ZebOS-XP VLAN SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:31 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 Struct Reference

Data Fields

- char **name** [SMI_BRIDGE_NAMSIZ+1]
- u_int8_t **type**
- u_int8_t **bridge_id**
- u_int8_t **is_default**
- u_int32_t **ageing_time**
- int **learning**
- struct smi_vlan_bmp **port_list**
- struct smi_vlan_bmp **vlanbmp**
- u_int8_t **traffic_class_enabled**
- enum smi_topology **topology_type**

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

- smi_vlan_msg.h

3.2 smi_bridge_vlan_summ Struct Reference

Data Fields

- char **bridge_name** [SMI_BRIDGE_MAX_VALUE][SMI_BRIDGE_NAMSIZ]
- int **bridge_count**
- struct smi_vlan_summ **vlan_summ** [SMI_BRIDGE_MAX_VALUE]

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

- smi_vlan_msg.h

3.3 smi_if_swport_br Struct Reference

Data Fields

- char **ifname** [IFNAMSIZ+1]
- char **port_mode** [20]
- char **in_filter** [20]
- char **acc_frames** [20]
- u_int16_t **default_vlan_id**
- struct smi_vlan_bmp **conf_vlan**

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

- smi_vlan_msg.h

3.4 smi_if_swport_br_list Struct Reference

Data Fields

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

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

- smi_vlan_msg.h

3.5 smi_if_vlan_info Struct Reference

Data Fields

- char **name** [INTERFACE_NAMSIZ+1]
- enum smi_vlan_port_mode **mode**
- enum smi_vlan_port_mode **sub_mode**
- u_int16_t **pvid**
- u_int16_t **native_vid**
- u_char **flags**
- enum smi_vlan_add_opt **config**
- struct smi_vlan_bmp **staticMemberBmp**
- struct smi_vlan_bmp **dynamicMemberBmp**

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

- smi_vlan_msg.h

3.6 smi_msg_vlan Struct Reference

Data Fields

- smi_cindex_t **cindex**
- smi_cindex_t **cindex_1**
- char **br_name** [SMI_BRIDGE_NAMSIZ]
- char **vlan_range** [255]
- char **vlan_name** [SMI_VLAN_NAMSIZ]
- char **if_name** [INTERFACE_NAMSIZ+1]
- enum smi_vlan_state **state**
- enum smi_vlan_type **type**
- enum smi_vlan_port_mode **mode**
- enum smi_vlan_port_mode **submode**
- enum smi_acceptable_frame_type **frame_type**
- enum smi_vlan_port_ingress_filter **ingress_filter**
- enum smi_vlan_egress_type **egress_type**
- struct smi_vlan_bmp **vlan_bmp**
- struct smi_vlan_info **vlan_info**
- struct smi_if_vlan_info **if_vlan_info**
- struct smi_bridge **bridge_info**
- enum smi_bridge_proto **protocol**
- enum smi_bridge_proto_process **process**
- struct smi_vlan_bmp **egressTypeBmp**
- u_int16_t **vid**
- struct smi_port_bmp **port_list**
- u_int16_t **ether_type**
- enum ha_switch **switch_to_state**
- enum smi_vlan_add_opt **vlan_add_opt**
- u_int16_t **lower_vid**
- u_int16_t **higher_vid**
- struct smi_vlan_info_list **vlan_info_list**
- struct smi_vlan_summ **vlan_summ**
- u_int16_t **eps_id**
- u_int32_t **vr_id**
- u_char **num_traffic_classes**
- u_char **traffic_class_value**
- u_char **regen_user_priority**
- u_char **user_priority**
- struct smi_traffic_class_table **traffic_class_table**
- struct smi_user_regen_prio **user_regen_prio**
- struct smi_if_swport_br_list **if_sw_list**
- struct smi_bridge_vlan_summ **bridge_vlan_summ**

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

- smi_vlan_msg.h

3.7 smi_traffic_class_table Struct Reference

Data Fields

- u_char **traffic_class_table** [SMI_HAL_BRIDGE_MAX_USER_-PRIO+1][SMI_HAL_BRIDGE_MAX_TRAFFIC_CLASS]

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

- smi_vlan_msg.h

3.8 smi_user_regen_prio Struct Reference

Data Fields

- unsigned int **user_priority** [SMI_HAL_BRIDGE_MAX_USER_PRIO]
- int **regen_prio** [SMI_HAL_BRIDGE_MAX_USER_PRIO]

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

- smi_vlan_msg.h

3.9 smi_vlan_info Struct Reference

Data Fields

- char **vlan_name** [SMI_VLAN_NAMSIZ+1]
- char **bridge_name** [SMI_BRIDGE_NAMSIZ+1]
- u_int16_t **vid**
- enum smi_vlan_type **type**
- enum smi_vlan_state **vlan_state**
- u_int32_t **mtu_val**
- struct smi_vlan_bmp **port_list**
- int **instance**

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

- smi_vlan_msg.h

3.10 smi_vlan_info_list Struct Reference

Data Fields

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

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

- smi_vlan_msg.h

3.11 smi_vlan_summ Struct Reference

Data Fields

- int **vlan_configured**
- int **vlan_active**
- int **vlan_suspend**

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

- smi_vlan_msg.h

Chapter 4

File Documentation

4.1 smi_vlan.h File Reference

Provides APIs for VLAN management. `#include "smi_client.h"`
`#include "smi_vlan_msg.h"`

Defines

- #define **SMI_NSM_VLAN_EPS_ID_MIN** 1
- #define **SMI_NSM_VLAN_EPS_ID_MAX** 4094
- #define **SMI_VR_ID_MIN** 0
- #define **SMI_VR_ID_MAX** 252
- #define **SMI_VLAN_ID_MIN** 2
- #define **SMI_VLAN_ID_MAX** 4094
- #define **SMI_BRIDGE_GROUP_MIN** 1
- #define **SMI_BRIDGE_GROUP_MAX** 32
- #define **SMI_VLAN_USER_PRIORITY_MIN** 0
- #define **SMI_VLAN_USER_PRIORITY_MAX** 7
- #define **VLAN_NUM_TRAFFIC_CLASS_VALUE_MAX** 8
- #define **VLAN_NUM_TRAFFIC_CLASS_VALUE_MIN** 1
- #define **VLAN_TRAFFIC_CLASS_VALUE_MAX** 7
- #define **VLAN_TRAFFIC_CLASS_VALUE_MIN** 0
- #define **VLAN_STATE_ENABLE** 1
- #define **VLAN_STATE_DISABLE** 0
- #define **DEFAULT_VLAN_NAME_LEN** 8
- #define **MIN_VLAN_NAME_LEN** 4
- #define **NSM_VLAN_STATIC** (1 << 0)
- #define **NSM_VLAN_CVLAN** (1 << 2)

Functions

- int smi_vlan_add (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, char ∗vlanName, u_int16_t vlanId, enum smi_vlan_state vlanState, enum smi_vlan_-type vlanType)

  Adds a vlan to specified bridge.

- int **smi_vlan_clear_allowed_vlanId_to_port_wrap** (struct smiclient_globals ∗azg, int vrId, char ∗ifname, u_int16_t vid, enum smi_vlan_port_mode mode, u_int32_t allowedFlag)
- int **smi_vlan_clear_allowed_vlanId_to_port_wrap_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifname, u_int16_t vid, enum smi_-vlan_port_mode mode, u_int32_t allowedFlag)
- int smi_vlan_delete (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_-int16_t vlanId, enum smi_vlan_type vlanType)

  Remove a vlan from specified bridge.

- int smi_vlan_range_add (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t lowerVlan, u_int16_t higherVlan, enum smi_vlan_state vlanState, enum smi_vlan_type vlanType)

  Adds a range of vlan to the specified bridge.

