ZebOS-XP Spanning Tree Protocol (xSTP) SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:29 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_bridge_id Struct Reference

Data Fields

- u_char **prio** [2]
- u_char **addr** [MSTP_ETHER_ADDR_LEN]

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

- smi_mstp_msg.h

3.2 smi_mac_addr Struct Reference

Data Fields

- u_char **addr** [MSTP_ETHER_ADDR_LEN]

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

- smi_mstp_msg.h

3.3 smi_msg_mstp Struct Reference

Data Fields

- smi_cindex_t **cindex**
- smi_cindex_t **extended_cindex_1**
- smi_cindex_t **extended_cindex_2**
- char **if_name** [INTERFACE_NAMSIZ]
- char **bridge_name** [BRIDGE_NAMESIZ]
- unsigned char **txholdcount**
- int **instance**
- int **enabled**
- int **bpduguard_en**
- int **bridge_bpdufilter**
- u_int16_t **vid**
- u_int16_t **revision_nu**
- u_int16_t **port_priority**
- s_int32_t **timeout**
- s_int16_t **portpriority_br**
- s_int32_t **age**
- s_int32_t **hello_time**
- s_int32_t **max_age**
- s_int32_t **port_edge**
- s_int32_t **version**
- u_int32_t **pt_method**
- s_int32_t **port_hello_time**
- u_int32_t **new_priority**
- u_int32_t **msti_priority**
- u_int32_t **forward_delay**
- u_int32_t **cost**
- u_int32_t **restricted_role**
- s_int16_t **restricted_prole**
- u_int32_t **restricted_tcn**
- s_int16_t **restricted_ptcn**
- u_int32_t **is_p2p**
- u_int32_t **path_cost**
- s_int32_t **max_hops**
- s_int32_t **port_rootg**
- u_char **portfast_bpdu**
- u_char **bpdu_guard**
- u_int8_t **extended_ctype**
- u_int32_t **debug**
- u_int32_t **blk_inc**
- u_int32_t **br_forward**
- u_int32_t **auto_edge**
- u_int32_t **in_start**

- u_int32_t **in_end**
- char **region_name** [MSTP_CONFIG_NAME_LEN]
- struct smi_mstp_spanning_tree_details **tree_details**
- struct smi_mstp_port_name_msg **port_name**
- struct smi_mstp_port_details **port_details**
- struct smi_mstp_instance_details **instance_details**
- struct smi_mstp_instance_port_details **instanceport_details**
- struct smi_mstp_config_details **config_details**
- struct smi_mstp_interface_all_msg **all_if**
- struct smi_mstp_instance_all_msg **all_inst**
- u_char **mac_addr** [MSTP_ETHER_ADDR_LEN ∗2+SMI_ETHER_DELIMETER_LEN]
- u_char **stp_enabled**
- struct smi_vlan_bmp **vlanBmp**
- enum ha_switch **switch_to_state**
- u_int8_t **path_cost_method**
- struct smi_mstp_mst_br_all_msg **bridge_info**
- struct smi_mstp_mst_config_all_msg **br_conf_details**
- struct smi_mstp_instance_detail_msg **br_inst_info**
- struct smi_mstp_br_inst_all_msg **br_inst_msg**
- struct smi_mstp_br_port_all_msg **br_port_msg**
- struct smi_mstp_stats_all **br_stats_msg**
- int **l2gp_status**
- u_int8_t **cisco_intr_status**
- int **role**
- int **state**

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

- smi_mstp_msg.h

3.4 smi_mstp_bmp Struct Reference

Data Fields

- u_int32_t **bitmap** [SMI_BMP_WORD_MAX]

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

- smi_mstp_msg.h

3.5 smi_mstp_br_inst Struct Reference

Data Fields

- u_int32_t **msti_bridge_priority**
- struct smi_bridge_id **msti_bridge_id**
- u_int16_t **tc_initiator**
- pal_time_t **time_last_topo_change**
- bool_t **tc_flag**
- u_char **topology_change_detected**:1
- u_char **tc_last_rcvd_from** [6]
- u_int32_t **total_num_topo_changes**

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

- smi_mstp_msg.h

3.6 smi_mstp_br_inst_all_msg Struct Reference

Data Fields

- int **have_more_flag**
- int **have_more**
- int **count**
- struct smi_mstp_br_inst **br_inst**
- struct list ∗ **inst_port_list**

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

- smi_mstp_msg.h

3.7 smi_mstp_br_port Struct Reference

Data Fields

- char **port_name** [24]
- int **ifindex**
- u_int32_t **conf_bpdu_sent**
- u_int32_t **conf_bpdu_rcvd**
- u_int32_t **tcn_bpdu_sent**
- u_int32_t **tcn_bpdu_rcvd**
- s_int32_t **cist_message_age**
- u_int32_t **cist_forward_transitions**

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

- smi_mstp_msg.h

3.8 smi_mstp_br_port_all_msg Struct Reference

Data Fields

- int **have_more**
- int **count**
- struct list ∗ **br_port_list**

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

- smi_mstp_msg.h

3.9 smi_mstp_bridge_detail Struct Reference

Data Fields

- char **name** [SMI_BRIDGE_NAMESIZ+1]
- u_int16_t **num_of_int**
- struct smi_mstp_config_details **config**
- u_int8_t **type**
- struct smi_mac_addr **bridge_addr**
- struct smi_bridge_id **cist_bridge_id**
- struct smi_bridge_id **cist_designated_root**
- struct smi_bridge_id **cist_reg_root**
- struct smi_bridge_id **cist_designated_bridge**
- u_int32_t **external_root_path_cost**
- u_int32_t **internal_root_path_cost**
- u_int16_t **cist_root_port_id**
- u_int32_t **cist_root_port_ifindex**
- s_int32_t **cist_max_age**
- s_int32_t **cist_message_age**
- s_int32_t **cist_hello_time**
- s_int32_t **cist_forward_delay**
- s_int32_t **hop_count**
- u_char **force_version**
- s_int32_t **bridge_max_age**
- s_int32_t **bridge_hello_time**
- s_int32_t **bridge_forward_delay**
- s_int32_t **bridge_max_hops**
- u_int32_t **cist_bridge_priority**
- s_int32_t **ageing_time**
- u_char **topology_change**:1
- u_char **topology_change_detected**:1
- unsigned char **bpduguard**
- s_int32_t **errdisable_timeout_interval**
- unsigned char **errdisable_timeout_enable**
- unsigned char **bpdu_filter**
- unsigned char **oper_cisco**
- pal_time_t **time_last_topo_change**
- u_int32_t **num_topo_changes**
- u_char **bridge_enabled**:1
- u_char **is_default**:1
- u_char **mstp_enabled**:1
- u_char **transmit_hold_count**
- u_int8_t **path_cost_method**
- u_int16_t **num_instances**
- struct smi_mstp_bmp **instanceBmp**
- struct smi_mstp_bmp **vlanMemberBmp**

- struct smi_mstp_interface_all_msg **all_port**
- struct list ∗ **mst_inst_list**

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

- smi_mstp_msg.h

3.10 smi_mstp_config_details Struct Reference

Data Fields

- u_char **cfg_format_id**
- char **cfg_name** [MSTP_CONFIG_NAME_LEN+1]
- u_int16_t **cfg_revision_lvl**
- char **cfg_digest** [SMI_MSTP_CONFIG_DIGEST_LEN]

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

- smi_mstp_msg.h

3.11 smi_mstp_instance_all_msg Struct Reference

Data Fields

- u_int16_t **have_more**
- u_int16_t **list_cnt**
- u_int16_t **port_cnt**
- struct list ∗ **all_instance_list**

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

- smi_mstp_msg.h

3.12 smi_mstp_instance_detail_msg Struct Reference

Data Fields

- int **have_more**
- char **bridge_name** [SMI_BRIDGE_NAMESIZ+1]
- u_int32_t **internal_root_path_cost**
- u_int32_t **msti_root_port_ifindex**
- u_int32_t **msti_bridge_priority**
- struct smi_mstp_interface_all_msg **all_port**

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

- smi_mstp_msg.h

3.13 smi_mstp_instance_details Struct Reference

Data Fields

- u_int8_t **instance_id**
- u_int16_t **num_of_int**
- u_char **master**
- struct smi_bridge_id **msti_bridge_id**
- u_int32_t **msti_bridge_priority**
- struct smi_bridge_id **msti_designated_root**
- struct smi_bridge_id **msti_designated_bridge**
- u_int32_t **internal_root_path_cost**
- u_int32_t **designated_internal_root_path_cost**
- u_int16_t **msti_root_port_id**
- u_int32_t **msti_root_port_ifindex**
- struct smi_mstp_bmp **vlanMemberBmp**
- s_int32_t **hop_count**
- struct smi_mstp_bmp **port_list**
- struct smi_mstp_interface_all_msg **all_port**
- s_int32_t **message_age**
- s_int32_t **max_age**
- s_int32_t **fwd_delay**
- s_int32_t **hello_time**
- u_int32_t **vid**

