# IXNETWORK-CLASSIC QUICK REFERENCE GUIDE



# **ÍXIA** A Keysight Business



# **PREFACE**

1.	Ove	rview	2
2.	Con	figure OSPFv2 through GUI	2
2	2.1	Add Chassis and Reserve Ports	3
2	2	Configure Interfaces	4
2	.3	Emulate OSPFv2 Protocol	5
2	.4	Configure OSPFv2	5
2	.5	Create OSPFv2 Route Ranges	6
2	.6	Start OSPFv2 Protocol	6
2	.7	Check Learned LSAs	7
2	.8	Configure Traffic	8
2	.9	Add Endpoints To Traffic	8
2	.10	Edit Packet	9
2	2.11	Setup Flow Group	9
2	.12	Setup Frame Size1	0
2	.13	Setup Line Rate1	0
2	.14	Setup Flow Tracking1	.1
2	.15	Preview Flow Groups1	2
2	.16	Validate Traffic Items1	2
2	.17	Apply Traffic, Start Traffic and Statistics View1	.3
3.	Con	figure OSPFv2 through Automation1	.4
З	8.1	Initialize Environment1	4
З	.2	Add Chassis and Reserve Ports1	.4
З	.3	Configure Ports1	.5
З	.4	Create OSPFv21	6
3	.5	Create OSPFv2 Route Ranges1	.7
З	.6	Start Protocols1	8
З	3.7	Check Learned LSAs1	9
З	8.8	Configure Traffic2	1
З	.9	Start and Stop Traffic2	2
3	.10	Get Statistics	.3
4.	Oth	er Utilities2	.4
4	.1	IxNetwork API Documentation Browser2	.4
4	.2	Script Gen2	.5
4	.3	F1 Option	6
5.	To k	Know More on IxNetwork Classic 2	:7
6.	Sup	port	:7



#### 1. Overview

IxNetwork is a comprehensive network infrastructure performance testing solution. It scales to handle most powerful devices and largest networks, from routing and switching to data center ethernet and software defined networking. IxNetwork has to protocol emulation frameworks for simulating networks. Both are capable of emulating a wide array of protocol activities running on simulated network devices. The next generation protocol framework (NGPF) is the latest generation of protocol emulation engine and supports the widest array of protocols with the highest scalability in the industry. The previous protocol framework (Classic) continues to be supported in IxNetwork, primarily to support existing customers with legacy configurations created prior to NGPF introduction. Ixia recommends using NGPF for all new network infrastructure test projects.

IxNetwork Classic protocol framework provides the following features and benefits:

- > IxNetwork Classic offers the performance and functionality testing of Routers/Switches.
- Provides a powerful, yet easy-to-use, graphical user interface (GUI) that you can use to configure and run complex tests.
- Offers the flexibility to customize the application to meet a wide range of requirements for testing complex network topologies, consisting of thousands of routing or switching devices.
- Emulate millions of routes and reachable hosts within the topology. Provides with the ability to customize millions of traffic flows to stress the data plane performance.
- Create sophisticated configurations using powerful wizards and grid controls in GUI.
- Capable of reporting comprehensive protocol status and detailed per-flow traffic performance metrics.

#### 2. <u>Configure OSPFv2 through GUI</u>

This section visualizes the scenario to configure OSPFv2 protocol through GUI and verify IPv4 traffic. Section includes the following tasks.

- Add chassis and reserve ports.
- Configure OSPFv2 protocol on interfaces. Disable 'Discard Learned LSAs' on port 1/1/7 for the peer (1/1/8) to learn LSAs. Create route ranges to advertise prefixes to the peer.
- Start OSPFv2 protocol. Once OSPFv2 protocol sessions UP, check for learned LSAs in the peer (1/1/8) side. Please refer Fig 7.1.
- > Create IPv4 unicast traffic  $(1/1/7 \rightarrow 1/1/8)$ . Start traffic and wait for some time for the traffic to flow. Stop the traffic. Verify Tx == Rx packets count from Traffic Item Statistics section.



# 2.1 Add Chassis and Reserve Ports

#### Add Chassis and reserve ports.

File	Home Automation Results / Reports Views		
Protocols	Is All QuickTest	e Test	Clear All
*	Traffic • • Ports • Protocols • Traffic • QuickTests • Manager	<ul> <li>Options</li> </ul>	Statistics 🔻
_ €	Add Ports Build		Statistics
÷	Add Offline Ports		
	Recent Port Resources		

Fig 1.1 Add Ports

Chassis 🕂 Add Chassis 🚺 🛃 📰 🛛	lore	Ports in configuration
Click Add Recently Used Chassis	Chassis	S Name
<ul> <li><u>10.39.64.132</u></li> <li><u>10.214.100.93</u></li> <li>Connect all checked</li> </ul>	<ul> <li>Chassis Name or IP 10.39.64.132 For multiple chassis us</li> <li>Recently Used Chassis</li> <li>10.39.64.132</li> <li>10.214.100.93</li> </ul>	se comine perated list.
	3 OK Canc	Click OK

