ZebOS-XP LLDP SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:22 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 smi_lldp_port Struct Reference

Data Fields

- bool_t **dcbx_state**
- u_int8_t **admin_status**
- u_int8_t **hw_addr** [INTERFACE_HWADDR_MAX]

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

- smi_lldp_msg.h

3.2 smi_msg_lldp Struct Reference

Data Fields

- smi_cindex_t **cindex**
- u_int32_t **lldp_debug**
- char **if_name** [INTERFACE_NAMSIZ]
- char **hwaddr** [SMI_ETHER_ADDR_LEN]
- u_int8_t **admin_status**
- u_char **sys_name** [LLDP_NAME_MAX_LEN]
- u_char **agt_ckt_id** [LLDP_NAME_MAX_LEN]
- u_char **descriptor** [LLDP_DESCR_MAX_LEN]
- u_char **name** [LLDP_LOCAL_MAX_LEN]
- u_char **mac** [SMI_ETHER_ADDR_LEN]
- u_char **remote_mac** [SMI_ETHER_ADDR_LEN]
- u_char **rem_mac_array** [SMI_NUM_REC][SMI_ETHER_ADDR_LEN]
- u_int16_t **tlv_flag**
- u_int32_t **tx_hold**
- u_int32_t **tx_interval**
- u_int32_t **delay**
- u_int32_t **limit**
- u_int32_t **interval**
- u_int32_t **credit**
- u_int32_t **fast_init**
- u_int8_t **type**
- u_int32_t **tx_delay**
- struct smi_remote_lldp **rlldp**
- struct smi_port_lldp_statistics **port_lldp_stat**
- enum smi_lldp_chassis_id_sub_type **chassis_id_type**
- char **ipaddress** [SMI_IPADDRESS_SIZE]
- int **first_call**
- int8_t **set_flag**
- struct smi_lldp_port **port_info**

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

- smi_lldp_msg.h

3.3 smi_port_lldp_statistics Struct Reference

Data Fields

- u_int32_t **frames_out_total**
- u_int32_t **ageouts_total**
- u_int32_t **frames_discarded_total**
- u_int32_t **frames_in_errors_total**
- u_int32_t **frames_in_total**
- u_int32_t **tlvs_discarded_total**
- u_int32_t **tlvs_unrecognized_total**

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

- smi_lldp_msg.h

3.4 smi_remote_lldp Struct Reference

Data Fields

- u_int8_t **remote_chassic_comp**
- u_int8_t **remote_port_comp**
- u_char **remote_mac_addr** [SMI_ETHER_ADDR_LEN]
- u_int16_t **rx_ttl**
- u_int32_t **rxinfo_ttl_rem_secs**
- u_char **remote_nw_addr** [SMI_NETWORK_ADDR_LENGTH]
- u_char **remote_ifname** [INTERFACE_NAMSIZ+1]
- u_char **remote_local** [LLDP_LOCAL_MAX_LEN+1]
- u_char **remote_descr** [LLDP_DESCR_MAX_LEN+1]
- u_int8_t **remote_if_numbering**
- u_int32_t **remote_if_number**
- u_int16_t **remote_port_vlan_id**
- u_int8_t **remote_ppvid_flag**
- u_int16_t **remote_pp_vlan_id**
- struct smi_vlan_bmp **remote_vlanbmap**
- u_int16_t **remote_protocol**
- u_int8_t **remote_autonego_support**
- u_int16_t **remote_autonego_cap**
- u_int16_t **remote_oper_mau_type**
- u_int8_t **remote_link_aggr_status**
- u_int32_t **remote_link_aggr_id**
- u_int16_t **remote_max_frame_size**
- u_char **remote_sys_name** [LLDP_NAME_MAX_LEN+1]
- u_char **remote_sys_descr** [LLDP_DESCR_MAX_LEN+1]
- u_int16_t **remote_sys_cap**
- u_int16_t **remote_sys_cap_enabled**
- u_char **remote_mgmt_addr** [SMI_MGMT_ADDR_LENGTH]
- u_char **remote_oid** [SMI_OID_LEN_MAX]
- smi_time_t **time_mark** [SMI_TIME_MARK_INDEX_MAX]
- u_char **time_mark_index**
- u_int32_t **remote_index**
- u_int32_t **remote_unknown_tlv_type**
- u_char * **remote_unknown_tlv_info**
- u_int32_t **remote_org_def_type**
- u_char * **remote_org_def_info**

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

- smi_lldp_msg.h

Chapter 4

File Documentation

4.1 smi_lldp.h File Reference

Provides APIs for managing Link Layer Discovery Protocol (LLDP) in ZebOS.
`#include "smi_client.h"`

Defines

- #define **SMI_LLDP_LOGICAL_PORT**(ifname)

Functions

- int smi_lldp_api_port_disable (struct smiclient_globals *azg, char *ifname)

  Disables LLDP on specified port.