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

- smi_mstp_msg.h

3.14 smi_mstp_instance_port_details Struct Reference

Data Fields

- char **name** [BRIDGE_NAMESIZ+1]
- u_int32_t **ifindex**
- u_int8_t **instance_id**
- u_char **admin_p2p_mac**:1
- u_char **oper_p2p_mac**:1
- char **ifname** [SMI_L2_IF_NAME_LEN]
- u_int16_t **msti_port_id**
- enum smi_port_role **msti_role**
- enum smi_port_state **msti_state**
- u_int32_t **internal_rpc**
- u_int32_t **msti_path_cost**
- s_int32_t **designated_port_id**
- s_int16_t **msti_priority**
- struct smi_bridge_id **msti_root**
- struct smi_bridge_id **msti_designated_bridge**
- s_int32_t **msti_message_age**
- s_int32_t **msti_max_age**
- s_int32_t **msti_fwd_delay**
- s_int32_t **msti_hello_time**
- u_int32_t **cist_forward_transitions**
- u_int32_t **conf_bpdu_sent**
- u_int32_t **conf_bpdu_rcvd**
- u_int32_t **tcn_bpdu_sent**
- u_int32_t **tcn_bpdu_rcvd**
- s_int32_t **forward_delay_time_remain**
- s_int32_t **hello_timer_remain**
- s_int32_t **message_age_timer_remain**
- s_int32_t **tc_timer_remain**

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

- smi_mstp_msg.h

3.15 smi_mstp_interface_all_msg Struct Reference

Data Fields

- u_int16_t **have_more**
- u_int16_t **list_cnt**
- struct list ∗ **all_port_list**

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

- smi_mstp_msg.h

3.16 smi_mstp_mst_br_all_msg Struct Reference

Data Fields

- int **have_more**
- int **num_of_inst**
- int **port_count**
- int **start_index**
- int **end_index**
- int **count**
- struct list ∗ **mst_msg_list**

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

- smi_mstp_msg.h

3.17 smi_mstp_mst_config Struct Reference

Data Fields

- char **bridge_name** [SMI_BRIDGE_NAMSIZ]
- struct smi_mstp_config_details **br_config_msg**

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

- smi_mstp_msg.h

3.18 smi_mstp_mst_config_all_msg Struct Reference

Data Fields

- int **have_more**
- int **start_index**
- int **end_index**
- int **count**
- struct list ∗ **mst_config_list**

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

- smi_mstp_msg.h

3.19 smi_mstp_port_details Struct Reference

Data Fields

- char **name** [BRIDGE_NAMESIZ+1]
- u_int32_t **ifindex**
- char **ifname** [SMI_L2_IF_NAME_LEN]
- u_char **force_version**
- u_char **admin_p2p_mac**:1
- u_char **oper_p2p_mac**:1
- u_char **admin_edge**:1
- u_char **oper_edge**:1
- u_char **auto_edge**:1
- u_char **portfast_conf**:1
- u_int16_t **cist_port_id**
- enum smi_port_role **cist_role**
- enum smi_port_state **cist_state**
- u_int32_t **cist_external_rpc**
- u_int32_t **cist_internal_rpc**
- u_int32_t **cist_path_cost**
- s_int32_t **cist_designated_port_id**
- s_int16_t **cist_priority**
- struct smi_bridge_id **cist_root**
- struct smi_bridge_id **cist_reg_root**
- struct smi_bridge_id **cist_designated_bridge**
- s_int32_t **cist_message_age**
- s_int32_t **cist_max_age**
- s_int32_t **cist_fwd_delay**
- s_int32_t **cist_hello_time**
- u_int32_t **cist_forward_transitions**
- s_int32_t **forward_delay_time_remain**
- s_int32_t **hello_timer_remain**
- s_int32_t **message_age_timer_remain**
- s_int32_t **tc_timer_remain**
- s_int32_t **admin_bpduguard**
- u_char **oper_bpduguard**
- s_int32_t **admin_bpdufilter**
- u_char **oper_bpdufilter**
- u_char **admin_rootguard**
- u_char **oper_rootguard**
- u_int32_t **tx_count**
- u_int32_t **rx_count**
- u_int32_t **total_tx_count**

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

- smi_mstp_msg.h

3.20 smi_mstp_port_name_msg Struct Reference

Data Fields

- u_int32_t **count**
- struct list ∗ **all_port_name**

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

- smi_mstp_msg.h

3.21 smi_mstp_spanning_tree_details Struct Reference

Data Fields

- u_int16_t **num_bridges**
- struct smi_mstp_bridge_detail **bridge**

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

- smi_mstp_msg.h

3.22 smi_mstp_stats Struct Reference

Data Fields

- struct smi_mstp_port_details **port_detail**
- u_char **port_enabled**:1
- char **cist_port_state** [15]
- u_char **tc_ack**:1
- u_char **config_bpdu_pending**:1
- u_int32_t **conf_bpdu_sent**
- u_int32_t **conf_bpdu_rcvd**
- u_int32_t **tcn_bpdu_sent**
- u_int32_t **tcn_bpdu_rcvd**
- s_int32_t **hello_time**
- int **is_hello_timer**
- int **is_forward_delay_timer**
- int **is_message_age_timer**
- int **is_tc_timer**
- int **is_hold_timer**
- int **hold_timer_value**
- u_int32_t **msg_age_timer_cnt**
- u_int32_t **similar_bpdu_cnt**
- u_int32_t **total_src_mac_count**
- int **cist_next_state**
- s_int32_t **topology_change_time**
- s_int32_t **max_age_count**
- u_int32_t **src_mac_count**

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

- smi_mstp_msg.h

3.23 smi_mstp_stats_all Struct Reference

Data Fields

- int **have_more**
- int **instance_have_more**
- int **count**
- struct smi_bridge_id **cist_bridge_id**
- u_int32_t **cist_bridge_priority**
- s_int32_t **bridge_hello_time**
- s_int32_t **bridge_forward_delay**
- u_int16_t **tc_initiator**
- pal_time_t **time_last_topo_change**
- bool_t **tc_flag**
- u_char **topology_change_detected**:1
- u_int32_t **total_num_topo_changes**
- u_char **tc_last_rcvd_from** [6]
- struct list ∗ **port_stat_list**

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

- smi_mstp_msg.h

3.24 smi_mstp_stats_inst_port Struct Reference

Data Fields

- int **ifindex**
- u_int32_t **conf_bpdu_sent**
- u_int32_t **conf_bpdu_rcvd**
- u_int32_t **tcn_bpdu_sent**
- u_int32_t **tcn_bpdu_rcvd**
- s_int32_t **message_age**
- u_int8_t **next_state**
- s_int32_t **topology_change_time**

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

- smi_mstp_msg.h

3.25 smi_port_name Struct Reference

Data Fields

- char **name** [SMI_L2_IF_NAME_LEN]

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

- smi_mstp_msg.h

Chapter 4

File Documentation

4.1 smi_mstp.h File Reference

Provides API for managing xSTP protocols. `#include "smi_client.h"` `#include "parser_utils.h"`

Functions

- int smi_mstp_add_instance (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, struct smi_vlan_bmp ∗vlanBmp, struct smi_vlan_bmp ∗successBmp)

  This API creates an MSTP instance on a bridge, which provides functionality to add VLANs to a MSTP instance running on a bridge. Multiple VLANs can be added to an instance at once.

- int smi_mstp_delete_instance (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, struct smi_vlan_bmp ∗vlanBmp, struct smi_vlan_bmp ∗successBmp)

  This API deletes provides the functionality to delete VLANs from an MSTP instance running on a bridge. Multiple VLANs can be deleted from an instance at once. In addition, multiple VLANs can be deleted from an STP instance based on the VLAN ID of each VLAN.

- int **smi_mstp_add_instance_wrap** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, char ∗range)
- int **smi_mstp_update_instance_wrap** (struct smiclient_globals ∗azg, char ∗name, int instance, char ∗range)
- int **smi_mstp_update_instance_wrap_validate** (struct smiclient_globals ∗azg, char ∗name, int instance, char ∗range)
- int **smi_mstp_add_instance_vlan_wrap** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, int vlanId)

- int **smi_mstp_delete_instance_wrap** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, char ∗range)
- int **smi_mstp_add_instance_wrap_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, char ∗range)
- int **smi_mstp_add_instance_vlan_wrap_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, int vlanId)
- int **smi_mstp_delete_instance_wrap_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, char ∗range)
- int **smi_mstp_api_add_instance_wrap** (struct smiclient_globals ∗azg, char ∗name, int instance)
- int **smi_mstp_api_add_instance_wrap_validate** (struct smiclient_globals ∗azg, char ∗name, int instance)
- int smi_mstp_add_port (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId)

  This API adds an interface to a MSTP instance on a bridge group. In addition, it configures the bridge port to the type "Explicit" if the port was implicitly configured in the bridge group.