- int **smi_vlan_add_multiple_vlans_validate** (struct smiclient_globals ∗azg, int vrId, char ∗br_name, char ∗vlan_range, enum smi_vlan_state state, enum smi_-vlan_type type)
- int **smi_vlan_update_multiple_vlans_state** (struct smiclient_globals ∗azg, int vrId, char ∗br_name, char ∗vlan_range, enum smi_vlan_state state, enum smi_-vlan_type type)
- int **smi_vlan_update_multiple_vlans_state_validate** (struct smiclient_globals ∗azg, int vrId, char ∗br_name, char ∗vlan_range, enum smi_vlan_state state, enum smi_vlan_type type)
- int **smi_vlan_add_multiple_vlans** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, char ∗vlanRange, enum smi_vlan_state vlanState, enum smi_vlan_-type vlanType)
- int **smi_vlan_delete_multiple_vlans_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, char ∗vlan_range, enum smi_vlan_type type)
- int **smi_vlan_delete_multiple_vlans** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, char ∗vlan_range, enum smi_vlan_type type)
- int smi_vlan_range_del (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t lowerVlan, u_int16_t higherVlan, enum smi_vlan_type vlanType)

  Remove a range of vlan from the specified bridge.

- int smi_vlan_api_set_port_mode (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode, enum smi_vlan_port_-mode vlanPortSubMode)

  This API sets the mode and sub mode for a port on a VLAN. A user will set the modes on a port to know what type of traffic it carries; for example, if the traffic is customer network, provider network, or etc. The use should make sure that the corresponding VLAN is already configured.

- int smi_vlan_api_get_port_mode (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode ∗vlanPortMode, enum smi_vlan_port_-mode ∗vlanPortSubMode)

  This API retrieves the mode and submode that were configured on a VLAN interface.

- int smi_vlan_set_acceptable_frame_type (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode, enum smi_-acceptable_frame_type frameType)

  This API sets the acceptable frame type for the VLAN port by providing the functionality to configure an acceptable frame type for a VLAN interface and mode.

- int smi_vlan_get_acceptable_frame_type (struct smiclient_globals ∗azg, int vrId, char ∗ifName, int ∗acceptableFrameType)

  This API provides the functionality to retrieve the type of acceptable frames that were configured on a VLAN port, such as a VLAN untagged frame, VLAN tagged frame or all.

- int smi_vlan_set_ingress_filter (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_ingress_filter vlanPortIngressFilter)

  This API sets the ingress filtering on a VLAN port. It provides the functionality for enabling/disabling the filtering for an incoming frame on a particular VLAN port. This API will look for what is the acceptable particular frame type defined for a particular mode and enable the filtering for the same, so that the rest of the frames are dropped. If the API is invoked with disable flag, then the filtering of the ingress frames will be stopped.

- int smi_vlan_get_ingress_filter (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode ∗vlanPortMode, enum smi_-vlan_port_mode ∗vlanPortSubMode, enum smi_vlan_port_ingress_filter ∗vlanPortIngressFilter)

  This API gets the ingress filtering status of a VLAN port by providing the functionality to retrieve filtering status on ingress side, such as enabled or disabled. It also gets the mode and the submode values along with the status of ingress filtering of a port.

- int smi_vlan_set_default_vid (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)

  API provides the functionality to configure a default VLAN identifier on an interface port.

- int smi_vlan_get_default_vid (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t ∗vlanId)

  API provides the functionality to configure a default VLAN identifier on an interface port.

- int smi_vlan_add_vlan_to_port (struct smiclient_globals ∗azg, int vrId, char ∗ifName, struct smi_vlan_bmp ∗vlanBmp, struct smi_vlan_bmp ∗egressTypeBmp, struct smi_vlan_bmp ∗successBmp)

  This API adds the VLANs to the given interface port.

- int smi_vlan_delete_vlan_from_port (struct smiclient_globals ∗azg, int vrId, char ∗ifName, struct smi_vlan_bmp ∗vlanBmp, struct smi_vlan_bmp ∗successBmp)

  This API deletes the VLANs that were added to a given interface name.

- int smi_vlan_clear_port (struct smiclient_globals ∗azg, int vrId, char ∗ifName)

  This API clears the VLAN configurations from an interface port, except VLAN 1. For a hybrid/access port, the default VID resets to VLAN 1.

- int smi_vlan_add_all_except_vid (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode, enum smi_vlan_port_mode vlanPortSubMode, struct smi_vlan_bmp ∗excludeBmp, enum smi_vlan_egress_type egressType, enum smi_vlan_add_opt vlanAddOpt)

  This API provides the functionality to add all VLANs (except a specified VLAN) to a trunk, hybrid or provider network port. The different type of VLAN add options include one of the following:-
  SMI_VLAN_CONFIGURED_ALL - To configure all the VLANs.
  SMI_VLAN_CONFIGURED_NONE - To unconfigure all the VLANs except specified VLANs.
  SMI_VLAN_CONFIGURED_SPECIFIC - To configure all the VLANs except specified VLANs.

- int smi_get_all_vlan_config (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, struct smi_vlan_bmp ∗vlanBmp)

  This API gets all VLAN IDs configured on a bridge. The bridge is identified by bridge name.

- int smi_get_vlan_by_id (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t vlanId, struct smi_vlan_info ∗vlanInfo)

  This API gets the VLAN information configured on a given interface.

- int smi_vlan_if_get (struct smiclient_globals ∗azg, int vrId, char ∗ifName, struct smi_if_vlan_info ∗vlanInfo)

  This API gets the VLAN information configured on a given interface.

- int smi_get_bridge (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, struct smi_bridge ∗bridgeInfo)

  This API provides the functionality to retrieve any bridge information configured on a given bridge name.

- int smi_get_vlan_summary (struct smiclient_globals ∗azg, int vrId, struct smi_bridge_vlan_summ ∗vlanSumm)

  Use this function to get all the interface's brief information.

- int smi_show_vlan (struct smiclient_globals ∗azg, int vrId, int startIndex, int endIndex, struct list ∗vlanInfo, int(∗callback)(struct list ∗vlanInfo))

  Use this function to get all the interface's brief information.

- s_int32_t smi_vlan_clear_trunk_port (struct smiclient_globals ∗azg, int vrId, char ∗ifName)

  This API provides the functionality to remove trunk port.

- s_int32_t smi_vlan_clear_hybrid_port (struct smiclient_globals ∗azg, int vrId, char ∗ifName)

  This API provides the functionality to remove trunk port.

- int smi_vlan_unset_hybrid_port_vlan (struct smiclient_globals ∗azg, int vrId, char ∗ifName)

  This API provides the functionality to unset hybrid port vlan.

- int smi_vlan_unset_access_port_vlan (struct smiclient_globals ∗azg, int vrId, char ∗ifName)

  This API provides the functionality to unset acess port vlan.

- int smi_vlan_unset_access_hybrid_port_vlan (struct smiclient_globals ∗azg, int vrId, char ∗ifname, int vlanId, int vlanPortMode, int modeFlag)

  This API provides the functionality to unset acess/hybrid port vlan.

- int **smi_vlan_unset_access_hybrid_port_vlan_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifname, int vlanId, int vlanPortMode, int modeFlag)

- int smi_trunk_allowed_vlan_all (struct smiclient_globals ∗azg, int vrId, char ∗ifName)

  This API provides the functionality to allowed all vlan to trunk port.

- int smi_trunk_allowed_vlan_none (struct smiclient_globals ∗azg, int vrId, char ∗ifName)

  This API provides the functionality to remove all vlan from trunk port.

- int smi_trunk_allowed_vlan (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)

  This API provides the functionality to allowed vlan in trunk port.

- int smi_trunk_unset_native_vlan (struct smiclient_globals ∗azg, int vrId, char ∗ifName)

  This API provides the functionality to unset the trunk native vlan .

- int smi_trunk_set_native_vlan (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)

  This API provides the functionality to set the trunk native vlan .

