ZebOS-XP Interface SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:32 2015

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 interfaceBriefInfo Struct Reference

Data Fields

- char **ifName** [INTERFACE_NAMSIZ]
- char **ifAddr** [16]
- enum smi_if_status **ifState**
- u_char **flags**
- u_char **gmpls_type**
- u_char **vrx_flag**
- u_int32_t **vrf_id**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.2 smi_bridge_info Struct Reference

Data Fields

- char **bridge_name** [SMI_BRIDGE_NAMSIZ]
- struct smi_fdb_list **fdb_list**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.3 smi_bridge_list Struct Reference

Data Fields

- u_int32_t **br_count**
- char **br_name** [32][SMI_BRIDGE_NAMSIZ]

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.4 smi_fdb_list Struct Reference

Data Fields

- u_int32_t **list_cnt**
- u_int32_t **have_more**
- struct list * **fdb_table**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.5 smi_fdb_mac_table Struct Reference

Data Fields

- u_int32_t **vlan_id**
- u_int32_t **ageing**
- char **name** [INTERFACE_NAMSIZ]
- u_int32_t **type**
- u_int8_t **hw_addr** [SMI_ETHER_ADDR_LEN]

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.6 smi_if Struct Reference

Data Fields

- char **name** [INTERFACE_NAMSIZ+1]
- char **hw_type** [40]
- u_int8_t **hw_addr** [INTERFACE_HWADDR_MAX]
- s_int32_t **hw_addr_len**
- s_int32_t **ifindex**
- u_int32_t **flags**
- s_int32_t **metric**
- s_int32_t **mtu**
- u_int32_t **duplex**
- u_int32_t **arp_ageing_timeout**
- char **vrfname** [INTERFACE_NAMSIZ+1]
- char **admin_group_name** [40]
- float64_t **bandwidth**
- int **bc_mode**
- struct prefix **ipv4_prefix**
- struct prefix **dest_ipv4**
- struct prefix **ipv6_prefix**
- char **connected_label** [40]
- struct smi_if_stats **ifstats**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.7 smi_if_desc Struct Reference

Data Fields

- char **ifname** [INTERFACE_NAMSIZ+1]
- bool_t **admin_shutdown**
- bool_t **protocol**
- char * **description**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.8 smi_if_desc_list Struct Reference

Data Fields

- int **have_more**
- int **start_index**
- int **end_index**
- int **count**
- struct list * **if_desc_list**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.9 smi_if_list Struct Reference

Data Fields

- int **have_more**
- int **start_index**
- int **end_index**
- int **count**
- struct list * **if_list**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.10 smi_if_stats Struct Reference

Data Fields

- u_int64_t good_octets_rcv
- u_int32_t bad_octets_rcv
- u_int32_t in_unicasts
- u_int32_t brdc_pkts_rcv
- u_int32_t mc_pkts_rcv
- u_int32_t in_pause
- u_int32_t undersize_pkts
- u_int32_t fragments_pkts
- u_int32_t oversize_pkts
- u_int32_t jabber_pkts
- u_int32_t in_rx_err
- u_int32_t bad_crc
- u_int32_t in_pkts_64_octets
- u_int32_t in_pkts_65_127_octets
- u_int32_t in_pkts_128_255_octets
- u_int32_t in_pkts_256_511_octets
- u_int32_t in_pkts_512_1023_octets
- u_int32_t in_pkts_1024_max_octets
- u_int64_t good_octets_sent
- u_int32_t out_unicasts
- u_int32_t brdc_pkts_sent
- u_int32_t mc_pkts_sent
- u_int32_t out_pause
- u_int32_t deferred
- u_int32_t collisions
- u_int32_t single_collision_frame
- u_int32_t multiple_collision_frame
- u_int32_t excess_collision_drop
- u_int32_t late_collision_detect
- u_int32_t out_fcs_err
- u_int32_t out_pkts_64_octets
- u_int32_t out_pkts_65_127_octets
- u_int32_t out_pkts_128_255_octets
- u_int32_t out_pkts_256_511_octets
- u_int32_t out_pkts_512_1023_octets
- u_int32_t out_pkts_1024_max_octets
- u_int32_t policy_in_discards
- u_int16_t policy_in_filtered
- u_int16_t policy_out_filtered

