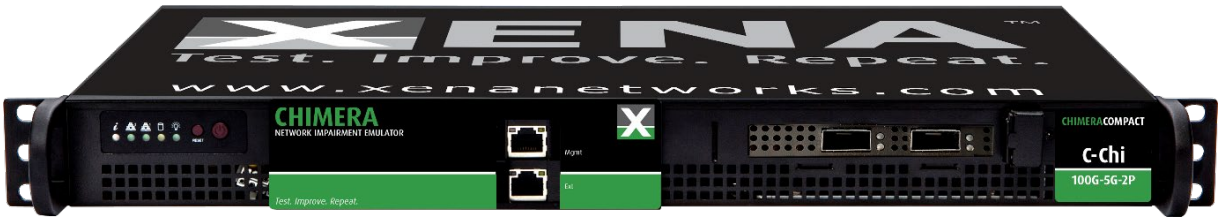
Xena script commands for



Chimera

NETWORK IMPAIRMENT EMULATOR

Updated May, 2021

Table of Contents

Overview

This document describes the Xena script commands for Chimera installed in ValkyrieBay and ValkyrieCompact chassis. The general syntax for the Xena script commands for Chimera is the same as for Valkyrie.

As an alternative to using the ValkyrieManager, you can interact with the Chimera using Xena script commands. This also allows the Chimera to be controlled from a scripting environment and be part of a larger automation environment.

Script commands can be defined at various levels – Chassis, Module, Port, and Flow.

Script Definition Format

The general syntax for the Xena script commands for Chimera is the same as for Valkyrie.

<entity> API [indices] value-types

Where,

- **entity** can be chassis number, module number or port number.
- **indices** are specified as integers. When there are more than one index to be specified, use comma as separator.
- **Value-types** are used to specific the parameters in case of SET/GET and return values in case of GET.

Value Type	Identifier
Byte	B
String	S
Integer	I
Long	L
Hex String	H
Address	A
Owner*	O

*special string for tracking reservations

Minimum and Maximum Latency

Speed	Minimum Latency	Maximum Latency NORMAL mode	Maximum Latency EXTENDED mode
100G (FEC)	7.0us	1.9s	10s
100G (No FEC)	7.0us	1.9s	10s
50G	7.0us	1.9s	10s
25G (FEC)	7.2us	1.9s	10s
25G (No FEC)	7.0us	1.9s	10s
40G	7.0us	1.9s	10s
10G	13.0us	1.9s	10s

Chassis Script Commands

This is same for all for Xena traffic generators. Refer to Chassis Script Commands.

Module Script Commands

This chapter describes the layer 2-3 module level script commands corresponding to the **Module Resource Properties** panel of the ValkyrieManager, and deal with basic information about, and configuration of, the module itself.

The module parameter names all have the form M_**<xxx>** and require a module index before the parameter name. In general the module must be reserved to use the 'set' functions in this section.

Identification

M_MODEL model

Obtains the specific Xena model of a module.

model: string, the Xena model designation for the module.

Summary: get only, value type: S

Example: 0 M_MODEL?

0 M_MODEL "XE2QSFP28V"

M_COMMENT comment

Obtains the user-defined description string of a module.

comment: string, The user-specified comment/description for the module.

Summary: get only, value type: S

Example: 0 M_COMMENT?

0 M_COMMENT ""

M_SERIALNO serialno

Obtains the unique serial number of a module.

serialno: integer, the serial number of this module.

Summary: get only, value type: I

Example: 0 M_SERIALNO?

0 M_SERIALNO 525083

M_VERSIONNO version

Obtains the version number of the hardware image installed on a module.

version: integer, the hardware image version number.

Summary: get only, value types: I

Example: 0 M_VERSIONNO?

0 M_VERSIONNO 307

M_PORTCOUNT count

Obtains the number of ports on a module for the selected speed.

count: integer, the max. number of ports.

Summary: get only, value types: I

Example: 0 M_PORTCOUNT?

0 M_PORTCOUNT 2

M_INFO

Multi-parameter query, obtaining all the non-settable parameters for a module.

Summary: get only

Example: 0 M_INFO?

0 M_RESERVATION RESERVED_BY_YOU
0 M_RESERVEDBY ”HH”
0 M_MODEL ” Module model ”
0 M_SERIALNO 240751
0 M_VERSIONNO 276
0 M_STATUS 45
0 M_CFPTYPE FLEXIBLE

M_CONFIG

Multi-parameter query, obtaining all the settable parameters for a module.

Summary: get only

Example: 0 M_CONFIG??

0 M_TIMESYNC?

0 M_TIMESYNC CHASSIS

0 M_CONFIG?

0 M_MEDIA QSFP28

0 M_CFPCONFIGEXT 2 100000 100000

0 M_CFPCONFIG 2 100

0 M_COMMENT ""

0 M_TIMESYNC CHASSIS

0 M_CLOCKPPB 0

0 M_SMAINPUT NOTUSED

0 M_SMAOUTPUT DISABLED

0 M_TXCLOCKSOURCE MODULELOCALCLOCK

0 M_TXCLOCKFILTER BW7019HZ

0 M_LATENCYMODE NORMAL

M_RESERVATION whattodo

You set this parameter to reserve, release, or relinquish a module itself (as opposed to its ports). The module must be reserved before its hardware image can be upgraded or to change the speed of the ports.

The owner of the session must already have been specified. Reservation will fail if the chassis or any ports are reserved to other users.

whattodo: coded byte, containing the operation to perform: [RELEASE (0) | RESERVE (1) | RELINQUISH (2)]

Note: The reservation parameters are slightly asymmetric with respect to set/get. When querying for the current reservation state, the chassis will use these values: [RELEASED (0) | RESERVED_BY_YOU (1) | RESERVED_BY_OTHER (2)]

Summary: set or get, value type: B

Example: 0 M_RESERVATION RELEASE

M_RESERVEDBY username

Identify the user who has a module reserved. Returns an empty string if the module is not currently reserved by anyone.

username: string, containing the name of the current owner of the module.

Summary: get only, value type: O

Example: 0 M_RESERVEDBY?

0 M_RESERVEDBY "michael"

M_STATUS temperature

Get status readings for the test module itself.

temperature: temperature of the main hardware chip, in degrees Celsius.

Summary: get only, I*

Example: 0 M_STATUS?

0 M_STATUS 45

Timing Configuration

M_CLOCKPPB ppb

Makes small adjustment to the local clock of the test module, which drives the TX rate of the test ports.

ppb: adjustment from nominal value, in parts-per-billion, positive or negative.

Summary: set and get, value type: I

Example: 0 M_CLOCKPPB -200000

M_TXCLOCKSOURCE source

Controls the clock source of the Tx port. The source can be configured to be either the recovered clock of one of the Rx port the local module oscillator.

source: clock source

- MODULELOCALCLOCK(0)

- SMAINPUT(1)
- P0RXCLK(2)
- P1RXCLK(3)
- P2RXCLK(4)
- P3RXCLK(5)
- P4RXCLK(6)
- P5RXCLK(7)
- P6RXCLK(8)
- P7RXCLK(9)

Only the valid ports at the given speed, can be given as input.

Summary: set and get, value type: I

Example: 0 M_TXCLOCKSOURCE 4

M_TXCLOCKSTATUS

Retrieves the current selected clock source of the Tx port.

Summary: get only, value type: I

- OK, indicates that an Rx port was selected as Tx clock source, and that the system can successfully recover this clock.

- NOVALIDTXCLK, indicates local clock source is selected or that the user configured an Rx clock as clock source, but the FPGA could not recover this clock and is therefore selecting the internal local clock.

Example: 0 M_TXCLOCKSTATUS?

0 M_TXCLOCKSTATUS NOVALIDTXCLK

Media Configuration

M_CFPTYPE info

Get information about the transceiver currently inserted into the cages of the Chimera HW module.

info:

coded byte, specifying the CFP state:

- NOTCFP (0): this is not a CFP-based test module
- NOTPRESENT (1): no transceiver, the CFP cage is empty
- NOTFLEXIBLE (2): transceiver present, supporting a fixed speed and port-count
- FLEXIBLE (3): transceiver present, supporting flexible speed and port-count

byte, specifiying the CFP type

Summary: get only, B,B

Example: ;8 M_CFPTYPE?

8 M_CFPTYPE NOTPRESENT 0

M_CFPCONFIG ports speed

The current number of ports and their speed of a CFP test module.

If the CFP type is NOTFLEXIBLE then it reflects the transceiver currently in the CFP cage. If the CFP type is FLEXIBLE (or NOTPRESENT) then the configuration can be changed explicitly.

The following combinations are possible: 8x10G, 8x25G, 2x40G, 4x50G, 2x100G

ports: number of ports.

speed: port speed, in Gbps.

Summary: set and get, value types: B,B

Example: 0 M_CFPCONFIG 2 40

Firmware Upgrade

M_UPGRADE magic imagename

Transfers a hardware image file from the chassis to a module. This image will take effect when the chassis is powered-on the next time. The transfer takes approximately 3 minutes, but no further action is required by the client.

magic: integer, must be the special value -1480937026.

imagename: string, the fully qualified name of a file previously uploaded to the chassis.

Summary: set only, value types: I,S

Example: 0 M_UPGRADE -1480937026 ”/xbin/xenaimageXE_18”

M_UPGRADEPROGRESS progress

Provides a value indicating the current stage of an on-going hardware image upgrade operation. This is for information only; the upgrade operation runs to completion by itself. The progress values are pushed to the client without it having to request them.

progress: integer, the current stage within the three phases: erase, write, verify.

Summary: pushed (get) only, value type: I

- 1-100: Erase completion percentage.

- 101-200: Write completion percentage + 100.
- 201-300: Verify completion percentage + 200.
- 0: Failure.

Example: 0 M_UPGRADEPROGRESS 277

Impairments

M_LATENCYMODE mode

Configures the latency mode. In extended latency mode, the FPGA allows all latency parameters to be 10 times higher, at the cost of reduced latency precision.

mode: coded byte, specifying latency mode:

- NORMAL (0)
- EXTENDED (1)

Summary: set and get, flow index, value type: B

NOTE:

- For Latency range see section “Minimum and Maximum Latency”.
- When change the latency mode, all latency configurations are reset on all ports in chimera module.

