ZebOS-XP DCB SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:22 2015

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Data Structure Documentation

3.1 dcb_msg_ Struct Reference

Data Fields

- smi_cindex_t **cindex**
- smi_cindex_t **cindex_extended**
- u_int32_t **vr_id**
- u_int8_t **cp_id**
- bool_t **header**
- u_int8_t **tcgid**
- u_int8_t **pri**
- u_int8_t **maxtcg**
- u_int8_t **sel**
- u_int16_t **proto_id**
- u_int8_t **prio_map**
- int **type**
- u_int32_t **sample_base**
- s_int32_t **weight**
- u_int32_t **min_hdr_octet**
- u_int8_t **flag**
- u_int8_t **priority_map**
- char **ifname** [SMI_INTERFACE_NAMSIZ]
- bool_t **willing**
- bool_t **advertise**
- char **bridge_name** [SMI_BRIDGE_NAMSIZ]
- u_int8_t **mode**
- struct interface * **ifp**
- u_int8_t **transmit_priority**
- u_int8_t **priority**
- u_int8_t **user_priority**
- u_int8_t **regen_priority**

- u_int8_t **cnpv**
- u_int8_t **cap**
- u_int64_t **lda**
- u_int8_t **defense_mode**
- u_int8_t **cnpv_xmit_ready**
- u_int8_t **remote_cnpv**
- u_int8_t **alternate_priority**
- u_int8_t **bw** [SMI_NSM_DCB_MAX_TCG_DEFAULT]
- struct smi_dcb_tcg_intf **tcg_intf**
- struct smi_dcb_pfc_intf **pfc_intf**
- struct smi_scb_qcn_intf **qcn_intf**
- struct smi_dcb_app_intf **app_intf**

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

- smi_dcb_msg.h

3.2 smi_app_config Struct Reference

Data Fields

- u_int8_t **flags**
- u_int8_t **wflags**
- struct smi_appl_prio **appl_prio_table** [SMI_DCB_MAX_APPL_PRIO]
- u_int8_t **appl_prio_count**

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

- smi_dcb_msg.h

3.3 smi_appl_prio Struct Reference

Data Fields

- bool_t **enable**
- u_int16_t **proto_id**
- u_int8_t **sel**
- u_int8_t **priority_map**
- enum smi_dcb_appl_pri_mapping_type **mapping_type**

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

- smi_dcb_msg.h

3.4 smi_dcb_app_by_bridge Struct Reference

Data Fields

- char * **name**
- char * **ifname**
- u_int8_t **app_mode**
- u_int8_t **flag_admin**
- u_int8_t **wflags_admin**
- char * **proto_str**
- char * **pri_str**
- char * **sel**

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

- smi_dcb_msg.h

3.5 smi_dcb_app_intf Struct Reference

Data Fields

- struct smi_app_config **app_config_admin**
- struct smi_app_config **app_config_oper**
- enum smi_nsm_dcb_mode **app_mode**

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

- smi_dcb_msg.h

3.6 smi_dcb_appl_priority_table Struct Reference

Data Fields

- char * **name**
- char * **ifname**
- u_int8_t **mode**
- u_int8_t **flag**
- u_int8_t **wflags**
- char * **pri_str**
- char * **sel**
- char * **proto_str**

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

- smi_dcb_msg.h

3.7 smi_dcb_pfc_details_bridge Struct Reference

Data Fields

- char * **ifname**
- u_int8_t **pfc_mode**
- u_int8_t **flag_admin**
- u_int8_t **wflags_admin**
- u_int8_t **cap**
- u_int8_t **lda**
- u_int8_t **flag_oper**
- char * **prio_map_str**

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

- smi_dcb_msg.h

3.8 smi_dcb_pfc_details_intf Struct Reference

Data Fields

- char * **bridge_name**
- char * **ifname**
- u_int8_t **flag**
- u_int8_t **pfc_mode**
- u_int8_t **flag_admin**
- u_int8_t **wflags_admin**
- u_int8_t **cap_admin**
- u_int8_t **lda_admin**
- char * **prio_map_str**
- u_int8_t **flags_en**
- u_int8_t **cap_oper**
- u_int8_t **lda_oper**

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

- smi_dcb_msg.h

3.9 smi_dcb_pfc_intf Struct Reference

Data Fields

- struct smi_pfc_config **pfc_config_admin**
- struct smi_pfc_config **pfc_config_oper**
- enum smi_nsm_dcb_mode **pfc_mode**

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

- smi_dcb_msg.h

3.10 smi_dcb_pfc_stats_bridge Struct Reference

Data Fields

- char * **name**
- char * **ifname**
- char * **pause_sent_str**
- char * **pause_rcvd_str**

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

- smi_dcb_msg.h

3.11 smi_dcb_pfc_stats_intf Struct Reference

Data Fields

- char * **bridge_name**
- char * **ifname**
- char * **int_name**
- u_int8_t **pause_sent1**
- u_int8_t **pause_rcvd1**
- u_int8_t **pause_sent2**
- u_int8_t **pause_rcvd2**
- char * **pause_sent_str**
- char * **pause_rcvd_str**

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

- smi_dcb_msg.h

3.12 smi_dcb_qcn_cnpv Struct Reference

Data Fields

- char * **ifname**
- u_int8_t **i**
- u_int8_t **j** [SMI_NSM_NUM_CNPV]
- u_int8_t **mode**
- u_int8_t **alt_prio**
- u_int8_t **defense_mode**

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

- smi_dcb_msg.h

3.13 smi_dcb_qcn_cnpv_global Struct Reference

Data Fields

- u_int8_t **is_cnpv** [7]
- u_int8_t **j** [SMI_NSM_NUM_CNPV]
- u_int8_t **mode** [SMI_NSM_NUM_CNPV]
- u_int8_t **alt_prio** [SMI_NSM_NUM_CNPV]
- u_int8_t **defense_mode** [SMI_NSM_NUM_CNPV]
- struct smi_dcb_qcn_cnpv * **qcn_cnpv**