3.10.1 Field Documentation

3.10.1.1 u_int32_t smi_if_stats::bad_crc

packets received with bad Frame Check Sequence

3.10.1.2 u_int32_t smi_if_stats::bad_octets_rcv

No. of bad octets received

3.10.1.3 u_int32_t smi_if_stats::brdc_pkts_rcv

Broadcast packets received

3.10.1.4 u_int32_t smi_if_stats::brdc_pkts_sent

Broadcast packets Sent

3.10.1.5 u_int32_t smi_if_stats::collisions

Number of times a collision occurred before the interface transmitted a frame to the media successfully

3.10.1.6 u_int32_t smi_if_stats::deferred

Number of frames that have been transmitted successfully after they wait because media was busy

3.10.1.7 u_int32_t smi_if_stats::excess_collision_drop

Number of times an interface made maximum number of attempts to transmit a packet, each attempt resulting into collision

3.10.1.8 u_int32_t smi_if_stats::fragments_pkts

Pakcets received $<$ 64 Octets with Bad Frame Check Sequence

3.10.1.9 u_int64_t smi_if_stats::good_octets_rcv

No.of good Octets received

3.10.1.10 u_int64_t smi_if_stats::good_octets_sent

Number of good octets sent from this interface

3.10.1.11 u_int32_t smi_if_stats::in_pause

Number of Pause frames received

3.10.1.12 u_int32_t smi_if_stats::in_pkts_1024_max_octets

Packets received with octet length 1024 to max octet length

3.10.1.13 u_int32_t smi_if_stats::in_pkts_128_255_octets

Packets received with Octet length 128 to 255

3.10.1.14 u_int32_t smi_if_stats::in_pkts_256_511_octets

Packets received with octet length 256 to 511

3.10.1.15 u_int32_t smi_if_stats::in_pkts_512_1023_octets

Packets received with octet length 512 to 1023

3.10.1.16 u_int32_t smi_if_stats::in_pkts_64_octets

packets having 64 Octets Of length

3.10.1.17 u_int32_t smi_if_stats::in_pkts_65_127_octets

Packets received with Octet length 65 to 127

3.10.1.18 u_int32_t smi_if_stats::in_rx_err

Frames received with RxErr from the PHY

3.10.1.19 u_int32_t smi_if_stats::in_unicasts

Unicast Packets Received

3.10.1.20 u_int32_t smi_if_stats::jabber_pkts

Packets with length $>$ 1518 Bad FCS

3.10.1.21 u_int32_t smi_if_stats::late_collision_detect

Number of times late collision has been detected by an interface

3.10.1.22 u_int32_t smi_if_stats::mc_pkts_rcv

MultiCast Packets Receive

3.10.1.23 u_int32_t smi_if_stats::mc_pkts_sent

MultiCast Packets sent

3.10.1.24 u_int32_t smi_if_stats::multiple_collision_frame

Number of times an interface has experienced multiple collisions when attempting to transmit a given frame

3.10.1.25 u_int32_t smi_if_stats::out_fcs_err

Frame check sequence error counter of an interface

3.10.1.26 u_int32_t smi_if_stats::out_pause

Number of Pause frames sent

3.10.1.27 u_int32_t smi_if_stats::out_pkts_1024_max_octets

Packets sent with octet length 1024 to max octet length

3.10.1.28 u_int32_t smi_if_stats::out_pkts_128_255_octets

Packets sent with octet length 128 to 255

3.10.1.29 u_int32_t smi_if_stats::out_pkts_256_511_octets

Packets sent with octet length 256 to 511

3.10.1.30 u_int32_t smi_if_stats::out_pkts_512_1023_octets

Packets sent with octet length 512 to 1023

3.10.1.31 u_int32_t smi_if_stats::out_pkts_64_octets

Packets sent with octet length 64

3.10.1.32 u_int32_t smi_if_stats::out_pkts_65_127_octets

Packets sent with octet length 65 to 127

3.10.1.33 u_int32_t smi_if_stats::out_unicasts

Unicast Packets Sent

3.10.1.34 u_int32_t smi_if_stats::oversize_pkts

Packets received having length > than 1518 Octets

3.10.1.35 u_int32_t smi_if_stats::policy_in_discards

Packets received on a discard interface

3.10.1.36 u_int16_t smi_if_stats::policy_in_filtered

Packets received passing filtering policy

3.10.1.37 u_int16_t smi_if_stats::policy_out_filtered

Packets sent passing filtering policy

3.10.1.38 u_int32_t smi_if_stats::single_collision_frame

Number of times an interface has experienced a single collision when attempting to transmit a given frame