Example or get: 0 M_LATENCYMODE NORMAL

Misc. settings

M_EMULBYPASS onoff

Set emulator bypass mode. Emulator bypass mode will bypass the entire emulator for minimum latency.

onoff: coded byte, whether the emulator bypass is enabled.

- OFF (0)
- ON (1)

Summary: set and get, value type: B

Example or get: 0/0 M_EMULBYPASS ON

Port Script Commands

This page describes the layer 2-3 port level script commands corresponding to the Port Resource Properties panel of the ValkyrieManager, and deal with basic information and configuration of the test ports. The port parameter names all have the form P_<xxx> and require both a module index and a port index before the parameter name. In general, port parameters require the port to be reserved, before changes can be made and parameters cannot be changed while emulation is activated.

General State and Identification

P_RESERVATION whattodo

You set this parameter to reserve, release, or relinquish a port. The port must be reserved before any of its configurations can be changed, including flows. Reservation will fail if the chassis or module is reserved to other users.

whattodo: coded byte, containing the operation to perform: RELEASE RESERVE RELINQUISH. The reservation parameters are slightly asymmetric with respect to set/get. When querying for the current reservation state, the chassis will use these values: RELEASED RESERVED_BY_YOU RESERVED_BY_OTHER

Summary: set or get, value type: B

Example: 0/0 P_RESERVATION RESERVE Example: 0/0 P_RESERVATION RESERVED_BY_YOU

P_RESERVEDBY username

Identify the user who has a port reserved. The empty string if the port is not currently reserved. Note that multiple connections can specify the same name with C_OWNER, but a resource can only be reserved to one connection. Therefore, you cannot count on having the port just because it is reserved in your name. The port is reserved to this connection only if P_RESERVATION returns RESERVED_BY_YOU.

username: string, containing the name of the current owner of the port.

Summary: get only, value type: O

Example: 0/0 P_RESERVEDBY ”HH”

P_INTERFACE interface

Obtains the physical interface type of a port.

interface: string value describing the interface

Summary: get only, value type: S

Example: 0/0 P_INTERFACE ”Interface type”

P_COMMENT comment

The description of a port.

comment: string, containing the description of the port.

Summary: set and get, value type: S

Example or get: 0/0 P_COMMENT ”This port emulates a GPON network.”

Tx Control

P_RESET

Reset port-level parameters to standard values, and delete all flow definitions.

Summary: set only.

Example: 0/1 P_RESET

P_TXENABLE onoff

Whether a port should enable its optical transmitter, or keep the outgoing link down.

(This does not have any effect in case of electrical cables).

onoff: coded byte, whether the transmitter is enabled.

OFF (0)

ON (1)

Summary: set and get, value type: B

Example or get: 0/0 P_TXENABLE ON

P_LOADMODE action

The action determines if config load mode is enabled or disabled

action: coded byte

OFF (0) (config load function disabled)

ON (1) (config load function enabled)

Summary: set and get, value type: B

Example or get: 0/0 P_LOADMODE OFF

P_INFO

Multi-parameter query, obtaining all the non-settable parameters for a port. These parameters should not be included if the port configuration is saved and reloaded at a later time.

Summary: get only.

Example: 0/0 P_INFO?

0/0 P_RESERVATION RESERVED_BY_YOU

0/0 P_RESERVEDBY "michael"

0/0 P_INTERFACE "QSFP28 100G CR4"

0/0 P_STATUS -1

0/0 P_SPEED 100000

0/0 PX_TEMPERATURE 0 0

0/0 P_EMULATE ON

P_CONFIG

Multi-parameter query, obtaining all the settable parameters for a port itself excluding flows.

Summary: get only.

Example: 0/0 P_CONFIG?

0/0 P_EMULATE ON

0/0 P_TXENABLE ON

0/0 P_COMMENT ”This port emulates a GPON network.”

… <etc>

P_FULLCONFIG

Multi-parameter query, obtaining all the settable parameters for a port, including flows. These parameters comprise the complete user-definable configuration for the port.

Summary: get only.

Example: P_FULLCONFIG?

0/0 P_EMULATE ON

0/0 P_TXENABLE ON

0/0 P_COMMENT ”This port emulates a GPON network.”

… <etc>

PE_COMMENT[fid] comment

Flow description.

comment: string, containing the description of the port.

Summary: set and get, value type: S

Example or get: 0/0 PE_COMMENT[0] ”eCPRI U-plane traffic.”

PE_CONFIG

Multi-parameter query, obtaining all the settable impairments parameters for a port per flows.

Summary: get only.

Example: 0/0 PE_CONFIG[0]?

0/0 PE_COMMENT [0] ""

0/0 PE_CORRUPT [0] OFF

0/0 PE_MISORDER [0] 1

0/0 PE_BANDPOLICER [0] OFF L1 0 0

0/0 PE_BANDSHAPER [0] OFF L1 0 0 0

PE_FULLCONFIG

Multi-parameter query, obtaining all the settable impairment parameters for a port, for all flows. These parameters comprise the complete user-definable configuration for the port.

Summary: get only.

Example: 0/0 P_FULLCONFIG?

0/0 PE_INDICES 0 1 2 3 4 5 6 7

0/0 PE_COMMENT [0] ""

0/0 PE_CORRUPT [0] OFF

0/0 PE_MISORDER [0] 1

0/0 PE_BANDPOLICER [0] OFF L1 0 0

0/0 PE_BANDSHAPER [0] OFF L1 0 0 0

0/0 PE_COMMENT [1] ""

0/0 PE_CORRUPT [1] OFF

0/0 PE_MISORDER [1] 1

0/0 PE_BANDPOLICER [1] OFF L1 0 0

0/0 PE_BANDSHAPER [1] OFF L1 0 0 0

PEC_FULLCONFIG

Multi-parameter query, obtaining all the settable custom distribution parameters for a port.

Summary: get only.

Example: 0/0 PEC_FULLCONFIG?

PEC_VAL [5] RANDOM SYMM 512 10000, 20000, ..., 10009

PEC_COMMENT[5] "COMMENT_5"

PEF_FULLCONFIG

Multi-parameter query, obtaining all the settable filter parameters for a port, including all flows. These parameters comprise the complete user-definable configuration for the port.

Summary: get only.

Example: 0/0 PEF_FULLCONFIG?

0/0 PEF_INIT[0]

0/0 PEF_IPV4DSCP[0,1] ON 63

0/0 PEF_ENABLE[0,1] ON

0/0 PEF_APPLY[0]

0/0 PEF_INIT[7]

0/0 PEF_L2PUSE[7,1] MPLS

0/0 PEF_APPLY[7]

… <etc>

P_STATUS

Get status readings for the port.

opticalpower: integer, received signal level for optical ports, in nanowatts, -1 when not available.

Summary: get only, value types: I

Example: 0/0 P_STATUS?

0/0 P_STATUS -1

PP_LINKFLAP_PARAMS duration period repetition

Set port 'link flap' parameters.

Duration: (0 ms – 1 s; increments of 1 ms; 0 = permanently link down)

Period: (10 ms – 50 s; number of ms – must be multiple of 10 ms)

Repetition: (1 – 64K; 0 = continuous)

Notice:

Period must be > duration.

Summary: Set and get value type: I, I, I

Example set: 0/1 PP_LINKFLAP_PARAMS 5 1000 5

PP_LINKFLAP_ENABLE enable

Enable / disable port 'link flap'.

enable: (0 = OFF, 1 = ON)

Summary: Set and get value type: B

Example set: 0/1 PP_LINKFLAP_ENABLE 1

PP_PMAERRPUL_PARAMS duration period repetition coeff exp

Set the 'PMA pulse error inject' parameters

Duration: (0 ms – 5 s; increments of 1 ms; 0 = constant BER)

Period: (10 ms – 50 s; number of ms – must be multiple of 10 ms)

Repetition: (1 – 64K; 0 = continuous)

Coeff: (0.01 < coeff < 9.99) * 100

Exp: (-3 < exp < -17)

Summary: set and get value type: I, I, I, l, I

Note:

- Period must be > duration.

- BER will be: coeff * 10exp

Example: 0/1 PP_PMAERRPUL_PARAMS 500 1000 123 234 -5

PP_PMAERRPUL_ENABLE enable

Enable / disable 'PMA pulse error inject'.

enable: (0 = OFF, 1 = ON)

Summary: set and get value type: B

Example: 0/1 PP_RMAERRPUL_ENABLE 1

P_EMULATE action

The action determines if emulate functionality is enabled or disabled

action: coded byte

OFF (0) (Emulate function disabled)

ON (1) (Emulate function enabled)

Summary: set and get, value type: B

Example: 0/0 P_EMULATE OFF

PEF_INDICES indices

Old Command Name: PW_INDICES

Set/Get the list of flow indices

Indices: array of int (size 8)

Summary: set and get, value type: array of int (size 8). Range 0 to 7

NOTE:

- Get: Returns list of flows defined on port - currently fixed at 0 .. 7
- Set: supports only setting to fixed list of 0 .. 7 (Intended for future use)

Example or get: 0/0 PE_INDICES 0 1 2 3 4 5 6 7

PE_FCSDROP action

The action on packets with FCS errors on a port.

action: coded byte

OFF (0) (all packets with FCS errors are associated with the default flow and treated accordingly) - Default

ON (1) (all packets with FCS errors are dropped)

Summary: set and get, value type: B

Example or get: 0/0 PE_FCSDROP OFF

PE_TPLDMODE size

The action indicates the TPLD mode to be used per port.

size: coded byte

NORMAL (0) (20 Bytes TPLD Mode is used) – Default

MICRO (1) (6 Bytes TPLD Mode is used)

Summary: set and get, value type: B

Example or get: 0/0 PE_TPLDMODE NORMAL

Impairments

PEC_VAL[cust_id] linear, symmetric, num_entries, data0... data_num_entries -1

Definition of custom distribution. Custom distributions can be defined for latency with 1024 entries and for non-latency impairments with 512 entries.