- int smi_vlan_api_set_portmode (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode)

  This API provides the functionality to set port mode .

- int smi_vlan_api_set_switchport_mode (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode)

  This API provides the functionality to set the switchport port mode .

- int smi_vlan_set_access_port_vlan (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)

  This API provides the functionality to allowed access port to vlan.

- int smi_vlan_set_hybrid_port_vlan (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)

  This API provides the functionality to allowed hybrid port to vlan.

- int smi_nsm_map_vlans_to_g8031_protection_group (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t epsId)

  This API provides the functionality to allowed access port to vlan.

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

  This API provides the functionality to allowed access port to vlan.

- int smi_nsm_vlan_br_name_word (struct smiclient_globals ∗azg, int vrId, u_-int16_t vlanId, char ∗bridgeId, enum smi_vlan_type vlanType, int vlanState, char ∗vlanName)

  Display VLAN prio regen.

- int smi_nsm_vlan_set_mtu (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t vlanId, enum smi_vlan_type vlanType, u_int32_t mtu-Val)

  set mtu in vlan

- int smi_get_vlan_by_name (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, char ∗vlanName, struct smi_vlan_info ∗vlanInfo)

  get vlan

- int smi_nsm_vlan_enable_disable (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t vlanId, int vlanState)

  set vlan

- int smi_nsm_vlan_port_set_regen_user_priority (struct smiclient_globals ∗azg, int vrId, char ∗if_name, u_char userPriority, u_char regenUserPriority)

  set regen user priority

- int smi_nsm_vlan_port_set_default_user_priority (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_char userPriority)

  set default priority

- int smi_nsm_vlan_port_set_traffic_class_table (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_char userPriority, u_char trafficClass, u_char trafficClassValue)

  Creates a set vlan traffic class table.

- int **smi_nsm_vlan_add_hybrid_port_none_sdkapi** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- int smi_nsm_vlan_add_hybrid_port_all_sdkapi (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- s_int32_t smi_show_api_traffic_class_table (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, struct smi_traffic_class_table ∗trafficClass)

  Show the information of the configured VLAN traffic class table.

- s_int32_t smi_show_api_default_priority (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, u_char ∗userPriority)

  Show the information of the configured user-priority.

- s_int32_t smi_show_api_user_prio_regen_table (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, struct smi_user_regen_prio ∗userRegenPrio)

  Show the information of the configured user-priority regen table.

- s_int32_t smi_show_api_interfaces_switchport_bridge (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗bridgeId, struct list ∗ifList)

  Show the information of the configured user-priority regen table.

- int **smi_vlan_set_ingress_filter_wrap** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode, enum smi_vlan_port_ingress_filter vlanPortIngressFilter)
- int **smi_vlan_set_ingress_filter_wrap_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode mode, enum smi_vlan_port_ingress_filter enable)
- int **smi_vlan_set_vlanId_to_port_wrap_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId, enum smi_vlan_port_mode vlanToPortMode)
- int **smi_vlan_set_vlanId_to_port_wrap** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId, enum smi_vlan_port_mode vlanToPortMode)
- int **smi_vlan_set_trunk_allowed_vlan_wrap** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId, enum smi_vlan_trunk_allow vlanTrunkAllow)
- int **smi_vlan_set_trunk_allowed_vlan_wrap_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId, enum smi_vlan_trunk_allow vlanTrunkAllow)
- int **smi_vlan_add_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, char ∗vlanName, u_int16_t vlanId, enum smi_vlan_state vlanState, enum smi_vlan_type vlanType)
- int **smi_vlan_delete_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t vlanId, enum smi_vlan_type vlanType)

- int **smi_vlan_range_add_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t lowerVlan, u_int16_t higherVlan, enum smi_vlan_state vlanState, enum smi_vlan_type vlanType)
- int **smi_vlan_range_del_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t lowerVlan, u_int16_t higherVlan, enum smi_vlan_type vlanType)
- int **smi_vlan_api_set_port_mode_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode, enum smi_-vlan_port_mode vlanPortSubMode)
- int **smi_vlan_set_acceptable_frame_type_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode, enum smi_acceptable_frame_type frameType)
- int **smi_vlan_set_ingress_filter_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_ingress_filter vlanPortIngressFilter)
- int **smi_vlan_set_default_vid_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)
- int **smi_vlan_add_vlan_to_port_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, struct smi_vlan_bmp ∗vlanBmp, struct smi_vlan_bmp ∗egressTypeBmp, struct smi_vlan_bmp ∗successBmp)
- int **smi_vlan_delete_vlan_from_port_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, struct smi_vlan_bmp ∗vlanBmp, struct smi_vlan_bmp ∗successBmp)
- int **smi_vlan_clear_port_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifname)
- int **smi_vlan_add_all_except_vid_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode, enum smi_-vlan_port_mode vlanPortSubMode, struct smi_vlan_bmp ∗excludeBmp, enum smi_vlan_egress_type egressType, enum smi_vlan_add_opt vlanAddOpt)
- int **smi_trunk_set_native_vlan_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)
- int **smi_trunk_allowed_vlan_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)
- int **smi_vlan_set_access_port_vlan_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)
- int **smi_vlan_set_hybrid_port_vlan_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_int16_t vlanId)
- int **smi_nsm_vlan_add_hybrid_port_none_sdkapi_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- int **smi_nsm_vlan_add_hybrid_port_all_sdkapi_validate** (struct smiclient_-globals ∗azg, int vrId, char ∗ifName)
- int **smi_nsm_vlan_port_set_regen_user_priority_validate** (struct smiclient_-globals ∗azg, int vrId, char ∗ifName, u_char userPriority, u_char regenUserPri-ority)
- int **smi_nsm_vlan_port_set_traffic_class_table_validate** (struct smiclient_-globals ∗azg, int vrId, char ∗ifName, u_char userPriority, u_char trafficClass, u_char trafficClassValue)
- int **smi_nsm_vlan_port_set_default_user_priority_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, u_char userPriority)

- int **smi_nsm_vlan_br_name_word_validate** (struct smiclient_globals ∗azg, int vrId, u_int16_t vlanId, char ∗bridgeId, enum smi_vlan_type vlanType, int vlanState, char ∗vlanName)
- int **smi_nsm_vlan_set_mtu_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t vlanId, enum smi_vlan_type vlanType, u_int32_t mtuVal)
- int **smi_nsm_vlan_unset_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, int vlanId)
- int **smi_nsm_vlan_enable_disable_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t vlanId, int vlanState)
- s_int32_t **smi_vlan_clear_trunk_port_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- s_int32_t **smi_vlan_clear_hybrid_port_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- int **smi_vlan_unset_hybrid_port_vlan_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- int **smi_vlan_unset_access_port_vlan_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- int **smi_trunk_allowed_vlan_all_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- int **smi_trunk_allowed_vlan_none_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- int **smi_trunk_unset_native_vlan_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName)
- int **smi_trunk_unset_native_vlan_wrap** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, int vlanId, int vlanPortMode, int nativedisableFlag)
- int **smi_trunk_unset_native_vlan_wrap_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, int vlanId, int vlanPortMode, int nativedisableFlag)
- int **smi_vlan_api_set_portmode_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode)
- int **smi_vlan_api_set_switchport_mode_validate** (struct smiclient_globals ∗azg, int vrId, char ∗ifName, enum smi_vlan_port_mode vlanPortMode)
- int **smi_nsm_map_vlans_to_g8031_protection_group_validate** (struct smiclient_globals ∗azg, int vrId, char ∗bridgeId, u_int16_t epsId)

4.1.1 Detailed Description

Provides APIs for VLAN management.