3.10.1.39 u_int32_t smi_if_stats::undersize_pkts

Packets received having length < 64 Octest long

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.11 smi_msg_if Struct Reference

Data Fields

- smi_cindex_t **cindex**
- smi_cindex_t **cindex_1**
- smi_cindex_t **cindex_2**
- smi_cindex_t **cindex_3**
- char **ifname** [INTERFACE_NAMSIZ]
- char **bridge_name** [SMI_BRIDGE_NAMSIZ]
- u_int32_t **mtu**
- u_int32_t **start_index**
- u_int32_t **end_index**
- float **bandwidth**
- int **autoneg**
- u_int8_t **hw_addr** [SMI_ETHER_ADDR_LEN]
- int **if_status**
- int **flag**
- int **mcast**
- u_int32_t **fdb_type**
- u_int32_t **duplex**
- enum smi_if_link_changed **smi_if_link_change**
- struct smi_if_stats **ifstats**
- enum smi_if_cross_over **cross_mode**
- u_int16_t **vid**
- u_int8_t **is_forward**
- enum smi_bridge_pri_ovr_mac_type **ovr_mac_type**
- u_int8_t **priority**
- u_int8_t **user_priority**
- u_char **traffic_class_table** [SMI_BRIDGE_MAX_USER_PRIO][SMI_BRIDGE_MAX_TRAFFIC_CLASS+1]
- enum smi_bridge_topo_type **topo_type**
- enum smi_bridge_type **type**
- enum smi_port_conf_state **port_conf_state**
- enum smi_port_conf_state **port_switch_state**
- enum smi_egress_port_mode **egress_mode**
- u_int8_t **spanning_tree_disable**
- enum smi_lacp_mode **if_lacp_mode**
- u_int32_t **if_lacp_admin_key**
- u_int8_t **if_lacp_load_balance_method**
- u_int8_t **dot1q_state**
- enum smi_dtag_mode **dtag_mode**
- enum smi_if_exist **exist**
- enum smi_port_learn_state **learn_state**
- struct smi_if_list **smi_iflist**
- struct smi_if_desc_list **smi_ifdesc**

- struct smiInterfaceBriefList **smiIfBriefList**
- struct smi_fdb_list **fdb_list**
- u_int32_t **vr_id**
- u_int32_t **mode**
- char **mac_add** [255]
- u_int16_t **v_id**
- int **instance**
- int **vlan_type**
- int **edge_type**
- u_int16_t **vlan**
- char **arp_name** [255]
- struct pal_in4_addr **ipv4_addr**
- u_char **prefixlen**
- char **label** [255]
- char **vrf_name** [255]
- int **secondary**
- char **ip_addr** [255]
- int **dad_attempts**
- char **group_name** [255]
- char **direct** [255]
- int **enabled**
- u_int32_t **valid_lifetime**
- u_int32_t **preferred_lifetime**
- char **ct_str** [255]
- char **bc_mode** [255]
- char **admin_name** [255]
- char **bw_str** [255]
- float32_t **bw_constraint** [SMI_MAX_BW_CONST]
- struct smi_static_channel_list **smiStaticChannelList**
- struct smi_bridge_list **br_list**
- int **vrrp**
- int **anycast**
- char **peer_str** [255]
- char **vcName** [255]
- char **vcType** [255]
- char **vcMode** [255]
- bool_t **vcStandbyMode**
- bool_t **vcRevertiveMode**
- char **groupName** [255]
- u_int32_t **groupId**
- char **pwName** [255]
- char **pwDescr** [255]
- bool_t **isAdd**
- u_int32_t **vcId**
- char **peer_address** [255]
- bool_t **control_word**