- int smi_lldp_api_port_enable (struct smiclient_globals *azg, char *ifname, u_int8_t admin_status)

  Enables LLDP on specified port.

- int smi_lldp_port_set_locally_assigned (struct smiclient_globals *azg, char *ifname, u_char *name)

  Sets the port name to a locally assigned alphanumeric string.

- int smi_lldp_port_get_locally_assigned (struct smiclient_globals *azg, char *ifname, u_char *name)

  Gets the locally assigned alphanumeric name of a given port.

- int smi_lldp_set_port_basic_tlvs_enable (struct smiclient_globals *azg, char *ifname, u_int16_t tlv_flag)

  Sets the TLVs to be enabled for transmission on a port.

- int smi_lldp_get_port_basic_tlvs_enable (struct smiclient_globals *azg, char *ifname, u_int16_t *tlv_flag)

  Gets the TLVs enabled for transmission on a given port.

- int smi_lldp_set_port_msg_tx_hold (struct smiclient_globals *azg, char *ifname, u_int32_t tx_hold)

  Sets the message transmit hold parameter that determines the TTL value for LLDP PDU to be transmitted by the port.

- int smi_lldp_get_port_msg_tx_hold (struct smiclient_globals *azg, char *ifname, u_int32_t *tx_hold)

  Gets the message transmit hold parameter that determines the TTL value for LLDP PDU to be transmitted by the port.

- int smi_lldp_set_port_msg_tx_interval (struct smiclient_globals *azg, char *ifname, u_int32_t tx_interval)

  Sets the interval at which LLDP frames are transmitted.

- int smi_lldp_get_port_msg_tx_interval (struct smiclient_globals *azg, char *ifname, u_int32_t *tx_interval)

  Gets the interval at which LLDP frames are transmitted.

- int smi_lldp_set_port_reinit_delay (struct smiclient_globals *azg, char *ifname, u_int32_t reinit_delay)

  Sets the delay time between when LLDP is disabled on a port and an attempt is made to reinitialize it.

- int smi_lldp_get_port_reinit_delay (struct smiclient_globals *azg, char *ifname, u_int32_t *reinit_delay)

  Gets the delay time between when LLDP is disabled on a port and an attempt is made to reinitialize it.

- int smi_lldp_set_port_too_many_neighbors (struct smiclient_globals *azg, char *ifname, u_int32_t limit, u_int8_t type, u_char *mac, u_int32_t interval)

  Sets the action to be taken when LLDP remote table is full.

- int smi_lldp_get_port_too_many_neighbors (struct smiclient_globals *azg, char *ifname, u_int32_t *limit, u_int8_t type, u_char *mac, u_int32_t *interval)

  Get the values related to too-many-neighbors configuration for a specified port.

- int smi_lldp_set_port_tx_delay (struct smiclient_globals *azg, char *ifname, u_int32_t tx_delay)

  Sets the delay time between successive transmission of LLDP frames.

- int smi_lldp_get_port_tx_delay (struct smiclient_globals *azg, char *ifname, u_int32_t *tx_delay)

  Gets the delay time between successive transmission of LLDP frames.

- int smi_lldp_set_system_description (struct smiclient_globals *azg, u_char *descriptor)

  Sets LLDP system description as provided.

- int smi_lldp_get_system_description (struct smiclient_globals *azg, u_char *descriptor)

  Gets LLDP system description.

- int smi_lldp_set_system_name (struct smiclient_globals *azg, u_char *sys_name)

  Sets LLDP system name as provided.

- int smi_lldp_get_system_name (struct smiclient_globals *azg, u_char *sys_name)

  Gets LLDP system name.

- int smi_lldp_get_port (struct smiclient_globals *azg, char *ifname, char *rem_mac, struct smi_remote_lldp *rlldp)

  Gets the Remote LLDP parameters for a known neighbor MAC on a port.

- int smi_lldp_get_rem_macs_on_port (struct smiclient_globals *azg, char *ifname, char rem_mac_arr[SMI_NUM_REC][SMI_ETHER_ADDR_LEN], int first_call, char *start_mac)

  Gets the bitmap of the remote ports on an interface.