Fig 1.2: Add Chassis to reserve ports

#### IXIA A Keysight Business

Nw Port Selection								-		Х
Chassis  Add Chassis	🗙 📚 Đ 📑 More 🗸	All ports 🔻		Ports in co	nfiguration	槹 Add Offlin	e Ports	6	All port	s 🔻
Chassis/Card/Port	Туре	Owner		S	Name	Chassis/Car	Resour	Туре	Owner	A
✓	ixos 8.40.1400.5 ea, IxNetwork 8.44 16 PORT 10/100/1000 LSM XMVD 10/100/1000 Base T	0 C								
<ul> <li>Port 01</li> <li>Port 02</li> <li>Port 03</li> </ul>	Click Select Ports		Add ports	$\mathbb{Q}_{2}$						
<ul> <li>Port 04</li> <li>Port 05</li> <li>Port 05</li> </ul>	10/100/1000 Base T 10/100/1000 Base T 100/1000 Base T		>>>> Assign to		Click	Add Ports				
<ul> <li>Port 07</li> <li>Port 08</li> </ul>	10/100/1000 Base T 10/100/1000 Base T		remaining							
Port 09 Port 10 Port 11	10/100/1000 Base T 10/100/1000 Base T 10/100/1000 Base T	lyNetwork/5	Assign to							
Port 12 Port 13	10/100/1000 Base T 10/100/1000 Base T	IxNetwork/5								
₩ Port 14 ₩ Port 15	10/100/1000 Base T 10/100/1000 Base T	lxNetwork/5 lxNetwork/4	Unassign selected							
🥥 Port 16	10/100/1000 Base I			Cliel	k OK					
•		•				3				
							ОК	ancel	He	lp

Fig 1.3: Select Ports and Reserve Ports

# 2.2 Configure Interfaces

		≰ 😂 🚮 🕨 🤁 Protocols 🕨 🤁 Protocol Interfaces 🕨							
	Connected Interfaces Unconnected Interfaces GRE Turnels Discovered Neightons Interface Addresses DHCPv4 Discovered Information DHCPv6 Discovered Information	n Rate Control Parameters							
■ Scenario	ARP on Link Up 🛛 Send Single ARP per Gateway 🖓 NS on Link Up 🖓 Send	I Single NS per Gateway	Filter By Unresolved Inte						
👻 阙 Ports	Port Description Port Lin Interface Description	Enable IPv4 Addres	s IPv4 Gateway						
Chaudia	1 1/1/7 - 10/100/1000 Base T  Connected - ProtocolInterface - 100:01 - 1	☑ 192.168.1.	1 24 192.168.1.2						
	2 1/1/8 - 10/100/1000 Base Connected - ProtocolInterface - 100:02 - 2	192.168.1.	2 24 192.168.1.1						
Protocols     Protocol Inter     Static     Traffic	Right Click and Add IPv4     3       Click Protocol Interfaces								

Fig 2.1: Configure Port's attributes



# 2.3 Emulate OSPFv2 Protocol

Enable OSPFv2 on interfaces.



Fig 3.1: Emulate OSPFv2





# 2.4 Configure OSPFv2



Fig 4.1: Map connected interfaces to configure OSPFv2 attributes



# 2.5 Create OSPFv2 Route Ranges



Fig 5.1: Configure number of route ranges required



Fig 5.2: Create Ipv4 prefix pool on selected port 1/1/7

# 2.6 Start OSPFv2 Protocol

File Hor	me Au	Itomation	Results /	Reports	Views	Configura	ition			
Protocols OSPF	OSPF Actions •	Traffic Group ID	Add Protocols	H Add	l Routers nove Router(s)	Grid Operatio	Filter Selected Por	<b>rts</b> ed Ports		
	ctions		Build		Edit		Grid			
0 Overview	~	Start OS Proto	PFv2 col	Protocols	Ber LSA Groups   User LSAs					
📭 🗧 Scenario			Port	Enable	Router ID	Number	Number of RouteRanges	Numb	er of UserLSA Groups	LSA Re
		1	1/1/7		190.29.0.1	1	0			0
Ports     Chass     Chass     Protocols     Protocols     Protoc     Pro	is :ol Interfaces 1/7 1 RID - 190.2	29.0.1	0.02		100.00.0.1			1		







Fig 6.2 OSPFv2 is in Running state and sessions are UP

# 2.7 Check Learned LSAs

Disable "Discard Learned LSAs" to view learned LSAs on 1/1/8. Please refer Fig 7.1

Control         Degree         Proce Nummer         Number         Number of RouteRanges         Number of UserLSA Groups         LSA Refresh ti         Discard Le           1         I         190.290.1         1         1         0         5         1,800         0	arned LSAs
Scenario	arned LSAs
	-
2 <sup>2</sup> 190.30.0.1 1 1 0 5 1,800	
Disabled Discard	2
Protocols Learned LSAs	-
H Protocol interfaces	
▲ & >> 🐴 Protocols > 🛱 OSPE > 🚺 1/1/8 Ruppi	ng 🕨 🐣
Overview	
Scenario	
Link State ID Advertising Router LSA T	уре
Ports 1 190.29.0.1 190.29.0.1 Route	er 8
Chassis 2 5.1.1.0 190.29.0.1 Summar	ry IP 8
<b>3</b> 5.1.2.0 190.29.0.1 Summar	ry IP 8
▼ 🔂 Protocols 4 5.1.3.0 190.29.0.1 Summar	ry IP 8
Protocol Interfaces     5   5.1.4.0   190.29.0.1   Summar	ry IP 8
← 🔂 OSPF 6 5.1.5.0 190.29.0.1 Summar	ry IP 8
- ↓ 1/1/7 Running 2	
▼ 🖧 RID - 190.29.0.1	
✓ ■ OSPF Interfaces	
▼ 🛃 192.168.1.1 - 0 - DUT	
🗈 Learned LSAs	
< RouteRanges	
🗲 User LSA Groups	
✓ Ⅰ RID - 190.30.0.1	
Click Learned LSAs	
Learned LSAs	
RouteRanges	

Fig 7.1 Disable discard Learned LSA and check Learned LSAs on 1/1/8 from peer router 1/1/7



# 2.8 Configure Traffic



Fig 8.1: Create traffic stream

# 2.9 Add Endpoints To Traffic



Fig 8.2: Setup source and destination endpoints



#### 2.10 Edit Packet

\*Edit Packet and Flow Group Setup are optional.