- bool_t **is_manual**
- bool_t **is_pw_status**
- bool_t **is_passive**
- u_int32_t **local_refresh_timer**
- u_int32_t **tunnel_id**
- bool_t **tunnel_direction**
- char **agi** [255]
- char **saii** [255]
- char **taii** [255]
- u_int32_t **cc_types**
- u_int32_t **cv_types**
- char **vcContainer1Name** [255]
- char **vcContainer2Name** [255]
- char **route_type** [255]
- char **mrouteifname** [255]

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.12 smi_static_channel_group Struct Reference

Data Fields

- char **port_channel_name** [INTERFACE_NAMSIZ]
- u_int32_t **linkcnt**
- char **member** [SMI_NSM_MAX_AGGREGATOR_LINKS][INTERFACE_-NAMSIZ]

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.13 smi_static_channel_list Struct Reference

Data Fields

- u_int32_t **list_cnt**
- u_int32_t **have_more**
- struct list * **static_channel**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

3.14 smiInterfaceBriefList Struct Reference

Data Fields

- int **have_more**
- int **start_index**
- int **end_index**
- int **count**
- struct list * **ifBriefInfoList**

The documentation for this struct was generated from the following file:

- smi_if_msg.h

Chapter 4

File Documentation

4.1 smi_if_msg.h File Reference

Provides APIs for managing interfaces. `#include "smi_message.h"`

Data Structures

- struct smi_if_stats
- struct smi_if
- struct smi_if_list
- struct smi_if_desc
- struct smi_if_desc_list
- struct interfaceBriefInfo
- struct smiInterfaceBriefList
- struct smi_fdb_mac_table
- struct smi_fdb_list
- struct smi_static_channel_group
- struct smi_static_channel_list
- struct smi_bridge_list
- struct smi_bridge_info
- struct smi_msg_if

Defines

- #define **SMI_MSG_IF_SIZE** 4
- #define **SMI_IF_LACP_LINK_ADMIN_KEY_MIN** 1
- #define **SMI_IF_LACP_LINK_ADMIN_KEY_MAX** 65535
- #define **DEFAULT_INTERFACE** "fe1"
- #define **SMI_BRIDGE_MIN_VAL** 1
- #define **SMI_BRIDGE_MAX_VAL** 32
- #define **VLAN_ID_MIN_VAL** 2

- #define **VLAN_ID_MAX_VAL** 2094
- #define **SMI_INSTANCE_MIN_VAL** 1
- #define **SMI_INSTANCE__MAX_VAL** 63
- #define **SMI_IF_PREFIX_VALID_LIFETIME_MIN** 0
- #define **SMI_IF_PREFIX_VALID_LIFETIME_MAX** 4294967295U
- #define **SMI_IF_PREFIX_PREFERRED_LIFETIME_MIN** 0
- #define **SMI_IF_PREFIX_PREFERRED_LIFETIME_MAX** 4294967295U
- #define **SMI_IF_IPV6_ND_DAD_ATTEMPTS_MIN** 0
- #define **SMI_IF_IPV6_ND_DAD_ATTEMPTS_MAX** 600
- #define **SMI_NSM_STATIC_AGG_KEY_MIN** 1
- #define **SMI_NSM_STATIC_AGG_KEY_MAX** 12
- #define **SMI_NSM_MAU_TYPE_ZERO_DOT_ZERO** 255
- #define **SMI_MIN_MTU** 64
- #define **SMI_MAX_MTU** 16360
- #define **SMI_BANDWIDTH_MIN_VAL** 1
- #define **SMI_BANDWIDTH_MAX_VAL** 10000000000U
- #define **SMI_MAX_BW_CONST** 8
- #define **SMI_NSM_MAX_AGGREGATOR_LINKS** 6
- #define **SMI_IF_CTYPE_NAME** 0
- #define **SMI_IF_CTYPE_MTU** 1
- #define **SMI_IF_CTYPE_BW** 2
- #define **SMI_IF_CTYPE_FLAG** 3
- #define **SMI_IF_CTYPE_AUTONEG** 4
- #define **SMI_IF_CTYPE_HWADDR** 5
- #define **SMI_IF_CTYPE_DUPLEX** 6
- #define **SMI_IF_CTYPE_MCAST** 7
- #define **SMI_IF_CTYPE_STATUS** 8
- #define **SMI_IF_CTYPE_IFSTATUS** 9
- #define **SMI_IF_CTYPE_STATISTICS** 10
- #define **SMI_IF_CTYPE_MDIX_CROSSOVER** 11
- #define **SMI_IF_CTYPE_BRIDGE_NAME** 12
- #define **SMI_IF_CTYPE_VID** 13
- #define **SMI_IF_CTYPE_FORWARD** 14
- #define **SMI_IF_CTYPE_TRAFFIC_CLASSTBL** 15
- #define **SMI_IF_CTYPE_USRPRIORITY** 16
- #define **SMI_IF_CTYPE_TOPOTYPE** 17
- #define **SMI_IF_CTYPE_BRIDGETYPE** 18
- #define **SMI_IF_CTYPE_PORT_STATE** 19
- #define **SMI_IF_CTYPE_EGRESS_PORT_MODE** 20
- #define **SMI_IF_CTYPE_PORT_SWITCH_STATE** 21
- #define **SMI_IF_CTYPE_SPAN_TREE_DISABLE** 22
- #define **SMI_IF_LACP_CTYPE_MODE** 23
- #define **SMI_IF_LACP_CTYPE_KEY** 24
- #define **SMI_IF_CTYPE_DOT1Q_STATE** 25
- #define **SMI_IF_CTYPE_DTAG_MODE** 26
- #define **SMI_IF_CTYPE_IF_EXIST** 27