- int smi_lldp_get_port_statistics (struct smiclient_globals *azg, char *ifname, struct smi_port_lldp_statistics *port_lldp_stat)

  Gets LLDP Port statictics.

- int smi_lldp_set_hwaddr (struct smiclient_globals *azg, char *hwaddr)

  Sets LLDP HW address.

- int smi_lldp_get_hwaddr (struct smiclient_globals *azg, char *hwaddr)

  Gets LLDP HW address.

- int smi_lldp_set_chassis_id_type (struct smiclient_globals *azg, char *ifname, enum smi_lldp_chassis_id_sub_type lldp_chassis_type)

  Sets the chassis ID subtype on a port.

- int smi_lldp_get_chassis_id_type (struct smiclient_globals *azg, char *ifname, enum smi_lldp_chassis_id_sub_type *lldp_chassis_type)

  Gets the chassis ID subtype set on a port.

- int smi_lldp_set_chassis_ip_address (struct smiclient_globals *azg, char *chassis_ipaddr)

  Sets the chassis IP address.

- int smi_lldp_get_chassis_ip_address (struct smiclient_globals *azg, char *chassis_ipaddr)

  Gets the chassis IP address.

- int smi_lldp_get_port_admin_status (struct smiclient_globals *azg, char *ifname, u_int8_t *admin_status)

  Retrieves the admin status of an LLDP port.

- s_int32_t smi_lldp_debug_on (struct smiclient_globals *azg, u_int32_t debug)

  Function enables the debug for lldp.

- s_int32_t smi_lldp_debug_off (struct smiclient_globals *azg, u_int32_t debug)

  Function disables the debug for lldp.

- s_int32_t smi_lldp_api_port_info (struct smiclient_globals *azg, char *ifname, struct smi_lldp_port *port_info)

  Retrieves the information of an LLDP port.

- int smi_lldp_get_all_port_statistics (struct smiclient_globals *azg, struct smi_port_lldp_statistics *port_lldp_stat)

  Retrieves the information of all LLDP port statistics.

4.1.1 Detailed Description

Provides APIs for managing Link Layer Discovery Protocol (LLDP) in ZebOS. LLDP is a link layer protocol used by network elements for advertising their indentity, capabilities and neighbours in an ethernet network. The APIs provided here enables a network management application to monitor and control ZebOS LLDP implementation.

4.1.2 Define Documentation

4.1.2.1 #define SMI_LLDP_LOGICAL_PORT(ifname)

Value:

```
((pal_strncmp (ifname, "po ", 2) == 0)                  \
        || (pal_strncmp (ifname, "sa ", 2) == 0)          \
        || (pal_strncmp (ifname, "vlan", 3) == 0)         \
        || (pal_strstr (ifname, "."))                     \
        || (pal_strncmp (ifname, "lo ", 2) == 0))
```

4.1.3 Function Documentation

4.1.3.1 int smi_lldp_api_port_disable (struct smiclient_globals * *azg*, char * *ifname*)

Disables LLDP on specified port. smi_lldp_api_port_disable

Parameters:

← ***azg*** SMI Client global structure
← ***ifname*** Interface name on which LLDP needs to be disabled

Returns:

0 on success, otherwise one of the following error codes
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.1.3.2 int smi_lldp_api_port_enable (struct smiclient_globals * *azg*, char * *ifname*, u_int8_t *admin_status*)

Enables LLDP on specified port. smi_lldp_api_port_enable

Parameters:

← ***azg*** SMI Client global structure
← ***ifname*** Interface name on which LLDP needs to be enabled
← ***admin_status*** Admin status

Returns:

0 on success, otherwise one of the following error codes
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.1.3.3 s_int32_t smi_lldp_api_port_info (struct smiclient_globals * *azg*, char * *ifname*, struct smi_lldp_port * *port_info*)

Retrieves the information of an LLDP port. smi_lldp_api_port_info

Parameters:

← ***azg*** Pointer to smiclient_globals structure
← ***ifname*** Interface name for which port information needs to be retrieved
→ ***port_info*** where the final port information resides

Returns:

On success 0, otherwise returns SMI_ERROR

4.1.3.4 s_int32_t smi_lldp_debug_off (struct smiclient_globals * *azg*, u_int32_t *debug*)

Function disables the debug for lldp. smi_lldp_debug_off

Parameters:

← ***azg*** Pointer to the SMI client global structure

← ***debug*** disables the debug <0>

Returns:

SET_SUCCESS when the function succeeds, otherwise one of the following error codes
SET_ERROR

4.1.3.5 s_int32_t smi_lldp_debug_on (struct smiclient_globals * *azg*, u_int32_t *debug*)

Function enables the debug for lldp. smi_lldp_debug_on

Parameters:

← ***azg*** Pointer to the SMI client global structure

← ***debug*** enables the debug <1>

Returns:

SET_SUCCESS when the function succeeds, otherwise one of the following error codes
SET_ERROR

4.1.3.6 int smi_lldp_get_all_port_statistics (struct smiclient_globals * *azg*, struct smi_port_lldp_statistics * *port_lldp_stat*)

Retrieves the information of all LLDP port statistics. smi_lldp_get_all_port_statistics

Parameters:

← ***azg*** Pointer to smiclient_globals structure

→ ***port_lldp_stats*** where the final port information resides

Returns:

On success 0, otherwise returns SMI_ERROR

4.1.3.7 int smi_lldp_get_chassis_id_type (struct smiclient_globals * *azg*, char * *ifname*, enum smi_lldp_chassis_id_sub_type * *lldp_chassis_type*)

Gets the chassis ID subtype set on a port. smi_lldp_get_chassis_id_type

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which chassis id type needs to be retrieved

→ ***lldp_chassis_id_type*** Pointer to chassis ID type

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST

4.1.3.8 int smi_lldp_get_chassis_ip_address (struct smiclient_globals * *azg*, char * *chassis_ipaddr*)

Gets the chassis IP address. smi_lldp_get_chassis_ip_address

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***chassis_ipaddr*** Pointer to Chassis IP address

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_IP_ADDR_ERR
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.1.3.9 int smi_lldp_get_hwaddr (struct smiclient_globals * *azg*, char * *hwaddr*)

Gets LLDP HW address. smi_lldp_set_hwaddr

Parameters:

← ***azg*** Pointer to smiclient_globals structure

→ ***hwaddr*** LLDP HW address

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.1.3.10 int smi_lldp_get_port (struct smiclient_globals * *azg*, char * *ifname*, char * *rem_mac*, struct smi_remote_lldp * *rlldp*)

Gets the Remote LLDP parameters for a known neighbor MAC on a port. smi_lldp_-get_port

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name associated with remote LLDP MAC

← ***rem_mac*** MAC address of the LLDP neighbor

→ ***rlldp*** Remote LLDP information structure

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST
LLDP_API_ERR_LLDP_REM_PORT_MAC_NOT_EXIST

4.1.3.11 int smi_lldp_get_port_admin_status (struct smiclient_globals * *azg*, char * *ifname*, u_int8_t * *admin_status*)

Retrieves the admin status of an LLDP port. smi_lldp_get_port_admin_status

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which admin status needs to be retrieved

→ ***admin_status*** Administrative status of the port. It can be one of the following
1 - LLDP_DISABLED 2 - LLDP_ENABLED_TX_ONLY
4 - LLDP_ENABLED_RX_ONLY
8 - LLDP_ENABLED_RX_TX

Returns:

On success 0, otherwise one of the following error codes
LLDP_API_ERR_LLDP_IF_NOT_EXIST

4.1.3.12 int smi_lldp_get_port_basic_tlvs_enable (struct smiclient_globals * *azg*, char * *ifname*, u_int16_t * *tlv_flag*)

Gets the TLVs enabled for transmission on a given port. smi_lldp_get_port_basic_-tlvs_enable

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name on which TLV should be enabled

→ ***tlv_flag*** Flad indentifying the TLVs enabled.

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS

4.1.3.13 int smi_lldp_get_port_msg_tx_hold (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t * *tx_hold*)

Gets the message transmit hold parameter that determines the TTL value for LLDP PDU to be transmitted by the port. smi_lldp_get_port_msg_tx_hold

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which tx_hold needs to be retrieved

→ ***tx_hold*** Pointer to Message Transmit hold time parameter value.

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS

4.1.3.14 int smi_lldp_get_port_msg_tx_interval (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t * *tx_interval*)

Gets the interval at which LLDP frames are transmitted. smi_lldp_get_port_msg_tx_-interval

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which tx_interval needs to be retrieved

→ ***tx_interval*** The value of tx_interval

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS

4.1.3.15 int smi_lldp_get_port_reinit_delay (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t * *reinit_delay*)

Gets the delay time between when LLDP is disabled on a port and an attempt is made to reinitialize it. smi_lldp_get_port_reinit_delay

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which reinitialization delay needs to be retrieved

→ ***reinit_delay*** The pointer to reinitialization delay

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS

4.1.3.16 int smi_lldp_get_port_statistics (struct smiclient_globals * *azg*, char * *ifname*, struct smi_port_lldp_statistics * *port_lldp_stat*)

Gets LLDP Port statictics. smi_lldp_get_port_statistics

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which LLDP port statistics needs to be retrieved.