Each port will maintain a list of defined custom distributions, identified by an CUST_ID. (Range: 1 – 40)

cust_id: integer, the sub-index value of the custom distribution definition.
linear: coded byte, defines the way the FPGA RAM content is played out.

- OFF (0) – Random address for playing RAM content
- ON (1) – Incrementing address for playing RAM content
- Default value: OFF (0)

symmetric: coded byte:

- Reserved for future use – must be set to 0.

num_entries: integer, defines the no. of entries in “dataX” (allowed value: 512,1024)

• NOTE: For Latency, 1024 entries are used, and for rest, 512 entries are used)

dataX: (array of long of size=”num_entries”), holds values to be filled in the RAM memory.

In case of iid = DELAY specifies delay.

Unit is ns. Valid range 0 → 30 ms (Normal timing mode) / 302 ms (Extended timing mode)

In case of iid != DELAY specifies distance between impairments in number of packets packets.

Unit is packets. Valid range 0 → 262,143 packets.

Summary: set and get, custom index, value-type:B,B,I,L*

NOTE:

If a custom distribution definition is updated, the updated contents is pushed to all the relevant impairment type specific RAMs after validation. If validation fails, custom distribution is not updated and an error is returned to the user. See 7.4.15

Example: 0/0 PEC_VAL [5] ON ON 512 10000 20000....30009

PEC_COMMENT[cust_id]

Defines the user-defined description string of a custom distribution.

cust_id: integer, the sub-index value of the custom distribution definition.

comment: string, The user-specified comment/description for the custom distribution.

Summary: get only, value type: S

Example: 0/0 PEC_COMMENT[5] ”This is Gauss Distr”

PEC_DELETE[cust_id]

Deletes the custom distribution definition.

cust_id: integer, the sub-index value of the custom distribution definition.

Summary: set only

NOTE: Once a customer has defined a customer distribution using PEC_VAL[CUST_ID], it is defined until it is explicitly deleted.

Only customer distributions which are not referenced by any impairments, can be deleted.

Example: 0/0 PEC_DELETE[5]

PEC_INDICES

The full list of which custom distributions which are defined for a port. These are the custom id values that are used for assigning the custom distributions to an impairment. Setting the value of this parameter creates a new custom distribution (default values) for each value that is not already in use, and deletes each custom distribution that is not mentioned in the list. The same can be accomplished one-custom-distribution-at-a-time using the PEC_VAL and PEC_DELETE commands.

NOTE:

- Custom distributions which are currently defined are not affected when mentioned in a PEC_INDICES set command.
- Custom distributions which are currently assigned to an impairment cannot be deleted and any attempt of deleting such a custom distribution using either PEC_DELETE or PEC_INDICES will result in an error.

Summary: set and get, value type: array of int (max size 40)

Example: 0/0 PEC_INDICES ?

0/0 PEC_INDICES 1 2 3 4 5 6 7

PEC_DISTTYPE

Retrieves if a custom distribution is defined for latency or non-latency.

Summary: set and get, value type: B (0 indicates Interpacket distribution, 1 indicates latency distribution)

NOTE: Using PEC_DISTTYPE as set has no effect. The disttype is determined upon custom distribution creation and cannot be modified later. However, it is legal to issue the PEC_DISTTYPE set command with no effect.

Example: 0/0 PEC_DISTTYPE?

0/0 PEC_DISTTYPE 0

Flow Script Commands

Flows in Chimera are identified using a 'flow ID' (fid). Valid fids are 0-7, where fid = 0 is the port default flow.

When configuring the flow filters and flow impairments, the fid of the flow to be configured must be provided to the API using square brackets after the API command. E.g: PEF_INIT[fid]

Filter Configurations

A filter configuration can either be Basic (the default), or Extended. Basic mode enables configuration of a basic subset of protocol segments in a fixed order whereas Extended allows a much high degree of flexibility to filter within the first 128 bytes of a packet. The mode is selected with PEF_MODE.

Basic mode allows filtering on:

- Ethernet header
- Up to two VLANs
- Up to one MPLS label
- The first IPv4 or IPv6 header
- Up to six sequential bytes of payload anywhere within the first 128 bytes of the packet
- Xena Test Payload IDs (TPID)

Extended mode allows filtering based on an initial Ethernet header followed by any combination of segments up to a total length of 128 bytes, as well as optionally Xena Test Payload IDs (TPID).

For example:

0/0 PEF_PROTOCOL [1,0] ETHERNET VLAN VLAN MPLS MPLS -4 ETHERNET VLAN

could specify a Metro Ethernet Forum provider-tagged (first VLAN) customer-tagged (second VLAN) frame containing an MPLS tunnel (first MPLS label) containing an MPLS Pseudo Wire (second MPLS label + four bytes of Control Word) containing a single-tagged Ethernet frame.

Once the protocol layout has been specified it is possible to set up a sequence of value + mask bytes that selects the relevant fields of each protocol segment; see PEF_VALUE and PEF_MASK.

PEF_INIT [fid]

Prepares for setting up a filter definition. When called, all filter definitions in the shadow-set which are not applied are discarded and replaced with the default values (DEFAULT).

NOTE: There are 2 register copies used to configure the filters:

- Shadow copy (type = 0)
  - Temporary copy configured by SW. Values stored in 'shadow registers' have no immediate effect on the flow filters. "PEF_APPLY" API will pass the values from the 'shadow copy' to the 'working copy'.
- Working registers (type = 1)
  - Reflects what is currently used for filtering in the FPGA. Working registers cannot be written directly – only using the 'shadow copy'.
- All SETs are performed on shadow registers ONLY.
- Only when PEF_APPLY is called, working registers and FPGA are updated with values from the 'shadow copy'.

fid: integer, the sub-index value of the flow definition.

Summary: set, flow index

Example: 0/0 PEF_INIT [5]

PEF_APPLY [fid]

Applies filter definitions from "shadow-registers" to "working-registers". This also pushes these settings to the FPGA.

fid: integer, the sub-index value of the flow definition.

Summary: set, flow index, value type: B

Example: 0/0 PEF_APPLY [5]

PEF_MODE [fid,filter_type] mode

Select Basic or Extended filter mode.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

Mode: BASIC or EXTENDED. Default is BASIC.

Summary: set and get, flow index, filter type, mode

Example: 0/0 PEF_MODE [1,0] EXTENDED

PEF_ENABLE [fid,filter_type] state

Defines if filtering is enabled for the flow.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

state: coded byte, specifies the state of the filter:

- OFF (0) (Filtering is disabled on the flow)
- ON (1) (Filtering is enabled on the flow)
- Default value: OFF (0)

Summary: set and get, flow index, filter type, value type: B

Example: 0/0 PEF_ENABLE [5,1] ON

PEF_ETHSETTINGS [fid, filter_type] usage action

Basic mode only. Defines what filter action is performed on the Ethernet header.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

usage: coded byte, specifies if Ethernet information is expected:

- OFF (0) (No filtering will be done on this header)
- AND (1) (AND - filtering will be done on this header)
- Default value: OFF (0)

action: coded byte, specifies the usage of Ethernet information:

- EXCLUDE (0) (Matching packets are excluded from the flow)
- INCLUDE (1) (Matching packets are included in the flow)
- Default value: EXCLUDE (0)

Summary: set and get, flow index, filter type, value type: B, B

Example: 0/0 PEF_ETHSETTINGS[5,0] AND INCLUDE

PEF_ETHSRCADDR [fid,filter_type] use value mask

Basic mode only. Defines the Ethernet Source Address settings for the Ethernet filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of Ethernet information:

- OFF (0) (Ethernet Source Address is not used for the filter)
- ON (1) (Ethernet Source Address is used for the filter)
- Default value: OFF (0)

value: hex bytes, specifying the six bytes of the address. Default value: 0x000000000000

mask: hex bytes, specifying the mask corresponding to the address. Default value: 0xFFFFFFFFFFFF

Summary: set and get, flow index, filter type, value type: B, HHHHHH, HHHHHH

Example:

0/0 PEF_ ETHSRCADDR [5,0] OFF 0x000000000000 0xFFFFFFFFFFFF

PEF_ETHDESTADDR [fid, filter_type] use value mask

Basic mode only. Defines Ethernet Destination Address is used for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of Ethernet information:

- OFF (0) (Ethernet Source Address is not used for the filter)
- ON (1) (Ethernet Source Address is used for the filter)
- Default value: OFF (0)

value: hex bytes, specifying the six bytes of the address. Default value: 0x000000000000

mask: hex bytes, specifying the mask corresponding to the address. Default value: 0xFFFFFFFFFFFF

Summary: set and get, flow index, filter type, value type: B, HHHHHH, HHHHHH

Example: 0/0 PEF_ ETHDESTADDR [5,0] OFF 0x000000000000 0xFFFFFFFFFFFF

PEF_L2PUSE [fid, filter_type] use

Basic mode only. Defines what Layer 2+ protocols that are present and may be used for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the presence of Layer 2+ protocols:

- NA (0) (No Layer 2+ protocols)
- VLAN1 (1) (1 VLAN Tag is present)
- VLAN2 (2) (2 VLAN Tags are present)
- MPLS (3) (MPLS label is present)
- Default value: NA (0)

Summary: set and get, flow index, filter type, value type: B

Example: 0/0 PEF_L2PUSE [5,0] VLAN1

PEF_VLANSETTINGS [fid, filter_type] usage action

Basic mode only. Defines what filter action is performed on the VLAN header.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

usage: coded byte, specifies if VLAN information is expected:

- OFF (0) (No filtering will be done on this header)
- AND (1) (AND-filtering will be done on this header)
- Default value: OFF (0)

action: coded byte, specifies the usage of VLAN information:

- EXCLUDE (0) (Matching packets are excluded from the flow)
- INCLUDE (1) (Matching packets are included in the flow)
- Default value: EXCLUDE (0)

Summary: set and get, flow index, filter type, value type: B, B

Example or get: 0/0 PEF_ VLANSETTINGS [5,0] AND INCLUDE

PEF_VLANTAG [fid, filter_type, vlan_type] use value mask

Basic mode only. Defines the VLAN TAG settings for the VLAN filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

vlan_type: coded byte, the sub-index value specifies the VLAN type:

- VLAN1 (0) (INNER VLAN Tag is specified for the filter – used also when only 1 VLAN)

- Indicates single/inner VLAN-TPID=0x8100

- VLAN2 (1) (OUTER VLAN Tag is specified for the filter)
    - Indicates outer VLAN-TPID=0x88A8

use: coded byte, specifies the usage of VLAN information:

- OFF (0) (VLAN Tag is not used for the filter)
- ON (1) (VLAN Tag is used for the filter)
- Default value: OFF (0)

value: decimal digits, specifying the 12 bit value of the tag. Default value: 0

mask: hex digits, specifying the 12 bit value of the tag. Default value: 0x0FFF

Summary: set and get, flow index, filter_type, vlan_type, value type: B, I, HH

Example: 0/0 PEF_ VLANTAG [5,0,0] OFF 1234 0x0FFF

PEF_VLANPCP [fid, filter_type, vlan_type] use value mask

Basic mode only. Defines the VLAN PCP settings for the VLAN filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

vlan_type: coded byte, the sub-index value specifies the VLAN type:

- VLAN1 (0) (VLAN1INNER VLAN Tag is specified for the filter)
    - Indicates single/inner VLAN-TPID=0x8100
- VLAN2 (1) (OUTER VAN Tag is specified for the filter)
    - Indicates outer VLAN-TPID=0x88A8

use: coded byte, specifies the usage of VLAN information:

- OFF (0) (VLAN PCP is not used for the filter)
- ON (1) (VLAN PCP is used for the filter)
- Default value: OFF (0)

value: 1 byte, specifying the value of the PCP. Default value: 0 (Range: 0 to 7)

mask: hex digit, specifying the 8 bit value mask. Default value: 0x7

Summary: set and get, flow index, filter_type, vlan_type, value type: B, B, H

Example: 0/0 PEF_ VLANPCP [5,0,0] OFF 5 0x7

PEF_MPLSSETTINGS [fid,filter_type] usage action

Basic mode only. Defines what filter action is performed on the MPLS header.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

usage: coded byte, specifies the usage of MPLS information:

- OFF (0) (MPLS information will not be used for the flow filtering)
- AND (1) (Filtered MPLS information will be used for flow filtering)
- Default value: OFF (0)

action: coded byte, specifies the usage of MPLS information:

- EXCLUDE (0) (Matching packets are excluded from the flow)
- INCLUDE (1) (Matching packets are included in the flow)
- Default value: INCLUDE (1)

Summary: set and get, flow index, filter type, value type: B, B

Example: 0/0 PEF_ MPLSSETTINGS [5,0] AND INCLUDE

PEF_MPLSLABEL [fid, filter_type] use value mask

Basic mode only. Defines the MPLS label settings for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of MPLS information:

- OFF (0) (MPLS label is not used for the filter)
- ON (1) (MPLS label is used for the filter)
- Default value: OFF (0)

value: decimal digits, specifying the 20 bit value of the label. Default value: 0

mask: hex digits, specifying the 20 bit value of the label. Default value: 0x0FFFFF

Summary: set and get, flow index, filter type, value type: B, I, HHH

Example: 0/0 PEF_ MPLSLABEL [5,0] ON 1234 0xFFFFF

PEF_MPLSTOC [fid, filter_type] use value mask

Basic mode only. Defines the MPLS TOC settings for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of MPLS TOC information:

- OFF (0) (MPLS TOC is not used for the filter)
- ON (1) (MPLS TOC is used for the filter)
- Default value: OFF (0)

value: 1 byte, specifying the value of the MPLS TOC. Default value: 0 (Range: 0 to 7)

mask: hex digit, specifying the filter mask for the value of the MPLS TOC. Default value: 0x7

Summary: set and get, flow index, filter type, value type: B, B, H

Example:

0/0 PEF_MPLSTOC [5,0] OFF 6 0x07

PEF_L3USE [fid, filter_type] use

Basic mode only. Defines what Layer 3 protocols that are present and may be used for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the presence of Layer 3 protocols:

- NA (0) (No Layer 3 protocols)
- IP4 (1) (IPv4 is present)
- IP6 (2) (IPv6 is present)
- Default value: NA (0)

Summary: set and get, flow index, filter type, value type: B

Example: 0/0 PEF_L3USE [5,0] IP4

PEF_IPV4SETTINGS [fid, filter_type] usage action

Basic mode only. Defines what filter action is performed on the IPv4 information.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

usage: coded byte, specifies the usage of IPv4 information:

- OFF (0) (IPv4 sub-filter is disabled and not used for filtering)
- AND (1) (Filtered IPv4 sub-filter is enabled and used for the flow filtering)
- Default value: OFF (0)

action: coded byte, specifies the usage of IPv4 information:

- EXCLUDE (0) (Packets which have IPv4 header, matching the sub-filter are excluded from the flow)
- INCLUDE (1) (Packets which have IPv4 header, matching the sub-filter are included in the flow)
- Default value: INCLUDE (1)

Summary: set and get, flow index, filter type, value type: B, B

Example: 0/0 PEF_ IPV4SETTINGS [5,0] AND INCLUDE

PEF_IPV4SRCADDR [fid, filter_type] use value mask

Basic mode only. Defines the IPv4 Source Address settings for the IPv4 filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of IPv4 information:

- OFF (0) (IPv4 Source Address is not used for the filter)
- ON (1) (IPv4 Source Address is used for the filter)
- Default value: OFF (0)

value: dot separated digits, specifying the four bytes of the address. Default value: 0.0.0.0

mask: hex bytes, specifying the filter mask of the value.. Default value: 0xFFFFFFFF

Summary: set and get, flow index, filter type, value type: B, A, HHHH

Example: 0/0 PEF_ IPV4SRCADDR [5,0] OFF 192.168.1.2 0xFFFFFFFF

PEF_IPV4DESTADDR [fid, filter_type] use value mask

Basic mode only. Defines the IPv4 Destination Address settings for the IPv4 filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of IPv4 information:

- OFF (0) (IPv4 Destination Address is not used for the filter)
- ON (1) (IPv4 Destination Address is used for the filter)
- Default value: OFF (0)

value: dot separated digits, specifying the four bytes of the address. Default value: 0.0.0.0

mask: hex bytes, specifying the mask corresponding to the address. Default value: 0xFFFFFFFF

Summary: set and get, flow index, filter type, value type: B, A, HHHH

Example: 0/0 PEF_ IPV4DESTADDR [5,0] OFF 192.168.1.2 0xFFFFFFFF

PEF_IPV4DSCP [fid] use value mask

Basic mode only. Defines if IPv4 DSCP/TOS settings used for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of IPv4 DSCP/TOS information:

- OFF (0) (IPv4 DSCP/TOS is not used for the filter)
- ON (1) (IPv4 DSCP/TOS is used for the filter)
- Default value: OFF (0)

value: 1 byte, specifying the value of the IPv4 DSCP/TOS in the upper 6 bits.

value[7:2] = DSCP/TOS.

value[1:0] = reserved (must be zero)

Default value: 0

mask: hex bytes, specifying the filter mask of the value in the upper 6 bits.

mask[7:2] = DSCP/TOS mask

mask[1:0] = reserved (must be zero)

Default value: 0xFC

Summary: set and get, flow index, filter type, value type: B, B, H

Example: 0/0 PEF_IPV4DSCP [5,0] OFF 15 0xFC

PEF_IPV6SETTINGS [fid, filter_type] usage action

Basic mode only. Defines what filter action is performed on the IPv6 information.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

usage: coded byte, specifies the usage of IPv6 information:

- OFF (0) (IPv6 sub-filter is disabled and not used for the flow filtering)
- AND (1) (IPv6 sub-filter is enabled and will be used for flow filtering)
- Default value: OFF (0)

action: coded byte, specifies the usage of IPv6 information:

- EXCLUDE (0) (Matching packets are excluded from the flow)
- INCLUDE (1) (Matching packets are included in the flow)
- Default value: INCLUDE (1)

Summary: set and get, flow index, filter type, value type: B, B

Example: 0/0 PEF_ IPV6SETTINGS [5,0] AND INCLUDE

PEF_IPV6SRCADDR [fid, filter_type] use value mask

Basic mode only. Defines the IPv6 Source Address settings for the IPv4 filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of IPv6 information:

- OFF (0) (IPv6 Source Address is not used for the filter)
- ON (1) (IPv6 Source Address is used for the filter)
- Default value: OFF (0)

value: hex bytes, specifying the address. Default value: 0x00000000000000000000000000000000

mask: hex bytes, specifying the six first bytes of the address. Default value:
0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Summary: set and get, flow index, filter type, value type:

B, HHHHHHHHHHHHHHHH, HHHHHHHHHHHHHHHH

Example: 0/0 PEF_IPV6SRCADDR [5,0] OFF 0x00000000000000000000000000000000
0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

PEF_IPV6DESTADDR [fid, filter_type] use value mask

Basic mode only. Defines the IPv6 Destination Address settings for the IPv6 filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of IPv6 information:

- OFF (0) (IPv6 Destination Address is not used for the filter)
- ON (1) (IPv6 Destination Address is used for the filter)
- Default value: OFF (0)

value: hex bytes, specifying the address. Default value: 0x00000000000000000000000000000000

mask: hex bytes, specifying the six first bytes of the address. Default value:
0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Summary: set and get, flow index, filter type, value type:

B, HHHHHHHHHHHHHHHH, HHHHHHHHHHHHHHHH

Example: 0/0 PEF_IPV6DESTADDR [5,0] OFF 0x00000000000000000000000000000000
0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

PEF_IPV6TC [fid, filter_type] use value mask

Basic mode only. Defines the IPv6 Traffic Class settings used for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of IPv6 Traffic Class information:

- OFF (0) (IPv6 Traffic Class is not used for the filter)
- ON (1) (IPv6 Traffic Class is used for the filter)
- Default value: OFF (0)

value: 1 byte, specifying the value of the IPv6 Traffic Class in the upper 6 bits.