- #define **SMI_IF_CTYPE_LEARN_STATE** 28
- #define **SMI_IF_CTYPE_OVR_MAC_TYPE** 29
- #define **SMI_IF_CTYPE_PRIORITY** 30
- #define **SMI_IF_CTYPE_EXTENDED_1** 31
- #define **SMI_IF_CTYPE_SMI_IFLIST** 0
- #define **SMI_IF_CTYPE_SMI_IFBRIEFLIST** 1
- #define **SMI_IF_LACP_CTYPE_LOAD_BALANCE_METHOD** 2
- #define **SMI_CTYPE_FDB_TYPE** 3
- #define **SMI_CTYPE_SHOW_FDB** 4
- #define **SMI_CTYPE_SHOW_AGEING** 5
- #define **SMI_CTYPE_INDEXING** 6
- #define **SMI_IF_CTYPE_VR_ID** 7
- #define **SMI_IF_CTYPE_MODE** 8
- #define **SMI_IF_CTYPE_MAC_ADD** 9
- #define **SMI_IF_CTYPE_VLAN_TYPE** 10
- #define **SMI_IF_CTYPE_V_ID** 11
- #define **SMI_IF_CTYPE_INSTANCE** 12
- #define **SMI_IF_CTYPE_EDGE_TYPE** 13
- #define **SMI_IF_CTYPE_VLAN** 14
- #define **SMI_IF_CTYPE_ARP_NAME** 15
- #define **SMI_IF_CTYPE_IPV4_ADDR** 16
- #define **SMI_IF_CTYPE_PREFIXLEN** 17
- #define **SMI_IF_CTYPE_LABEL** 18
- #define **SMI_IF_CTYPE_VRF_NAME** 19
- #define **SMI_IF_CTYPE_SECONDARY** 20
- #define **SMI_IF_CTYPE_IP_ADDR** 21
- #define **SMI_IF_CTYPE_DAD_ATTEMPTS** 22
- #define **SMI_IF_CTYPE_GROUP_NAME** 23
- #define **SMI_IF_CTYPE_DIRECT** 24
- #define **SMI_IF_CTYPE_ENABLED** 25
- #define **SMI_IF_CTYPE_VALID_LIFETIME** 26
- #define **SMI_IF_CTYPE_PREFERRED_LIFETIME** 27
- #define **SMI_IF_CTYPE_BC_MODE** 28
- #define **SMI_IF_CTYPE_CT_STR** 29
- #define **SMI_IF_CTYPE_BW_STR** 30
- #define **SMI_IF_CTYPE_EXTENDED_2** 31
- #define **SMI_IF_CTYPE_ADMIN_NAME** 0
- #define **SMI_IF_CTYPE_SMI_IFDESCLIST** 1
- #define **SMI_IF_CTYPE_BW_CONSTRAINT** 2
- #define **SMI_IF_CTYPE_STATIC_CHANNEL_LIST** 3
- #define **SMI_IF_CTYPE_BRIDGE_LIST** 4
- #define **SMI_IF_CTYPE_MPLS_VC_NAME** 5
- #define **SMI_IF_CTYPE_MPLS_VC_TYPE** 6
- #define **SMI_IF_CTYPE_MPLS_VC_MODE** 7
- #define **SMI_IF_CTYPE_MPLS_VC_STANDBY_MODE** 8
- #define **SMI_IF_CTYPE_MPLS_VC_REVERTIVE_MODE** 9