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

- smi_dcb_msg.h

3.14 smi_dcb_qcn_config_global Struct Reference

Data Fields

- u_int8_t **cnm_transmit_priority**
- u_int8_t **discarded_frames**
- u_int8_t **err_port_list** [100]
- struct smi_dcb_qcn_cnpv_global * **qcn_cnpv_global**
- struct smi_dcb_qcn_cp_global * **qcn_cp_global**

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

- smi_dcb_msg.h

3.15 smi_dcb_qcn_cp Struct Reference

Data Fields

- char * **ifname**
- float **weight**
- u_int32_t **samplebase**
- u_int32_t **minhdroctet**
- u_int32_t **qsp**

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

- smi_dcb_msg.h

3.16 smi_dcb_qcn_cp_cpid Struct Reference

Data Fields

- u_int32_t **cp_id**
- u_char **ifname** [SMI_INTERFACE_NAMSIZ]
- u_char **cp_mac_addr** [6]
- float **weight**
- u_int32_t **qsp**
- u_int32_t **qlen**
- u_int32_t **qlenold**
- u_int32_t **samplebase**
- u_int32_t **discarded_frames**
- u_int32_t **transmitted_frames**
- u_int32_t **minhdroctet**
- s_int32_t **qoffset**
- s_int32_t **qdelta**
- s_int32_t **fb**
- s_int32_t **enqued**

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

- smi_dcb_msg.h

3.17 smi_dcb_qcn_cp_global Struct Reference

Data Fields

- struct smi_dcb_qcn_cp * **qcn_cp**

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

- smi_dcb_msg.h

3.18 smi_dcb_tcg_bridge Struct Reference

Data Fields

- char * **name**
- char * **ifname**
- u_int8_t **mode**
- u_int8_t **flag_admin**
- u_int8_t **wflags_admin**
- u_int8_t **flag_oper**
- u_int8_t **max_tcgs**
- u_int8_t **pri_str**
- u_int8_t **tcgid**
- u_int8_t **tcg_bw** [SMI_NSM_DCB_MAX_TCG_DEFAULT]
- u_int8_t **ets_mode**

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

- smi_dcb_msg.h

3.19 smi_dcb_tcg_intf Struct Reference

Data Fields

- struct smi_ets_config **ets_config_admin**
- struct smi_ets_config **ets_config_oper**
- enum smi_nsm_dcb_mode **ets_mode**

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

- smi_dcb_msg.h

3.20 smi_ets_config Struct Reference

Data Fields

- u_int8_t **flags**
- u_int8_t **wflags**
- u_int8_t **cbs**
- u_int8_t **max_tcgs**
- u_int8_t **prio_tcg_map** [SMI_DCB_MAX_PRI]
- u_int8_t **tcg_bw_table** [SMI_DCB_MAX_TCG]
- u_int8_t **tsa_tc_table** [SMI_DCB_MAX_TSA]

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

- smi_dcb_msg.h

3.21 smi_nsm_cp_if_data Struct Reference

Data Fields

- bool_t **is_cp_enable**
- u_int32_t **samplebase**
- s_int8_t **weight**
- u_int32_t **minhdroctet**

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

- smi_dcb_msg.h

3.22 smi_nsm_qcn_cnpv_data Struct Reference

Data Fields

- bool_t **is_a_cnpv**
- enum smi_nsm_dcb_qcn_mode **mode**
- u_int8_t **alternate_priority**
- u_int32_t **defense_mode**
- struct smi_nsm_cp_if_data * **cpd**

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

- smi_dcb_msg.h

3.23 smi_nsm_qcn_data Struct Reference

Data Fields

- struct nsm_dcb_bridge * **dcbg**
- struct smi_nsm_qcn_cnpv_data **nsm_qcn_cnpv_data** [SMI_NSM_NUM_CNPV]
- u_int8_t **cnm_transmit_priority**

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

- smi_dcb_msg.h

3.24 smi_pfc_config Struct Reference

Data Fields

- u_int8_t **flags**
- u_int8_t **wflags**
- u_int8_t **priority_map**
- u_int8_t **pfc_mbc**
- u_int8_t **cap**
- u_int32_t **link_delay_allowance**

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

- smi_dcb_msg.h

3.25 smi_scb_qcn_intf Struct Reference

Data Fields

- struct smi_nsm_qcn_cnpv_data **nsm_qcn_cnpv_data_if** [SMI_NSM_NUM_-CNPV]
- struct smi_nsm_cp_if_data **cpd_if** [SMI_NSM_NUM_CNPV]
- struct smi_nsm_qcn_data **qcnd**
- struct smi_nsm_cp_if_data **cpd_bridge** [SMI_NSM_NUM_CNPV]

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

- smi_dcb_msg.h

Chapter 4

File Documentation

4.1 smi_dcb.h File Reference

Provides API for managing Data Center Bridging. `#include "smi_client.h"`

`#include "smi_dcb_msg.h"`

Functions

- s_int32_t **smi_client_create_n_send_dcb_msg** (struct smi_client_handler *async, int vrid, dcb_msg *msg, int optype)
- s_int32_t smi_dcb_set_bridge (struct smiclient_globals *azg, u_int32_t vr_id, char *bridge_name, int set_dcbbr)

  Enables/ Disables the DCB on bridge.

- s_int32_t smi_dcb_set_interface (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int set_dcbif)

  Enables/ Disables the DCB on interface.