value[7:2] = IPv6 Traffic Class.

value[1:0] = reserved (must be zero)

Default value: 0

mask: hex byte, specifying the filter mask for the value in the upper 6 bits.

mask[7:2] = IPv6 Traffic Class mask

mask[1:0] = reserved (must be zero)

Default value: 0xFC

Summary: set and get, flow index, filter type, value type: B, B, H

Example: 0/0 PEF_IPV6TC [5,0] OFF 15 0xFC

PEF_UDPSETTINGS [fid, filter_type] usage action

Basic mode only. Controls if UDP packet information is used for flow filtering.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

usage: coded byte, specifies the usage of UDP information:

- OFF (0) (UDP information will not be used for the flow filtering)
- AND (1) (Filtered UDP information will be used for flow filtering)
- Default value: OFF (0)

action: coded byte, specifies the usage of IPv6 information:

- EXCLUDE (0) (Matching packets are excluded from the flow)
- INCLUDE (1) (Matching packets are included in the flow)

- Default value: INCLUDE (1)

Summary: set and get, flow index, filter type, value type: B, B

Example: 0/0 PEF_UDPSETTINGS [5,0] AND INCLUDE

PEF_UDPSRCPORT [fid, filter_type] use value mask

Basic mode only. Defines UDP Source Port settings used for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of UDP Source Port information:

- OFF (0) (UDP Source Port is not used for the filter)
- ON (1) (UDP Source Port is used for the filter)
- Default value: OFF (0)

value: integer , specifying the value of the UDP Source Port. Default value: 0

mask: hex digits , specifying the filter mask for the value. Default value: 0xFFFF

Summary: set and get, flow index, filter type, value type: B, B, HH

Example: 0/0 PEF_UDPSRCPORT [5,0] OFF 10 0xFFFF

PEF_UDPDESTPORT [fid, filter_type] use value mask

Basic mode only. Defines UDP Destination Port settings used for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of UDP information:

- OFF (0) (UDP Destination Port is not used for the filter)
- ON (1) (UDP Destination Port is used for the filter)

- Default value: OFF (0)

value: integer , specifying the value of the UDP Destination Port. Default value: 0

mask: hex digits , specifying the filter mask for the value. Default value: 0xFFFF

Summary: set and get, flow index, filter type, value type: B, B, HH

Example: 0/0 PEF_UDPDESTPORT [5,0] OFF 10 0xFFFF

PEF_TCPSETTINGS [fid, filter_type] usage action

Basic mode only. Defines if filtering on TCP information is used for flow filtering.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

usage: coded byte, specifies the usage of TCP information:

- OFF (0) (TCP information will not be used for the flow)
- AND (1) (Filtered TCP information will be used for flow filtering)
- Default value: OFF (0)

action: coded byte, specifies the usage of IPv6 information:

- EXCLUDE (0) (Matching packets are excluded from the flow)
- INCLUDE (1) (Matching packets are included in the flow)
- Default value: INCLUDE (1)

Summary: set and get, flow index, filter type, value type: B, B

Example: 0/0 PEF_TCPSETTINGS [5,0] AND INCLUDE

PEF_TCPSRCPORT [fid, filter_type] use value mask

Basic mode only. Defines TCP Source Port settings used for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of TCP information:

- OFF (0) (TCP Source Port is not used for the filter)
- ON (1) (TCP Source Port is used for the filter)
- Default value: OFF (0)

value: integer, specifying the value of the TCP Source Port. Default value: 0

mask: hex digits , specifying the filter mask for the value. Default value: 0xFFFF

Summary: set and get, flow index, filter type, value type: B

Example: 0/0 PEF_TCPSRCPORT [5] OFF 10 0xFFFF

PEF_TCPDESTPORT [fid, filter_type] use value mask

Basic mode only. Defines TCP Destination Port settingsused for the filter.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

use: coded byte, specifies the usage of TCP information:

- OFF (0) (TCP Destination Port is not used for the filter)
- ON (1) (TCP Destination Port is used for the filter)
- Default value: OFF (0)

value: integer , specifying the value of the TCP Destination Port. Default value: 0

mask: hex digits , specifying the filter mask for the value. Default value: 0xFFFF

Summary: set and get, flow index, filter type, value type: B, B, HH

Example: 0/0 PEF_TCPDESTPORT [5,0] OFF 10 0xFF

PEF_ANYSETTINGS [fid, filter_type] usage action

Basic mode only. Defines if filtering on ANY field in a packet is used for flow filtering.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

usage: coded byte, specifies the usage of ANY field information:

- OFF (0) (ANY field information will not be used for the flow filtering)
- AND (1) (Filtered ANY field information will be used for flow filtering)
- Default value: OFF (0)

action: coded byte, specifies the usage of IPv6 information:

- EXCLUDE (0) (Matching packets are excluded from the flow)
- INCLUDE (1) (Matching packets are included in the flow)
- Default value: INCLUDE (1)

Summary: set and get, flow index, filter type, value type: B, B

Example: 0/0 PEF_ANYSETTINGS [5,0] AND INCLUDE

PEF_ANYCONFIG [fid, filter_type] position value mask

Basic mode only. Defines the ANY field filter configuration. The “ANY field” filter will match 6 consecutive bytes in the incoming packets at a programmable offset. Applying a mask, allows to only filter based on selected bits within the 6 bytes.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

position: byte, specifies the start position of the ANY field. Default value: 0, Range:0-127

value: hex bytes, specifying the six bytes of the field. Default value: 0x000000000000

mask: hex bytes, specifying the six bytes of the field. Default value: 0xFFFFFFFFFFFF

Summary: set and get, flow index, filter type, value type: B, HHHHHH, HHHHHH

Example: 0/0 PEF_ANYCONFIG [5,0] 12 0x8100 0x00000000FFFF

PEF_TPLDSETTINGS [fid, filter_type] action

Defines if filtering on TPLD field in a packet is used for flow filtering. The TPLD filter allows filtering based on the Xena Testpayload ID. The Testpayload ID is meta data, which can be inserted into the Ethernet packets by Xena traffic generators. Each flow filter, can filter based on 16 TPLD ID values.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

action: coded byte, specifies the usage of IPv6 information:

- EXCLUDE (0) (Matching TPLD field contained packets are excluded from the flow)
- INCLUDE (1) (Matching TPLD field contained packets are included in the flow)
- Default value: INCLUDE (1)

Summary: set and get, flow index, filter type value type: B

Example: 0/0 PEF_TPLDSETTINGS [5,0] INCLUDE

PEF_TPLDCONFIG [fid, filter_type, tpld_index] usage id

Defines the TLPD filter configuration.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

NOTE:

- For SET, only allowed fid_type is “shadow-copy”(0)

tpld_index: integer, the sub-index value which indicates the tpld filter index (range 0 to 15)

usage: coded byte, specifies the usage of ANY field information:

- OFF (0) (TPLD field information will not be used for the flow)
- ON (1) (Filtered TPLD field information will be used for the flow)
- Default value: OFF (0)

id: int, specifies the TPLD ID. Range: 0-2015, Default value: 0

Summary: set and get, flow index, filter type, tpld index value type: B,I

Example: 0/0 PEF_TPLDCONFIG [5,0,0] ON 1

PEF_PROTOCOL [fid, filter_type] segment_list

Extended mode only. Defines the sequence of protocol segments that can be matched.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

segment_list: List of protocol segment types in the order they are expected in a frame. First segment type must be ETHERNET; the following can be chosen freely from:

Segment type	Length in bytes	Segment type	Length in bytes	Segment type	Length in bytes
ETHERNET	12	SCTP	12	TCPCHECK	20
ETHERTYPE	2	MACCTRL	4	GTPV1L0	8
VLAN	4	MPLS	4	GTPV1L1	12
ARP	28	PBBTAG	4	GTPV2L0	8
IP	20	FCOEHEAD	14	GTPV2L1	12
IPV6	40	FC	24	IGMPV1	8
UDP	8	FCOETAIL	4	PWETHCTRL	4
TCP	20	IGMPV3L0	12	VXLAN	8
LLC	3	IGMPV3L1	16	ETHERNET_8023	14
SNAP	5	UDPCHECK	8	NVGRE	8
GTP	20	IGMPV2	8	GENEVE	8
ICMP	8	MPLS_TP_OAM	8	MACCTRL_PFC	20
RTP	12	GRE_NOCHECK	4	ECPRI	8
RTCP	4	GRE_CHECK	8	ETHERTYPE	2
STP	35	ROE		*-n* (raw segment)	*+n*

The total length of the specified segments cannot exceed 128 bytes.

If an existing sequence of segments is changed (using PEF_PROTOCOL) the underlying value and mask bytes remain unchanged, even though the semantics of those bytes may have changed. However, if the total length, in bytes, of the segments is reduced, then the excess bytes of value and mask are set to zero.

I.e. to update an existing filter, you must first correct the list of segments (using PEF_PROTOCOL) and subsequently update the filtering value (using PEF_VALUE) and filtering mask (PEF_MASK).

Summary: set and get, flow index, filter type, segment list

Example: 0/0 PEF_PROTOCOL [5,0] ETHERNET VLAN ETHERTYPE ECPRI

PEF_VALUE [fid, filter_type, segment_index] value

Extended mode only. Defines byte values that can be matched if selected by the mask; see PEF_MASK.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

segment_index: integer, the index into list of segments specified by PEF_PROTOCOL, starting with the first ETHERNET at index 1, or 0 for the full 128 byte of match values.

If *segment_index* is zero, the maximum number of match value bytes that can be set is determined by the total length of the segments specified with PEF_PROTOCOL. E.g. if PEF_PROTOCOL is set to ETHERNET then only 12 bytes can be set. In order to set the full 128 bytes, either specify a detailed segment list, or use the raw segment type, e.g.

0/0 PEF_PROTOCOL [5,0] ETHERNET -116

This specifies 12 + 116 = 128 bytes.

If *segment_index* is non-zero, only the bytes covered by that segment are manipulated, so if PEF_PROTOCOL is set to ETHERNET VLAN ETHERTYPE ECPRI, then *segment_index* = 4 selects the 8 bytes of the eCPRI header starting at byte position (12 + 2 + 4) = 18.