→ ***port_lldp_stat*** Pointer to LLDP port statistics structure

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST

4.1.3.17 int smi_lldp_get_port_too_many_neighbors (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t * *limit*, u_int8_t *type*, u_char * *mac*, u_int32_t * *interval*)

Get the values related to too-many-neighbors configuration for a specified port. smi_lldp_get_port_too_many_neighbors

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which too-many-neighbors configuration needs to be retrieved

→ ***limit*** Pointer to the upper limit value for too many neighbours

→ ***discard_type*** Too many neighbours discard value.

→ ***mac*** Pointer MAC address of the remote LLDP agent for which too-many-neighbors configuration needs to be retrieved.

→ ***interval*** Pointer to too-many-neighbors time interval

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS

4.1.3.18 int smi_lldp_get_port_tx_delay (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t * *tx_delay*)

Gets the delay time between successive transmission of LLDP frames. smi_lldp_get_port_tx_delay

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which LLDP tranmission delay needs to be set

→ ***tx_delay*** Pointer to the LLDP transmission delay

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS

4.1.3.19 int smi_lldp_get_rem_macs_on_port (struct smiclient_globals * *azg*, char * *ifname*, char *rem_mac_arr*[SMI_NUM_REC][SMI_ETHER_ADDR_LEN], int *first_call*, char * *start_mac*)

Gets the bitmap of the remote ports on an interface. smi_lldp_get_rem_macs_on_port

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which the remote port bitmap needs to be retrieved.

→ ***rem_mac_arr*** bitmap of remote mac addresses

← ***first_call*** Whether this is the fist call to this API

← ***Starting*** MAC address

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST
LLDP_API_ERR_LLDP_REM_PORT_MAC_NOT_EXIST

4.1.3.20 int smi_lldp_get_system_description (struct smiclient_globals * *azg*, u_char * *descriptor*)

Gets LLDP system description. smi_lldp_get_system_description

Parameters:

← ***azg*** Pointer to smiclient_globals structure

→ ***sys_description*** LLDP system description string retrieved

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.1.3.21 int smi_lldp_get_system_name (struct smiclient_globals * *azg*, u_char * *sys_name*)

Gets LLDP system name. smi_lldp_get_system_name

Parameters:

← ***azg*** Pointer to smiclient_globals structure

→ ***sys_name*** Pointer to LLDP system name

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.1.3.22 int smi_lldp_port_get_locally_assigned (struct smiclient_globals * *azg*, char * *ifname*, u_char * *name*)

Gets the locally assigned alphanumeric name of a given port. smi_lldp_port_get_-locally_assigned

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Name of the interface

→ ***name*** Locally assigned name of the port

Returns:

0 on success. The output param name is filled with local. In case of error, one of the following error code is returned
LLDP_API_ERR_ONM_IF_NOT_EXIST LLDP_API_ERR_LLDP_IF_NOT_-EXIST

4.1.3.23 int smi_lldp_port_set_locally_assigned (struct smiclient_globals * *azg*, char * *ifname*, u_char * *name*)

Sets the port name to a locally assigned alphanumeric string. smi_lldp_port_set_-locally_assigned

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Name of the interface

← ***name*** Locally assigned alphanumeric string that identifies the port

Returns:

0 on success, otherwise one of the followign error code
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST

4.1.3.24 int smi_lldp_set_chassis_id_type (struct smiclient_globals * *azg*, char * *ifname*, enum smi_lldp_chassis_id_sub_type *lldp_chassis_type*)

Sets the chassis ID subtype on a port. smi_lldp_set_chassis_id_type

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name on which chassis ID subtype needs to be set

← ***lldp_chassis_id_type*** Chassis ID type to be set

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_INVALID_CHASSIS_SUBTYPE
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST

4.1.3.25 int smi_lldp_set_chassis_ip_address (struct smiclient_globals * *azg*, char * *chassis_ipaddr*)

Sets the chassis IP address. smi_lldp_set_chassis_ip_address

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***chassis_ipaddr*** Chassis IP address to be set

