

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** · **Csv Logging** · **Csv Snapshot** · **Get All Test Result** · **Get All Views All Csv Logs** · **Get All Views Csv Log** · **Get Csv Log** · **Get Quicktest List** · **Get Quicktest Result** · **Get Quicktest Result Path** · **Get Test Composer Result** · **Get Test Report** · **Get Test Result** · **Get View All Csv Logs** · **Get View Csv Log** · **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 <code>real-port</code> 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> , <i>flow_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>pattern</code>: a regular expression to identify traffic item name, default is everything <code>.*</code><code>flow_pattern</code>: a regular expression to identify flow group inside the item
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>

		<ul style="list-style-type: none">▪ <code>pattern</code>: a regular expression to identify traffic item name, default is everything <code>.</code>										
Change Frame Size	<code>self, type, value, pattern=.*, flow_pattern=.*</code>	<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>pattern</code>: a regular expression to identify traffic item name, default is everything <code>.</code>▪ <code>flow_pattern</code>: a regular expression to identify flow group inside the item										
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										
Csv Logging	<code>self, enabled=True, *views</code>	<p>Toggles enable/disable CSV login for a view</p> <p>Parameters:</p> <ul style="list-style-type: none">▪ <code>views</code>: is a list of views. <code>None</code> means all views. <p>Result files will have format <View name>.index.csv, when index is automatically increased everytime the view is disable and re-enable again in the same test.</p> <p>Samples:</p> <table><tr><td>Tester.CSV Logging</td><td>\$(TRUE)</td><td>Flow Statistics</td></tr><tr><td>Sleep</td><td>10s</td><td></td></tr><tr><td>Tester.CSV Logging</td><td>\${FALSE}</td><td>Flow Statistics</td></tr></table> <p>Note: Long time enable for CSV login could returns in very big file</p>	Tester. CSV Logging	\$(TRUE)	Flow Statistics	Sleep	10s		Tester. CSV Logging	\${FALSE}	Flow Statistics	
Tester. CSV Logging	\$(TRUE)	Flow Statistics										
Sleep	10s											
Tester. CSV Logging	\${FALSE}	Flow Statistics										
Csv Snapshot	<code>self, prefix=snapshot_, *views</code>	<p>Get current CSV snapshot</p> <p>Parameters:</p> <ul style="list-style-type: none">▪ <code>prefix</code>: prefix that be added to the filename. Default is <code>snapshot_</code>▪ <code>views</code>: list of target views (eg: <code>Port Statistics</code>, <code>Flow Statistics</code> ...). If <code>view</code> is <code>None</code>, all current available views will be target <p>Samples:</p> <table><tr><td>Tester.CSV Snapshot</td><td>snapshot03_</td><td># collect all views</td><td></td><td></td></tr><tr><td>Tester.CSV Snapshot</td><td>snapshot03_</td><td>Port Statistics</td><td>Flow Statistics</td><td># collect specific views</td></tr></table> <p>Note: the name of result file will be modified so <code>space</code> will be replaced by <code>underbar</code>.</p> <p>Depending on the traffic status, the available views could be varied. For example, the view <code>Flow Statistics</code> is not available when there is no traffic.</p>	Tester. CSV Snapshot	snapshot03_	# collect all views			Tester. CSV Snapshot	snapshot03_	Port Statistics	Flow Statistics	# collect specific views
Tester. CSV Snapshot	snapshot03_	# collect all views										
Tester. CSV Snapshot	snapshot03_	Port Statistics	Flow Statistics	# collect specific views								
Get All Test Result	<code>self, prefix=stat_</code>	<p>Collects all Ixia traffic data after traffic is stopped.</p> <p>Results are CSV files that are stored in <code>result</code> folder. The prefix <code>prefix</code> is appended to the original view name</p>										
Get All Views All Csv Logs	<code>self, prefix=</code>	Gets all CSV logs for all available views										
Get All Views Csv Log	<code>self, prefix=</code>	Gets the newest CSV log of all available views										
Get Csv Log	<code>self, prefix=, index=0, *views</code>	<p>Gets all CSV log for a specific views or all from current test folder</p> <p>Parameters:</p> <ul style="list-style-type: none">▪ <code>views</code> is a list of views. <code>None</code> is all views.▪ <code>prefix</code> will be appended automatically to the beginning of the result▪ <code>range</code>: number of files from the newest data. <code>0</code> for only 1 newest and <code>-1</code> for all files. <p>Samples:</p> <table><tr><td>Tester.Get CSV Log</td><td>all_</td><td>0</td><td># get the newest CSV logging data for all views</td><td></td></tr><tr><td>Tester.Get CSV Log</td><td>single_</td><td>-1</td><td>Flow Statistics</td><td># get all CSV logging data for one view</td></tr></table>	Tester. Get CSV Log	all_	0	# get the newest CSV logging data for all views		Tester. Get CSV Log	single_	-1	Flow Statistics	# get all CSV logging data for one view
Tester. Get CSV Log	all_	0	# get the newest CSV logging data for all views									
Tester. Get CSV Log	single_	-1	Flow Statistics	# get all CSV logging data for one view								
Get Quicktest List	<code>self</code>	Returns current loaded Quicktest list										
Get Quicktest Result	<code>self, test_index=-1, prefix=, enable_all=True</code>	<p>Get the result.csv file from the latest Quicktests</p> <p><code>test_index</code> is a index of the current Quicktest. <code>-1</code> means that last one.</p>										
Get Quicktest Result Path	<code>self, test_index=-1</code>	<p>Returns the path of the newest run of a Quicktest</p> <p><code>test_index</code> is a index of the current Quicktest. <code>-1</code> means that last one.</p>										
Get Test Composer Result	<code>self, result_file=composer.log</code>	Get the result of test composer script										
Get Test Report	<code>self, local_name=ixnet_report.pdf, enable_all=True</code>	<p>Generates and get report of the current active test in PDF format</p> <p><code>local_name</code>: name of the report on local machine. Default is <code>ixnet_report.pdf</code></p>										
Get Test Result	<code>self, view, prefix=stat_</code>	<p>Collects traffic data of a <code>view</code> and export to a CSV file in <code>result</code> folder</p> <p>Currently, supported views are:</p> <p><code>Port Statistics</code>, <code>Global Protocol Statistics</code>, <code>BGP Aggregated Statistics</code>, <code>BGP Aggregated State Counts</code>, <code>OSPF</code></p>										