- int smi_mstp_delete_port (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId)

  This API deletes an interface from the MSTP instance of a bridge group.

- int smi_mstp_set_msti_bridgepriority (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, u_int32_t priority)

  This API configures the bridge priority for an MSTI bridge.

- int smi_mstp_get_msti_bridgepriority (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, u_int32_t ∗priority)

  This API configures the bridge priority for an MSTI bridge.

- int smi_mstp_set_msti_portpathcost (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, u_int32_t pathCost)

  This API configures the path cost for an MSTI port on a specific bridge.

- int smi_mstp_get_msti_portpathcost (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, u_int32_t ∗pathCost)

  This API retrieves the path cost currently configured on a MSTI port.

- int smi_mstp_set_msti_instance_restrictedrole (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, bool_t restrictedRole)

  This API configures restricted role on a port for an MSTP instance. Enabling restricted role for a port prevents it from becoming the root port. By default, restricted role is disabled on an MSTI port.

- int smi_mstp_get_msti_instance_restrictedrole (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, u_int32_t ∗restrictedRole)

  This API retrieves the restricted role configuration on a MSTI port.

- int smi_mstp_set_msti_instance_restrictedtcn (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, bool_t restrictedTCN)

  This API configures the instance to ignore any TCN (Topology Change Notification) received on a port for an MSTP instance.

- int smi_mstp_get_msti_instance_restrictedtcn (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, u_int32_t ∗restrictedTCN)

  This API retrieves the restricted TCN status of a port for a MSTP instance.

- int smi_mstp_set_msti_portpriority (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, s_int16_t portPriority)

  This API configures the port priority for a bridge.

- int smi_mstp_get_msti_portpriority (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, s_int16_t ∗portPriority)

  This API retrieves the port priority for a bridge.

- int smi_mstp_get_spanningtree_details (struct smiclient_globals ∗azg, char ∗bridgeId, struct smi_mstp_spanning_tree_details ∗treeDet)

  This API retrieves the spanning tree details of all the bridges configured in the device.

- int smi_mstp_get_spanningtree_interface (struct smiclient_globals ∗azg, char ∗ifName, struct smi_mstp_port_details ∗portDet)

  This API retrieves the state of the spanning tree for all named bridge-groups of the specified interface.

- int smi_mstp_get_spanningtree_mst (struct smiclient_globals ∗azg, char ∗bridgeId, u_int16_t instanceId, struct smi_mstp_instance_details ∗instanceDet)

  This API retrieves the MSTP instance details, which includes identifying information, master and root information, port list, and other information.

- int smi_mstp_get_spanningtree_mstconfig (struct smiclient_globals ∗azg, char ∗bridgeId, struct smi_mstp_config_details ∗configDet)

  This API retrieves the MSTP configuration information for a bridge.

- int smi_mstp_get_spanningtree_mstdetail_interface (struct smiclient_globals ∗azg, char ∗bridgeId, u_int16_t instanceId, struct smi_mstp_instance_port_details ∗details)

  This API retrieves detailed information on MSTP bridge instance to which an interface is attached.

- int smi_mstp_get_spanningtree_mstdetail (struct smiclient_globals ∗azg, char ∗bridgeId, u_int16_t instanceId, struct smi_mstp_instance_details ∗details)

  This API retrieves detailed information about an MSTP instance.

- s_int32_t smi_mstp_show_port_all (struct smiclient_globals ∗azg, char ∗bridgeId, struct list ∗readmsg, u_int32_t(∗callback)(struct smi_mstp_spanning_tree_details ∗treeDet))

  This API retrieves the all bridge port info in one IPC.

- s_int32_t smi_mstp_show_inc_blk_ports (struct smiclient_globals ∗azg, char ∗bridgeId, struct list ∗readmsg, int flag, u_int32_t(∗callback)(struct smi_mstp_port_name_msg ∗getmsg))

  This API retrieves the inconsistent and blocked ports.

- s_int32_t smi_mstp_show_instance_all (struct smiclient_globals ∗azg, char ∗bridgeId, struct list ∗readmsg, u_int32_t in_start, u_int32_t in_end, u_int32_t(∗callback)(struct smi_mstp_instance_all_msg ∗getmsg))

  This API retrieves the all bridge instance info in one IPC.

- int smi_mstp_api_unset_msti_bridge_priority (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId)

  This API unset the bridge priority for an MSTI bridge.

- int smi_mstp_api_unset_msti_port_path_cost (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId)

  This API unset the path cost for an MSTI port on a specific bridge.

- int smi_mstp_api_delete_instance (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId)

  This API delete the instance.

- int smi_mstp_stats_clear_by_ifname (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName)

  This API unset a BPDU-filter configuration for a bridge port.

- int smi_mstp_stats_clear_by_ifname_instance (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId)

  This API unset a BPDU-filter configuration for a bridge port.

- int smi_mstp_stats_clear_by_ifname_vlan (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, u_int16_t vlanId)

  This API unset a BPDU-filter configuration for a bridge port.

- int smi_mstp_stats_clear_by_vlan (struct smiclient_globals ∗azg, char ∗bridgeId, u_int16_t vlanId)

  This API unset a BPDU-filter configuration for a bridge port.

- int smi_mstp_stats_clear_by_instance (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId)

  This API unset a BPDU-filter configuration for a bridge port.

- int smi_mstp_stats_clear (struct smiclient_globals ∗azg, char ∗bridgeId)

  This API unset a BPDU-filter configuration for a bridge port.

- int smi_mstp_api_get_interface_role_by_vlan (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId, char ∗ifName, int ∗role)

  Gets This function gets state of RPVST+ bridge vlan .

- int smi_mstp_api_get_interface_state_by_vlan (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId, char ∗ifName, int ∗state)

  Gets This function gets state of of RPVST+ bridge vlan.

- int smi_mstp_api_get_interface_role_by_instance (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, char ∗ifName, int ∗role)

  Gets This function gets state of interface in bridge.

- int smi_mstp_api_get_interface_state_by_instance (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, char ∗ifName, int ∗state)

  Gets This function gets state of interface in bridge.

- int **smi_mstp_api_show_stat** (struct smiclient_globals ∗azg, struct smi_mstp_br_inst_all_msg ∗brInstMsg, struct smi_mstp_br_port_all_msg ∗brPortMsg, int msgtype)
- s_int32_t **smi_mstp_stats_show** (struct smiclient_globals ∗azg, char ∗bridgeId, struct smi_mstp_stats_all ∗brStatsMsg)
- s_int32_t **smi_mstp_stats_show_by_inst** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, struct smi_mstp_br_inst_all_msg ∗brInstMsg, struct smi_mstp_br_port_all_msg ∗brPortMsg)
- s_int32_t **smi_mstp_stats_show_by_vlan** (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId, struct smi_mstp_br_inst_all_msg ∗brInstMsg, struct smi_mstp_br_port_all_msg ∗brPortMsg)
- s_int32_t **smi_mstp_stats_show_by_if_inst** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, struct smi_mstp_br_inst_all_msg ∗brInstMsg, struct smi_mstp_br_port_all_msg ∗brPortMsg)
- s_int32_t **smi_mstp_stats_show_by_if_vlan** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, struct smi_mstp_br_inst_all_msg ∗brInstMsg, struct smi_mstp_br_port_all_msg ∗brPortMsg)
- s_int32_t **smi_mstp_stats_show_by_if** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, struct smi_mstp_stats_all ∗brStatsMsg, struct list ∗brInstAll)
- int **smi_mstp_show_st_mst** (struct smiclient_globals ∗azg, int inStart, int inEnd, struct list ∗mstMsgList)
- int **smi_mstp_show_st_mst_config** (struct smiclient_globals ∗azg, int in_start, int inEnd, struct list ∗mstConfigList)
- int **smi_mstp_show_st_mst_inst_if** (struct smiclient_globals ∗azg, int instanceId, char ∗ifName, struct list ∗mstMsgList, struct smi_mstp_instance_detail_msg ∗brInstInfo)
- int **smi_mstp_show_st_mst_inst** (struct smiclient_globals ∗azg, int instanceId, struct list ∗mstMsgList, struct list ∗instMsgList)