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_IP_ADDR_ERR
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.1.3.26 int smi_lldp_set_hwaddr (struct smiclient_globals * *azg*, char * *hwaddr*)

Sets LLDP HW address. smi_lldp_set_hwaddr

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***hwaddr*** LLDP HW address to be set

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST
LLDP_API_ERR_LLDP_MAC_ADDR_ERR

4.1.3.27 int smi_lldp_set_port_basic_tlvs_enable (struct smiclient_globals * *azg*, char * *ifname*, u_int16_t *tlv_flag*)

Sets the TLVs to be enabled for transmission on a port. smi_lldp_set_port_basic_tlvs_-enable

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name on which TLV should be enabled

← ***tlv_flag*** Flad indentifying the TLV to be enabled.

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS

4.1.3.28 int smi_lldp_set_port_msg_tx_hold (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t *tx_hold*)

Sets the message transmit hold parameter that determines the TTL value for LLDP PDU to be transmitted by the port. smi_lldp_set_port_msg_tx_hold

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which tx_hold needs to be set

← ***tx_hold*** Message Transmit hold time parameter value.

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS
LLDP_API_ERR_MSG_TX_INTERVAL_ERR

4.1.3.29 int smi_lldp_set_port_msg_tx_interval (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t *tx_interval*)

Sets the interval at which LLDP frames are transmitted. smi_lldp_set_port_msg_tx_-interval

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which tx_interval needs to be set

← ***tx_interval*** The value of tx_interval to be set

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS
LLDP_API_ERR_MSG_TX_INTERVAL_ERR

4.1.3.30 int smi_lldp_set_port_reinit_delay (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t *reinit_delay*)

Sets the delay time between when LLDP is disabled on a port and an attempt is made to reinitialize it. smi_lldp_set_port_reinit_delay

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which reinitialization delay needs to be set

← ***reinit_delay*** The value of reinitialization delay

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS

4.1.3.31 int smi_lldp_set_port_too_many_neighbors (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t *limit*, u_int8_t *type*, u_char * *mac*, u_int32_t *interval*)

Sets the action to be taken when LLDP remote table is full. smi_lldp_set_port_too_-many_neighbors

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which the set action will be applied

← ***limit*** the upper limit value for too many neighbours

← ***discard_type*** Too many neighbours discard types. Possible values are
TOO_MANY_NEIGHBORS_DISCARD_NONE (0)
TOO_MANY_NEIGHBORS_EXISTING_INFO (1)
TOO_MANY_NEIGHBORS_RECEIVED_INFO (2)

← ***mac*** The MAC address of the remote LLDP agent for which the specified action will be taken

← ***interval*** The period in seconds after which specified action will be taken

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_MAC_ADDR_ERR

4.1.3.32 int smi_lldp_set_port_tx_delay (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t *tx_delay*)

Sets the delay time between successive transmission of LLDP frames. smi_lldp_set_-port_tx_delay

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***ifname*** Interface name for which LLDP tranmission delay needs to be set

← ***tx_delay*** The value of LLDP transmission delay

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_ONM_IF_NOT_EXISTS
LLDP_API_ERR_LLDP_IF_NOT_EXISTS
LLDP_API_ERR_TX_DELAY_ERR

4.1.3.33 int smi_lldp_set_system_description (struct smiclient_globals * *azg*, u_char * *descriptor*)

Sets LLDP system description as provided. smi_lldp_set_system_description

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***sys_description*** LLDP system description string to be set

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.1.3.34 int smi_lldp_set_system_name (struct smiclient_globals * *azg*, u_char * *sys_name*)

Sets LLDP system name as provided. smi_lldp_set_system_name

Parameters:

← ***azg*** Pointer to smiclient_globals structure

← ***sys_name*** LLDP system name to be set

Returns:

0 on success, otherwise one of the following error code
LLDP_API_ERR_LLDP_MASTER_NOT_EXIST

4.2 smi_lldp_msg.h File Reference

Defines data structures used by LLDP SMI APIs. `#include "smi_message.h"`

Data Structures

- struct smi_remote_lldp
- struct smi_port_lldp_statistics
- struct smi_lldp_port
- struct smi_msg_lldp