		<div>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</div> <div>Result were store as CSV files in <code>result</code> folder. If there is no valid data, view will be silently ignored</div> <div>The prefix <code>prefix</code> is appended to the view name for the CSV file.</div> <div>Note: the name of the result files are modified so that <i>space</i> will become <i>underbar</i>, <i>hyphen</i> will be deleted.</div>										
Get View All Csv Logs	<code>self, view, prefix=</code>	Gets the newest CSV log file of ALL available views										
Get View Csv Log	<code>self, view, prefix=</code>	Gets the newest CSV log file of the specific view										
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, wait_time3=30s</code>	<div>loads traffic configuration, applies and start protocol if necessary.</div> <div>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.</div> <div>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.</div> <div>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>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.▪ <code>wait_time3</code>: default waiting time after config file is loaded (30s)</div> <div>More information about ports could be define in <code>real_port</code> section like this:</div> <div><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></div> <div>Configurable port parameters ares:<ul style="list-style-type: none">▪ <code>tx_mode</code>: <i>sequential</i> or <i>interleaved</i>(default)▪ <code>media</code> : <i>copper</i> or <i>fiber</i> (Note: no default value)</div> <div>See Common for more details about the yaml configuration files.</div>										
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, index=0</code>	<div>Returns <code>packet loss</code> by miliseconds and <i>delta frame</i>.</div> <div>Parameters:<ul style="list-style-type: none">▪ <code>file_name</code>: flow information (csv format). Default is <code>Flow_Statistics.csv</code>▪ <code>index</code>: row index of the result(counted from zero)</div> <div>Samples:</div> <div><table><tr><td><code>\$(LOSS)</code></td><td><code>\$(DELTA)=</code></td><td>Tester.Loss From File</td><td>Flow_Statistics.csv</td><td></td></tr><tr><td><code>\$(LOSS)</code></td><td><code>\$(DELTA)=</code></td><td>Tester.Loss From File</td><td>Flow_Statistics.csv</td><td>index=1</td></tr></table></div> <div>Note: The calculation should be performed when traffic is stopped. The calculation supposed traffic is configured by frame per second.</div>	<code>\$(LOSS)</code>	<code>\$(DELTA)=</code>	Tester. Loss From File	Flow_Statistics.csv		<code>\$(LOSS)</code>	<code>\$(DELTA)=</code>	Tester. Loss From File	Flow_Statistics.csv	index=1
<code>\$(LOSS)</code>	<code>\$(DELTA)=</code>	Tester. Loss From File	Flow_Statistics.csv									
<code>\$(LOSS)</code>	<code>\$(DELTA)=</code>	Tester. Loss From File	Flow_Statistics.csv	index=1								
Ping	<code>self, dst_ip, src_port_index=0, src_intf_index=0</code>	<div>Ping from Ixia to <code>dst_ip</code></div> <div>The keyword return the output string as it is. The return could be</div> <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> <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)</div> <div>Examples:</div> <div><table><tr><td>Tester.Ping</td><td>1.1.1.1</td><td>0</td><td>0</td></tr></table></div>	Tester. Ping	1.1.1.1	0	0						
Tester. Ping	1.1.1.1	0	0									