- #define **SMI_LDP_CTYPE_PW_NAME** 10
- #define **SMI_LDP_CTYPE_PW_DESCR** 11
- #define **SMI_LDP_CTYPE_IS_ADD** 12
- #define **SMI_LDP_CTYPE_GROUPNAME** 13
- #define **SMI_LDP_CTYPE_GROUPID** 14
- #define **SMI_LDP_CTYPE_VCID** 15
- #define **SMI_LDP_CTYPE_PEER_ADDR** 16
- #define **SMI_LDP_CTYPE_CONTROL_WORD** 17
- #define **SMI_LDP_CTYPE_IS_MANUAL** 18
- #define **SMI_LDP_CTYPE_IS_PW_STATUS** 19
- #define **SMI_LDP_CTYPE_IS_PASSIVE** 20
- #define **SMI_LDP_CTYPE_LOCAL_REFRESH_TIMER** 21
- #define **SMI_LDP_CTYPE_TUNNEL_ID** 22
- #define **SMI_LDP_CTYPE_TUNNEL_DIRECTION** 23
- #define **SMI_LDP_CTYPE_AGI** 24
- #define **SMI_LDP_CTYPE_SAII** 25
- #define **SMI_LDP_CTYPE_TAII** 26
- #define **SMI_LDP_CTYPE_CCTYPES** 27
- #define **SMI_LDP_CTYPE_CVTYPES** 28
- #define **SMI_LDP_CTYPE_VC1_NAME** 29
- #define **SMI_LDP_CTYPE_VC2_NAME** 30
- #define **SMI_IF_CTYPE_EXTENDED_3** 31

Enumerations

- enum **port_mode** { **SMI_PORT_MODE_UNKNOWN**, **SMI_PORT_MODE_SGMII**, **SMI_PORT_MODE_1000BASEX** }
- enum smi_if_duplex { SMI_IF_HALF_DUPLEX, SMI_IF_FULL_DUPLEX, SMI_IF_AUTO_NEGO }

  Specifies duplex operation of an interface.

- enum smi_if_status { SMI_IF_DOWN, SMI_IF_UP }

  Specify administrative state of a port.

- enum smi_if_link_changed { SMI_IF_LINK_UNCHANGED, SMI_IF_LINK_CHANGED }
- enum smi_if_cross_over { SMI_IF_CROSS_OVER, SMI_IF_CROSS_OVER_NONE, SMI_IF_CROSS_OVER_AUTO }
- enum **smi_bridge_pri_ovr_mac_type** {

  SMI_BRIDGE_MAC_PRI_OVR_NONE, **SMI_BRIDGE_MAC_STATIC**, **SMI_BRIDGE_MAC_STATIC_PRI_OVR**, **SMI_BRIDGE_MAC_STATIC_MGMT**,

  SMI_BRIDGE_MAC_STATIC_MGMT_PRI_OVR, **SMI_BRIDGE_MAC_PRI_OVR_MAX** }
- enum smi_port_conf_state { SMI_PORT_DISABLE, SMI_PORT_ENABLE }

- enum smi_egress_port_mode { SMI_EGRESS_PORT_TAGGED, SMI_EGRESS_PORT_UNTAGGED, SMI_EGRESS_PORT_UNMODIFIED }
- enum smi_port_learn_state { SMI_PORT_LEARN_ENABLE, SMI_PORT_LEARN_DISABLE }
- enum **smi_dtag_mode** { **SMI_DTAG_MODE_INTERNAL**, **SMI_DTAG_MODE_EXTERNAL**, **SMI_DTAG_MODE_INVALID** }
- enum **smi_if_exist** { **SMI_EXIST_NO**, **SMI_EXIST_YES** }
- enum smi_if_lacp_load_balance_method {