- int **smi_mstp_show_st_mst_detail** (struct smiclient_globals ∗azg, struct list ∗mstMsgList)
- int **smi_mstp_show_st_mst_detail_if** (struct smiclient_globals ∗azg, char ∗ifName, struct list ∗mstMsgList)
- int **smi_mstp_add_instance_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, struct smi_vlan_bmp ∗vlanBmp, struct smi_vlan_bmp ∗successBmp)
- int **smi_mstp_delete_instance_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, struct smi_vlan_bmp ∗vlanBmp, struct smi_vlan_bmp ∗successBmp)
- int **smi_mstp_add_port_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId)
- int **smi_mstp_delete_port_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId)
- int **smi_mstp_set_msti_bridgepriority_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId, u_int32_t priority)
- int **smi_mstp_set_msti_portpathcost_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, u_int32_t cost)
- int **smi_mstp_set_msti_instance_restrictedrole_validate** (struct smiclient_-globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, bool_t restricted-Role)
- int **smi_mstp_set_msti_instance_restrictedtcn_validate** (struct smiclient_-globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, bool_t restrict-edTCN)
- int **smi_mstp_set_msti_portpriority_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId, s_int16_t portPriority)
- int **smi_mstp_api_unset_msti_bridge_priority_validate** (struct smiclient_-globals ∗azg, char ∗bridgeId, int instanceId)
- int **smi_mstp_api_unset_msti_port_path_cost_validate** (struct smiclient_-globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId)
- int **smi_mstp_api_delete_instance_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId)
- int **smi_mstp_stats_clear_by_ifname_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName)
- int **smi_mstp_stats_clear_by_ifname_instance_validate** (struct smiclient_-globals ∗azg, char ∗bridgeId, char ∗ifName, int instanceId)
- int **smi_mstp_stats_clear_by_ifname_vlan_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, u_int16_t vlanId)
- int **smi_mstp_stats_clear_by_vlan_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, u_int16_t vlanId)
- int **smi_mstp_stats_clear_by_instance_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int instanceId)
- int **smi_mstp_stats_clear_validate** (struct smiclient_globals ∗azg, char ∗bridgeId)

4.1.1 Detailed Description

Provides API for managing xSTP protocols.

4.1.2 Function Documentation

4.1.2.1 int smi_mstp_add_instance (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *instanceId*, struct smi_vlan_bmp ∗ *vlanBmp*, struct smi_vlan_bmp ∗ *successBmp*)

This API creates an MSTP instance on a bridge, which provides functionality to add VLANs to a MSTP instance running on a bridge. Multiple VLANs can be added to an instance at once. smi_mstp_add_instance

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_-ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.

← ***vlanBmp*** Structure that stores the VLAN ID of the VLAN to be added to an instance. If the added vlanBmp parameter is empty, the MSTP instance is created without a VLAN

→ ***successBmp*** Structure that stores the VLAN ID of the VLAN that was successfully added.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.2 int smi_mstp_add_port (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*)

This API adds an interface to a MSTP instance on a bridge group. In addition, it configures the bridge port to the type “Explicit” if the port was implicitly configured in the bridge group. smi_mstp_add_port

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_-ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.3 int smi_mstp_api_delete_instance (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *instanceId*)

This API delete the instance. smi_mstp_api_delete_instance

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** Instance ID

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_INSTANCE_NOT_FOUND

4.1.2.4 int smi_mstp_api_get_interface_role_by_instance (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *instanceId*, char ∗ *ifName*, int ∗ *role*)

Gets This function gets state of interface in bridge. smi_mstp_api_get_interface_state

Parameters:

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

← ***bridgeId*** Bridge Name

← ***instanceId*** Instance

← ***ifName*** Interface Name

→ ***role*** role of interface

Returns:

RESULT_OK, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.5 int smi_mstp_api_get_interface_role_by_vlan (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *vlanId*, char ∗ *ifName*, int ∗ *role*)

Gets This function gets state of RPVST+ bridge vlan . smi_mstp_api_get_interface_role_by_vlan

Parameters:

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

← ***bridgeId*** Bridge Name

← ***vlanId*** vlan id

← ***ifName*** Interface Name

→ ***role*** role of interface

Returns:

RESULT_OK, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.6 int smi_mstp_api_get_interface_state_by_instance (struct smiclient_globals * *azg*, char * *bridgeId*, int *instanceId*, char * *ifName*, int * *state*)

Gets This function gets state of interface in bridge. smi_mstp_api_get_interface_state

Parameters:

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

← ***bridgeId*** Bridge Name

← ***instanceId*** Instance

← ***ifName*** Interface Name

→ ***state*** Portstate in bridge

Returns:

RESULT_OK, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.7 int smi_mstp_api_get_interface_state_by_vlan (struct smiclient_globals * *azg*, char * *bridgeId*, int *vlanId*, char * *ifName*, int * *state*)

Gets This function gets state of of RPVST+ bridge vlan. smi_mstp_api_get_interface_state_by_vlan

Parameters:

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

← ***bridgeId*** Bridge Name

← ***vlanId*** vlan id

← ***ifName*** Interface Name

→ ***state*** Portstate in bridge

Returns:

RESULT_OK, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.8 int smi_mstp_api_unset_msti_bridge_priority (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *instanceId*)

This API unset the bridge priority for an MSTI bridge. smi_mstp_api_unset_msti_bridge_priority

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_INSTANCE_OUTOFBOUNDS
MSTP_ERR_INSTANCE_NOT_FOUN
MSTP_ERR_PRIORITY_VALUE_WRONG
MSTP_ERR_PRIORITY_OUTOFBOUNDS

4.1.2.9 int smi_mstp_api_unset_msti_port_path_cost (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*)

This API unset the path cost for an MSTI port on a specific bridge. smi_mstp_api_unset_msti_port_path_cost

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_IF_NOT_FOUND
MSTP_ERR_BRIDGE_NOT_FOUND
RESULT_ERROR

4.1.2.10 int smi_mstp_delete_instance (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *instanceId*, struct smi_vlan_bmp ∗ *vlanBmp*, struct smi_vlan_bmp ∗ *successBmp*)

This API deletes provides the functionality to delete VLANs from an MSTP instance running on a bridge. Multiple VLANs can be deleted from an instance at once. In addition, multiple VLANs can be deleted from an STP instance based on the VLAN ID of each VLAN. smi_mstp_delete_instance

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_-ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.

← ***vlanBmp*** Structure that stores the VLAN ID of the VLAN to be added to an instance. If the added vlanBmp parameter is empty, the MSTP instance is created without a VLAN

→ ***successBmp*** Structure that stores the VLAN IDs of the VLANs that were successfully deleted.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.11 int smi_mstp_delete_port (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*)

This API deletes an interface from the MSTP instance of a bridge group. smi_mstp_-delete_port

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_-ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.12 int smi_mstp_get_msti_bridgepriority (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *instanceId*, u_int32_t ∗ *priority*)

This API configures the bridge priority for an MSTI bridge. smi_mstp_get_msti_-bridgepriority

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

→ ***priority*** Pointer to the retrieved MSTI bridge priority value. Values can be in the range of <SMI_MSTP_BRIDGE_PR_MIN and="" smi_mstp_bridge_pr_max>="">. Memory allocation for this parameter must be done by the caller of this function.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN

4.1.2.13 int smi_mstp_get_msti_instance_restrictedrole (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*, u_int32_t ∗ *restrictedRole*)

This API retrieves the restricted role configuration on a MSTI port. smi_mstp_get_msti_instance_restrictedrole

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>

→ ***restrictedRole*** Value indicates the restricted role status of the MSTI port. Memory allocation for this parameter must be done by the caller

PAL_TRUE Enable restricted role on a port.

PAL_FALSE Disable restricted role on a port

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN

4.1.2.14 int smi_mstp_get_msti_instance_restrictedtcn (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*, u_int32_t ∗ *restrictedTCN*)

This API retrieves the restricted TCN status of a port for a MSTP instance. smi_mstp_get_msti_instance_restrictedtcn

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_-ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>

→ ***restrictedTCN*** Pointer to retrieve restricted TCN value currently configured on the port. Memory allocation for this parameter must be done by caller of this function. Values can include one of the following:

PAL_TRUE Remove restricted TCN on a port for an MSTP instance.

PAL_FALSE Do not remove restricted TCN on a port for an MSTP instance.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN

4.1.2.15 int smi_mstp_get_msti_portpathcost (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*, u_int32_t ∗ *pathCost*)

This API retrieves the path cost currently configured on a MSTI port. smi_mstp_get_-msti_portpathcost

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_-ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.