Fig 8.3: Edit packet header

#### 2.11 Setup Flow Group



Fig 8.4: Setup flow group



# 2.12 Setup Frame Size

\*Setup frame Size and Line rate are optional.

Main Advanced Traffic Wizard *		— 🗆 🗙
Endpoints	Frame Setup	lxNetwork
Packet / QoS	Iterapsulations         Per Encapsulation           Tx Port         Frame Size         CRC         Payload         Preamble Size	
Flow Group Setup	Image: Non-Section     Image: Non-Section       Image: Image: Non-Section     Image: Non-Section       Image: Image: Image: Non-Section     Image: Non-Section       Image: Image: Image: Non-Section     Image: Non-Section       Image: Image: Image: Image: Non-Section     Image: Non-Section       Image: Image: Image: Image: Image: Image: Non-Section     Image: Image: Non-Section       Image: I	
Rate Setup		
Flow Tracking	Per Encapsulation - Settings will be applied to: EndpointSet-1 / Ethernet II.IPv4	
Protocol Behaviors	Fixed Size 64     Fixed F	▼
Validate	© Random	
	IMIX     O Custom IMIX     O Custom IMIX	
$\leq$	Quad Gaussian  Auto Disparity Errors	
<u> </u>	Retrieve Frame Size	
Ū Ū	Use Control Plane Frame Size, if available  Preamble Size	
	Auto     Custom     S bytes	
$\sim$	2 Click Next	
	<u>Prev</u> <u>Next</u> <u>Einish</u> <u>C</u>	ancel Help

Fig 8.5: Setup Frame size as per test scenario

# 2.13 Setup Line Rate

Me Advanced Traffic Wizard *			— 🗆 X
Endpoints	Rate Setup		IxNetwork
	All Encapsulations     O Per Encapsulation		
Packet / QoS	전물 Tx Port Encapsulation Transmiss	sion Target Rate	Target Rate Distribution
Flow Group Setup	V Name: EndpointSet-1		
Frame Setup		s   10% Line Rate   Apply Ra	te to all ports. Eveniy split port rate among the How Group
Rate Setup	•	1111	•
Elow Tracking	Per Encapsulation - Settings will be applied to: EndpointSet-1	I / Ethernet II.IPv4	
	Traffic Item Transmission Mode		sion Mode
Protocol Behaviors	<ul> <li>Interleaved</li> </ul>	Continuous	Stop After 1 iterations
Preview Preview	O Sequential	Fixed Packet Count	Start Delay 0 bytes 🔻
Validate	The Interleaved Transmit mode will interleave the packets from each Flow Group when sending Traffic	Fixed Iteration Count	Minimum Gap 12 bytes
	Select Line Rate	Fixed Duration Burst (Auto)	
	Round obin Packet Ordering	O Burst (Custom)	
	O Line rate 10.0000 %	How it will look on the wire:	1 2 3 1 2 3 1 2 3 1 2 3
	O Packet rate 100000.00 per second		
(1)	C Layer2 Bit Rate 1000.00 bps 🔻	Ports:	
<u> </u>		<ul> <li>Apply rate on all ports</li> <li>Split rate evenly among pr</li> </ul>	orts
		Flow Groups:	
	V Use Control Plane Rate, if available	Apply port rate to all Flow	Groups
		Split port rate evenly amo	ng Flow Groups
		2	Click Next
		<u>P</u> rev	ext Einish Cancel Help

8.6: Setup line rate



# 2.14 Setup Flow Tracking

\*Setup Flow Tracking and Protocol Behaviors are optional.

Nw /	Advanced Traffic Wizard *				— 🗆	×
	•					
	Endpoints	Flow Tracking	<u>}</u>		IxNetv	vork
		Track Flows by	Custom Overri	de		
	Packet / Qos	Traffic Item	One - One meshed	đ		
	Flow Group Setup	Source/Dest Endpoint Pair	Offset from Root	-		
		Source/Dest Port Pair	Official			
Lide	Frame Setup	Source Endpoint		U DIS		
68	Rate Setup	Dest Endpoint	Field width 32 Bits	· · ·		
		Source Port	Values			
	Flow Tracking	Intended Rx Ports		▶ 0		
R	Protocol Rehaviora	MRIS Flow Descriptor		<1 ype to add value>		
¢	Protocorbenaviors	Frame Size		Select Option Flow Trackin	ig )	
	Preview	Flow Group		-		
		AVB Stream Name				
	Validate	Dest Session Description				
		IPv4 Destination MAC Flow Descriptor				
		Ethernet II : Destination MAC Address	)			
		- Egress Fracking -		- Latency Bin Measurements		
1				Enable Latency Bin Measurements		
	*Optional step	. Use this window to configure	flow tracking	mbers of Bins 🛛 🔻 Minin	num step size: 0.02 us	
	for all flow are	ups For every field 'Track Flo	we by' a flow	Greater Than (us) Less Tha		
	ior an now gro	Jups. For every neid, Track no		• 0.00		-
	will be created	d for each flow group. For exa	nple, if track	1.00		
	flows anablad	on two fields in an ondepoint	cot with two	1.42		
	nows enabled	on two news in an enupoint		2,00	4.00	
	flow groups fo	ur flows will be created. These	flows can be	Click N	ext since	
	viewed individ	ually in the Statistics Window	section		ess 🔜 Egress 🔜 Latency Bi	ns 🔁
l			Dro	Next Einid	h Cancel H	lelo
			Ele	EV NEXT Enis		icip