For ‘set’ commands where fewer value bytes are provided than specified by the protocol segment, those unspecified bytes are set to zero.

‘get’ commands always returns the number of bytes specified by the segment.

Summary: set and get, flow index, filter type, segment index

Example:

0/0 PEF_PROTOCOL [5,0] ETHERNET VLAN VLAN
0/0 PEF_VALUE [5,0**,0**] 0x000000000000000000000000088a8012381000abc
0/0 PEF_MASK [5,0**,0**] 0x00000000000000000000000ffffffffffffffff
specifies two VLAN tags, the first one with TPID 0x88A8 and VID 0x123; the second with TPID 0x8100 and VID 0xABC. This can also be accomplished with a sequence of commands, selecting each segment index in turn:

0/0 PEF_VALUE [5,0**,1**] 0x00 *sets all 12 bytes to zero*
0/0 PEF_MASK [5,0**,1**] 0x00
0/0 PEF_VALUE [5,0**,2**] 0x88a80123 *value + mask of first VLAN*
0/0 PEF_MASK [5,0**,2**] 0xffffffff
0/0 PEF_VALUE [5,0**,3**] 0x81000abc *value + mask of second VLAN*
0/0 PEF_MASK [5,0**,3**] 0xffffffff

PEF_MASK [fid, filter_type, segment_index] value

Extended mode only. Defines mask byte values that select the values specified by PEF_VALUE.

fid: integer, the sub-index value of the flow definition.

filter_type: integer, the sub-index value which indicates the filter type - “shadow-copy”(0) or “working-copy”(1).

segment_index: integer, the index into list of segments specified by PEF_PROTOCOL, starting with the first ETHERNET at index 1, or 0 for the full 128 byte of mask values. See description under PEF_VALUE for more details.

For a chosen *segment_index* the first byte in the value masks the first byte of the corresponding PEF_VALUE, and so on.

Get/set semantics are similar to PEF_VALUE; please see description there for details.

Summary: set and get, flow index, filter type, segment index

Example:

0/0 PEF_VALUE [5,0,3] 0x81000abc

0/0 PEF_MASK [5,0,3] 0x00000fff

PEF_CONFIG [fid, filter_type]

Multi-parameter query, obtaining all the settable filter parameters for a flow per port.

Summary: get only.

Example: 8/0 PEF_CONFIG[1,0]?

0/0 PEF_INIT

0/0 PEF_MODE [0, 1] BASIC

0/0 PEF_IPV4DSCP[0,1] ON 63

… <etc>

Flow Impairments Script Commands

Indices Used

fid denotes the flow identifier. It ranges from 0 to 7. (Default Flow is 0)

iid denotes the impairment identifier. Valid values : 0 to 6 (DROP=0, MISO=1, DELAY=2, DUPLICATION=3, CORRUPTION=4, POLICER=5, SHAPER=6)

NOTE: When using script commands, symbolic constants cannot be used for the ‘impairment identifiers’. Use only numbers.

Impairment specific script commands

This defines the impairment specific configuration, if any.

PE_CORRUPT [fid] type

Configures impairment corruption type.

fid: integer, the sub-index value of the flow definition. See Indices Used

type: coded byte, specifying corruption type:

- ETH (1) (Corruption of Ethernet Frame FCS)
- IP(2) (Corruption of IP header Check Sum)
- UDP (3) (Corruption of UDP Check Sum)
- TCP (4) (Corruption of TCP Check Sum)
- Default value: ETH(1)

Summary: set and get, flow index, value type: B

NOTE: IP / TCP / UDP corruption modes are not supported on default flow (0).

Example or get: 0/0 PE_CORRUPT [5] ETH

PE_MISORDER [fid] depth

Configures the misordering depth in number of packets.

fid: integer, the sub-index value of the flow definition. See Indices Used

depth: integer, specifies the misordering depth (Range 1 – 32). Default value: 1

Note: probability [see PED_FIXED] * (depth + 1) should be less than 1,000,000.

Summary: set and get, flow index, value type: I

Example or get: 0/0 PE_MISORDER[5] 10

PE_BANDPOLICER [fid] use mode cir cbs

Configures the policer.

fid: integer, the sub-index value of the flow definition. See Indices Used

use: coded byte, enables/disables policer:

OFF (0) (Policer control is disabled)

ON (1) (Policer control is enabled)

Default value: OFF (0)

mode: coded byte, sets policer mode:

- L1 (0) (Policer performed at Layer 1 level. I.e. including the preamble and min interpacket gap)
- L2 (1) (Policer performed at Layer 2 level. I.e. excluding the preamble and min interpacket gap)
- Default value: L2 (0)

cir: integer, policer rate in units of 100kbps (range 0 to 1000000), default is 0.

cbs: integer, policer burst in bytes (range 0 to 4194304), default is 0.

Note: Because the PE_BANDPOLICER command has a dedicated “use” parameter, PED_ENABLE is not supported for the policer.

Summary: set and get, flow index, impairment type,B,I,I

Example: 0/0 PE_BANDPOLICER [5] ON L1 1000 64

PE_BANDSHAPER [fid] use mode cir cbs buffersize

Configures the shaper.

fid: integer, the sub-index value of the flow definition. See Indices Used

use: coded byte, enables/disables shaper:

OFF (0) (Shaper control is disabled)

ON (1) (Shaper control is enabled)

Default value: OFF (0)

mode: coded byte, sets shaper mode:

- L1 (0) (Shaper performed at Layer 1 level. I.e. including the preamble and min interpacket gap)
- L2 (1) (Shaper performed at Layer 2 level . I.e. excluding the preamble and min interpacket gap)
- Default value: L2 (0)

cir: integer, policer rate in units of 100kbps (range 0 to 1000000), default is 0.

cbs: integer, shaper burst in bytes (range 0 to 4194304), default is 0:

buffersize: integer, shaper buffer size in bytes (range 0 to 2097152), default is 0:

Note: Because the PE_BANDSHAPER command has a dedicated “use” parameter, PED_ENABLE is not supported for the shaper.

Summary: set and get, flow index, impairment type,B,I,I

Example: 0/0 PE_BANDSHAPER [5] ON L1 1000 64 2048

PE_LATENCYRANGE [fid] min_latency max_latency

Old Command Name: PW_LATENCYRANGE

Retrieve minimum and maximum configurable latency per flow in nanoseconds.

fid: integer, the sub-index value of the flow definition. See Indices Used

Summary: get, flow index, value types: L, L

Example: 0/0 PE_LATENCYRANGE [5] 7000 1900000000

Impairments Distribution Script Commands

PED_OFF [fid,iid]

Configuration of “OFF” distribution. Off distribution clears impairment of all configuration and turns off the impairment. To de-activate the impairment assign distribution ‘OFF’.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used

Summary: set and get, flow index, impairment type

Example: 0/0 PED_OFF [5,3]

PED_ENABLE [fid,iid] action

Control if impairment is enabled or dis-abled.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used

action: coded byte

OFF (0) (Impairment disabled)

ON (1) (Impairment enabled)

Note: This command is not applicable for PE_BANDPOLICER and PE_BANDSHAPER because they have a separate ON / OFF parameter.

Summary: set and get, flow index, impairment type, value type: L

Example: 0/0 PED_ENABLE [5,3] ON

PED_CONST[fid,iid] delay

Configuration of “Constant Delay” distribution (DELAY only).

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used (Only valid for impairment type = DELAY)

delay: long, specifies the constant delay/latency time.

Unit is ns (must be in steps of 100ns).

Default value: Minimum supported per speed and FEC mode.

NOTE:

- If the latency is less than minimum latency (see section “Minimum and Maximum Latency”), value is set to minimum latency.
- If the latency is greater than maximum latency (see section “Minimum and Maximum Latency”), value is set to maximum latency.

Summary: set and get, flow index, impairment type, value type: L

Example: 0/0 PED_CONST [5,2] 90000

PED_ACCBURST[fid,iid] delay

Configuration of “Accumulate & Burst” distribution (DELAY only).

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used (Only valid for impairment type = DELAY)

delay: long, specifies the burst delay time.

Units = ns (must in steps of 100 ns).

Default value: minimum latency

Note:

- If the delay is less than minimum latency (see section “Minimum and Maximum Latency”), value is set to minimum latency.
- If the delay is greater than maximum latency (see “ see section “Minimum and Maximum Latency”), value is set to maximum latency.

Summary: set and get, flow index, impairment type, value type: L

Example: 0/0 PED_ACCBURST [5,2] 15000

PED_STEP[fid,iid] low high

Configuration of “Step” distribution (DELAY only).

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used(Only valid for impairment type = DELAY)

low: long, specifies the packet delay in the ‘low’ state of the step

Units = ns (must be in steps of 100ns).

Default value: minimum latency.

high: long, specifies the packet delay in the ‘high’ state of the step.

Units = ns (must be in steps of 100ns).

Default value: minimum latency.

Note:

- If the low/high is less than minimum latency (see section “Minimum and Maximum Latency”), value is set to minimum latency.
- If the low/high is greater than maximum latency (see section “Minimum and Maximum Latency”), value is set to maximum latency.

Summary: set and get, flow index, impairment type, value type: L,L

Example: 0/0 PED_STEP [5,2] 15000 16000000000

PED_FIXED [fid,iid] probability

Configuration of “Fixed Rate” distribution. This is predictable distribution with nearly equal distance between impairments, to match the configured probability.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except DELAY)

probability: integer, specifies the fixed probability in % * 10000. Default value: 0.

Note:

In case of Mis-ordering a special limit applies:

probability * (depth + 1) should be less than 1000000 (see PE_MISORDER [fid] depth).

Summary: set and get, flow index, impairment type, value type: I

Example: 0/0 PED_FIXED [5,3] 1234

PED_RANDOM [fid,iid] probability

Configuration of "Random Rate" distribution. Packets are impaired randomly based on a per packet probability. This way the impaired fraction of packets will be equal to the configured probability over time.

Random probability in % * 10000 (i.e. 1 means 0.0001%)

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except DELAY and MISO).

probability: integer, specifies the random probability in % * 10000. Default value: 0.