- s_int32_t smi_dcb_ets_set_bridge (struct smiclient_globals *azg, u_int32_t vr_id, char *bridge_name, int etsbr)

  This function enables/disables ETS on Bridge.

- s_int32_t smi_dcb_ets_set_interface (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, u_int8_t mode)

  This function enables/disables ETS on interface.

- s_int32_t smi_dcb_ets_set_willing (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, bool_t willing)

  This function Enables/ Disables the ETS willing configurations on the interface.

- s_int32_t smi_dcb_ets_set_advertise (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, bool_t advertise)

This function Enables/ Disables the ETS advertise local configurations on the interface.

- s_int32_t smi_dcb_ets_set_pri_to_tcg (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, u_int8_t tcgid, u_int8_t pri, int set_pritcg)

  This function Adds/removes priorities in traffic-class-group on the interface.

- s_int32_t smi_dcb_ets_set_max_tcg (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, u_int8_t maxtcg)

  This function Set/Unset the maximum traffic-class-group to be configured on the interface.

- s_int32_t smi_dcb_ets_delete_tcgs (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)

  This function Deletes the traffic-class-groups from the ETS interface.

- s_int32_t smi_dcb_get_tcg_by_interface (struct smiclient_globals *azg, struct smi_dcb_tcg_intf *tcg_intf, u_int32_t vr_id, char *ifname)

  This function Get traffic-class-groups information from the ETS interface.

- s_int32_t smi_dcb_pfc_set_bridge (struct smiclient_globals *azg, u_int32_t vr_id, char *bridge_name, int set_pfcbr)

  This function Enables/ Disables the PFC on bridge.

- s_int32_t smi_dcb_pfc_set_interface (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, u_int8_t mode)

  This function Enables/ Disables the PFC on interface.

- s_int32_t smi_dcb_pfc_set_priority (struct smiclient_globals *azg, char *ifname, u_int8_t priority_map, int setpri)

  This function Enable/Disables priority flow control for a given priority on the interface.

- s_int32_t smi_dcb_pfc_set_cap (struct smiclient_globals *azg, char *ifname, u_int8_t cap)

  This function Set/Unset PFC cap for the interface.

- s_int32_t smi_dcb_pfc_set_lda (struct smiclient_globals *azg, char *ifname, u_int32_t lda)

  This function Set/Unset PFC link delay allowance for the interface.

- s_int32_t smi_dcb_pfc_set_willing (struct smiclient_globals *azg, char *ifname, bool_t willing)

  Enables/ Disables the PFC willing configurations on the interface.

- s_int32_t smi_dcb_pfc_set_advertise (struct smiclient_globals *azg, char *ifname, bool_t advertise)

Enables/ Disables the PFC advertise local configurations on the interface.

- s_int32_t smi_dcb_get_pfc_details_by_interface (struct smiclient_globals *azg, dcb_msg *getmsg, u_int32_t vr_id, char *ifname)

  Gets PFC information from the interface.

- s_int32_t smi_dcb_app_set_bridge (struct smiclient_globals *azg, u_int32_t vr_id, char *bridge_name, int app_set)

  Enables/ Disables the Application priority on the bridge.

- s_int32_t smi_dcb_app_set_interface (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, u_int8_t mode)

  Enables/ Disables the Application priority mode on interface.

- s_int32_t smi_dcb_app_set_priority (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, u_int8_t sel, u_int16_t app_port_no, char *app_serv_name, u_int8_t prio_map, int type, int app_pri_set)

  Set the application priority for well known tcp/udp ports ethertype based on port number/service name/hex value/protocol name on the interface.

- s_int32_t smi_dcb_app_set_advertise (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, bool_t advertise)

  Enables/ Disables the APP to advertise local configurations on the interface.

- s_int32_t smi_dcb_set_tcg_bandwidth (struct smiclient_globals *azg, u_int32_t vr_id, char *if_name, u_int8_t *bw)

  Assign the bandwidth percentage to traffic-class-groups on the interface.

- s_int32_t smi_dcb_get_app_by_interface (struct smiclient_globals *azg, struct smi_dcb_app_intf *app_intf, u_int32_t vr_id, char *ifname)

  Get Appplication Priority information from the interface.

- s_int32_t smi_dcb_qcn_set_bridge (struct smiclient_globals *azg, u_int32_t vr_id, char *bridge_name, u_int8_t transmit_priority, int set_qcn)

  Enables/ Disables the QCN on bridge.

- s_int32_t smi_dcb_qcn_create_cp (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, u_int8_t cnpv, u_int32_t sample_base, s_int32_t weight, u_int32_t min_hdr_octet, int set_cp)

  Create/Remove cp at an interface and associated parameters with it.

- s_int32_t smi_dcb_qcn_set_priority (struct smiclient_globals *azg, u_int32_t vr_id, char *bridge_name, u_int8_t priority, int set_qcn_pri)

  Set/Unset the Congestion Notification Priority Value.

- s_int32_t smi_dcb_qcn_set_mode_global (struct smiclient_globals *azg, u_int32_t vr_id, char *bridge_name, u_int8_t cnpv, u_int8_t mode, u_int8_t defense_mode, u_int8_t alternate_priority)

Create the defense mode per CNPV on a component level.

- s_int32_t smi_dcb_qcn_set_mode (struct smiclient_globals ∗azg, u_int32_t vr_id, char ∗ifname, u_int8_t cnpv, u_int8_t mode, u_int8_t defense_mode, u_int8_t alternate_priority)

  Create the defense mode per CNPV on a port level.

- s_int32_t smi_dcb_get_qcn_by_intf (struct smiclient_globals ∗azg, struct smi_scb_qcn_intf ∗qcn_intf, u_int32_t vr_id, char ∗ifname)

  QCN get Command.