Fig 8.7 Setup flow tracking to track specific field

New Advanced Traffic Wizard *		×
Protocol Behaviors         Packet / QoS         Flow Group Setup         Frame Setup         Flow Tracking         Preview         Validate         Preview         Validate         Transport LSP:         RSVP, Basic LDP         Inter AS/Region LSP:         BGP (RFC 3107), Targeted LDP         Ordinal Value         Or For out of bound value last available RLOC will be used	<u>IxNetw</u>	ork
Copen Flow Preference     Enable Open Flow     Nexthop selection for EVPN unicast traffic destined to Multi-homed CE     Ordinal Value     O In dassic, first nexthop is used for out of bound ordinal. NGPF will give it multicast treatment     Multicast     Multicast     Multicast forwarding mode:	H	elp

Fig 8.8 Update traffic on the fly with information learned from protocols



# 2.15 Preview Flow Groups

Nw Advanced Traffic Wizard *		— 🗆 X
Endpoints	Preview	lxNetwork
Packet / QoS	Flow Groups/Packets	OCurrent Traffic Item O All Traffic Items View Flow Groups/Packets
Flow Group Setup	Flow Group	Traffic Item
Frame Setup		Click View Flow Groups/Packets
Rate Setup	No Flow	r Groups to display.
Flow Tracking	Please click 'View Flow Groups/Pa	ckets' button above to show the Flow Groups.
Protocol Behaviors		
Preview	Flow Group packets	
Validate		
6	$\searrow$	
	No pa	icket to display.
		ackets button above to snow the packets.
	- 44 bb bbi -d	Click Next
	x	
		Prev Next Einish Cancel Help

Fig 8.9 View flow group which is currently configured

# 2.16 Validate Traffic Items

Market Ma	- D X
Endpoints	Validate IxNetwork
Packet / QoS	
Flow Group Setup	High level view to quickly identify category of errors detected per Traffic Item     Traffic Item     Tracking Bits
Frame Setup	
Flow Tracking	
Protocol Behaviors	
Preview	🙆 0 Errors 🏠 0 Warnings 🕕 0 Messages 😒 0 Test Inspector Events 📄 Show Details 📑 Conv
Validate	Time     Errors
<pre>&gt;</pre>	
$\sim$	2 Click Finish
	Prev Next Finish Cancel Help

Fig 8.10 Validate the traffic item to identify errors

#### **ÍXÍA** A Keysight Business

# 2.17 Apply Traffic, Start Traffic and Statistics View



Click this button to regenerate the traffic before apply. If this button clicked, IxNetwork detects conflicts between existing and newly-generated flow groups (If created) within the traffic item and accept the newly-generated flow groups if needed.

Apply the L2-L3 Traffic by selecting **Apply L2-L3 Traffic** from the L2-3Traffic drop-down menu (on the **Traffic Tools** tab) and Start L2-L3 Traffic.



Fig 9.1 Apply and start traffic







# 3. Configure OSPFv2 through Automation

IxNetwork provides a wide array of automation APIs. The REST API is the recommended method for creating new test scripts for IxNetwork. Please refer to the IxNetwork REST API quick reference guide for more details. The legacy API libraries, including Tcl, and Python traditional API libraries continue to be supported for existing test scripts.

This section explains the method to automate OSPFv2 emulation scenario through High Level TCL/Python APIs. Covers common and OSPFv2 specific HLAPIs used in IxNetwork classic framework.

#### 3.1 Initialize Environment

Source Ixia package and proceed with HLTAPI execution.

<u>TCL</u>	<u>PYTHON</u>
> package require Ixia	import sys, os
Tcl 8.5 is installed on 64bit architecture.	import time, re
IXIA_VERSION env variable is set to 8.20.0.10, but this	from ixiatcl import IxiaTcl
value is not matching any HLTSET.	from ixiahlt import IxiaHlt
Using default HLTSET (HLTSET210) instead.	from ixiaerror import IxiaError
Using products based on HLTSET210	tcl_dependencies = ['/usr/local/lib/', '/usr/lib/',
IxTclHal is not be used for current HLTSET.	'/usr/share/tcl8.5', '/usr/lib/tcl8.5', '/usr/lib/tk8.5',
Loaded IxTclNetwork 8.20.1071.8	'/usr/share/tk8.5' ]
Loaded Mpexpr 1.0	ixiatcl = IxiaTcl(tcl_autopath=tcl_dependencies)
HLT release 8.20.136.2	ixia = IxiaHlt(ixiatcl, use_legacy_api = 1)
Loaded ixia_hl_lib-8.20	
8.20	<i>Note:</i> If python version > 3.4, call IxiaTcl with Tcl 8.6
> package require Tclx	path. Example: tcl_dependencies = ['/path/to/tcl8.6'];
8.4	ixiatcl = lxiaTcl(tcl_autopath=tcl_dependencies)
	ixiatcl = IxiaTcl(); ixia = IxiaHlt(ixiaTcl)

#### 3.2 Add Chassis and Reserve Ports

**::ixia::connect** - Connects to the Ixia Chassis, takes ownership of selected ports and optionally loads a configuration on the chassis or resets the targeted ports to factory defaults.