  SMI_LACP_LOAD_BALANCE_DST_MAC = 1, SMI_LACP_LOAD_BALANCE_SRC_MAC, SMI_LACP_LOAD_BALANCE_SRC_DST_MAC, SMI_LACP_LOAD_BALANCE_SRC_IP,

  SMI_LACP_LOAD_BALANCE_DST_IP, SMI_LACP_LOAD_BALANCE_SRC_DST_IP, SMI_LACP_LOAD_BALANCE_SRC_PORT, SMI_LACP_LOAD_BALANCE_DST_PORT,

  SMI_LACP_LOAD_BALANCE_SRC_DST_PORT }

  Supported load balance methods for aggregators.

Functions

- void **smi_interface_dump** (struct lib_globals *zg, struct smi_msg_if *msg)
- int **smi_encode_ifmsg** (u_char **, u_int16_t *size, struct smi_msg_if *msg)
- int **smi_decode_ifmsg** (u_char **pnt, u_int16_t *size, struct smi_msg_if *msg)
- int **smi_parse_if** (u_char **, u_int16_t *, struct smi_msg_header *, void *, SMI_CALLBACK)

4.1.1 Detailed Description

Provides APIs for managing interfaces.

4.1.2 Enumeration Type Documentation

4.1.2.1 enum smi_egress_port_mode

Enumerator:

SMI_EGRESS_PORT_TAGGED Specifies tagged egress port mode

SMI_EGRESS_PORT_UNTAGGED Specifies untagged egress port mode

SMI_EGRESS_PORT_UNMODIFIED Specifies default egress port mode

4.1.2.2 enum smi_if_cross_over

Enumerator:

SMI_IF_CROSS_OVER Specifies MDIX interface

SMI_IF_CROSS_OVER_NONE Specify an interface without any crossover

SMI_IF_CROSS_OVER_AUTO Specifies an Auto MDIX interface

4.1.2.3 enum smi_if_duplex

Specifies duplex operation of an interface.

Enumerator:

SMI_IF_HALF_DUPLEX Specifies half duplex operation

SMI_IF_FULL_DUPLEX Specifies full duplex operation

SMI_IF_AUTO_NEGO Specifies auto negotiation capability. The interface automatically operates at half or full duplex depending on environmental factors

4.1.2.4 enum smi_if_lacp_load_balance_method

Supported load balance methods for aggregators.

Enumerator:

SMI_LACP_LOAD_BALANCE_DST_MAC Destination MAC Address based load balancing

SMI_LACP_LOAD_BALANCE_SRC_MAC Source MAC address based load balancing

SMI_LACP_LOAD_BALANCE_SRC_DST_MAC Source and destination MAC address based load balancing

SMI_LACP_LOAD_BALANCE_SRC_IP Source IP address based load balancing

SMI_LACP_LOAD_BALANCE_DST_IP Destination IP address based load balancing

SMI_LACP_LOAD_BALANCE_SRC_DST_IP Source and destination IP address based load balancing

SMI_LACP_LOAD_BALANCE_SRC_PORT Source TCP/UDP port based load balancing

SMI_LACP_LOAD_BALANCE_DST_PORT Destination TCP/UDP port based load balancing

SMI_LACP_LOAD_BALANCE_SRC_DST_PORT Source and destination TCP/UDP port based load balancing

4.1.2.5 enum smi_if_link_changed

Enumerator:

SMI_IF_LINK_UNCHANGED Specifies no change in interface state

SMI_IF_LINK_CHANGED Specifies change in interface state

4.1.2.6 enum smi_if_status

Specify administrative state of a port.

Enumerator:

SMI_IF_DOWN Specifies port is down

SMI_IF_UP Specifies port is Up

4.1.2.7 enum smi_port_conf_state

Enumerator:

SMI_PORT_DISABLE Specify if port configuration state is enabled

SMI_PORT_ENABLE Specifies if port configuration state is disabled

4.1.2.8 enum smi_port_learn_state

Enumerator:

SMI_PORT_LEARN_ENABLE Specifies Port learning state enabled

SMI_PORT_LEARN_DISABLE Specifies Port learning state disabled

Index