Defines

- #define **SMI_MSG_LLDP_SIZE** 4
- #define **LLDP_LOCAL_MAX_LEN** 255
- #define **LLDP_DESCR_MAX_LEN** 255
- #define **LLDP_NAME_MAX_LEN** 255
- #define **SMI_LLDP_HOLD_MIN** 2
- #define **SMI_LLDP_HOLD_MAX** 10
- #define **SMI_LLDP_INTERVAL_MIN** 5
- #define **SMI_LLDP_INTERVAL_MAX** 32768
- #define **SMI_REINIT_DELAY_MIN** 1
- #define **SMI_REINIT_DELAY_MAX** 10
- #define **SMI_LLDP_NEIGHBORS_LIMIT_MIN** 1
- #define **SMI_LLDP_NEIGHBORS_LIMIT_MAX** 65535
- #define **SMI_LLDP_NEIGHBORS_INTERVAL_MIN** 1
- #define **SMI_LLDP_NEIGHBORS_INTERVAL_MAX** 65535
- #define **SMI_TX_DELAY_MIN** 1
- #define **SMI_TX_DELAY_MAX** 8192
- #define **SMI_NETWORK_ADDR_LENGTH** 4
- #define **SMI_TIME_MARK_INDEX_MAX** 10
- #define **SMI_MGMT_ADDR_LENGTH** 31
- #define **SMI_OID_LEN_MAX** 128
- #define **SMI_LLDP_ENABLED_RX_ONLY** 2
- #define **SMI_LLDP_ENABLED_TX_ONLY** 4
- #define **SMI_LLDP_ENABLED_RX_TX** 8
- #define **SMI_LLDPV2_MSG_FAST_TX_MIN** 1
- #define **SMI_LLDPV2_MSG_FAST_TX_MAX** 3600
- #define **SMI_LLDPV2_MSG_TX_HOLD_MIN** 1
- #define **SMI_LLDPV2_MSG_TX_HOLD_MAX** 100
- #define **SMI_LLDPV2_MSG_TX_INTERVAL_MIN** 1
- #define **SMI_LLDPV2_MSG_TX_INTERVAL_MAX** 3600
- #define **SMI_LLDPV2_TX_CREDIT_MAX_MIN** 1
- #define **SMI_LLDPV2_TX_CREDIT_MAX_MAX** 10
- #define **SMI_LLDPV2_TX_FAST_INIT_MIN** 1

- #define **SMI_LLDPV2_TX_FAST_INIT_MAX** 8
- #define **SMI_LLDPV2_TOO_MANY_NBR_LIMIT_MIN** 1
- #define **SMI_LLDPV2_TOO_MANY_NBR_LIMIT_MAX** 65535
- #define **SMI_LLDPV2_TOO_MANY_NBR_TIMER_MIN** 1
- #define **SMI_LLDPV2_TOO_MANY_NBR_TIMER_MAX** 65535
- #define **SMI_LLDPV2_REINIT_DELAY_MIN** 1
- #define **SMI_LLDPV2_REINIT_DELAY_MAX** 10
- #define **IF_MAU_AUTONEG_SUPPORTED** (1 << 0)
- #define **IF_MAU_AUTONEG_ENABLED** (1 << 1)
- #define **AUTONEGO_BOTHER** 0
- #define **AUTONEGO_B10BASET** 1
- #define **AUTONEGO_B10BASETFD** 2
- #define **AUTONEGO_B100BASET4** 3
- #define **AUTONEGO_B100BASETX** 4
- #define **AUTONEGO_B100BASETXFD** 5
- #define **AUTONEGO_B100BASET2** 6
- #define **AUTONEGO_B100BASET2FD** 7
- #define **AUTONEGO_BFDXPAUSE** 8
- #define **AUTONEGO_BFDXAPAUSE** 9
- #define **AUTONEGO_BFDXSPAUSE** 10
- #define **AUTONEGO_BFDXBPAUSE** 11
- #define **AUTONEGO_B1000BASEX** 12
- #define **AUTONEGO_B1000BASEXFD** 13
- #define **AUTONEGO_B1000BASET** 14
- #define **AUTONEGO_B1000BASETFD** 15
- #define **ONM_AGGREGATION_CAPABLE** (1 << 0)
- #define **ONM_AGGREGATION_ENABLE** (1 << 1)
- #define **TOO_MANY_NEIGHBOURS_DISCARD_NONE** 0
- #define **TOO_MANY_NEIGHBOURS_EXISTING_INFO** 1
- #define **TOO_MANY_NEIGHBOURS_RECIEVED_INFO** 2
- #define **SMI_CHASSIS_ID_TLV_TX_ENABLE** (1 << 0)
- #define **SMI_PORT_ID_TLV_TX_ENABLE** (1 << 1)
- #define **SMI_TTL_TLV_TX_ENABLE** (1 << 2)
- #define **SMI_PORT_DESCRIPTION_TLV_TX_ENABLE** (1 << 3)
- #define **SMI_SYSTEM_NAME_TLV_TX_ENABLE** (1 << 4)
- #define **SMI_SYSTEM_DESCRIPTION_TLV_TX_ENABLE** (1 << 5)
- #define **SMI_SYSTEM_CAPABILITIES_TLV_TX_ENABLE** (1 << 6)
- #define **SMI_MANAGEMENT_ADDRESS_TLV_TX_ENABLE** (1 << 7)
- #define **SMI_IEEE_8021_ORG_SPECIFIC_TLV_TX_ENABLE** (1 << 8)
- #define **SMI_IEEE_8023_ORG_SPECIFIC_TLV_TX_ENABLE** (1 << 9)
- #define **SMI_IPADDRESS_SIZE** 16
- #define **IFHWASIZ** 20
- #define **INTERFACE_HWADDR_MAX** (IFHWASIZ)
- #define **SMI_LLDP_CTYPE_IFNAME** 0
- #define **SMI_LLDP_CTYPE_ADMINSTATUS** 1
- #define **SMI_LLDP_CTYPE_LOCALSTRING** 2