<u>TCL</u>		<u>PYTHON</u>	
<pre>&gt; set connect_result [::ixia::connect</pre>	1	> connect_result = ixia.connect (	١
-device 10.39.64.132	1	device = 10.39.64.132,	١
-port_list {1/7 1/8}	1	port_list = "1/7 1/8",	١
-reset 1	١	reset = 1	١
-ixnetwork_tcl_server 10.154.161.223:800	9 \	ixnetwork_tcl_server = "10.154.161.223:8009"	", \
-tcl_server 10.154.161.223]		tcl_server = 10.154.161.223 <mark>)</mark>	
> set ports [keylget connect_result vport_list]		<pre>&gt; ports = connect_result['vport_list'].split()</pre>	
> set port1 [lindex \$ports 0]		> port1 = ports[0]	
> set port2 [lindex \$ports 1]		> port2 = ports[1]	

\*Note: High Level API's are highlighted in **Red** and all other handles are highlighted in **Green**.

Now chassis are connected and reserved ports (1/7 & 1/8). Please refer Fig 10.1



Overview	≰ 📚 🚮 > \varTheta Ports >				
	S	Name	Connection Status		
<b>¤</b> <sup>4</sup> Scenario	1 0	1/1/7 1/1/8	10.39.64.132:01:07 10.39.64.132:01:08		
Ports     Chassis			Ports Reserved		

Fig 10.1 Connected to chassis and reserved ports

#### 3.3 Configure Ports

**::ixia::interface\_config** - Configures an interface and accommodates addressing schemes such as IPv4, IPv6, MAC and VLAN.



Fig 11.1 Configure ports attributes using HLAPI

<b>D</b> Overview	Comment Menters     Commenters     Comment Menters     Commenters     Comment Menters     Comment Menters     Comment Menters     Comment Menters     Comment Menters     Commenters     C									
<ul> <li>Ports</li> <li>Chassis</li> <li>Chocols</li> <li>Protocol Interfaces</li> <li>Static</li> <li>Traffic</li> <li>Impairments</li> <li>QuickTests</li> <li>Captures</li> </ul>	Port Description         Port Link         Interface Description         IPV4 Address         IPv4 Mask Width         Gateway           1         11/17 - 100/1000 Base X         11/17 - 00 00 00 00 00 4 - 198.168.1.2         24         198.168.1.1           2         11/18 - 100/1000 Base X         11/18 - 00 00 00 00 00 05 - 5         198.168.1.1         24         198.168.1.2           2         11/18 - 100/1000 Base X         11/18 - 00 00 00 00 00 05 - 5         198.168.1.1         24         198.168.1.2           2         11/18 - 100/1000 Base X         11/18 - 00 00 00 00 00 05 - 5         198.168.1.1         24         198.168.1.2           1         Interface         MAC ) IPv4         DHCPv6   VLAN   ATM /         Configured Ports Attributes									

Fig 11.2 View configured attributes in GUI



# 3.4 Create OSPFv2

**::ixia::emulation\_ospf\_config** - Add ospf router(s) to a port and configure OSPFv2 attributes. Disable 'discard learned lsa' to view learned lsa's on peer router.

<u>TCL</u>		<u>PYTHON</u>	
<pre>&gt; set ospf_emul_res1 \</pre>		> ospf_emul_res1 = \	
[::ixia::emulation_ospf_config	1	ixia.emulation_ospf_config (	١
-port_handle \$port1	١	port_handle = port1,	١
-mode create	١	mode = create,	١
-session_type ospfv2	1	session_type = 'ospfv2',	١
-area_id 0.0.0.1	١	area_id = 0.0.0.1,	١
-router_id 190.29.0.1	1	router_id = 190.29.0.1,	١
-network_type ptop	١	network_type = 'ptop',	1
-lsa_discard_mode 0	1	lsa_discard_mode = 0,	١
-intf_ip_addr 198.168.1.2	١	intf_ip_addr = 198.168.1.2,	١
-neighbor_intf_ip_addr 192.168.1.1]		neighbor_intf_ip_addr = 192.168.1.1)	
<pre>&gt; set ospf_handle1 [keylget ospf_emul_res1</pre>	١	<pre>&gt; ospf_handle1 = ospf_emul_res1['handle']</pre>	
handle]			
<pre>&gt; set ospf_emul_res2 \</pre>		> ospf_emul_res2 = \	
[::ixia::emulation_ospf_config	١	ixia.emulation_ospf_config (	١
-port_handle \$port2	١	port_handle = port2,	١
-mode create	١	mode = create,	١
<pre>-session_type ospfv2</pre>	١	session_type = 'ospfv2',	١
-area_id 0.0.0.1	١	area_id = 0.0.0.1,	١
-router_id 190.30.0.1	١	router_id = 190.30.0.1,	1
-network_type ptop	١	network_type = 'ptop',	١
-lsa_discard_mode 0	١	lsa_discard_mode = 0 ,	1
-intf_ip_addr 198.168.1.1	١	intf_ip_addr = 198.168.1.1,	1
-neighbor_intf_ip_addr 192.168.1.2]		neighbor_intf_ip_addr = 192.168.1.2)	
> set ospf_handle2 [keylget ospf_emul_res2 \		<pre>&gt; ospf_handle2 = ospf_emul_res2['handle']</pre>	
handle]			