→ ***cost*** Path cost value to be set for an MSTI port. Value must be in the range of <SMI_MSTP_PORTPATHCOST_MIN and="" smi_mstp_portpathcost_-max>="">.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN

4.1.2.16 int smi_mstp_get_msti_portpriority (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*, s_int16_t ∗ *portPriority*)

This API retrieves the port priority for a bridge. smi_mstp_get_msti_portpriority

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** Instance ID of MSTP in the range of, including <SMI_MSTP_-INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.

← ***portPriority*** Memory for this parameter must be allocated by the caller. On returning SMI_SUCCESS, the priority value is populated in the following range of <SMI_MSTP_PORT_PRIORITY_MIN - SMI_MSTP_PORT_-PRIORITY_MAX>.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
MSTP_ERR_PRIORITY_VALUE_WRONG
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_PORT_NOT_FOUND

4.1.2.17 int smi_mstp_get_spanningtree_details (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, struct smi_mstp_spanning_tree_details ∗ *treeDet*)

This API retrieves the spanning tree details of all the bridges configured in the device. smi_mstp_get_spanningtree_details

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

→ ***treeDet*** Pointer to structure . The caller must allocate memory for this parameter before invoking this API. On success the values populate with spanning tree details returning

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.18 int smi_mstp_get_spanningtree_interface (struct smiclient_globals ∗ *azg*, char ∗ *ifName*, struct smi_mstp_port_details ∗ *portDet*)

This API retrieves the state of the spanning tree for all named bridge-groups of the specified interface. smi_mstp_get_spanningtree_interface

Parameters:

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

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ

→ ***portDet*** Pointer to structure . The caller must allocate memory for this parameter before invoking this API. on returning success, the values populate with MSTP bridge port details

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.19 int smi_mstp_get_spanningtree_mst (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, u_int16_t *instanceId*, struct smi_mstp_instance_details ∗ *instanceDet*)

This API retrieves the MSTP instance details, which includes identifying information, master and root information, port list, and other information. smi_mstp_get_-spanningtree_mst

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** MSTP instance ID of the range, including <SMI_MSTP_-INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>

→ ***instanceDet*** Pointer to structure . The caller must allocate memory for this parameter before invoking this API. On returning success, the values populate with bridge instance details. If there is no such instance, all the attributes in the structure are set to zero (0). The caller can check the instance_det->instance_id to the input parameter instance to validate the instance availability

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.20 int smi_mstp_get_spanningtree_mstconfig (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, struct smi_mstp_config_details ∗ *configDet*)

This API retrieves the MSTP configuration information for a bridge. smi_mstp_get_-spanningtree_mstconfig

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

→ ***configDet*** Pointer to structure . The caller must allocate memory for this parameter before invoking this API. On returning success, the values populate with the bridge configuration details

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.21 int smi_mstp_get_spanningtree_mstdetail (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, u_int16_t *instanceId*, struct smi_mstp_instance_details ∗ *details*)

This API retrieves detailed information about an MSTP instance. smi_mstp_get_-spanningtree_mstdetail

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** MSTP instance ID of the range, including <SMI_MSTP_-INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>

→ ***details*** Pointer to structure . The caller must allocate memory for this parameter before invoking this API. On returning success, values populate with MSTP bridge instance details

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.22 int smi_mstp_get_spanningtree_mstdetail_interface (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, u_int16_t *instanceId*, struct smi_mstp_instance_port_details ∗ *details*)

This API retrieves detailed information on MSTP bridge instance to which an interface is attached. smi_mstp_get_spanningtree_mstdetail_interface

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** MSTP instance ID of the range, including <SMI_MSTP_INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>

→ ***details*** Pointer to strucutre . The caller must allocate memory for this parameter before invoking this API. On returning success, values populate with MSTP bridge instance details

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.23 int smi_mstp_set_msti_bridgepriority (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *instanceId*, u_int32_t *priority*)

This API configures the bridge priority for an MSTI bridge. smi_mstp_set_msti_bridgepriority

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***priority*** Bridge priority to be set for an MSTI bridge. Values must be in the range of <SMI_MSTP_BRIDGE_PR_MIN-SMI_MSTP_BRIDGE_PR_MAX>. Value must be in multiples of MSTP_BRIDGE_PRIORITY_MULTIPLIER.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.24 int smi_mstp_set_msti_instance_restrictedrole (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*, bool_t *restrictedRole*)

This API configures restricted role on a port for an MSTP instance. Enabling restricted role for a port prevents it from becoming the root port. By default, restricted role is disabled on an MSTI port. smi_mstp_set_msti_instance_restrictedrole

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>

← ***restrictedRole*** Value to be set for restricted role of an MSTI port. Value can include one of the following:

PAL_TRUE Enable restricted role on a port.

PAL_FALSE Disable restricted role on a port

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.25 int smi_mstp_set_msti_instance_restrictedtcn (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*, bool_t *restrictedTCN*)

This API configures the instance to ignore any TCN (Topology Change Notification) received on a port for an MSTP instance. smi_mstp_set_msti_instance_restrictedtcn

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>

← ***restrictedTCN*** Value indicating whether TCN is restricted. Allowed values include one of the following:

PAL_TRUE Remove restricted TCN on a port for an MSTP instance.

PAL_FALSE Do not remove restricted TCN on a port for an MSTP instance.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.26 int smi_mstp_set_msti_portpathcost (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*, u_int32_t *pathCost*)

This API configures the path cost for an MSTI port on a specific bridge. smi_mstp_set_msti_portpathcost

Parameters:

- ← ***azg*** Pointer to smiclient_globals structure
- ← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ
- ← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.
- ← ***instanceId*** MSTP instance ID in the range of <SMI_MSTP_INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.
- ← ***cost*** Path cost value to be set for an MSTI port. Value must be in the range of <SMI_MSTP_PORTPATHCOST_MIN and="" smi_mstp_portpathcost_max>="">.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.27 int smi_mstp_set_msti_portpriority (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*, s_int16_t *portPriority*)

This API configures the port priority for a bridge. smi_mstp_set_msti_portpriority

Parameters:

- ← ***azg*** Pointer to smiclient_globals structure
- ← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ
- ← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.
- ← ***instanceId*** Instance ID of MSTP in the range of, including <SMI_MSTP_INSTANCE_ID_MIN - SMI_MSTP_INSTANCE_ID_MAX>.
- ← ***portPriority*** Priority value in the range of, including <SMI_MSTP_PORT_PRIORITY_MIN - SMI_MSTP_PORT_PRIORITY_MAX>.

Returns:

0 in case of success, otherwise one of the following errors
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
MSTP_ERR_PRIORITY_VALUE_WRONG
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_PORT_NOT_FOUND

4.1.2.28 s_int32_t smi_mstp_show_inc_blk_ports (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, struct list ∗ *readmsg*, int *flag*, u_int32_t(∗)(struct smi_mstp_port_name_msg ∗getmsg) *callback*)

This API retrieves the inconsistent and blocked ports. smi_mstp_show_inc_blk_ports

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

→ ***readmsg*** List where the information stored.

← ***flag*** 1 - blocked 0 - inconsistent

→ ***getmsg*** For user, to call a callback function upon reception of the information from the server

Returns:

SMI_SUCCESS in case of success, otherwise one of the following errors SMI_-ERROR

4.1.2.29 s_int32_t smi_mstp_show_instance_all (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, struct list ∗ *readmsg*, u_int32_t *in_start*, u_int32_t *in_end*, u_int32_t(∗)(struct smi_mstp_instance_all_msg ∗getmsg) *callback*)

This API retrieves the all bridge instance info in one IPC. smi_mstp_show_instance_all

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

→ ***readmsg*** List where the information stored

← ***in_start*** Instance start

← ***in_end*** Instance end

→ ***getmsg*** For user, to call a callback function upon reception of the information from the server

Returns:

SMI_SUCCESS in case of success, otherwise one of the following errors SMI_-ERROR

4.1.2.30 s_int32_t smi_mstp_show_port_all (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, struct list ∗ *readmsg*, u_int32_t(∗)(struct smi_mstp_spanning_tree_details ∗treeDet) *callback*)

This API retrieves the all bridge port info in one IPC. smi_mstp_show_port_all

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

→ ***readmsg*** List where the information stored.

→ ***treeDet*** For user, to call a callback function upon reception of the information from the server

Returns:

SMI_SUCCESS in case of success, otherwise one of the following errors SMI_ERROR

4.1.2.31 int smi_mstp_stats_clear (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*)

This API unset a BPDU-filter configuration for a bridge port. smi_mstp_stats_clear

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.32 int smi_mstp_stats_clear_by_ifname (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*)

This API unset a BPDU-filter configuration for a bridge port. smi_mstp_stats_clear_by_ifname

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_IF_NOT_FOUND
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.33 int smi_mstp_stats_clear_by_ifname_instance (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *instanceId*)

This API unset a BPDU-filter configuration for a bridge port. smi_mstp_stats_clear_by_ifname_instance

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***instanceId*** MSTP Instance ID in the range of <1-63> or 4092 for sbpm .