4.1.1 Detailed Description

Provides API for managing Data Center Bridging. The APIs provided in this file forms the basis of ZebOS DCB management. These APIs are used by various north bound management interfaces like CLI, SNMP etc. for managing DCB.

4.1.2 Function Documentation

4.1.2.1 s_int32_t smi_dcb_app_set_advertise (struct smiclient_globals ∗ *azg*, u_int32_t *vr_id*, char ∗ *ifname*, bool_t *advertise*)

Enables/ Disables the APP to advertise local configurations on the interface. smi_dcb_app_set_advertise

Parameters:

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

← ***ifname*** Interface name

← ***vr_id*** Virtual router id <1-255>

← ***advertise*** 1 - Enables APP advertise on interface 0 - disables APP advertise on interface

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_INTERFACE
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_APP
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_APP

4.1.2.2 s_int32_t smi_dcb_app_set_bridge (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *bridge_name*, int *app_set*)

Enables/ Disables the Application priority on the bridge. smi_dcb_app_set_bridge

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***bridge_name*** Bridge name <1-32>

← ***app_set*** 1 - Enable application priority on bridge 0 - Disable application priority on bridge

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_VLAN
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_APP_EXISTS
NSM_DCB_API_SET_ERR_NO_APP

4.1.2.3 s_int32_t smi_dcb_app_set_interface (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, u_int8_t *mode*)

Enables/ Disables the Application priority mode on interface. smi_dcb_app_set_interface

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***mode*** 1 - Enable application priority mode on interface 0 - Disable application priority mode on interface

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_INTERFACE
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_NO_APP
NSM_DCB_API_SET_ERR_APP_EXISTS
NSM_DCB_API_SET_ERR_NO_MEM
NSM_DCB_API_SET_ERR_INTERFACE_NO_APP

4.1.2.4 s_int32_t smi_dcb_app_set_priority (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, u_int8_t *sel*, u_int16_t *app_port_no*, char * *app_serv_name*, u_int8_t *prio_map*, int *type*, int *app_pri_set*)

Set the application priority for well known tcp/udp ports ethertype based on port number/service name/hex value/protocol name on the interface. smi_dcb_app_set_priority

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***sel*** 1 - Ethertype 2 - tcp 3 - udp 4 - tcp and udp

← ***app_port_no*** if type = 1, port number of the service if type = 2, hex value of the ethetype

← ***app_serv_name*** if type = 0, service name if type = 3, protocol name of the ethertype

← ***prio_map*** Priority

← ***type*** 0 - app_serv_name containing service name for tcp/udp ports is taken as input 1 - app_port_no containing port number for tcp/udp ports is taken as input 2 - app_port_no containing hex value of the ethetype is taken as input 3 - app_serv_name containing protocol name of the ethertype is taken as input

← ***app_pri_set*** 1 - Set the application priority 0 - Unset the application priority

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_INTERFACE
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_NO_APP
NSM_DCB_API_SET_ERR_INTERFACE_NO_APP
NSM_DCB_API_SET_ERR_EXCEED_APP_PRIO

4.1.2.5 s_int32_t smi_dcb_ets_delete_tcgs (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*)

This function Deletes the traffic-class-groups from the ETS interface. smi_dcb_ets_delete_tcgs

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_DCB_API_SET_ERR_INTERFACE
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_ETS
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS
NSM_DCB_API_SET_ERR_ETS_NO_TCGS

4.1.2.6 s_int32_t smi_dcb_ets_set_advertise (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, bool_t *advertise*)

This function Enables/ Disables the ETS advertise local configurations on the interface. smi_dcb_ets_set_advertise

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***advertise*** 1 - Enables ETS advertise on interface 0 - disables ETS advertise on interface

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_DCB_API_SET_ERR_INTERFACE
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_ETS
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS

4.1.2.7 s_int32_t smi_dcb_ets_set_bridge (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *bridge_name*, int *etsbr*)

This function enables/disables ETS on Bridge. smi_dcb_ets_set_bridge

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ∗***bridge_name*** Bridge name

← ***etsbr*** 1-Enable ETS on bridge 0-Disable ETS on bridge

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_VLAN
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_ETS_EXISTS
NSM_DCB_API_SET_ERR_HW_NO_SUPPORT
NSM_DCB_API_SET_ERR_ETS_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS

4.1.2.8 s_int32_t smi_dcb_ets_set_interface (struct smiclient_globals ∗ *azg*, u_int32_t *vr_id*, char ∗ *ifname*, u_int8_t *mode*)

This function enables/disables ETS on interface. smi_dcb_ets_set_interface

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ∗***ifname*** Interface name

← ***mode*** 2-mode auto,1-mode on,0-disable

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_DCB_API_SET_ERR_INTERFACE
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_ERR_AGG_PORT
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_ETS
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_ETS_EXISTS
NSM_DCB_API_SET_ERR_NO_MEM
NSM_DCB_QOS_ENABLED_ERROR
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS_REC

4.1.2.9 s_int32_t smi_dcb_ets_set_max_tcg (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, u_int8_t *maxtcg*)

This function Set/Unset the maximum traffic-class-group to be configured on the interface. smi_dcb_ets_set_max_tcg

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***maxtcg*** Maximum TCGs to be configured on the interface 0 - Unsets max TCG

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_DCB_API_SET_ERR_INTERFACE
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_ETS
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS
NSM_DCB_API_SET_ERR_CONFIGURED_TCGS

4.1.2.10 s_int32_t smi_dcb_ets_set_pri_to_tcg (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, u_int8_t *tcgid*, u_int8_t *pri*, int *set_pritcg*)

This function Adds/removes priorities in traffic-class-group on the interface. smi_dcb_ets_set_pri_to_tcg

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***tcgid*** TCG-ID

← ***pri*** Priority

← ***set_pritcg*** 1 - Adds priorities in TCG on the interface 0 - Removes priorities in TCG on the interface

