

ixnet

Library scope: global
Named arguments: supported

Introduction

provides functions for IxNetwork

To use IxNetwork module, a IxNetwork TCL server should be started properly.

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 Composer Result** · **Get Test Report** · **Get Test Result** · **Load And Start Traffic** · **Load Config** · **Load Traffic** · **Loss From File** · **Ping** · **Regenerate** · **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 Test Composer** · **Start Traffic** · **Stop All Protocols** · **Stop And Save Capture** · **Stop Quicktest** · **Stop Test Composer** · **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"><code>time_out</code> is the wait time until port is connected (default is 2m)<code>learn_time</code> 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: <code>rfc2544throughput</code>, <code>rfc2544frameLoss</code>, <code>rfc2544back2back</code>. Use Tester.Load Config to load a customized quicktest</p> <p>When <code>clear_all</code> is True, any existed quicktests will be cleared.</p> <p>Transmit mode <code>tx_mode</code> takes following values: <code>interleaved</code> (default) or <code>sequential</code>. 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> , <i>refresh=True</i>	<p>Applies the current traffic configuration</p> <p><code>refresh</code>: Refreshed the learned information before apply the traffic or not 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"><code>value</code>: value to set. Depends on the current configuration, this could be <code>percent line rate</code> or <code>bit per second</code> etc.<code>traffic_pattern</code>: a regular expression to identify traffic item name, default is everything <code>.*</code>
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"><code>value</code>: value to set. Depend on the current configuration, this could be <code>percent line rate</code> or <code>bit per second</code> etc.<code>pattern</code>: a regular expression to identify traffic item name, default is everything <code>.*</code>
Change Frame Size	<i>self</i> , <i>type</i> , <i>value</i> , <i>pattern=.*</i>	<p>Changes the frame size</p>

Size		Parameter: <ul style="list-style-type: none"> type: could be fixed size, increment_from, increment_step or increment_to value: value to set traffic_pattern: a regular expression to identify traffic item name, default is everything .*
Close	self	Disconnects the current tester client
Collect All Data	self, prefix=stat_	Deprecated. Use
Collect Data	self, view, prefix=stat_	Deprecated. Use Get Test Result
Get All Test Result	self, prefix=stat_	Collects all Ixia traffic data after traffic is stopped. Results are CSV files that are stored in result folder. The prefix prefix is appended to the original view name
Get Quicktest List	self	Returns current loaded Quicktest list
Get Quicktest Result	self, test_index=-1, prefix=, enable_all=True	Get the result.csv file from the latest Quicktests test_index is a index of the current Quicktest. -1 means that last one.
Get Quicktest Result Path	self, test_index=-1	Returns the path of the newest run of a Quicktest test_index is a index of the current Quicktest. -1 means that last one.
Get Test Composer Result	self, result_file=composer.log	Get the result of test composer script
Get Test Report	self, local_name=ixnet_report.pdf, enable_all=True	Generates and get report of the current active test in PDF format local_name: name of the report on local machine. Default is ixnet_report.pdf
Get Test Result	self, view, prefix=stat_	Collects traffic data of a view and export to a CSV file in result 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 result folder. If there is no valid data, view will be silently ignored The prefix prefix is appended to the view name for the CSV file.
Load And Start Traffic	self, wait_time1=10s, wait_time2=10s	Combines Load Traffic and Start Traffic to one keyword.
Load Config	self, config_name=, wait_time=2m, wait_time2=2m, apply=True, protocol=True, force=True, wait_time3=30s	loads traffic configuration, applies and start protocol if necessary. The config file name was defined in the local.yaml which is a Ixia Network configuration file and located in the config 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 tx_mode to set the TxMode of the remapped ports. Parameters: <ul style="list-style-type: none"> apply: applies traffic when True otherwise protocol: starts all protocols when True otherwise force: force to reclaim the ports when True otherwise wait_time: wait time after applying protocols wait_time2: maximum wait time befor all ports become available. In common case, this is calculated automatically so user does not need to change this value. wait_time3: default waiting time after config file is loaded (30s) More information about ports could be define in real_port section like this: <pre># tester information tester: tester: device: ixnet03_8009 config: bgp.ixncfg real-port: - chassis: 10.128.4.41 card: 4 port: 7 media: fiber tx_mode: interleaved</pre> Configurable port parameters ares: <ul style="list-style-type: none"> tx_mode: sequential or interleaved(default) media : copper or fiber (Note: no default value) See Common for more details about the yaml configuration files.
Load Traffic	self, wait_time=2m, wait_time2=2m, apply=True, protocol=True, force=True, tx_mode=interleaved	
Loss From File	self,	Returns packet loss by milliseconds and delta frame.