		Tester. Ping 1.1.1.1
Regenerate	<i>self</i>	Regenerates all flow of current traffic items
Reset Config	<i>self</i>	Clears current config and creates new blank config
Run Quicktest	<i>self, test_index=0, wait_until_finish=True</i>	Runs the Quicktest and wait until it finishes Warning: it could take a long time to finish a quicktest
Set All Traffic Item	<i>self, enabled=True</i>	Enables/Disables all traffic items at once
Set Bgp Items	<i>self, port_index, neighbor_index, route_range_index, is_enable</i>	Enables/Disables BGP entry by a set of port,neighbor,route_range index Parameters: <ul style="list-style-type: none"> port_index: index of the port neighbor_index: index of the neighbor or * route_range_index: index of the route range or * is_enable: \${TRUE} or \${FALSE} Note Examples: Tester. Set BGP Items 0 * * \${FALSE} Tester. Set BGP Items 0 * * \${TRUE}
Set Bgp Neighbor	<i>self, *indexes, **kwargs</i>	Enables/Disables BGP entry by neighbor index kwargs contains following parameters: <ul style="list-style-type: none"> indexes: is a list of index of BGP neighbor (index is started from zero) vport_index: is the target vport index enabled: TRUE or FALSE Examples: 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	<i>self, data_mode=True, control_mode=True, port_index=0</i>	Capture packets for follow port port_index: is a index of current test port (start from 0) data_mode: capture data packets and save in <intf>_HW.cap file control_mode: capture controls packets and save in <intf>_SW.cap file Note: control_mode saves all control packets and data_mode only saves data packets. Note: control_mode saves all control packets and data_mode only saves data packet Examples: Tester. Set Capture Port 0 Tester. Set Capture Port control_mode=\${TRUE} 0 1
Set Traffic Item	<i>self, *items, **kwargs</i>	Enables/Disables some traffic items items Parameters: <ul style="list-style-type: none"> items: a list of Ixia traffic item name enabled: False or True ,the mode to set traffic item to, default is True (enabled) Note: traffic item could be specified by ::<num> format. In this case the num is the order of traffic item count from zero. Returns True if all items are set coordinately or otherwise Examples: 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	<i>self, dst_ip, src_port_index=0, src_intf_index=0</i>	Ping from Ixia and raise an error if ping fails The keyword return True if succeeds
Start Capture	<i>self, wait_time=30s</i>	Start packet capture Target ports are set by the configuration file or by [Set Capture] keyword
Start Protocol	<i>self, wait_time=1m</i>	Starts all protocols and wait for wait_time Default wait_time is 1 minute. Make sure wait_time is big enough to start all protocols.
Start Test Composer	<i>self, script_name=Main_Procedure, run_num=1, wait_for_test=True, parameter=, wait=10s</i>	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 Main_Procedure. wait_for_test: if \${TRUE} then wait until the script finishes. parameter: parameter that is passed to the script. Parameter could be in 2 formats: {{VAR1 VALUE1}} {VAR2 VALUE2}} or simply as VALUE1 VALUE2. The script must prepare VAR1 and VAR2 properly by Test parameter. See Ixia Network anout composer script for more details.

		<ul style="list-style-type: none">wait: wait time before go to next keyword <p>Examples:</p> <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	<p>Starts the current traffic settiing and wait for wait_time .</p> <p>Note: This is a asynchronous action. After called, the keyword finishes immediatly but it will take a while before traffic starts</p> <p>By default the keyword will wait for 30 seconds.</p>								
Stop All Protocols	self, wait_time=30s	Stop all running protocols								
Stop And Save Capture	self, prefix=, wait_until_finish=True, monitor_interval=5s	<p>Stop current capture and save the results to folder specified by path</p> <p>Captured files will be saved in current result folder with prefix appended in their names.</p> <p>Examples:</p> <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	<p>Stop a running composer</p> <p>Do nothing when a test composer has already stopped or no composer has been prepared.</p>								
Stop Traffic	self, stop_protocol=False, wait_time=10s	<p>Stops the current traffic and wait for wait_time</p> <p>Parameters:</p> <ul style="list-style-type: none">stop_protocol: if True 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								

Altogether 47 keywords.

Generated by [Libdoc](#) on 2019-01-31 10:22:40.