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_IF_NOT_FOUND
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.34 int smi_mstp_stats_clear_by_ifname_vlan (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, u_int16_t *vlanId*)

This API unset a BPDU-filter configuration for a bridge port. smi_mstp_stats_clear_by_ifname_vlan

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***ifName*** Interface name string of maximum length INTERFACE_NAMSIZ.

← ***vlanId*** vlan id in the range of <1-4094>.

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_IF_NOT_FOUND
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.35 int smi_mstp_stats_clear_by_instance (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *instanceId*)

This API unset a BPDU-filter configuration for a bridge port. smi_mstp_stats_clear_by_instance

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***instanceId*** MSTP Instance ID in the range of <1-63> or 4092 for sbpm .

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_BRIDGE_NOT_FOUND

4.1.2.36 int smi_mstp_stats_clear_by_vlan (struct smiclient_globals * *azg*, char * *bridgeId*, u_int16_t *vlanId*)

This API unset a BPDU-filter configuration for a bridge port. smi_mstp_stats_clear_by_vlan

Parameters:

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

← ***bridgeId*** Bridge name string of maximum length SMI_BRIDGE_NAMSIZ

← ***vlanId*** vlan id in the range of <1-4094>.

Returns:

0 in case of success, otherwise one of the following errors
MSTP_ERR_BRIDGE_NOT_FOUND

4.2 smi_mstp_msg.h File Reference

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

Data Structures

- struct smi_mstp_bmp
- struct smi_bridge_id
- struct smi_mstp_mst_config_all_msg
- struct smi_mstp_br_port
- struct smi_mstp_stats_inst_port
- struct smi_mstp_br_inst
- struct smi_mstp_br_port_all_msg
- struct smi_mstp_br_inst_all_msg
- struct smi_mstp_stats_all
- struct smi_mac_addr
- struct smi_mstp_config_details
- struct smi_mstp_mst_config
- struct smi_port_name
- struct smi_mstp_mst_br_all_msg
- struct smi_mstp_port_name_msg
- struct smi_mstp_interface_all_msg
- struct smi_mstp_instance_detail_msg
- struct smi_mstp_instance_all_msg
- struct smi_mstp_bridge_detail
- struct smi_mstp_spanning_tree_details
- struct smi_mstp_port_details
- struct smi_mstp_stats
- struct smi_mstp_instance_details
- struct smi_mstp_instance_port_details
- struct smi_msg_mstp

Defines

- #define **SMI_MSG_MSTP_SIZE** 4
- #define **MSTP_CONFIG_NAME_LEN** 32
- #define **SMI_ETHER_DELIMETER_LEN** 6
- #define **SMI_MSTP_PORT_STATE_LEN** 32
- #define **SMI_MSTP_INSTANCE_ID_MIN** 1
- #define **SMI_MSTP_INSTANCE_ID_MAX** 63
- #define **SMI_MSTP_AGEINGTIME_MIN** 0
- #define **SMI_MSTP_AGEINGTIME_MAX** 1000000
- #define **SMI_MSTP_AGEINGTIME_DEFAULT** 300
- #define **SMI_MSTP_BRIDGE_PR_MIN** 0
- #define **SMI_MSTP_BRIDGE_PR_MAX** 61440

- #define **SMI_MSTP_BRIDGE_PR_DEFAULT** 32768
- #define **SMI_MSTP_FORWARD_DELAY_MIN** 4
- #define **SMI_MSTP_FORWARD_DELAY_MAX** 30
- #define **SMI_MSTP_FORWARD_DELAY_DEFAULT** 15
- #define **SMI_MSTP_HELLOTIME_MIN** 1
- #define **SMI_MSTP_HELLOTIME_MAX** 10
- #define **SMI_MSTP_HELLOTIME_DEFAULT** 2
- #define **SMI_MSTP_MAXAGE_MIN** 6
- #define **SMI_MSTP_MAXAGE_MAX** 40
- #define **SMI_MSTP_MAXAGE_DEFAULT** 20
- #define **SMI_MSTP_PORTPATHCOST_MIN** 1
- #define **SMI_MSTP_PORTPATHCOST_MAX** 200000000
- #define **SMI_MSTP_PORT_SHARED** 0
- #define **SMI_MSTP_PORT_P2P** 1
- #define **SMI_MSTP_ADMIN_LINK_TYPE_AUTO** 2
- #define **SMI_MSTP_PORT_PRIORITY_MIN** 0
- #define **SMI_MSTP_PORT_PRIORITY_MAX** 240
- #define **SMI_MSTP_MAXHOPS_MIN** 1
- #define **SMI_MSTP_MAXHOPS_MAX** 40
- #define **SMI_MSTP_MAXHOPS_DEFAULT** 20
- #define **SMI_MSTP_PORTROOTGUARD_DISABLE** 0
- #define **SMI_MSTP_PORTROOTGUARD_ENABLE** 1
- #define **SMI_MSTP_TRANSMIT_HC_MIN** 1
- #define **SMI_MSTP_TRANSMIT_HC_MAX** 10
- #define **SMI_MSTP_TRANSMIT_HC_DEFAULT** 3
- #define **SMI_MSTP_ERRDISABLE_TIMEOUT_MIN** 10
- #define **SMI_MSTP_ERRDISABLE_TIMEOUT_MAX** 1000000
- #define **SMI_MSTP_REVISION_NUM_MIN** 0
- #define **SMI_MSTP_REVISION_NUM_MAX** 65535
- #define **SMI_MSTP_PORTBPDUGUARD_DISABLE** 0
- #define **SMI_MSTP_PORTBPDUGUARD_ENABLE** 1
- #define **SMI_MSTP_PORTBPDUGUARD_DEFAULT** 2
- #define **SMI_MSTP_BPDUDISABLE** 0
- #define **SMI_MSTP_BPDUENABLE** 1
- #define **SMI_MSTP_BPDUDEFAULT** 2
- #define **SMI_MSTP_PORT_PORTFAST_BPDUGUARD_ENABLED** 0
- #define **SMI_MSTP_PORT_PORTFAST_BPDUGUARD_DISABLED** 1
- #define **SMI_MSTP_PORT_PORTFAST_BPDUGUARD_DEFAULT** 2
- #define **SMI_MSTP_PORT_PORTFAST_BPDUFILTER_ENABLED** 0
- #define **SMI_MSTP_PORT_PORTFAST_BPDUFILTER_DISABLED** 1
- #define **SMI_MSTP_PORT_PORTFAST_BPDUFILTER_DEFAULT** 2
- #define **SMI_MSTP_CONFIG_DIGEST_LEN** 16
- #define **SMI_DONT_CHK_TYPE** 99
- #define **SMI_VLAN_ID_START** 1
- #define **SMI_VLAN_ID_END** 4094
- #define **SMI_MSTP_PATHCOST_SHORT** 0

- #define **SMI_MSTP_PATHCOST_LONG** 1
- #define **SMI_MSTP_PATHCOST_DEFAULT** 2
- #define **DEBUG_MSTP_TIMER** 0x01
- #define **DEBUG_MSTP_TIMER_DETAIL** 0x02
- #define **DEBUG_MSTP_PROTOCOL** 0x04
- #define **DEBUG_MSTP_PROTOCOL_DETAIL** 0x08
- #define **DEBUG_MSTP_CLI** 0x10
- #define **DEBUG_MSTP_TX** 0x20
- #define **DEBUG_MSTP_RX** 0x40
- #define **DEBUG_MSTP_ALL** 0
- #define **SMI_NSM_BRIDGE_TYPE_STP** 1
- #define **SMI_NSM_BRIDGE_TYPE_RSTP** 3
- #define **SMI_NSM_BRIDGE_TYPE_MSTP** 5
- #define **SMI_NSM_BRIDGE_TYPE_PROVIDER_RSTP** 6
- #define **SMI_NSM_BRIDGE_TYPE_CE** 8
- #define **SMI_NSM_BRIDGE_TYPE_RPVST_PLUS** 12
- #define **SMI_IS_BRIDGE_MSTP**(B)
- #define **SMI_IS_BRIDGE_RSTP**(B)
- #define **SMI_MST_INSTANCE_IST** 0
- #define **SMI_MST_DEFAULT_VLAN** 1
- #define **SMI_L2_TIMER_SCALE_FACT** 256
- #define **SMI_MSTP_ADMIN_LINK_TYPE_AUTO** 2
- #define **SMI_DEBUG_MSTP_WMI_ALL** 0
- #define **SMI_DEBUG_MSTP_WMI_EVENT** 1
- #define **SMI_DEBUG_MSTP_WMI_RECV** 2
- #define **SMI_DEBUG_MSTP_WMI_SEND** 3
- #define **SMI_DEBUG_MSTP_WMI_DETAIL** 4
- #define **SMI_DEBUG_MSTP_WMI_MSG** 5
- #define **SMI_MSTP_MIN_BRIDGE_PRIORITY** 0
- #define **SMI_MSTP_MAX_BRIDGE_PRIORITY** 61440
- #define **SMI_MIN_WMI_DEBUG** 0
- #define **SMI_MAX_WMI_DEBUG** 5
- #define **SMI_MSTP_CTYPE_IFNAME** 0
- #define **SMI_MSTP_CTYPE_NAME** 1
- #define **SMI_MSTP_CTYPE_VID** 2
- #define **SMI_MSTP_CTYPE_AGE** 3
- #define **SMI_MSTP_CTYPE_INSTANCE** 4
- #define **SMI_MSTP_CTYPE_HELLOTIME** 5
- #define **SMI_MSTP_CTYPE_MAXAGE** 6
- #define **SMI_MSTP_CTYPE_PORTEDGE** 7
- #define **SMI_MSTP_CTYPE_VERSION** 8
- #define **SMI_MSTP_CTYPE_PRIORITY** 9
- #define **SMI_MSTP_CTYPE_MSTIPRIORITY** 10
- #define **SMI_MSTP_CTYPE_FWDDELAY** 11
- #define **SMI_MSTP_CTYPE_COST** 12
- #define **SMI_MSTP_CTYPE_RST** 13

