

Router

Library version: RENAT 0.1.8
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>									