Fig 12.2 View configured ospfv2 attributes on 1/1/7 in GUI



✓		Protocol Interface	rface Connected - Protocolinterface - 🗸
€ RouteRanges € User LSA Groups	Arguments Configured	Interface IP	192 . 168 . 1 . 1
<ul> <li>✓ ↓ 1/1/8 Running</li> <li>✓ ♣ RID - 190.30.0.1</li> </ul>	gumento comiguren	IP Mask	255 255 255 0
		Area ID	0 . 0 . 0 . 0
E Learned LSAs		Neiwork Type	Point-Point V

Fig 12.3 View configured ospfv3 attributes on 1/1/8 in GUI

B Overview	Dayan P	Image:     Image:									
Scenario		Enable	Router ID	Number	Number of RouteRanges	Number of UserLSA Groups	LSA Retransmi	LSA Refresh ti	Discard Learned LSAs		
7	1		190.29.0.1	1	1	0	5	1,800			
- Date	2	Ø	190.30.0.1	1	1	0	5	1,800			
Chassis  Chassis  Protocols  Protocol Interfaces  Cospr							Disabled Di Learned I	iscard SAs			

Fig 12.4 View Discard Learned LSAs disbaled in GUI

#### 3.5 Create OSPFv2 Route Ranges

**::ixia::emulation\_ospf\_topology\_route\_config** - Add OSPFv2 route(s) to a simulated OSPFv2 router interface.

<u>TCL</u>	<u>PYTHON</u>
<pre>&gt; set ospf_routes \</pre>	<pre>&gt; ospf_routes = \</pre>
[::ixia::emulation_ospf_topology_route_config \	ixia.emulation_ospf_topology_route_config ( \
-mode create	mode = 'create',
-handle \$ospf_handle1	handle = ospf_handle1,
-type ext_routes	type = ext_routes,
<pre>-external_number_of_prefix 1 \</pre>	external_number_of_prefix = 1,  \
<pre>-external_prefix_length 24 \</pre>	external_prefix_length = 24, \
<pre>-external_prefix_start 5.1.1.0 \</pre>	external_prefix_start = 5.1.1.0, \
<pre>-external_prefix_step 1]</pre>	external_prefix_step 1)

Fig 13.1 Configure OSPFv2 Route Ranges on 1/1/7 ospf handle using HLAPI





Fig 13.2 View configured route ranges in 1/1/7 (Ethernet – 001)

#### 3.6 Start Protocols

::ixia::emulation\_ospf\_control - Start OSPF protocol.







Fig 14.2 View ospfv2 session states details in GUI



# 3.7 Check Learned LSAs

::ixia::emulation\_ospf\_info - Retrieves information about the OSPF sessions. Please refer Fig 15.2

TCL	PYTHON
> set learned info [::ixia::emulation ospf info \	> learned info = ixia.emulation ospf info ( \
-mode learned info	mode = 'learned_info'.
-handle \$ospf_handle2	handle = $ospf$ handle2)
> set sessions [keylaet learned_info	> sessions =
Sospf handle2.S Sospf handle2/interface:1	learned infolospf handle21[ospf handle2+
> puts "Learned LSAs: Ssessions"	'/interface:1']
1:	> print "Learned LSAs: %s" % sessions
adv router id: 190.29.0.1	1:
 age: 30	adv router id: 190.29.0.1
link state id: 190.29.0.1	 age: 30
lsa type: router	link state id: 190.29.0.1
seg_number: 80000002	lsa type: router
prefix_v4_address: ::ixNet::OK	seg_number: 80000002
prefix_v6_address: ::ixNet::OK	prefix_v4_address: ::ixNet::OK
prefix_length: ::ixNet::OK	prefix_v6_address: ::ixNet::OK
2:	prefix_length: ::ixNet::OK
adv_router_id: 190.29.0.1	2:
age: 41	adv_router_id: 190.29.0.1
link_state_id: 5.1.1.0	age: 41
lsa_type: external	link_state_id: 5.1.1.0
seq_number: 80000001	lsa_type: external
prefix_v4_address: ::ixNet::OK	seq_number: 80000001
prefix_v6_address: ::ixNet::OK	prefix_v4_address: ::ixNet::OK
prefix_length: ::ixNet::OK	prefix_v6_address: ::ixNet::OK
3:	prefix_length: ::ixNet::OK
adv_router_id: 190.29.0.1	3:
age: 41	adv_router_id: 190.29.0.1
link_state_id: 5.1.2.0	age: 41
lsa_type: external	link_state_id: 5.1.2.0
seq_number: 80000001	lsa_type: external
prefix_v4_address: ::ixNet::OK	seq_number: 80000001
	prefix_v4_address: ::ixNet::OK
Stripped off Long output	
	Stripped off Long output
<i>9</i> :	9: 
aav_router_1a: 190.29.0.1	aav_router_1a: 190.29.0.1
age: 41	age: 41
link_state_la: 5.1.8.0	link_state_la: 5.1.8.0
isa_type: external	isa_type: external
seq_number: 80000001	seq_number: 80000001
prefix_v4_aaaress: ::IXINet::UK	prefix_v4_adaress: ::IXINet::UK
prejix_vo_adaress: ::IXINET::UK	prejix_vo_adaress: ::IXINet::OK
	prejix_iength: ::ixivet::OK