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM

NSM_DCB_API_SET_ERR_INTERFACE
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_ETS
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS
NSM_DCB_API_SET_ERR_MAXTCGS_CROSSED

4.1.2.11 s_int32_t smi_dcb_ets_set_willing (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, bool_t *willing*)

This function Enables/ Disables the ETS willing configurations on the interface. smi_-dcb_ets_set_willing

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***willing*** 1-Enables ETS willing on interface 0 - disables ETS willing on interface

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_DCB_API_SET_ERR_INTERFACE
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_ETS
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS

4.1.2.12 s_int32_t smi_dcb_get_app_by_interface (struct smiclient_globals * *azg*, struct smi_dcb_app_intf * *app_intf*, u_int32_t *vr_id*, char * *ifname*)

Get Appplication Priority information from the interface. smi_dcb_get_app_by_-interface

Parameters:

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

← ***ifname*** Interface name

← ***vr_id*** Virtual router id <1-255>

← ***app_intf*** Structure to be filled

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_INTERFACE
NSM_DCB_API_SET_ERR_NO_DCBG
SMI_DCB_API_SET_ERR_INTERFACE_NO_APP
NSM_DCB_API_SET_ERR_DCB_INTERFACE

4.1.2.13 s_int32_t smi_dcb_get_pfc_details_by_interface (struct smiclient_globals * *azg*, dcb_msg * *getmsg*, u_int32_t *vr_id*, char * *ifname*)

Gets PFC information from the interface. smi_dcb_get_pfc_details_by_interface

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***getmsg*** Structure to be filled

Returns:

0 on success, otherwise one of the following error codes
SMI_DCB_API_SET_ERR_INTERFACE
SMI_BRIDGE_ERR_NOTFOUND
SMI_DCB_API_SET_ERR_NO_DCBG
SMI_DCB_API_SET_ERR_NO_PFC
SMI_DCB_API_SET_ERR_DCB_INTERFACE
SMI_DCB_API_SET_ERR_INTERFACE_NO_PFC

4.1.2.14 s_int32_t smi_dcb_get_qcn_by_intf (struct smiclient_globals * *azg*, struct smi_scb_qcn_intf * *qcn_intf*, u_int32_t *vr_id*, char * *ifname*)

QCN get Command. smi_dcb_get_qcn_by_intf

Parameters:

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

← ***ifname*** Interface name

← ***vr_id*** Virtual router id <1-255>

← ***qcn_intf*** Structure to be filled

Returns:

0 on success, otherwise one of the following error codes

NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
SMI_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_NO_DCBG SMI_DCB_API_SET_ERR_-
INTERFACE

4.1.2.15 s_int32_t smi_dcb_get_tcg_by_interface (struct smiclient_globals * *azg*, struct smi_dcb_tcg_intf * *tcg_intf*, u_int32_t *vr_id*, char * *ifname*)

This function Get traffic-class-groups information from the ETS interface. smi_dcb_-get_tcg_by_interface

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***Structure*** to be filled

← ***ifname*** Interface name

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
SMI_DCB_API_SET_ERR_NO_NM
SMI_DCB_API_SET_ERR_INTERFACE
SMI_BRIDGE_ERR_NOTFOUND
SMI_DCB_API_SET_ERR_NO_DCBG
SMI_DCB_API_SET_ERR_DCB_INTERFACE
SMI_DCB_API_SET_ERR_INTERFACE_NO_ETS
SMI_DCB_API_SET_ERR_INTERFACE_NO_ETS

4.1.2.16 s_int32_t smi_dcb_pfc_set_advertise (struct smiclient_globals * *azg*, char * *ifname*, bool_t *advertise*)

Enables/ Disables the PFC advertise local configurations on the interface. smi_dcb_-pfc_set_advertise

Parameters:

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

← ***ifname*** Interface name

← ***advertise*** 1 - Enables PFC advertise on interface 0 - disables PFC advertise on interface

Returns:

0 on success, otherwise one of the following error codes

NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_PFC
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_PFC
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS

4.1.2.17 s_int32_t smi_dcb_pfc_set_bridge (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *bridge_name*, int *set_pfcbr*)

This function Enables/ Disables the PFC on bridge. smi_dcb_pfc_set_bridge

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***bridge_name*** Bridge name

← ***set_pfcbr*** 1 - Enable PFC on bridge 0 - Disable PFC on bridge

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_VLAN
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_PFC_EXISTS

4.1.2.18 s_int32_t smi_dcb_pfc_set_cap (struct smiclient_globals * *azg*, char * *ifname*, u_int8_t *cap*)

This function Set/Unset PFC cap for the interface. smi_dcb_pfc_set_cap

Parameters:

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

← ***ifname*** Interface name

← ***cap*** PFC cap value on the interface 0 - Unsets PFC cap

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_PFC

NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_PFC
NSM_DCB_API_SET_ERR_PFC_CAP

4.1.2.19 s_int32_t smi_dcb_pfc_set_interface (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, u_int8_t *mode*)

This function Enables/ Disables the PFC on interface. smi_dcb_pfc_set_interface

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***mode*** 2 - mode auto 1 - mode on 0 - disable

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_PFC
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_PFC
NSM_DCB_ERR_AGG_PORT
NSM_DCB_API_ERR_LINK_FLOW_CTRL_ENABLE
NSM_DCB_API_SET_ERR_INTERFACE_PFC_EXISTS
NSM_DCB_API_SET_ERR_NO_MEM

4.1.2.20 s_int32_t smi_dcb_pfc_set_lda (struct smiclient_globals * *azg*, char * *ifname*, u_int32_t *lda*)

This function Set/Unset PFC link delay allowance for the interface. smi_dcb_pfc_set_lda