Summary: set and get, flow index, impairment type, value type: I.

Example: 0/0 PED_RANDOM [5,3] 10.

PED_ONESHOTSTATUS[fid,iid] one_shot_status

Retrieves the one-shot completion status.

Note: The return value is only valid, if the configured distribution is either accumulate & burst (DELAY) or fixed burst (non-DELAY).

Summary: get only, flow index, impairment type, value type: B

Example: 0/0 PED_ONESHOTSTATUS [5,3] ?

0/0 PED_ONESHOTSTATUS [5,3] 0

PED_FIXEDBURST [fid,iid] burstsize

Configuration of "Fixed Burst" distribution.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except DELAY)

burstsize: integer, specifies the burst size (Range 1 – 16383). Default value: 1

NOTE: In case of iid = MISO, burstsize is fixed to 1

Summary: set and get, flow index, impairment type, value type: I

Example: 0/0 PED_FIXEDBURST [5,3] 1

PED_RANDOMBURST [fid,iid] minimum maximum probability

Configuration of "Random Burst" distribution.

fid: integer, the sub-index value of the flow definition. See Indices Used.

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except DELAY and MISO)

minimum: integer, specifies minimum burst size. Default value: 0 (Range 0 to 65535)

maximum: integer, specifies maximum burst size. Default value: 0 (Range 0 to 65535)

probability: integer, specifies the per packet probability of initiating a burst in % * 10000. Default value: 0.

Summary: set and get, flow index, impairment type, value type: I,I,I

Example: 0/0 PED_RANDOMBURST[5,3] 10 100 50

PED_GE [fid,iid] goodprob goodtransprob badprob badtransprob

Configuration of "Gilbert-Elliot" distribution.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except DELAY and MISO)

goodprob: integer, specifies the good state probability in % * 10000. Default value: 0.

goodtransprob: integer, specifies the good state transition probability in % * 10000. Default value: 0.

badprob: integer, specifies the bad state probability in % * 10000. Default value: 0.

badtransprob: integer, specifies the bad state transition probability in % * 10000. Default value: 0.

Summary: set and get, flow index, impairment type, value type: I,I,I,I

Example: 0/0 PED_GE [5,3] 10 100 10 100

PED_BER [fid,iid] coef exp

Configuration of "Bit Error Rate" distribution.

fid: integer, the sub-index value of the flow definition. See Indices Used.

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except DELAY and MISO).

coef: integer, specifies the coeffient for BER. Default value: 1 (Range is 1 to 9).

exp: integer, specifies the exponent for BER. Default value: -10 (Range is -18 to -1).

Summary: set and get, flow index, impairment type, value type: I,I

Example: 0/0 PED_BER [5,3] 1 -10

PED_UNI [fid,iid] minimum maximum

Configuration of “Uniform” distribution.

fid: integer, the sub-index value of the flow definition. See Indices Used.

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except MISO).

minimum: long,

In case of iid != DELAY, specifies the minimum no. of packets. Default value: 0 (Range 0 to 4194288)

In case of iid = DELAY, specifies the minimum latency limit.

Unit is ns (must be in steps of 100ns).

Default value: minimum latency (see Minimum supported per speed and FEC mode).

Note:

If minimum is less than minimum latency (see section “Minimum and Maximum Latency”), value is set to minimum latency.

If minimum is greater than maximum latency (see section “Minimum and Maximum Latency”), value is set to maximum latency.

maximum: long,

In case of iid != DELAY, specifies the maximum no. of packets . Default value: 0 (Range 0 to 4194288)

In case of iid = DELAY, specifies the maximum latency limit.

Units is ns (must be in steps of 100ns).

Default value: minimum latency (Maximum supported per speed and FEC mode).

Summary: set and get, flow index, impairment type, value type: L,L

Example: 0/0 PED_UNI [5,3] 10 100

PED_GAUSS [fid,iid] mean std_deviation

Configuration of “Gaussian” distribution.

fid: integer, the sub-index value of the flow definition. See Indices Used.

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except MISO)

mean: long, specifies the Gaussian mean:

In case of iid != DELAY, specifies the Gaussian mean value as number of packets.

Default value: 0 packets (Range 0 to 4194288).

In case of iid = DELAY, specifies the Gaussian mean value.

Units is ns (must be in steps of 100ns).

Default value: minimum latency (Maximum supported per speed and FEC mode).

std_deviation: long, specifies the Gaussian standard deviation:

In case of iid != DELAY, specifies the standard deviation as number of packets.

Default value: 0 packets (Range 0 to 4194288).

In case of iid = DELAY, specifies the the Gaussian standard deviation.

Units is ns (must be in steps of 100ns).

Default value: 0 ns.

Note:

In case of iid != DELAY,

- Mean plus 3 times standard deviation shoule be less than or equal to max allowed (4194288).
- Mean should always be at least 3 times the standard deviation, this to ensure that the impairment distance is always positive.

In case of iid = DELAY,

- mean plus 3 times standard deviation shoule be less than or equal to the maximum latency
- mean minus 3 times the standard deviation should be greater than or equal to minimum latency (see see section “Minimum and Maximum Latency”)

Summary: set and get, flow index, impairment type, value type: L,L

Example: 0/0 PED_GAUSS [5,3] 90 30

PED_POISSON [fid,iid] mean

Configuration of “Poisson” distribution.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except MISO).

mean: long, specifies the Poisson mean value:

In case of iid = DELAY specifies the Poisson mean.

Unit is ns (must be in steps of 100ns).

Default value: 0 ns

In case of iid != DELAY specifies the Poisson mean in number of packets packets.

Default value: 9 packets (Range 0 to 4194288).

Note:

standard deviation is derived from mean. Ie., standard deviation = SQRT(mean)

In case of iid != DELAY,

- mean plus 3 times standard deviation shoule be less than or equal to max allowed (4194288)

In case of iid = DELAY,

- mean plus 3 times standard deviation should be less than or equal to the maximum latency.

Summary: set and get, flow index, impairment type, value type: L

Example: 0/0 PED_POISSON [5,3] 90

PED_GAMMA [fid, iid] shape scale

Configuration of “Gamma” distribution.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used (Valid for all impairment types except MISO)

shape: long, specifies the shape:

Units: none.

Default value: 0

scale: long, specifies the Gamma function scaleparameter

In case of iid = DELAY,

Units: ns (must be in steps of 100 ns).

Default value: 0 ns.

In case of iid != DELAY,

Units: Number of packets.

Default value: 0 packets.

Note:

Mean and Standard deviation are calculated from Shape and Scale parameters and validation is performed using those.

standard deviation = [SQRT(shape * scale * scale)]

mean = [shape * scale]

In case of iid != DELAY,

- mean plus 4 times standard deviation shoule be less than or equal to max allowed(4194288)
- shape and scale should be greater than or equal to 0.

In case of iid = DELAY,

- mean plus 4 times standard deviation shoule be less than or equal to the maximum latency

Summary: set and get, flow index, impairment type, value type: L,L

Example: 0/0 PED_GAMMA [5,3] 100 700

PED_CUST [fid,iid] cust_id

Associate a custom distribution to a flow and impairment type.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used

cust_id: integer, custom distribution identifier

Before associating a custom distribution, the below validation checks are applied.

In case of iid != DELAY,

- Custom values shoule be less than or equal to max allowed (4194288).
- Custom distribution bust contain 512 values.

In case of iid = DELAY,

- Custom values shoule be less than or equal to the maximum latency
- Custom values should be greater than or equal to minimum latency (see section “Minimum and Maximum Latency”).
- Custom distribution should contain 1024 values.

Summary: set and get, flow index, impairment type, value type: I

Example: 0/0 PED_CUST[5,3] 5

PED_GET[fid, iid]

Retrieves the enabled impairment distribution information for a particular type of impairment.

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used

Summary: get only, flow index, impairment type

Example: ;8/0 PED_GET[0,0]?

8/0 PED_RANDOM [0,0] 10000

Impairments distribution scheduler

PED_SCHEDULE [fid,iid] duration period

Configure the impairment scheduler function.

The configuration of the scheduler depends on the kind of distribution to schedule:

- Burst distributions: “Fixed Burst” and “Accumulate and Burst”.
- Non-Burst distributions: All others

For burst distributions, the scheduler can be configured for “One-shot” operation or “Repeat Operation”. When running in “Repeat Operation” the “Repeat Period” must be configured.

For non-burst distributions, the scheduler can be configured operate in either “Continuous” or “Repeat Period” modes. When running in “Repeat Period” configuration of “Duration” and “Repeat Period” is required.

See the table below for the valid combinations

Distribution Type	State	Duration	Period
Burst	One-shot	1	0
Burst	Repeat	1	“Repeat Period”
Non-Burst	Continuous	1	0
Non-Burst	Repeat pattern	“Duration”	“Repeat Period”

fid: integer, the sub-index value of the flow definition. See Indices Used

iid: integer, the sub-index value of the impairment type. See Indices Used

duration: integer, specifies the “on” period. Units = steps of 10ms (range 1 to 65535), default is 1

period: integer, specifies the “total” period. Units = steps of 10ms (range 0 to 65535), default is 0

Summary: set and get, flow index, value type: I,I

Example: 8/0 PED_SCHEDULE [5,0] 10 50

Statistics Script Commands

The statistics parameters correspond to the Emulate Transmit Statistics, Emulate Receive Statistics and Emulate Impairment Statistics of the XenaManager, and provide quantitative information about the transmitted, received and impaired packets on a port.

The statistics parameter names all have the form **PT_<xxx>, PR_<xxx>** and **PI_<xxx>** and require both a module index and a port index. Those parameters dealing with a specific flow also have a sub-index identifying the flow.

Flow Statistics

PT_FLOWCLEAR[fid]

Clear all the transmit statistics on a particular flow for a Chimera port. The byte and packet counts will restart at zero.

fid: integer, the sub-index value of the flow definition.

Summary: set only.

Example: 0/0 PT_FLOWCLEAR

PR_FLOWCLEAR[fid]

Clear all the receive statistics on a particular flow for a Chimera port. The byte and packet counts will restart at zero.

fid: integer, the sub-index value of the flow definition.