#### **ÍXÍA** A Keysight Business

10:	10:
adv_router_id: 190.29.0.1	adv_router_id: 190.29.0.1
age: 41	age: 41
link_state_id: 5.1.9.0	link_state_id: 5.1.9.0
lsa_type: external	lsa_type: external
seq_number: 80000001	seq_number: 80000001
prefix_v4_address: ::ixNet::OK	prefix_v4_address: ::ixNet::OK
prefix_v6_address: ::ixNet::OK	prefix_v6_address: ::ixNet::OK
prefix_length: ::ixNet::OK	prefix_length: ::ixNet::OK
11:	11:
adv_router_id: 190.29.0.1	adv_router_id: 190.29.0.1
age: 41	age: 41
link_state_id: 5.1.10.0	link_state_id: 5.1.10.0
lsa_type: external	lsa_type: external
seq_number: 80000001	seq_number: 80000001
prefix_v4_address: ::ixNet::OK	prefix_v4_address: ::ixNet::OK
prefix_v6_address: ::ixNet::OK	prefix_v6_address: ::ixNet::OK
prefix_length: ::ixNet::OK	prefix_length: ::ixNet::OK
status: 1	status: 1
log:	log:

#### Fig 15.1 Retrieve learned LSA info using HLAPI

Overview	Kumber	▶ 🔁 Protocols of LSAs 11	) 🤁 OSPF 🕨 🚺 1/*	1/8 Running 🕨 🛃
		Link State ID	Advertising Router	LSA Type
▼ U Ports	1	190.29.0.1	190.29.0.1	Router
Chassis	2	5.1.1.0	190.29.0.1	Summary IP
	3	5.1.2.0	190.29.0.1	Summary IP
👻 🤁 Protocols	4	5.1.3.0	190.29.0.1	Summary IP
Reprotocol Interfaces	5	5.1.4.0	190.29.0.1	Summary IP
V 💮 OSPF	6	5.1.5.0	190.29.0.1	Summary IP
1/1/7 Running	7	5.1.6.0	190.29.0.1	Summary IP
<b>BID</b> - 190 29 0 1	8	5.1.7.0	190.29.0.1	Summary IP
	9	5.1.8.0	190.29.0.1	Summary IP
	10	5.1.9.0	190.29.0.1	Summary IP
	11	5.1.10.0	190.29.0.1	Summary IP
<ul> <li>✓ User LSA Groups</li> <li>✓ 1/1/8 Running</li> <li>✓ ♣ RID - 190.30.0.1</li> <li>✓ ➡ OSPF Interfaces</li> <li>✓ ♣ 192.168.1.1 - 0 - DUT</li> </ul>	Learne	ed LSAs	Learned LSA Peer	s from
RouteRanges	Stat N	Select Views	Port CPU Statistics	Port Statistics

Fig 15.2: View learned LSAs info in GUI



# 3.8 Configure Traffic

::ixia::traffic\_config - Configures traffic streams on the specified ports with the specified options. Here created the traffic flow from  $1/1/8 \rightarrow 1/1/7$ .

<u>TCL</u>	<u>PYTHON</u>			
<pre>&gt; set traffic_res [::ixia::traffic_config \</pre>	<pre>&gt; traffic_res = ixia.traffic_config (</pre>			
-mode create	mode = 'create', \			
-transmit_mode continuous \	<pre>transmit_mode = 'continuos', \</pre>			
<pre>-track_by {traffic_item}</pre>	<pre>track_by = 'traffic_item',</pre>			
-rate_pps 1000 \	rate_pps = 1000, \			
-port_handle \$port2	port_handle = port2, \			
-port_handle2 \$port1 \	port_handle2 = port1, \			
-I3_protocol ipv4	I3_protocol = 'ipv4', \			
-ip_src_addr 192.168.1.1 \	ip_src_addr = 192.168.1.1,			
-ip_dst_addr 192.168.1.2 \	ip_dst_addr = 192.168.1.2,			
-mac_dst 00:00:19:d5:54:74 \	mac_dst = 00:00:19:d5:54:74, \			
-mac_src 00:00:19:d5:54:75 \	mac_src = 00:00:19:d5:54:75, \			
-l3_length 64]	l3_length = 64 <mark>)</mark>			

Fig 16.1: Create traffic stream from 1/1/8 to 1/1/7 port handle using HLAPI

< 📎 🚮 🗯 Traffic 🕨 🗱 L2-3 Traffic Items 🕨 🔀 TI13-HLTAPI_TRAFFICITEM_540							
Flow Groups Topology Endpoint Sets							
En Trans S Tx Port Rx Ports Flow Group Encaps * Configure Applied Fra Frame Rate Pream CRC Frame P Transm	ission Tx p						
Traffic Item Name: TI13-HLTAPI_TRAFFICITEM_540 TX Mode: Interleaved, Src/Dst Mesh: OneToOne, Route Mesh: OneToOne, Uni-directional							
1 🗹 🔍 💷 1/1/8 1/1/7; TI13-HLTAPI Ethernet Fixed: 82 1000 fps Auto No E Incremen Continu	ous Ethe						
Arguments Configured							

Fig 16.2: View traffic item configuration in GUI



#### 3.9 Start and Stop Traffic

**::ixia::traffic\_control** - Starts or Stops traffic on a given port list. As per below code, start the traffic, wait for some time for the traffic to flow and stop the traffic.