4.1.2 Function Documentation

4.1.2.1 int smi_get_all_vlan_config (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, struct smi_vlan_bmp ∗ *vlanBmp*)

This API gets all VLAN IDs configured on a bridge. The bridge is identified by bridge name. smi_get_all_vlan_config

Parameters:

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

← ***bridgeId*** Bridge name pass 0 for default bridge.

→ ***vlanBmp*** Bitmap of the VLAN IDs that are configured. Before invoking this API, the user must allocate memory for this parameter

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN

4.1.2.2 int smi_get_bridge (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, struct smi_bridge ∗ *bridgeInfo*)

This API provides the functionality to retrieve any bridge information configured on a given bridge name. smi_get_bridge

Parameters:

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

← ***bridgeId*** Bridge Name

→ ***bridgeInfo*** Used to store the retrieved bridge information. Before invoking this API, the user must allocate memory for this parameter.

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL
NSM_VLAN_ERR_BRIDGE_NOT_FOUND

4.1.2.3 int smi_get_vlan_by_id (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, u_int16_t *vlanId*, struct smi_vlan_info ∗ *vlanInfo*)

This API gets the VLAN information configured on a given interface. smi_get_vlan_-by_id

Parameters:

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

← ***bridgeId*** Bridge name. Pass 0 for default bridge.

← ***vlanId*** VLAN identifier. This is a 16-bit unsigned integer.

→ ***vlanInfo*** VLAN related information for a particular VLAN ID. Before invoking this API, the user must allocate memory for this parameter.

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL
NSM_VLAN_ERR_BRIDGE_NOT_FOUND

4.1.2.4 int smi_get_vlan_by_name (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, char ∗ *vlanName*, struct smi_vlan_info ∗ *vlanInfo*)

get vlan smi_get_vlan_by_name

Parameters:

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

← ***nsm_bridge_master***

← ***bridgeId*** bridge name

← ***vlanName*** vlan name

→ ***vlanInfo*** vlan info

Returns:

0 on success, otherwise one of the following error codes
NSM_VLAN_ERR_BRIDGE_NOT_FOUND
NSM_VLAN_ERR_VLAN_NOT_FOUND
NSM_VLAN_ERR_IFP_NOT_BOUND

4.1.2.5 int smi_get_vlan_summary (struct smiclient_globals ∗ *azg*, int *vrId*, struct smi_bridge_vlan_summ ∗ *vlanSumm*)

Use this function to get all the interface's brief information. smi_get_vlan_summary

Parameters:

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

← ***vrId*** virtual router ID

→ ***vlanSumm*** Pointer to that structure smi_vlan_summ that contains vlan info. The caller must allocate memory for this parameter before invoking this API.

Returns:

RESULT_OK on success

4.1.2.6 int smi_nsm_map_vlans_to_g8031_protection_group (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, u_int16_t *epsId*)

This API provides the functionality to allowed access port to vlan. smi_nsm_map_-vlans_to_g8031_protection_group

Parameters:

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

← ***bridgeId***

← ***epsId***

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.7 int smi_nsm_vlan_add_hybrid_port_all_sdkapi (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

smi_nsm_vlan_add_hybrid_port_all_sdkapi

Parameters:

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

← ***ifName*** interface name

Returns:

RESULT_OK on success, otherwise one of the following error code NSM_-VLAN_ERR_IFP_NOT_BOUND
NSM_VLAN_ERR_INVALID_MODE
AGG_MEM_NO_SWITCHPORT
AGG_MEM_BRIDGE_NOT_VLAN_AWARE
NSM_VLAN_ERR_CONFIG_PVID_TAG
NSM_VLAN_ERR_VLAN_NOT_FOUND

4.1.2.8 int smi_nsm_vlan_br_name_word (struct smiclient_globals ∗ *azg*, int *vrId*, u_int16_t *vlanId*, char ∗ *bridgeId*, enum smi_vlan_type *vlanType*, int *vlanState*, char ∗ *vlanName*)

Display VLAN prio regen. smi_nsm_vlan_br_name_word

Parameters:

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

← ***vlanId*** vlan id <2-4094>

← ***bridgeId*** bridge name <1-32>
← ***vlanState*** enable or disable
← ***vlanType*** vlan type
← ***vlanName*** vlan name

Returns:

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

4.1.2.9 int smi_nsm_vlan_enable_disable (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, u_int16_t *vlanId*, int *vlanState*)

set vlan smi_nsm_vlan_enable_disable

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***nsm_bridge_master***
← ***bridgeId*** bridge name
← ***vlanId*** vlan id
← ***vlanState*** vlan state

Returns:

0 on success, otherwise one of the following error codes
NSM_VLAN_ERR_BRIDGE_NOT_FOUND
NSM_VLAN_ERR_VLAN_NOT_FOUND
NSM_VLAN_ERR_IFP_NOT_BOUND

4.1.2.10 int smi_nsm_vlan_port_set_default_user_priority (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, u_char *userPriority*)

set default priority smi_nsm_vlan_port_set_default_user_priority

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***ifName*** interface name
← ***userPriority*** <0-7>

Returns:

0 if success, otherwise one of the following error codes NSM_VLAN_ERR_IFP_-
NOT_BOUND
NSM_VLAN_ERR_BRIDGE_NOT_VLAN_AWARE
NSM_VLAN_ERR_IFP_INVALID
NSM_VLAN_ERR_BRIDGE_NOT_FOUND
NSM_DCB_API_SET_ERR_PRI_IS_CNPV
NSM_VLAN_ERR_BRIDGE_NOT_FOUND

4.1.2.11 int smi_nsm_vlan_port_set_regen_user_priority (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *if_name*, u_char *userPriority*, u_char *regenUserPriority*)

set regen user priority smi_nsm_vlan_port_set_regen_user_priority

Parameters:

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

← ***userPriority*** <0-7>

← ***regenUserPriority*** <0-7>

Returns:

0 if success, otherwise one of the following error codes NSM_VLAN_ERR_IFP_-NOT_BOUND
NSM_VLAN_ERR_BRIDGE_NOT_VLAN_AWARE
NSM_VLAN_ERR_IFP_INVALID
NSM_VLAN_ERR_BRIDGE_NOT_FOUND
NSM_DCB_API_SET_ERR_PRI_IS_CNPV
NSM_VLAN_ERR_BRIDGE_NOT_FOUND

4.1.2.12 int smi_nsm_vlan_port_set_traffic_class_table (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, u_char *userPriority*, u_char *trafficClass*, u_char *trafficClassValue*)

Creates a set vlan traffic class table. smi_nsm_vlan_port_set_traffic_class_table

Parameters:

← ***ifName*** Interface name

← ***userPriority*** <0-7>

← ***numTrafficClasses*** <1-8>

← ***trafficClassValue*** <0-7>

Returns:

0 if success, otherwise one of the following error codes NSM_VLAN_ERR_IFP_-NOT_BOUND
NSM_VLAN_ERR_BRIDGE_NOT_VLAN_AWARE
NSM_VLAN_ERR_IFP_INVALID
NSM_VLAN_ERR_BRIDGE_NOT_FOUND

4.1.2.13 int smi_nsm_vlan_set_mtu (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, u_int16_t *vlanId*, enum smi_vlan_type *vlanType*, u_int32_t *mtuVal*)

set mtu in vlan smi_nsm_vlan_set_mtu

Parameters:

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

← ***nsm_bridge_master***

← ***bridgeId*** bridge name

← ***vlanId*** vlan id

← ***vlanType*** vlan type

← ***mtuVal*** maximum transmission value

Returns:

0 on success, otherwise one of the following error codes
NSM_VLAN_ERR_BRIDGE_NOT_FOUND
NSM_VLAN_ERR_VLAN_NOT_FOUND
NSM_VLAN_ERR_IFP_NOT_BOUND

4.1.2.14 int smi_nsm_vlan_unset (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, int *vlanId*)

This API provides the functionality to allowed access port to vlan. smi_nsm_vlan_-unset

Parameters:

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

← ***bridgeId***

← ***vlanId***

Returns:

0 in case of success,

4.1.2.15 s_int32_t smi_show_api_default_priority (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, u_char ∗ *userPriority*)

Show the information of the configured user-priority. smi_show_api_default_priority

Parameters:

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

← ***vrId*** Virtual router id

← ***ifName*** Interface name

→ ***userPriority*** User priority

Returns:

RESULT_OK on success, otherwise one of the following error code NSM_API_-ERR_NO_NSM_MASTER
SMI_NSM_ERR_IF_NOT_EXIST

4.1.2.16 s_int32_t smi_show_api_interfaces_switchport_bridge (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *bridgeId*, struct list ∗ *ifList*)

Show the information of the configured user-priority regen table. smi_show_api_user_-prio_regen_table

Parameters:

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

← ***vrId*** Virtual router id

← ***bridgeId*** bridge name

→ ***ifList*** interface list

Returns:

RESULT_OK on success, otherwise one of the following error code NSM_API_-ERR_NO_NSM_MASTER
NSM_API_ERR_NO_NSM_BRIDGE_MASTER
NSM_API_ERR_BRIDGE_LOOKUP_FAIL
NSM_API_ERR_NO_BRIDGE_PORT_TREE
NSM_ERR_IF_NOT_BOUND
NSM_API_ERR_IF_NOT_SWITCHPORT_MODE
NSM_VLAN_ERR_NO_VLAN_PORT

4.1.2.17 s_int32_t smi_show_api_traffic_class_table (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, struct smi_traffic_class_table ∗ *trafficClass*)

Show the information of the configured VLAN traffic class table. smi_show_api_-traffic_class_table

Parameters:

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

← ***vrId*** Virtual router id

← ***ifName*** Interface name

→ ***trafficClass*** Traffic class table

Returns:

RESULT_OK on success, otherwise one of the following error code NSM_API_-ERR_NO_NSM_MASTER
SMI_NSM_ERR_IF_NOT_EXIST
NSM_VLAN_ERR_IFP_NOT_BOUND
NSM_VLAN_ERR_BRIDGE_NOT_VLAN_AWARE
NSM_VLAN_ERR_BRIDGE_NOT_FOUND

4.1.2.18 s_int32_t smi_show_api_user_prio_regen_table (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, struct smi_user_regen_prio ∗ *userRegenPrio*)

Show the information of the configured user-priority regen table. smi_show_api_user_-prio_regen_table

Parameters:

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

← ***vrId*** Virtual router id

← ***ifName*** Interface name

→ ***userRegenPrio*** User priority regen table

Returns:

RESULT_OK on success, otherwise one of the following error code NSM_API_-ERR_NO_NSM_MASTER
SMI_NSM_ERR_IF_NOT_EXIST
NSM_VLAN_ERR_IFP_NOT_BOUND
NSM_VLAN_ERR_BRIDGE_NOT_VLAN_AWARE
NSM_VLAN_ERR_BRIDGE_NOT_FOUND

4.1.2.19 int smi_show_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, int *startIndex*, int *endIndex*, struct list ∗ *vlanInfo*, int(∗)(struct list ∗vlanInfo) *callback*)

Use this function to get all the interface's brief information. smi_show_vlan

Parameters:

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

← ***startIndex*** start index

← ***endIndex*** end index

← ***vlanInfo*** Link list of structure smi_vlan_info. smi_vlan_info structure holds details of a specific vlan. List should be intialized by caller.

→ ***callback*** Callback function which take list as input parameter, here the list will be containing the nodes of struct smi_vlan_info. Pass NULL in case of no callback function required.

Returns:

RESULT_OK on success

4.1.2.20 int smi_trunk_allowed_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, u_int16_t *vlanId*)

This API provides the functionality to allowed vlan in trunk port. smi_trunk_allowed_-vlan

Parameters:

← ***azg*** Pointer to smiclient_globals structure
← ***ifName*** Interface name
← ***vlanId*** Vlan ID

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.21 int smi_trunk_allowed_vlan_all (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

This API provides the functionality to allowed all vlan to trunk port. smi_trunk_-allowed_vlan_all

Parameters:

← ***azg*** Pointer to smiclient_globals structure
← ***ifName*** Interface name

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.22 int smi_trunk_allowed_vlan_none (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

This API provides the functionality to remove all vlan from trunk port. smi_trunk_-allowed_vlan_none

Parameters:

← ***azg*** Pointer to smiclient_globals structure
← ***ifName*** Interface name

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.23 int smi_trunk_set_native_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, u_int16_t *vlanId*)

This API provides the functionality to set the trunk native vlan . smi_trunk_set_-native_vlan

Parameters:

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

← ***ifName*** Interface name

← ***vlanId*** Vlan ID

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.24 int smi_trunk_unset_native_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

This API provides the functionality to unset the trunk native vlan . smi_trunk_unset_-native_vlan

Parameters:

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

← ***ifName*** Interface name

← ***vlanId*** Vlan ID

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.25 int smi_vlan_add (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, char ∗ *vlanName*, u_int16_t *vlanId*, enum smi_vlan_state *vlanState*, enum smi_vlan_type *vlanType*)

Adds a vlan to specified bridge. smi_vlan_add

Parameters:

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

← ***bridgeId*** Bridge Name

← ***vlanName*** Vlan Name

← ***vlanId*** Vlan ID. Range is <SMI_VLAN_ID_START-SMI_VLAN_ID_-END>

← ***vlanState*** VLAN state, including:-

SMI_VLAN_INVALID - This is an INVALID state, so it should not be used.

SMI_VLAN_DISABLED - VLAN is under the suspended state. There is no VLAN tagging/untagging done.

SMI_VLAN_ACTIVE - VLAN is under active state. This should be passed as a parameter by the user.

← ***type*** Vlan type, including:-

VLAN_CVLAN

VLAN_SVLAN

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_vlan_add_all_except_vid (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, enum smi_vlan_port_mode *vlanPortMode*, enum smi_vlan_port_mode *vlanPortSubMode*, struct smi_vlan_bmp ∗ *excludeBmp*, enum smi_vlan_egress_type *egressType*, enum smi_vlan_add_opt *vlanAddOpt*)

This API provides the functionality to add all VLANs (except a specified VLAN) to a trunk, hybrid or provider network port. The different type of VLAN add options include one of the following:-

SMI_VLAN_CONFIGURED_ALL - To configure all the VLANs.

SMI_VLAN_CONFIGURED_NONE - To unconfigure all the VLANs except specified VLANs.

SMI_VLAN_CONFIGURED_SPECIFIC - To configure all the VLANs except specified VLANs. smi_vlan_add_all_except_vid

Parameters:

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

← ***ifName*** Interface Name

← ***vlanPortMode*** VLAN port mode, including:-

SMI_VLAN_PORT_MODE_INVALID
SMI_VLAN_PORT_MODE_ACCESS
SMI_VLAN_PORT_MODE_HYBRID
SMI_VLAN_PORT_MODE_TRUNK
SMI_VLAN_PORT_MODE_CE
SMI_VLAN_PORT_MODE_CN

SMI_VLAN_PORT_MODE_PN
SMI_VLAN_PORT_MODE_PE

← ***vlanPortSubMode*** Sub-mode of a VLAN port

← ***excludeBmp*** Bitmap of VLANs to be excluded

← ***egressType*** Type of egress, including:-
SMI_FRAME_TYPE_UNTAGGED
SMI_FRAME_TYPE_TAGGED

← ***vlanAddOpt*** Bitmap of VLANs to be excluded:-
SMI_VLAN_CONFIGURED_ALL=0
SMI_VLAN_CONFIGURED_NONE
SMI_VLAN_CONFIGURED_SPECIFIC

← ***egressTypeBmp*** Bitmap of the egressType for the corresponding VLAN, including: SMI_VLAN_EGRESS_UNTAGGED (0), SMI_VLAN_-EGRESS_TAGGED (1)