- #define **SMI_MSTP_CTYPE_RSTCN** 14
- #define **SMI_MSTP_CTYPE_PORTP2P** 15
- #define **SMI_MSTP_CTYPE_PPATHCOST** 16
- #define **SMI_MSTP_CTYPE_PORTHELLO** 17
- #define **SMI_MSTP_CTYPE_PORTPRIORITY** 18
- #define **SMI_MSTP_CTYPE_PORTRSTROLE** 19
- #define **SMI_MSTP_CTYPE_PORTRSTTCN** 20
- #define **SMI_MSTP_CTYPE_PORTROOTG** 21
- #define **SMI_MSTP_CTYPE_PORTBPDUF** 22
- #define **SMI_MSTP_CTYPE_BRIDGEFORWARD** 23
- #define **SMI_MSTP_CTYPE_BPDUGUARD** 24
- #define **SMI_MSTP_CTYPE_TXHOLDCOUNT** 25
- #define **SMI_MSTP_CTYPE_BRIDGEBPDUG** 26
- #define **SMI_MSTP_CTYPE_TIMEOUTEN** 27
- #define **SMI_MSTP_CTYPE_TIMEOUTINT** 28
- #define **SMI_MSTP_CTYPE_BRPRIORITY** 29
- #define **SMI_MSTP_CTYPE_REVISION** 30
- #define **SMI_MSTP_CTYPE_EXTENDED** 31
- #define **SMI_MSTP_CTYPE_MAXHOPS** 0
- #define **SMI_MSTP_CTYPE_AUTOEDGE** 1
- #define **SMI_MSTP_CTYPE_REGIONNAME** 2
- #define **SMI_MSTP_CTYPE_SPANNINGTREE** 3
- #define **SMI_MSTP_CTYPE_PORT_DETAILS** 4
- #define **SMI_MSTP_CTYPE_INSTANCE_DETAILS** 5
- #define **SMI_MSTP_CTYPE_INSTANCEPORT_DETAILS** 6
- #define **SMI_MSTP_CTYPE_CONFIG_DETAILS** 7
- #define **SMI_MSTP_CTYPE_MACADDR** 8
- #define **SMI_MSTP_CTYPE_BRIDGEBPDUFILTER** 9
- #define **SMI_MSTP_CTYPE_STP_ENABLED** 10
- #define **SMI_MSTP_CTYPE_VLAN_BMP** 11
- #define **SMI_MSTP_SWITCH** 12
- #define **SMI_MSTP_CTYPE_PORT_ALL** 13
- #define **SMI_MSTP_CTYPE_DEBUG_ON** 14
- #define **SMI_MSTP_CTYPE_DEBUG_OFF** 15
- #define **SMI_MSTP_CTYPE_INSTANCE_ALL** 16
- #define **SMI_MSTP_CTYPE_INSTANCE_START** 17
- #define **SMI_MSTP_CTYPE_BL_INC** 18
- #define **SMI_MSTP_CTYPE_BLK_INC_FLAG** 19
- #define **SMI_MSTP_CTYPE_PT_METHOD** 20
- #define **SMI_MSTP_CTYPE_INSTANCE_END** 21
- #define **SMI_MSTP_CTYPE_PATH_COST_METHOD** 22
- #define **SMI_MSTP_CTYPE_ENABLED** 23
- #define **SMI_MSTP_CTYPE_MST** 24
- #define **SMI_MSTP_CTYPE_MST_CONFIG** 26
- #define **SMI_MSTP_CTYPE_MST_INST_IF** 27
- #define **SMI_MSTP_CTYPE_STAT_BR_INST** 28

- #define **SMI_MSTP_CTYPE_STAT_BR_PORT** 29
- #define **SMI_MSTP_CTYPE_STAT_BR_INFO** 30
- #define **SMI_MSTP_CTYPE_EXTENDED_2** 31
- #define **SMI_MSTP_CTYPE_L2GP_STATUS** 0
- #define **SMI_MSTP_CTYPE_CISCO_INTR_STATUS** 1
- #define **SMI_MSTP_CTYPE_IF_ROLE** 2
- #define **SMI_MSTP_CTYPE_IF_STATE** 4

Enumerations

- enum **smi_stp_version** { **SMI_VERSION_STP**, **SMI_VERSION_NOT_SUPPORTED**, **SMI_VERSION_RSTP**, **SMI_VERSION_MSTP** }
- enum **smi_port_role** {

  SMI_ROLE_MASTERPORT, **SMI_ROLE_ALTERNATE**, **SMI_ROLE_ROOTPORT**, **SMI_ROLE_DESIGNATED**,

  SMI_ROLE_DISABLED, **SMI_ROLE_BACKUP** }
- enum **smi_port_state** {

  SMI_STATE_DISCARDING, **SMI_STATE_LISTENING**, **SMI_STATE_LEARNING**, **SMI_STATE_FORWARDING**,

  SMI_STATE_BLOCKING, **SMI_STATE_DISABLED**, **SMI_STATE_ERROR** }
- enum **smi_stp_port_state** {

  SMI_STP_PORT_STATE_0, **SMI_STP_PORT_STATE_1**, **SMI_STP_PORT_STATE_2**, **SMI_STP_PORT_STATE_3**,

  SMI_STP_PORT_STATE_4 }
- enum **smi_port_forward_state** { **SMI_BRIDGE_PORT_NONFORWARD_STATE**, **SMI_BRIDGE_PORT_FORWARD_STATE** }

Functions

- void **smi_mstp_dump** (struct lib_globals ∗zg, struct smi_msg_mstp ∗msg)
- int **smi_encode_mstpmsg** (u_char ∗∗pnt, u_int16_t ∗size, struct smi_msg_mstp ∗msg)
- int **smi_decode_mstpmsg** (u_char ∗∗pnt, u_int16_t ∗size, struct smi_msg_mstp ∗msg)
- int **smi_parse_mstp** (u_char ∗∗, u_int16_t ∗, struct smi_msg_header ∗, void ∗, SMI_CALLBACK)

4.2.1 Detailed Description

Defines data structures used by xSTP SMI APIs.

4.2.2 Define Documentation

4.2.2.1 #define SMI_IS_BRIDGE_MSTP(B)

Value:

```
((B) && (((B)->type == NSM_BRIDGE_TYPE_MSTP)                          \
         || ((B)->type == NSM_BRIDGE_TYPE_PROVIDER_MSTP)                       \
         || ((B)->type == NSM_BRIDGE_TYPE_BACKBONE_MSTP)))
```

4.2.2.2 #define SMI_IS_BRIDGE_RSTP(B)

Value:

```
((B) && (((B)->type == NSM_BRIDGE_TYPE_RSTP)                          \
         || ((B)->type == NSM_BRIDGE_TYPE_PROVIDER_RSTP)                       \
         || ((B)->type == NSM_BRIDGE_TYPE_CE)                                  \
         || ((B)->type == NSM_BRIDGE_TYPE_RSTP_VLANAWARE)))
```

4.3 smi_rpvst_plus.h File Reference

Provides APIs for managing Bidirectional Forwarding Detection(BFD) in ZebOS.

```
#include "smi_client.h"
#include "smi_rpvst_plus_msg.h"
```

Functions

- int smi_rpvst_plus_api_add_vlan (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId)

  This function sets a VLAN to be associated with a spanning tree instance.