Summary: set only.

Example: 0/0 PR_FLOWCLEAR

PT_FLOWTOTAL [fid] bps2 pps bytes packets

Obtains statistics concerning all the packets transmitted from a between this receive port and its partner TX port.

fid: integer, the sub-index value of the flow definition.

bps2: long integer, number of bits transmitted at layer 2 in the last second for the flow.

pps: long integer, number of packets transmitted in the last second for the flow.

bytes: long integer, number of bytes transmitted since statistics were cleared for the flow.

packets: long integer, number of packets transmitted since statistics were cleared for the flow.

Summary: get only, flow index, value types: L, L, L, L

Example: 0/0 PT_FLOWTOTAL [5] ?

0/0 PT_FLOWTOTAL [5] 8000000000 15000000 12345678987654 123456789876

PR_FLOWTOTAL [fid] bps2 pps bytes packets

Obtains statistics concerning all the packets received from a flow between this receive port and its partner TX port.

fid: integer, the sub-index value of the flow definition.

bps2: long integer, number of bits received at layer 2 in the last second for the flow.

pps: long integer, number of packets received in the last second for the flow.

bytes: long integer, number of bytes received since statistics were cleared for the flow.

packets: long integer, number of packets received since statistics were cleared for the flow.

Summary: get only, flow index, value types: L, L, L, L

Example: 0/0 PR_FLOWTOTAL [5] ?

0/0 PR_FLOWTOTAL [5] 8000000000 15000000 12345678987654 123456789876

PE_CLEAR

Clear all the impairment (duplicate, drop, mis-ordered, corrupted, latency and jitter) statistics for a Chimera port and flows on the port. The byte and packet counts will restart at zero.

Summary: set only.

Example: 0/0 PE_CLEAR

PE_DUPTOTAL packets ratio

Obtains statistics concerning all the packets duplicated between this receive port and its partner

TX port.

packets: long integer, number of packets duplicated in all flows.

ratio: long integer, ratio of number of packets duplicated in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

Summary: get only, value types: L, L

Example: 0/0 PE_DUPTOTAL ?

0/0 PE_DUPTOTAL 100000 1

PE_DROPTOTAL packets prog band other ratio progratio bandratio otherratio

Obtains statistics concerning all the packets dropped between this receive port and its partner

TX port.

packets: long integer, total number of packets dropped in all flows.

prog: long integer, total number of packets dropped as programmed in all flows.

band: long integer, total number of packets dropped due to bandwidth control in all flows.

other: long integer, total number of packets dropped for other reasons in all flows.

ratio: long integer, ratio of number of packets dropped in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

progratio: long integer, ratio of number of packets dropped as programmed in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

bandratio: long integer, ratio of number of packets dropped due to bandwidth control in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

otherratio: long integer, ratio of number of packets dropped for other reasons in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

Summary: get only, value types: L, L, L, L, L, L, L, L

Example: 0/0 PE_DROPTOTAL ?

0/0 PE_DROPTOTAL 100000 100000 100000 100000 100000 100000 100000 100000

PE_MISTOTAL packets ratio

Obtains statistics concerning all the packets mis-ordered between this receive port and its partner

TX port.

packets: long integer, number of packets mis-ordered in all flows.

ratio: long integer, number of packets mis-ordered in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

Summary: get only, value types: L, L

Example: 0/0 PE_MISTOTAL ?

0/0 PE_MISTOTAL 100000 1

PE_CORTOTAL packets fcs ip udp tcp ratio fcsratio ipratio udpratio tcpratio

Obtains statistics concerning all the packets corrupted on between this receive port and its partner

TX port.

packets: long integer, number of packets corrupted in all flows.

fcs: long integer, number of packets with Ethernet FCS corrupted in all flows.

ip: long integer, number of packets with IP Header check sum corrupted in all flows.

udp: long integer, number of packets with UDP check sum corrupted in all flows.

tcp: long integer, number of packets with TCP check sum corrupted in all flows.

ratio: long integer, ratio of number of packets corrupted in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

fcsratio: long integer, ratio of number of packets with Ethernet FCS corrupted in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

ipratio: long integer, ratio of number of packets with IP Header check sum corrupted in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

udpratio: long integer, ratio of number of packets with UDP check sum corrupted in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

tcpratio: long integer, ratio of number of packets with TCP check sum corrupted in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

Summary: get only, value types: L, L, L, L, L, L, L, L, L, L

Example: 0/0 PE_CORTOTAL?

0/0 PE_CORTOTAL 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000

PE_JITTERTOTAL packets ratio

Obtains statistics concerning all the packets jittered between this receive port and its partner TX port.

packets: long integer, number of packets jittered in all flows.

ratio: long integer, ratio of number of packets jittered in all flows. Expressed in % * 10000 (i.e. 1 means 0.0001%).

Summary: get only, value types: L, L

Example: 0/0 PE_JITTERTOTAL?

0/0 PE_JITTERTOTAL 1000 1000

Flow Impairments Statistics

PE_FLOWCLEAR[fid]

Clear all the impairment (duplicate, drop, mis-ordered, corrupted, latency and jitter) statistics on a particular flow on the port. The byte and packet counts will restart at zero.

fid: integer, the sub-index value of the flow definition.

Summary: set only.

Example: 0/0 PE_FLOWCLEAR [0]

PE_FLOWDUPTOTAL [fid] packets ratio

Obtains statistics concerning all the packets duplicated in a flow between this receive port and its partner TX port.

fid: integer, the sub-index value of the flow definition.

packets: long integer, number of packets duplicated for the flow.

ratio: long integer, ratio of number of packets * 10000 duplicated for the flow.

Summary: get only, value types: L, L, L, L

Example: 0/0 PE_FLOWDUPTOTAL [5]?

0/0 PE_FLOWDUPTOTAL [5] 100000 1

PE_FLOWDROPTOTAL [fid] packets prog band other ratio progratio bandratio otherratio

Obtains statistics concerning all the packets dropped in a flow between this receive port and its partner TX port.

fid: integer, the sub-index value of the flow definition.

packets: long integer, total number of packets dropped for the flow.

prog: long integer, total number of packets dropped as programmed for the flow.

band: long integer, total number of packets dropped due to bandwidth control for flow.

other: long integer, total number of packets dropped for other reasons for the flow.

ratio: long integer, ratio of number of packets * 10000 dropped for the flow.

progratio: long integer, ratio of number of packets * 10000 dropped as programmed for flow.

bandratio: long integer, ratio of number of packets * 10000 dropped due to bandwidth control for the flow.

otherratio: long integer, ratio of number of packets * 10000 dropped for other reasons for the flow.

Summary: get only, flow index, value types: L, L, L, L, L, L, L

Example: 0/0 PE_FLOWDROPTOTAL [5]?

0/0 PE_FLOWDROPTOTAL [5] 100000 100000 100000 100000 100000 100000 100000 100000

PE_FLOWMISTOTAL [fid] packets ratio

Obtains statistics concerning all the packets mis-ordered in a flow between this receive port and its partner TX port.

fid: integer, the sub-index value of the flow definition.

packets: long integer, number of packets mis-ordered for the flow.

ratio: long integer, ratio of number of packets * 10000 mis-ordered for the flow.

Summary: get only, flow index, value types: L, L

Example: 0/0 PE_FLOWMISTOTAL [5] ?

0/0 PE_FLOWMISTOTAL [5] 100000 1

PE_FLOWCORTOTAL [fid] packets fcs ip udp tcp ratio fcsratio ipratio udpratio tcpratio

Obtains statistics concerning all the packets corrupted in a flow between this receive port and its partner TX port.

fid: integer, the sub-index value of the flow definition.

packets: long integer, number of packets corrupted for the flow.

fcs: long integer, number of packets with Ethernet FCS corrupted for the flow.

ip: long integer, number of packets with IP Header check sum corrupted for the flow.

udp: long integer, number of packets with UDP check sum corrupted for the flow.

tcp: long integer, number of packets with TCP check sum corrupted for the flow.

ratio: long integer, ratio of number of packets * 10000 corrupted for the flow.

fcsratio: long integer, ratio of number of packets * 10000 with Ethernet FCS corrupted for the flow.

ipratio: long integer, ratio of number of packets * 10000 with IP Header check sum corrupted for the flow.

udpratio: long integer, ratio of number of packets * 10000 with UDP check sum corrupted for the flow.

tcpratio: long integer, ratio of number of packets * 10000 with TCP check sum corrupted for the flow.

Summary: get only, flow index, value types: I, L, L, L, L, L, L, L, L, L

Example: 0/0 PE_FLOWCORTOTAL [5]?

0/0 PE_FLOWCORTOTAL [5] 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000

PE_LATENCYTOTAL packets ratio

Obtains statistics concerning all the packets delayed this receive port and its partner TX port.

packets: long integer, number of packets delayed in all flows.

ratio: long integer, ratio of number of packets * 10000 delayed in all flows.

Summary: get only, value types: L, L

Example: 0/0 PE_LATENCYTOTAL?

0/0 PE_LATENCYTOTAL 1000 1000

PE_FLOWLATENCYTOTAL [fid] packets ratio

Obtains statistics concerning all the packets delayed between this receive port and its partner TX port.

fid: integer, the sub-index value of the flow definition.

packets: long integer, number of packets delayed in the flow.

ratio: long integer, ratio of number of packets * 10000 delayed in the flow.

Summary: get only, flow index, value types: L, L

Example: 0/0 PR_FLOWLATENCYTOTAL [5]?

0/0 PR_FLOWLATENCYTOTAL [5] 1000 1000

PE_FLOWJITTERTOTAL [fid] packets ratio

Obtains statistics concerning all the packets jittered in a flow between this receive port and its partner TX port.

fid: integer, the sub-index value of the flow definition.

packets: long integer, number of packets jittered in the flow.

ratio: long integer, ratio of number of packets * 10000 jittered in flow.

Summary: get only, flow index, value types: I, L, L

Example: 0/0 PR_FLOWJITTERTOTAL [5]?

0/0 PR_FLOWJITTERTOTAL [5] 1000 1000 1000 1000