current traffic settiing and wait for <code>wait_time</code> . Note: This is a asynchronus action. After called, the keyword finishes immediatly but it will take a while before traffic starts By default the keyword will wait for 30 seconds.										
Stop All Protocols	<code>self, wait_time=30s</code>	Stop all running protocols										
Stop And Save Capture	<code>self, prefix=, wait_until_finish=True, monitor_interval=5s</code>	Stop current capture and save the results to folder specified by <code>path</code> Captured files will be saved in current <code>result</code> folder with <code>prefix</code> appended in their names. Examples: <table><tr><td>Tester.Start Capture</td><td></td></tr><tr><td>Sleep</td><td>10s</td></tr><tr><td>Tester.Stop And Save Capture</td><td>\${RESULT_FOLDER}/capture.zip</td></tr></table>	Tester. Start Capture		Sleep	10s	Tester. Stop And Save Capture	\${RESULT_FOLDER}/capture.zip				
Tester. Start Capture												
Sleep	10s											
Tester. Stop And Save Capture	\${RESULT_FOLDER}/capture.zip											
Stop Quicktest	<code>self, test_index=0</code>	Stops a running test										

Stop Traffic	<i>self, stop_protocol=False, wait_time=10s</i>	Stops the current traffic and wait for <code>wait_time</code> Parameters: <ul style="list-style-type: none"> ▪ <code>stop_protocol</code>: if <code>True</code> also stops all running protocols ▪ <code>wait_time</code>: time to wait after apply the command
Wait Until Connected	<i>self, timeout_str=5m</i>	Waits until ports become enabled and connected

Altogether 36 keywords.

Generated by [Libdoc](#) on 2018-04-12 19:20:26.