Parameters:

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

← ***ifname*** Interface name

← ***lda*** PFC link delay allowance value on the interface 0 - Unsets PFC lda value to default

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_BRIDGE_ERR_NOTFOUND

NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_PFC
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_PFC
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS

4.1.2.21 s_int32_t smi_dcb_pfc_set_priority (struct smiclient_globals * *azg*, char * *ifname*, u_int8_t *priority_map*, int *setpri*)

This function Enable/Disables priority flow control for a given priority on the interface. smi_dcb_pfc_set_priority

Parameters:

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

← ***ifname*** Interface name

← ***priority_map*** Priority

← ***set_pri*** 1 - Enable PFC for a given priority on the interface 0 - Disable PFC for a given priority on the interface

Returns:

NSM_DCB_API_SET_SUCCESS on success, otherwise one of the following error codes
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_PFC
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_PFC
NSM_DCB_API_SET_ERR_EXCEED_PFC_CAP
NSM_DCB_API_SET_WRONG_PRIORITY_MAP

4.1.2.22 s_int32_t smi_dcb_pfc_set_willing (struct smiclient_globals * *azg*, char * *ifname*, bool_t *willing*)

Enables/ Disables the PFC willing configurations on the interface. smi_dcb_pfc_set_-willing

Parameters:

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

← ***ifname*** Interface name

← ***willing*** 1 - Enables PFC willing on interface 0 - disables PFC willing on interface

Returns:

0 on success, otherwise one of the following error codes

NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_PFC
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_INTERFACE_NO_PFC
NSM_DCB_API_SET_ERR_INTERFACE_NO_ETS

4.1.2.23 s_int32_t smi_dcb_qcn_create_cp (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, u_int8_t *cnpv*, u_int32_t *sample_base*, s_int32_t *weight*, u_int32_t *min_hdr_octet*, int *set_cp*)

Create/Remove cp at an interface and associated parameters with it. smi_dcb_qcn_-create_cp

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***cnpv*** Congestion Notification Priority Value

← ***sample_base*** Sample base value

← ***weight*** Weight in offset calculation

← ***min_hdr_octet*** Minimum Header Octet

← ***set_cp*** 1 - Create cp at an interface 0 - Remove cp at an interface

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_INTERFACE
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_NO_MEM
NSM_DCB_API_SET_ERR_HW
NSM_DCB_API_SET_ERR_NO_CNPV
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_QCN

4.1.2.24 s_int32_t smi_dcb_qcn_set_bridge (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *bridge_name*, u_int8_t *transmit_priority*, int *set_qcn*)

Enables/ Disables the QCN on bridge. smi_dcb_qcn_set_bridge

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***bridge_name*** Bridge name

← ***transmit_priority*** Priority

← ***set_qcn*** 1 - Enable QCN on bridge 0 - DIsable QCN on bridge

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_MEM
NSM_VLAN_ERR_BRIDGE_NOT_VLAN_AWARE
NSM_DCB_API_SET_ERR_HW
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_QCN

4.1.2.25 s_int32_t smi_dcb_qcn_set_mode (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, u_int8_t *cnpv*, u_int8_t *mode*, u_int8_t *defense_mode*, u_int8_t *alternate_priority*)

Create the defense mode per CNPV on a port level. smi_dcb_qcn_set_mode

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***ifname*** Interface name

← ***cnpv*** Congestion Notification Priority Value

in mode 1 - admin mode 2 - auto mode 3 - component mode

← ***defense_mode*** 1 - disable mode 2 - interior mode 3 - interior ready mode 4 - edge mode

← ***qcn_pri*** 1 - Set CNPV 0 - Unset CNPV

← ***alternate_priority*** - priority

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_INTERFACE
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_QCN
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_NO_CNPV

4.1.2.26 s_int32_t smi_dcb_qcn_set_mode_global (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *bridge_name*, u_int8_t *cnpv*, u_int8_t *mode*, u_int8_t *defense_mode*, u_int8_t *alternate_priority*)

Create the defense mode per CNPV on a component level. smi_dcb_qcn_set_mode_-global

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***bridge_name*** Bridge name

← ***cnpv*** Congestion Notification Priority Value

in mode 1 - admin mode 2 - auto mode 3 - component mode

← ***defense_mode*** 1 - disable mode 2 - interior mode 3 - interior ready mode 4 - edge mode

← ***qcn_pri*** 1 - Set CNPV 0 - Unset CNPV

← ***alternate_priority*** - priority

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_QCN
NSM_DCB_API_SET_ERR_NO_CNPV
NSM_DCB_API_SET_ERR_HW

4.1.2.27 s_int32_t smi_dcb_qcn_set_priority (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *bridge_name*, u_int8_t *priority*, int *set_qcn_pri*)

Set/Unset the Congestion Notification Priority Value. smi_dcb_qcn_set_priority

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***bridge_name*** Bridge name

← ***priority*** Priority value

← ***quc_pri*** 1 - Set CNPV 0 - Unset CNPV

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND

NSM_DCB_API_SET_ERR_HW
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_QCN
NSM_DCB_API_SET_ERR_QCN_ALT_PRIORITY

4.1.2.28 s_int32_t smi_dcb_set_bridge (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *bridge_name*, int *set_dcbbr*)

Enables/ Disables the DCB on bridge. smi_dcb_set_bridge

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ***bridge_name*** Bridge name <1-32>

← ***set_dcbbr*** 1-set 0-unset

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_NO_VLAN
NSM_DCB_API_SET_ERR_NO_DCBG NSM_DCB_API_SET_ERR_DCB_-EXISTS
NSM_DCB_API_SET_ERR_HW_NO_SUPPORT
NSM_DCB_API_SET_ERR_DCB_IF_INIT