← ***successBmp*** Bitmap of the VLAN that was successfully added to the port. Before invoking this API, the user must allocate memory for this parameter

Returns:

0 in case of success, otherwise SMI_ERROR, SMI_ERROR_NULL_STRING, SMI_INVALID_STRLEN, SMI_INVALID_VAL, NSM_VLAN_ERR_IFP_-NOT_BOUND, NSM_VLAN_ERR_INVALID_MODE, NSM_VLAN_ERR_-GENERAL

4.1.2.27 int smi_vlan_add_vlan_to_port (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, struct smi_vlan_bmp ∗ *vlanBmp*, struct smi_vlan_bmp ∗ *egressTypeBmp*, struct smi_vlan_bmp ∗ *successBmp*)

This API adds the VLANs to the given interface port. smi_vlan_add_vlan_to_port

Parameters:

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

← ***ifName*** Interface Name

← ***vlanBmp*** Bitmap of the VLAN IDs to be deleted

← ***egressTypeBmp*** Bitmap of the egressType for the corresponding VLAN, including:
SMI_VLAN_EGRESS_UNTAGGED (0)
SMI_VLAN_EGRESS_TAGGED (1)

← ***successBmp*** Bitmap of the VLAN that was successfully added to the port. Before invoking this API, the user must allocate memory for this parameter

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN

4.1.2.28 int smi_vlan_api_get_port_mode (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, enum smi_vlan_port_mode ∗ *vlanPortMode*, enum smi_vlan_port_mode ∗ *vlanPortSubMode*)

This API retrieves the mode and submode that were configured on a VLAN interface. smi_vlan_api_get_port_mode

Parameters:

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

← ***ifName*** Interface Name

→ ***vlanPortMode*** VLAN port mode, which stores the retrieved mode value.

→ ***vlanPortSubMode*** VLAN port submode, which stores the retrieved submode value.

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.29 int smi_vlan_api_set_port_mode (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, enum smi_vlan_port_mode *vlanPortMode*, enum smi_vlan_port_mode *vlanPortSubMode*)

This API sets the mode and sub mode for a port on a VLAN. A user will set the modes on a port to know what type of traffic it carries; for example, if the traffic is customer network, provider network, or etc. The use should make sure that the corresponding VLAN is already configured. smi_vlan_api_set_port_mode

Parameters:

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

← ***ifName*** Interface Name

← ***vlanPortMode*** VLAN mode of a port

← ***vlanPortSubMode*** VLAN submode of a port. This is applicable for customer ports in the provider

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.30 int smi_vlan_api_set_portmode (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, enum smi_vlan_port_mode *vlanPortMode*)

This API provides the functionality to set port mode . smi_vlan_api_set_portmode

Parameters:

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

← ***ifName*** Interface name

← ***vlanPortMode*** Interface Port Mode

7 SMI_VLAN_PORT_MODE_PN

8 SMI_VLAN_PORT_MODE_CNP

10 SMI_VLAN_PORT_MODE_PIP

11 SMI_VLAN_PORT_MODE_CBP

12 SMI_VLAN_PORT_MODE_UAP

13 SMI_VLAN_PORT_MODE_CAP

14 SMI_VLAN_PORT_MODE_SBP

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.31 int smi_vlan_api_set_switchport_mode (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, enum smi_vlan_port_mode *vlanPortMode*)

This API provides the functionality to set the switchport port mode . smi_vlan_api_-set_switchport_mode

Parameters:

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

← ***ifName*** Interface name

← ***vlanPortMode*** Interface Port Mode

1 SMI_VLAN_PORT_MODE_ACCESS

2 SMI_VLAN_PORT_MODE_HYBRID

3 SMI_VLAN_PORT_MODE_TRUNK

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.32 s_int32_t smi_vlan_clear_hybrid_port (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

This API provides the functionality to remove trunk port. smi_vlan_clear_hybrid_port

Parameters:

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

← ***ifName*** Interface name

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.33 int smi_vlan_clear_port (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

This API clears the VLAN configurations from an interface port, except VLAN 1. For a hybrid/access port, the default VID resets to VLAN 1. smi_vlan_clear_port

Parameters:

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

← ***ifName*** Interface Name

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN

4.1.2.34 s_int32_t smi_vlan_clear_trunk_port (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

This API provides the functionality to remove trunk port. smi_vlan_clear_trunk_port

Parameters:

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

← ***ifName*** Interface name

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.35 int smi_vlan_delete (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, u_int16_t *vlanId*, enum smi_vlan_type *vlanType*)

Remove a vlan from specified bridge. smi_vlan_delete

Parameters:

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

← ***bridgeId*** Bridge Name

← ***vlanId*** Vlan ID. Range is <SMI_VLAN_ID_START-SMI_VLAN_ID_-END>. VLAN 1 cannot be deleted.

← ***vlanType*** Vlan type include:-

VLAN_CVLAN - VLAN with managed switch mode. Provides the monitoring of traffic pass through a particular port.

VLAN_SVLAN - VLAN with metro switch mode, Used by telecom/service providers to provide Ethernet features such as, OAM, Double VLAN, QOS, etc.

Returns:

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

4.1.2.36 int smi_vlan_delete_vlan_from_port (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, struct smi_vlan_bmp ∗ *vlanBmp*, struct smi_vlan_bmp ∗ *successBmp*)

This API deletes the VLANs that were added to a given interface name. smi_vlan_-delete_vlan_from_port

Parameters:

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

← ***ifName*** Interface Name

← ***vlanBmp*** Bitmap of the VLAN IDs to be deleted

← ***successBmp*** Bitmap of the VLAN that was successfully added to the port. Before invoking this API, the user must allocate memory for this parameter

Returns:

0 in case of success, otherwise SMI_ERROR

4.1.2.37 int smi_vlan_get_acceptable_frame_type (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, int ∗ *acceptableFrameType*)

This API provides the functionality to retrieve the type of acceptable frames that were configured on a VLAN port, such as a VLAN untagged frame, VLAN tagged frame or all. smi_vlan_get_acceptable_frame_type

Parameters:

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

← ***ifName*** Interface Name

→ ***acceptableFrameType*** VLAN Acceptable frame type (that is, untagged, tagged and all).

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.38 int smi_vlan_get_default_vid (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, u_int16_t ∗ *vlanId*)

API provides the functionality to configure a default VLAN identifier on an interface port. smi_vlan_get_default_vid

Parameters:

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

← ***ifName*** Interface Name

→ ***vlanId*** An integer pointer variable, which stores the retrieved default VLAN ID. Before invoking this API, the user must allocate memory for this parameter.