	<code>file_name=Flow_Statistics.csv,</code> <code>tx_frame=4, frame_delta=6,</code> <code>time1=21, time2=22</code>	<code>tx_frame, frame_delta,time1,time2</code> is indexes of <i>TX frame, Frame Delta, Start Time, Stop Time</i> in the csv (counted from zero) 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,</code> <code>src_intf_index=0</code>	<p>Ping from Ixia to <code>dst_ip</code></p> <p>The keyword return the output string as it is. The return could be</p> <div><ul style="list-style-type: none">- 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></div> <p>Parameters:</p> <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) <p>Examples:</p> <table><tr><td>Tester.Ping</td><td>1.1.1.1</td><td>0</td><td>0</td></tr><tr><td>Tester.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											
Regenerate	<code>self</code>	Regenerates all flow of current traffic items										
Reset Config	<code>self</code>	Clears current config and creates new blank config										
Run Quicktest	<code>self, test_index=0,</code> <code>wait_until_finish=True</code>	Runs the Quicktest and wait until it finishes 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,</code> <code>neighbor_index,</code> <code>route_range_index, is_enable</code>	<p>Enables/Disables BGP entry by a set of port,neighbor,route_range index</p> <p>Parameters:</p> <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} <p>Note</p> <p>Examples:</p> <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>	<p>Enables/Disables BGP entry by neighbor index</p> <p><code>kwargs</code> contains following parameters:</p> <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 <p>Examples:</p> <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,</code> <code>control_mode=True,</code> <code>port_index=0</code>	<p>Capture packets for follow port</p> <p><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</p> <p>Note: <code>control_mode</code> saves all control packets and <code>data_mode</code> only saves data packets.</p> <p>Note: <code>control_mode</code> saves all control packets and <code>data_mode</code> only saves data packet</p> <p>Examples:</p> <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>	<p>Enables/Disables some traffic items <code>items</code></p> <p>Parameters:</p> <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) <p>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.</p> <p>Returns <code>True</code> if all items are set coordinately or otherwise</p> <p>Examples:</p> <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	<code>self, dst_ip, src_port_index=0</code>	Ping from Ixia to destination ip configuration follow										

Should Be Pingable	self, dst_ip, src_port_index=0, src_intf_index=0	Ping from Ixia and raise an error if ping fails The keyword return <i>True</i> if succeeds								
Start Capture	self, wait_time=30s	Start packet capture Target ports are set by the configuration file or by [Set Capture] keyword								
Start Protocol	self, wait_time=1m	Starts all protocols and wait for <i>wait_time</i> Default <i>wait_time</i> is 1 minute. Make sure <i>wait_time</i> is big engouh to start all protocols.								
Start Test Composer	self, script_name=Main_Procedure, run_num=1, wait_for_test=True, parameter=, wait=10s	Run a test composer script. The test composer script should be included in an Ixia Network configuration file and loaded properly with Load Config Parameters: <ul style="list-style-type: none">script_name is the name of the script to run. Default value is <i>Main_Procedure</i>.wait_for_test: if <i>\$(TRUE)</i> then wait until the script finishes.parameter: parameter that is passed to the script. Parameter could be in 2 formats: <i>{{VAR1 VALUE1} {VAR2 VALUE2}}</i> or simply as <i>VALUE1 VALUE2</i>. The script must prepare <i>VAR1</i> and <i>VAR2</i> properly by <i>Test parameter</i> . See Ixia Network anout composer script for more details. <ul style="list-style-type: none">wait: wait time before go to next keyword Examples: <table><tr><td>Tester.Start Test Composer</td><td>parameter=XXX YYY</td></tr><tr><td>Tester.Get Test Composer Result</td><td>result_file=script1.log</td></tr><tr><td>Tester.Start Test Composer</td><td>parameter={{VAR1 AAA} {VAR2 BBB}}</td></tr><tr><td>Tester.Get Test Composer Result</td><td>result_file=script1.log</td></tr></table>	Tester. Start Test Composer	parameter=XXX YYY	Tester. Get Test Composer Result	result_file=script1.log	Tester. Start Test Composer	parameter={{VAR1 AAA} {VAR2 BBB}}	Tester. Get Test Composer Result	result_file=script1.log
Tester. Start Test Composer	parameter=XXX YYY									
Tester. Get Test Composer Result	result_file=script1.log									
Tester. Start Test Composer	parameter={{VAR1 AAA} {VAR2 BBB}}									
Tester. Get Test Composer Result	result_file=script1.log									
Start Traffic	self, wait_time=30s	Starts the current traffic settiing and wait for <i>wait_time</i> . Note: This is a asynchroneus 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	self, wait_time=30s	Stop all running protocols								
Stop And Save Capture	self, prefix=, wait_until_finish=True, monitor_interval=5s	Stop current capture and save the results to folder specified by <i>path</i> Captured files will be saved in current <i>result</i> folder with <i>prefix</i> 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	self, test_index=0	Stops a running test								
Stop Test Composer	self, wait=10s	Stop a running composer Do nothing when a test composer has already stopped or no composer has been prepared.								
Stop Traffic	self, stop_protocol=False, wait_time=10s	Stops the current traffic and wait for <i>wait_time</i> Parameters: <ul style="list-style-type: none">stop_protocol: if <i>True</i> also stops all running protocolswait_time: time to wait after apply the command								
Wait Until Connected	self, timeout_str=5m	Waits until ports become enabled and connected								