- int smi_rpvst_plus_api_vlan_delete (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId)

  This function unsets a VLAN that was associated with a spanning tree instance.

- int smi_rpvst_plus_api_set_msti_bridge_priority (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId, u_int32_t priority)

  This function sets the priority on a spanning tree instance of a particular VLAN.

- int smi_rpvst_plus_api_set_msti_port_priority (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, s_int16_t portPriority)

  This function sets the priority level of a port on a VLAN.

- int smi_rpvst_plus_api_set_msti_port_path_cost (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, u_int32_t pathCost)

  This function sets the path cost recalculation to automatic mode for an instance of MSTI related to RPVST.

- int smi_rpvst_plus_api_add_port (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, u_int16_t vlanId, u_int8_t spanningTreeDisable)

  This function sets a port as a spanning-tree instance of a VLAN.

- int smi_rpvst_plus_delete_port (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, u_int16_t vlanId, int force, bool_t notifyForward)

  This function removes a port from a VLAN of a spanning tree.

- int smi_rpvst_plus_api_set_msti_vlan_restricted_role (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, bool_t restrictedRole)

  This function sets the priority level of a port on a VLAN.

- int smi_rpvst_plus_api_set_msti_vlan_restricted_tcn (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, bool_t restrictedTCN)

  This function sets restricted TCN on a VLAN.

- int **smi_rpvst_plus_api_unset_msti_bridge_priority** (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId)
- int **smi_rpvst_plus_api_unset_msti_port_priority** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId)
- int **smi_rpvst_plus_api_add_vlan_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId)
- int **smi_rpvst_plus_api_vlan_delete_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId)
- int **smi_rpvst_plus_api_set_msti_bridge_priority_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId, u_int32_t priority)
- int **smi_rpvst_plus_api_unset_msti_bridge_priority_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, int vlanId)
- int **smi_rpvst_plus_api_set_msti_port_priority_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, s_int16_t portPriority)
- int **smi_rpvst_plus_api_unset_msti_port_priority_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId)
- int **smi_rpvst_plus_api_set_msti_port_path_cost_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, u_int32_t pathCost)
- int **smi_rpvst_plus_api_add_port_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, u_int16_t vlanId, u_int8_t spanningTreeDisable)
- int **smi_rpvst_plus_delete_port_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, u_int16_t vlanId, int force, bool_t notifyForward)
- int **smi_rpvst_plus_api_set_msti_vlan_restricted_role_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, bool_t restrictedRole)
- int **smi_rpvst_plus_api_set_msti_vlan_restricted_tcn_validate** (struct smiclient_globals ∗azg, char ∗bridgeId, char ∗ifName, int vlanId, bool_t restrictedTCN)

4.3.1 Detailed Description

Provides APIs for managing Bidirectional Forwarding Detection(BFD) in ZebOS.

4.3.2 Function Documentation

4.3.2.1 int smi_rpvst_plus_api_add_port (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, u_int16_t *vlanId*, u_int8_t *spanningTreeDisable*)

This function sets a port as a spanning-tree instance of a VLAN. smi_rpvst_plus_api_add_port

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***ifName*** Interface Name

← ***vlanId*** Vlan id

← ***spanning_tree_disable*** Disables spanning tree on a port

Returns:

0 on success, otherwise one of the following error codes
MSTP_ERR_RPVST_VLAN_CONFIG_ERR
MSTP_ERR_NOT_RPVST_BRIDGE

4.3.2.2 int smi_rpvst_plus_api_add_vlan (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *vlanId*)

This function sets a VLAN to be associated with a spanning tree instance. smi_rpvst_plus_api_add_vlan

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***vlanId*** Vlan id

Returns:

0 on success, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_NOT_RPVST_BRIDGE
MSTP_ERR_RPVST_BRIDGE_NO_VLAN
MSTP_ERR_RPVST_BRIDGE_MAX_VLAN
MSTP_ERR_RPVST_BRIDGE_VLAN_EXISTS

4.3.2.3 int smi_rpvst_plus_api_set_msti_bridge_priority (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *vlanId*, u_int32_t *priority*)

This function sets the priority on a spanning tree instance of a particular VLAN. smi_rpvst_plus_api_set_msti_bridge_priority

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***vlanId*** Vlan id

← ***priority*** Proirity

Returns:

0 on success, otherwise one of the following error codes

MSTP_ERR_NOT_RPVST_BRIDGE
MSTP_ERR_RPVST_BRIDGE_NO_VLAN
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_NOT_SPB_BRIDGE
MSTP_ERR_INSTANCE_OUTOFBOUNDS
MSTP_ERR_INSTANCE_NOT_FOUND
MSTP_ERR_PRIORITY_VALUE_WRONG
MSTP_ERR_PRIORITY_OUTOFBOUNDS

4.3.2.4 int smi_rpvst_plus_api_set_msti_port_path_cost (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *vlanId*, u_int32_t *pathCost*)

This function sets the path cost recalculation to automatic mode for an instance of MSTI related to RPVST. smi_rpvst_plus_api_set_msti_port_path_cost

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***ifName*** Interface Name

← ***vlanId*** Vlan id

← ***cost*** The cost

Returns:

RESULT_OK on success, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_RPVST_VLAN_CONFIG_ERR
RESULT_ERROR

4.3.2.5 int smi_rpvst_plus_api_set_msti_port_priority (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *vlanId*, s_int16_t *portPriority*)

This function sets the priority level of a port on a VLAN. smi_rpvst_plus_api_set_-msti_port_priority

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***ifName*** Interface Name

← ***vlanId*** Vlan id

← ***portPriority*** Proirity

Returns:

0 on success, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_RPVST_VLAN_CONFIG_ERR
MSTP_ERR_RPVST_VLAN_CONFIG_ERR
MSTP_ERR_PORT_NOT_FOUND
MSTP_ERR_PORT_PRIORITY_VALUE_WRONG
MSTP_ERR_INSTANCE_OUTOFBOUNDS

4.3.2.6 int smi_rpvst_plus_api_set_msti_vlan_restricted_role (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *vlanId*, bool_t *restrictedRole*)

This function sets the priority level of a port on a VLAN. smi_rpvst_plus_api_set_-msti_vlan_restricted_role

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***ifName*** Interface Name

← ***vlanId*** Vlan id

← ***restrictedRole*** Flag to be set

Returns:

0 on success, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_INSTANCE_NOT_FOUND
MSTP_ERR_PORT_NOT_FOUND
MSTP_ERR_RPVST_VLAN_CONFIG_ERR
MSTP_ERR_NOT_RPVST_BRIDGE
MSTP_ERR_PORT_PRIORITY_VALUE_WRONG
MSTP_ERR_INSTANCE_OUTOFBOUNDS

4.3.2.7 int smi_rpvst_plus_api_set_msti_vlan_restricted_tcn (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, int *vlanId*, bool_t *restrictedTCN*)

This function sets restricted TCN on a VLAN. smi_rpvst_plus_api_set_msti_vlan_-restricted_tcn

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***ifName*** Interface Name

← ***vlanId*** Vlan id

← ***restrictedTCN*** Flag to be set

Returns:

0 on success, otherwise one of the following error codes
MSTP_ERR_RPVST_VLAN_CONFIG_ERR
MSTP_ERR_RPVST_VLAN_CONFIG_ERR

4.3.2.8 int smi_rpvst_plus_api_vlan_delete (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, int *vlanId*)

This function unsets a VLAN that was associated with a spanning tree instance. smi_rpvst_plus_api_vlan_delete

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***vlanId*** Vlan id

Returns:

0 on success, otherwise one of the following error codes
MSTP_ERR_BRIDGE_NOT_FOUND
MSTP_ERR_RPVST_BRIDGE_NO_VLAN
MSTP_ERR_INSTANCE_NOT_FOUND

4.3.2.9 int smi_rpvst_plus_delete_port (struct smiclient_globals ∗ *azg*, char ∗ *bridgeId*, char ∗ *ifName*, u_int16_t *vlanId*, int *force*, bool_t *notifyForward*)

This function removes a port from a VLAN of a spanning tree. smi_rpvst_plus_delete_port

Parameters:

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

← ***bridgeId*** Name of the bridge

← ***ifName*** Interface Name

← ***vlanId*** Vlan id

← ***force***

← ***notifyForward***

Returns:

0 on success, otherwise one of the following error codes
MSTP_ERR_RPVST_VLAN_CONFIG_ERR
MSTP_ERR_NOT_RPVST_BRIDGE

Index