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING

4.1.2.39 int smi_vlan_get_ingress_filter (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, enum smi_vlan_port_mode ∗ *vlanPortMode*, enum smi_vlan_port_mode ∗ *vlanPortSubMode*, enum smi_vlan_port_ingress_filter ∗ *vlanPortIngressFilter*)

This API gets the ingress filtering status of a VLAN port by providing the functionality to retrieve filtering status on ingress side, such as enabled or disabled. It also gets the mode and the submode values along with the status of ingress filtering of a port. smi_vlan_get_ingress_filter

Parameters:

- $\leftarrow$ ***azg*** Pointer to smiclient_globals structure
- $\leftarrow$ ***ifName*** Interface Name
- $\rightarrow$ ***vlanPortMode*** Pointer to the smi_vlan_port_mode enum, which stores the retrieved mode value
- $\rightarrow$ ***vlanPortSubMode*** The smi_vlan_port_mode enum, which stores the retrieved submode value
- $\rightarrow$ ***vlanPortIngressFilter*** The smi_vlan_port_ingress_filter enum, which stores the retrieved status of ingress filtering like if it is enabled or disabled. Before invoking this API, the user must allocate memory for this parameter

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.40 int smi_vlan_if_get (struct smiclient_globals $*$ *azg*, int *vrId*, char $*$ *ifName*, struct smi_if_vlan_info $*$ *vlanInfo*)

This API gets the VLAN information configured on a given interface. smi_vlan_if_get

Parameters:

- $\leftarrow$ ***azg*** Pointer to smiclient_globals structure
- $\leftarrow$ ***ifName*** Interface Name
- $\rightarrow$ ***vlanInfo*** VLAN related information for a particular interface. Before invoking this API, the user must allocate memory for this parameter

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL
NSM_VLAN_ERR_BRIDGE_NOT_FOUND

4.1.2.41 int smi_vlan_range_add (struct smiclient_globals $*$ *azg*, int *vrId*, char $*$ *bridgeId*, u_int16_t *lowerVlan*, u_int16_t *higherVlan*, enum smi_vlan_state *vlanState*, enum smi_vlan_type *vlanType*)

Adds a range of vlan to the specified bridge. smi_vlan_range_add

Parameters:

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

← ***bridgeId*** Bridge Name

← ***vlanName*** Vlan Name

← ***lowerVlan*** Lower VLAN identifier of the range.

← ***higherVlan*** Higher VLAN identifier of the range.

← ***vlanState*** VLAN state, including:-

SMI_VLAN_INVALID - This is an INVALID state, so it should not be used.

SMI_VLAN_DISABLED - VLAN is under the suspended state. There is no VLAN tagging/untagging done.

SMI_VLAN_ACTIVE - VLAN is under active state. This should be passed as a parameter by the user.

← ***vlanType*** Vlan type include:-

VLAN_CVLAN - VLAN with managed switch mode. Provides the monitoring of traffic pass through a particular port.

VLAN_SVLAN - VLAN with metro switch mode, Used by telecom/service providers to provide Ethernet features such as, OAM, Double VLAN, QOS, etc.

Returns:

0 in case of success, otherwise
SMI_INVALID_STRLEN
SMI_INVALID_VAL
SMI_ERROR

4.1.2.42 int smi_vlan_range_del (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *bridgeId*, u_int16_t *lowerVlan*, u_int16_t *higherVlan*, enum smi_vlan_type *vlanType*)

Remove a range of vlan from the specified bridge. smi_vlan_range_del

Parameters:

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

← ***bridgeId*** Bridge Name

← ***lowerVlan*** lower index of the range

← ***higherVlan*** higher index of the range

← ***vlanType*** Vlan type include:-

VLAN_CVLAN - VLAN with managed switch mode. Provides the monitoring of traffic pass through a particular port.

VLAN_SVLAN - VLAN with metro switch mode, Used by telecom/service providers to provide Ethernet features such as, OAM, Double VLAN, QOS, etc.

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.43 int smi_vlan_set_acceptable_frame_type (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, enum smi_vlan_port_mode *vlanPortMode*, enum smi_acceptable_frame_type *frameType*)

This API sets the acceptable frame type for the VLAN port by providing the functionality to configure an acceptable frame type for a VLAN interface and mode. smi_-vlan_set_acceptable_frame_type

Parameters:

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

← ***ifName*** Interface Name

← ***vlanPortMode*** VLAN port mode, which stores the retrieved mode value.

← ***frameType*** VLAN Acceptable frame type (that is, untagged, tagged and all).

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.44 int smi_vlan_set_access_port_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, u_int16_t *vlanId*)

This API provides the functionality to allowed access port to vlan. smi_vlan_set_-access_port_vlan

Parameters:

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

← ***ifName*** Interface name

← ***vlanId*** Vlan ID

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.45 int smi_vlan_set_default_vid (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, u_int16_t *vlanId*)

API provides the functionality to configure a default VLAN identifier on an interface port. smi_vlan_set_default_vid

Parameters:

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

← ***ifName*** Interface Name

← ***vlanId*** Default VLAN identifier to be set, which is the type 16-bit unsigned integer.

Returns:

0 in case of success, otherwise
SMI_ERROR
SMI_ERROR_NULL_STRING
SMI_INVALID_STRLEN
SMI_INVALID_VAL

4.1.2.46 int smi_vlan_set_hybrid_port_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, u_int16_t *vlanId*)

This API provides the functionality to allowed hybrid port to vlan. smi_vlan_set_-hybrid_port_vlan

Parameters:

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

← ***ifName*** Interface name

← ***vlanId*** Vlan ID

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.47 int smi_vlan_set_ingress_filter (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*, enum smi_vlan_port_ingress_filter *vlanPortIngressFilter*)

This API sets the ingress filtering on a VLAN port. It provides the functionality for enabling/disabling the filtering for an incoming frame on a particular VLAN port. This API will look for what is the acceptable particular frame type defined for a particular mode and enable the filtering for the same, so that the rest of the frames are dropped. If the API is invoked with disable flag, then the filtering of the ingress frames will be stopped. smi_vlan_set_ingress_filter

Parameters:

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

← ***ifName*** Interface Name

← ***vlanPortIngressFilter*** Enum variable. It holds the enable/disable flag for ingress filtering.

Returns:

0 in case of success, otherwise SMI_ERROR

4.1.2.48 int smi_vlan_unset_access_hybrid_port_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifname*, int *vlanId*, int *vlanPortMode*, int *modeFlag*)

This API provides the functionality to unset acess/hybrid port vlan. smi_vlan_unset_-access_hybrid_port_vlan

Parameters:

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

← ***ifName*** Interface name

← ***vlanId***

← ***vlanportMode*** mode of port

← ***disableFlag*** access/hybrid

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.49 int smi_vlan_unset_access_port_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

This API provides the functionality to unset acess port vlan. smi_vlan_unset_access_-port_vlan

Parameters:

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

← ***ifName*** Interface name

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.1.2.50 int smi_vlan_unset_hybrid_port_vlan (struct smiclient_globals ∗ *azg*, int *vrId*, char ∗ *ifName*)

This API provides the functionality to unset hybrid port vlan. smi_vlan_unset_hybrid_-port_vlan

Parameters:

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

← ***ifName*** Interface name

Returns:

0 in case of success, otherwise one of the following error codes
SMI_ERROR
SMI_INVALID_STRLEN

4.2 smi_vlan_msg.h File Reference

Defines the data structure used by VLAN SMI APIs. `#include "smi_-message.h"`

Data Structures

- struct smi_if_vlan_info
- struct smi_bridge
- struct smi_vlan_summ
- struct smi_bridge_vlan_summ
- struct smi_vlan_info
- struct smi_vlan_info_list
- struct smi_user_regen_prio
- struct smi_if_swport_br
- struct smi_if_swport_br_list
- struct smi_traffic_class_table
- struct smi_msg_vlan

Defines

- #define **SMI_MSG_VLAN_SIZE** 4
- #define **SMI_VLAN_ID_START** 1
- #define **SMI_VLAN_ID_END** 4094
- #define **SMI_NSM_VLAN_NONE** 0
- #define **SMI_NSM_VLAN_DEFAULT_VID** 1
- #define **SMI_NSM_VLAN_ALL** SMI_VLAN_ID_END + 1
- #define **SMI_BRIDGE_GROUP_MIN** 1
- #define **SMI_BRIDGE_GROUP_MAX** 32
- #define **VLAN_NUM_TRAFFIC_CLASS_VALUE_MAX** 8
- #define **VLAN_NUM_TRAFFIC_CLASS_VALUE_MIN** 1
- #define **VLAN_TRAFFIC_CLASS_VALUE_MAX** 7
- #define **VLAN_TRAFFIC_CLASS_VALUE_MIN** 0
- #define **VLAN_STATE_ENABLE** 1
- #define **VLAN_STATE_DISABLE** 0
- #define **SMI_BRIDGE_MIN_VALUE** 1
- #define **SMI_BRIDGE_MAX_VALUE** 32
- #define **SMI_VLAN_MTU_MIN** 0
- #define **SMI_GVRP_ENABLED** 1
- #define **SMI_GVRP_DISABLED** 0
- #define **SMI_GMRP_ENABLED** 1
- #define **SMI_GMRP_DISABLED** 0
- #define **SMI_BRIDGE_MAX_TRAFFIC_CLASS** 8
- #define **SMI_BRIDGE_MAX_USER_PRIO** 7
- #define **VLAN_TRAFFIC_CLASS_VALUE_MAX** 7