4.1.2.29 s_int32_t smi_dcb_set_interface (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *ifname*, int *set_dcbif*)

Enables/ Disables the DCB on interface. smi_dcb_set_interface

Parameters:

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

← ***vr_id*** Virtual router id <1-255>

← ∗***ifname*** Interface name

← ***set_dcbif*** 1 - Enable DCBX on the specified port 0 - disable DCBX on the specified port

Returns:

LLDP_API_SUCCESS on success, otherwise one of the following error codes
LLDP_API_ERR_ONM_IF_NOT_EXIST
LLDP_API_ERR_LLDP_IF_NOT_EXIST
LLDP_API_ERR_AGG_IF

LLDP_API_ERR_DCBX_ENABLE_EXIST
LLDP_API_ERR_DCBX_ENABLE_NO_RXTX
LLDP_API_ERR_DCBX_ENABLE_NOT_EXIST

4.1.2.30 s_int32_t smi_dcb_set_tcg_bandwidth (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *if_name*, u_int8_t * *bw*)

Assign the bandwidth percentage to traffic-class-groups on the interface. smi_dcb_-set_tcg_bandwidth

Parameters:

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

← ***ifname*** Interface name

← ***vr_id*** Virtual router id <1-255>

← ***bw*** Array containing the bandwidth allocation for all the TCGs

Returns:

0 on success, otherwise one of the following error codes
NSM_DCB_API_SET_ERR_NO_NM
NSM_BRIDGE_ERR_NOTFOUND
NSM_DCB_API_SET_ERR_INTERFACE
NSM_DCB_API_SET_ERR_NO_DCBG
NSM_DCB_API_SET_ERR_NO_ETS
NSM_DCB_API_SET_ERR_DCB_INTERFACE
NSM_DCB_API_SET_ERR_WRONG_BW
NSM_DCB_API_SET_ERR_NO_TCG_BW
NSM_DCB_API_SET_ERR_NO_PRI_BW

4.2 smi_dcb_msg.h File Reference

Defines data structures used by DCB SMI APIs. `#include "pal.h"`

```
#include "nsm_message.h"
#include "smi_message.h"
#include "if.h"
#include "message.h"
#include "thread.h"
#include "network.h"
#include "log.h"
#include "tlv.h"
#include "syslog.h"
#include <sys/types.h>
#include "pal_types.h"
#include "pal_socket.h"
#include "prefix.h"
```

Data Structures

- struct smi_ets_config
- struct smi_pfc_config
- struct smi_nsm_cp_if_data
- struct smi_nsm_qcn_cnpv_data
- struct smi_nsm_qcn_data
- struct smi_appl_prio
- struct smi_app_config
- struct smi_dcb_tcg_intf
- struct smi_dcb_pfc_intf
- struct smi_scb_qcn_intf
- struct smi_dcb_app_intf
- struct dcb_msg_
- struct smi_dcb_tcg_bridge
- struct smi_dcb_appl_priority_table
- struct smi_dcb_app_by_bridge
- struct smi_dcb_pfc_stats_intf
- struct smi_dcb_pfc_stats_bridge
- struct smi_dcb_pfc_details_intf
- struct smi_dcb_pfc_details_bridge
- struct smi_dcb_qcn_cp_cpid
- struct smi_dcb_qcn_cp
- struct smi_dcb_qcn_cp_global

- struct smi_dcb_qcn_cnpv
- struct smi_dcb_qcn_cnpv_global
- struct smi_dcb_qcn_config_global

Defines

- #define **SMI_NSM_DCB_MAX_TCG_DEFAULT** 8
- #define **SMI_NSM_NUM_CNPV** 8
- #define **SMI_NSM_DCB_NUM_USER_PRI** 8
- #define **SMI_DCB_MAX_TCG** 15
- #define **SMI_DCB_MAX_TSA** 8
- #define **SMI_DCB_MAX_PRI** 8
- #define **SMI_DCB_MAX_APPL_PRIO** 15
- #define **SMI_TCG_ID_START** 0
- #define **SMI_TCG_ID_END** 7
- #define **SMI_MAX_TCG_START** 0
- #define **SMI_MAX_TCG_END** 7
- #define **SMI_TCG_PRI_START** 0
- #define **SMI_TCG_PRI_END** 255
- #define **SMI_PFC_CAP_START** 0
- #define **SMI_PFC_CAP_END** 8
- #define **SMI_APPL_PROTO_ID_START** 1
- #define **SMI_APPL_PROTO_ID_END** 1023
- #define **SMI_APPL_PRIORITY_START** 0
- #define **SMI_APPL_PRIORITY_END** 255
- #define **SMI_PFC_PRIORITY_START** 0
- #define **SMI_PFC_PRIORITY_END** 255
- #define **SMI_PFC_LDA_START** 0
- #define **SMI_PFC_LDA_END** 4294967295U
- #define **SMI_TRANSMIT_PRIORITY_START** 0
- #define **SMI_TRANSMIT_PRIORITY_END** 7
- #define **SMI_QCN_PRIORITY_START** 0
- #define **SMI_QCN_PRIORITY_END** 255
- #define **SMI_QCN_MODE_START** 1
- #define **SMI_QCN_MODE_END** 3
- #define **SMI_QCN_DEFENSE_MODE_START** 1
- #define **SMI_QCN_DEFENSE_MODE_END** 4
- #define **SMI_QCN_ALT_PRIO_START** 0
- #define **SMI_QCN_ALT_PRIO_END** 7
- #define **SMI_CNPV_START** 0
- #define **SMI_CNPV_END** 7
- #define **SMI_SAMPLE_BASE_START** 10000
- #define **SMI_SAMPLE_BASE_END** 4294967295U
- #define **SMI_WEIGHT_START** -10
- #define **SMI_WEIGHT_END** 10
- #define **SMI_MIN_HDR_OCTET_START** 0