<u>TCL</u>	<u>PYTHON</u>
<pre>&gt; ::ixia::traffic_control -action run &gt; puts "Wait for some time for the traffic to flow" &gt; ::ixia::traffic_control -action stop</pre>	<pre>&gt; ixia.traffic_control (action = 'run') &gt; print "Wait for some time for the traffic to flow" &gt; ixia.traffic_control (action = 'stop')</pre>





Fig 17.2: View traffic start status in GUI



Fig 17.3: View traffic stop status in GUI



# 3.10 Get Statistics

**::ixia::traffic\_stats** - Gathers statistics depends on the mode suchss as per\_port\_flows, session, stream, streams, all etc., Below code used mods as 'streams'

<u>TCL</u>	ΡΥΤΗΟΝ				
> set stats [::ixia::traffic_stats \	> stats = ixia.traffic_stats (				
-mode streams	mode = streams,				
-streams TI13-HLTAPI_TRAFFICITEM_540]	streams = TI13-HLTAPI_TRAFFICITEM_540)				
> puts "Traffic Stats: \$stats"	> print "Traffic stats: %s" % stats				
status: 1	status: 1				
measure_mode: mixed	measure_mode: mixed				
waiting_for_stats: 0	waiting_for_stats: 0				
1/1/7:	1/1/7:				
stream:	stream:				
TI13-HLTAPI_TRAFFICITEM_540:	TI13-HLTAPI_TRAFFICITEM_540:				
rx:	rx:				
total_pkt_rate: 0.000	total_pkt_rate: 0.000				
total_pkt_byte_rate: 0.000	total_pkt_byte_rate: 0.000				
loss_percent: 0.000	loss_percent: 0.000				
small_error: N/A	small_error: N/A				
total_pkts_bytes: 150153856	total_pkts_bytes: 150153856				
expected_pkts: N/A	expected_pkts: N/A				
pkt_loss_duration: N/A	pkt_loss_duration: N/A				
last_tstamp: 00:00:16.334	last_tstamp: 00:00:16.334				
total_pkts: 368497	total_pkts: 368497				
reverse_error: N/A	reverse_error: N/A				
Stripped off long output	Stripped off long output				
1/1/8:	1/1/8:				
stream:	stream:				
TI13-HLTAPI_TRAFFICITEM_540:	TI13-HLTAPI_TRAFFICITEM_540:				
tx:	tx:				
total_pkts: 368497	total_pkts: 368497				
total_pkt_rate: 0.000	total_pkt_rate: 0.000				

#### Fig 18.1: Retrieve traffic stats using HLAPI

Protocols     Protocol Interfaces     B OSPF	Summary Flow groups Frame Setup D						
<ul> <li>→ ⊕ Static</li> <li>→ X Traffic</li> </ul>	Select Views. Port CPU Statistics Port Statistics Global Protocol Statistics L2-L3 Test Summary Statistics Flow Statistics Traffic Item Statistics						
✓ L2-3 Traffic Items     ✓ T113-HLTAPI_TRAFFICITEM_540     ✓ L2-3 Thur Course	Traffic Item         Tx Frames         Rx Frames         Prames         Delta         Loss %         Tx Frame Rate         Rx Frame Rate         Tx L1 Rate (bps)         Rx L1 Rate (bps)         Rx Bytes         Tx Rate (Bps)         Rx Rate (Bps)         Rx Rate (Bps)         Tx Rate (Bps)         Rx Rate						





# 4. Other Utilities

#### 4.1 IxNetwork API Documentation Browser

The main feature of this application is the ability to browse the API meta data in a hierarchical format. Access each level of the hierarchy with a view of siblings, attributes, execs, errors, and children by on clicking on BROWSE.



Fig 19.1: IxNetwork API Documentation Browser



# 4.2 Script Gen

For complex configuration use SCRIPTGEN. Reverse-engineer the scriptgen scripts as per the requirement.



Fig 20.1: Tool to generate script gen



#### 4.3 <u>F1 Option</u>

Move the mouse pointer over any field in the GUI, and then press F1 to get more information about the field. From Classic Protocols section, users can explore all the fields of all protocols. OSPFv2 protocol tree structure shown below.

File	Home	Autor	mation	Results /	Rep	oorts	Views		
Protocols	All Traffic •	uickTest	Compos Compos	er Script	•	Add Ports •	Add Protocols •	Add Traffic	Add QuickTests •
		Run							Build
Coverview Scenario Coverview O Ports									
💮 Por	rts		Pre	ess F1		)			
💮 Pro	otocols								
<b>↔</b> → <b>∓</b>	<i>cc</i> :						boro	+	

Fig 20.1 Press F1 on Overview field









Fig 20.2.1 Shows the contents of OSPFv2 Protocol Tree

# 5. To Know More on IxNetwork Classic

<u>https://www.youtube.com/watch?v=gWjgFndvSAI</u> <u>http://openixia.com/sampleScripts//lxNetwork/HighLevelApi/Classic/Tcl</u>

#### 6. Support

For more information, visit <u>https://support.ixiacom.com/</u> For support assistance, contact <u>support.ix@keysight.com</u>





https://github.com/openixia For queries : support.ix@keysight.com