- #define **SMI_LLDP_CTYPE_ENABLETLV** 3
- #define **SMI_LLDP_CTYPE_TXHOLDVALUE** 4
- #define **SMI_LLDP_CTYPE_TXINTERVAL** 5
- #define **SMI_LLDP_CTYPE_REINITDELAY** 6
- #define **SMI_LLDP_CTYPE_LIMIT** 7
- #define **SMI_LLDP_CTYPE_TYPE** 8
- #define **SMI_LLDP_CTYPE_MAC** 9
- #define **SMI_LLDP_CTYPE_INTERVAL** 10
- #define **SMI_LLDP_CTYPE_TXDELAY** 11
- #define **SMI_LLDP_CTYPE_DESCRIPTOR** 12
- #define **SMI_LLDP_CTYPE_SYSTEMNAME** 13
- #define **SMI_LLDP_CTYPE_HWADDR** 14
- #define **SMI_LLDP_CTYPE_PORT** 15
- #define **SMI_LLDP_CTYPE_STATPORT** 16
- #define **SMI_LLDP_CTYPE_REM_MAC_ARRAY** 17
- #define **SMI_LLDP_CTYPE_CHASSISID_TYPE** 18
- #define **SMI_LLDP_CTYPE_CHASSIS_IP** 19
- #define **SMI_LLDP_CTYPE_FIRST_CALL** 20
- #define **SMI_LLDP_CTYPE_SET_FLAG** 21
- #define **SMI_LLDP_CTYPE_AGT_CKT_ID** 22
- #define **SMI_LLDP_CTYPE_CREDIT** 23
- #define **SMI_LLDP_FAST_INIT** 24
- #define **SMI_LLDP_CTYPE_REMOTE_MAC** 25
- #define **SMI_LLDP_CTYPE_DEBUG** 26
- #define **SMI_LLDP_CTYPE_PORT_INFO** 27
- #define **SMI_LLDP_CTYPE_ALL_STATPORT** 28

Enumerations

- enum **smi_lldp_chassis_id_sub_type** {

  SMI_LLDP_CHASSIS_ID_CHASSIS_COMPONENT, **SMI_LLDP_-CHASSIS_ID_IF_ALIAS**, **SMI_LLDP_CHASSIS_ID_PORT_-COMPONENT**, **SMI_LLDP_CHASSIS_ID_MAC_ADDRESS**,

  SMI_LLDP_CHASSIS_ID_IP_ADDRESS, **SMI_LLDP_CHASSIS_ID_-IF_NAME**, **SMI_LLDP_CHASSIS_ID_LOCAL**, **SMI_LLDP_CHASSIS_-ID_INVALID** }

Functions

- void **smi_lldp_dump** (struct lib_globals *zg, struct smi_msg_lldp *msg)
- int **smi_encode_lldpmsg** (u_char **pnt, u_int16_t *size, struct smi_msg_lldp *msg)
- int **smi_decode_lldpmsg** (u_char **pnt, u_int16_t *size, struct smi_msg_lldp *msg)
- int **smi_parse_lldp** (u_char **pnt, u_int16_t *size, struct smi_msg_header *header, void *arg, SMI_CALLBACK callback)

4.2.1 Detailed Description

Defines data structures used by LLDP SMI APIs.

Index