- #define **SMI_MIN_HDR_OCTET_END** 64
- #define **SMI_PFC_MODE_START** 0
- #define **SMI_PFC_MODE_END** 1
- #define **SMI_DCB_MODE_START** 0
- #define **SMI_DCB_MODE_END** 2
- #define **SMI_DCB_ETS_DEFAULT_TCGID** 15
- #define **SMI_DCB_NUM_USER_PRI** 8
- #define **SMI_DCB_PFC_LINK_DELAY_ALLOW_DEFAULT** 0
- #define **SMI_DCB_API_SET_SUCCESS** 0
- #define **SMI_DCB_QCN_SAMPLE_BASE_INVLAID** 9999
- #define **SMI_DCB_QCN_WEIGHT_INVALID** -11
- #define **SMI_DCB_QCN_MIN_HDR_OCTATE_INVALID** 65
- #define **SMI_DCB_QCN_PRIORITY_INVALID** 8
- #define **SMI_DCB_CTYPE_VR_ID** 0
- #define **SMI_DCB_CTYPE_IFNAME** 1
- #define **SMI_DCB_CTYPE_BRIDGE_NAME** 2
- #define **SMI_DCB_CTYPE_TYPE** 3
- #define **SMI_DCB_CTYPE_FLAG** 4
- #define **SMI_DCB_CTYPE_WILLING** 5
- #define **SMI_DCB_CTYPE_ADVERTISE** 6
- #define **SMI_DCB_CTYPE_MODE** 7
- #define **SMI_DCB_CTYPE_PRIORITY_MAP** 8
- #define **SMI_DCB_CTYPE_PRIORITY** 9
- #define **SMI_DCB_CTYPE_TCGID** 10
- #define **SMI_DCB_CTYPE_MAXTCG** 11
- #define **SMI_DCB_CTYPE_TCG_BY_INTERFACE** 12
- #define **SMI_DCB_CTYPE_CAP** 13
- #define **SMI_DCB_CTYPE_LDA** 14
- #define **SMI_DCB_CTYPE_SEL** 15
- #define **SMI_DCB_CTYPE_PRI** 16
- #define **SMI_DCB_CTYPE_PRIO_MAP** 17
- #define **SMI_DCB_CTYPE_PROTO_ID** 18
- #define **SMI_DCB_CTYPE_APPL_PRIORITY_TABLE** 19
- #define **SMI_DCB_CTYPE_APP_BY_BRIDGE** 20
- #define **SMI_DCB_CTYPE_HEADER** 21
- #define **SMI_DCB_CTYPE_QCN_CONFIG_GLOBAL** 22
- #define **SMI_DCB_CTYPE_QCN_CNPV_GLOBAL** 23
- #define **SMI_DCB_CTYPE_QCN_CNPV** 24
- #define **SMI_DCB_CTYPE_CNPV** 25
- #define **SMI_DCB_CTYPE_WEIGHT** 26
- #define **SMI_DCB_CTYPE_MIN_HDR_OCTET** 27
- #define **SMI_DCB_CTYPE_DEFENSE_MODE** 28
- #define **SMI_DCB_CTYPE_ALTERNATE_PRIORITY** 29
- #define **SMI_DCB_CTYPE_TRANSMIT_PRIORITY** 30
- #define **SMI_DCB_CTYPE_EXTENDED** 31
- #define **SMI_DCB_CTYPE_QCN_BY_INTERFACE** 1

- #define **SMI_DCB_CTYPE_SAMPLE_BASE** 2
- #define **SMI_DCB_CTYPE_PFC_STATS_BY_INTERFACE** 3
- #define **SMI_DCB_CTYPE_PFC_DETAILS_BY_INTERFACE** 4
- #define **SMI_DCB_CTYPE_ASSIGN_TCG_BW** 5

Typedefs

- typedef struct dcb_msg_ **dcb_msg**

Enumerations

- enum **smi_dcb_appl_pri_mapping_type** { **SMI_DCB_MAP_SERV_NAME** = 0, **SMI_DCB_MAP_PORT_NO** = 1, **SMI_DCB_MAP_ETHERTYPE_VALUE** = 2, **SMI_DCB_MAP_ETHERTYPE_STR** = 3 }
- enum **smi_nsm_dcb_qcn_defense_mode** { **SMI_NSM_QCN_DEFENSE_MODE_DISABLE** = 1, **SMI_NSM_QCN_DEFENSE_MODE_INTERIOR** = 2, **SMI_NSM_QCN_DEFENSE_MODE_INTERIOR_READY** = 3, **SMI_NSM_QCN_DEFENSE_MODE_EDGE** = 4 }
- enum **smi_nsm_dcb_mode** { **SMI_NSM_DCB_MODE_DISABLED** = 0, **SMI_NSM_DCB_MODE_ON** = 1, **SMI_NSM_DCB_MODE_AUTO** = 2 }
- enum **smi_nsm_dcb_qcn_mode** { **SMI_NSM_DCB_QCN_MODE_ADMIN** = 1, **SMI_NSM_DCB_QCN_MODE_AUTO** = 2, **SMI_NSM_DCB_QCN_MODE_COMP** = 3 }

Functions

- int **smi_parse_dcb** (u_char **pnt, u_int16_t *size, struct smi_msg_header *header, void *arg, SMI_CALLBACK callback)
- int **smi_encode_dcb** (u_char **pnt, u_int16_t *size, dcb_msg *msg)
- int **smi_decode_dcb** (u_char **pnt, u_int16_t *size, dcb_msg *msg)

4.2.1 Detailed Description

Defines data structures used by DCB SMI APIs.

Index