- #define **VLAN_TRAFFIC_CLASS_VALUE_MIN** 0
- #define **SMI_HAL_BRIDGE_MIN_USER_PRIO** 0
- #define **SMI_HAL_BRIDGE_MAX_USER_PRIO** 7
- #define **SMI_HAL_BRIDGE_MIN_TRAFFIC_CLASS** 1
- #define **SMI_HAL_BRIDGE_MAX_TRAFFIC_CLASS** 8
- #define **SMI_VLAN_ENABLE_INGRESS_FILTER** (1 << 0)
- #define **SMI_VLAN_ACCEPTABLE_FRAME_TYPE_TAGGED** (1 << 1)
- #define **SMI_VLAN_ACCEPTABLE_FRAME_TYPE_UNTAGGED** (1 << 2)
- #define **SMI_BRIDGE_AGEING_DEFAULT** 300
- #define **SMI_LEARNING_BRIDGE_SET** 1
- #define **SMI_LEARNING_BRIDGE_UNSET** 0
- #define **SMI_VLAN_NAMSIZ** 32
- #define **SMI_VLAN_CTYPE_BR_NAME** 0
- #define **SMI_VLAN_CTYPE_VLAN_NAME** 1
- #define **SMI_VLAN_CTYPE_VLAN_ID** 2
- #define **SMI_VLAN_CTYPE_VLAN_STATE** 3
- #define **SMI_VLAN_CTYPE_VLAN_TYPE** 4
- #define **SMI_VLAN_CTYPE_PORT_MODE** 5
- #define **SMI_VLAN_CTYPE_PORT_SUBMODE** 6
- #define **SMI_VLAN_CTYPE_IF_NAME** 7
- #define **SMI_VLAN_CTYPE_ACC_FRAME_TYPE** 8
- #define **SMI_VLAN_CTYPE_INGRESS_FILTER** 9
- #define **SMI_VLAN_CTYPE_EGRESS_TYPE** 10
- #define **SMI_VLAN_CTYPE_BITMAP** 11
- #define **SMI_VLAN_CTYPE_NATIVE_VLAN** 12
- #define **SMI_VLAN_CTYPE_VLAN_INFO** 13
- #define **SMI_VLAN_CTYPE_IF_VLAN_INFO** 14
- #define **SMI_VLAN_CTYPE_BR_INFO** 15
- #define **SMI_VLAN_CTYPE_BR_PROTO** 16
- #define **SMI_VLAN_CTYPE_BR_PROTO_PROCESS** 17
- #define **SMI_VLAN_CTYPE_EGRESS_TYPE_BMAP** 18
- #define **SMI_VLAN_CTYPE_PORT_BMAP** 19
- #define **SMI_VLAN_CTYPE_PORT_ETHER_TYPE** 20
- #define **SMI_NSM_SWITCH** 21
- #define **SMI_VLAN_ADD_OPT** 22
- #define **SMI_VLAN_CTYPE_LOWER_VLAN_ID** 23
- #define **SMI_VLAN_CTYPE_HIGHER_VLAN_ID** 24
- #define **SMI_VLAN_INFO_LIST** 25
- #define **SMI_VLAN_SUMM** 26
- #define **SMI_VLAN_CTYPE_VR_ID** 27
- #define **SMI_VLAN_CTYPE_EPS_ID** 28
- #define **SMI_VLAN_CTYPE_NUM_TRAFFIC_CLASSES** 29
- #define **SMI_VLAN_CTYPE_EXTENDED_1** 31
- #define **SMI_VLAN_CTYPE_TRAFFIC_CLASS_VALUE_MAX** 0
- #define **SMI_VLAN_CTYPE_REGEN_USER_PRIORITY** 1

- #define **SMI_VLAN_CTYPE_TRAFFIC_CLASS_TABLE** 2
- #define **SMI_VLAN_CTYPE_USER_REGEN_PRIO** 3
- #define **SMI_VLAN_CTYPE_IF_SWPORT_BR_LIST** 4
- #define **SMI_BRIDGE_CTYPE_VLAN_SUMM** 5
- #define **SMI_VLAN_CTYPE_USER_PRIORITY** 6
- #define **SMI_VLAN_CTYPE_RANGE_STRING** 7

Enumerations

- enum **smi_vlan_trunk_allow** { **SMI_VLAN_NO_TRUNK_ALLOW** = 0, **SMI_VLAN_TRUNK_ALLOW** }
- enum **smi_vlan_packet_type** { **SMI_VLAN_PACKET_UNTAGGED** = 0, **SMI_VLAN_PACKET_TAGGED** }
- enum **smi_vlan_state** { **SMI_VLAN_INVALID**, **SMI_VLAN_DISABLED**, **SMI_VLAN_ACTIVE** }
- enum **smi_vlan_egress_type** { **SMI_VLAN_EGRESS_UNTAGGED** = 0, **SMI_VLAN_EGRESS_TAGGED** = 1 }
- enum **smi_vlan_type** { **VLAN_CVLAN**, **VLAN_SVLAN** }
- enum **smi_vlan_add_opt** { **SMI_VLAN_CONFIGURED_ALL** = 0, **SMI_-VLAN_CONFIGURED_NONE**, **SMI_VLAN_CONFIGURED_SPECIFIC** }
- enum **smi_acceptable_frame_type** { **SMI_FRAME_TYPE_UNTAGGED**, **SMI_FRAME_TYPE_TAGGED**, **SMI_FRAME_TYPE_ALL** }
- enum **smi_vlan_port_ingress_filter** { **SMI_VLAN_PORT_DISABLE_-INGRESS_FILTER**, **SMI_VLAN_PORT_ENABLE_INGRESS_FILTER** }
- enum **smi_topology** { **SMI_TOPOLOGY_NONE**, **SMI_TOPOLOGY_-RING** }
- enum **smi_bridge_proto** {

  SMI_PROTO_STP, **SMI_PROTO_RSTP**, **SMI_PROTO_MSTP**, **SMI_-PROTO_GMRP**,

  SMI_PROTO_GVRP, **SMI_PROTO_MMRP**, **SMI_PROTO_MVRP**, **SMI_PROTO_LACP**,

  SMI_PROTO_DOT1X, **SMI_PROTO_LLDP**, **SMI_PROTO_CFM**, **SMI_-PROTO_TRILL**,

  SMI_PROTO_SPB, **SMI_PROTO_MAX** }
- enum **smi_bridge_proto_process** { **SMI_PROTO_PROCESS_PEER**, **SMI_-PROTO_PROCESS_TUNNEL**, **SMI_PROTO_PROCESS_DISCARD**, **SMI_PROTO_PROCESS_MAX** }

Functions

- void **smi_vlan_dump** (struct lib_globals ∗zg, struct smi_msg_vlan ∗msg)
- int **smi_encode_vlan_msg** (u_char ∗∗pnt, u_int16_t ∗size, struct smi_msg_vlan ∗msg)

- int **smi_decode_vlan_msg** (u_char ∗∗pnt, u_int16_t ∗size, struct smi_msg_vlan ∗msg)
- int **smi_parse_vlan** (u_char ∗∗, u_int16_t ∗, struct smi_msg_header ∗, void ∗, SMI_CALLBACK)

4.2.1 Detailed Description

Defines the data structure used by VLAN SMI APIs.

Index