ZebOS-XP 802.1x SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:21 2015

Contents

1 File Index **1**

1.1 File List 1

2 File Documentation **3**

2.1 smi_auth.h File Reference 3

2.1.1 Detailed Description 10

2.1.2 Function Documentation 10

2.1.2.1 smi_auth_debug_off 10

2.1.2.2 smi_auth_debug_on 10

2.1.2.3 smi_auth_key_transmit_set 11

2.1.2.4 smi_auth_mac_port_auth_fail_action_set 11

2.1.2.5 smi_auth_mac_port_ctrl_set 12

2.1.2.6 smi_auth_mac_port_dynamic_vlan_creation_set . . 12

2.1.2.7 smi_auth_mac_port_mac_aging_set 13

2.1.2.8 smi_auth_mac_system_ctrl_set 13

2.1.2.9 smi_auth_mac_system_ctrl_unset 14

2.1.2.10 smi_auth_port_ctrl_dir_set 14

2.1.2.11 smi_auth_port_ctrl_set 14

2.1.2.12 smi_auth_port_ctrl_unset 15

2.1.2.13 smi_auth_port_initialize_set 15

2.1.2.14 smi_auth_port_protocol_version_set 16

2.1.2.15 smi_auth_port_protocol_version_unset 16

2.1.2.16 smi_auth_port_quiet_period_set 17

2.1.2.17 smi_auth_port_quiet_period_unset 17

2.1.2.18 smi_auth_port_reauth_max_set 18

2.1.2.19 smi_auth_port_reauth_max_unset 18

2.1.2.20 smi_auth_port_reauth_period_set 18
2.1.2.21 smi_auth_port_reauth_period_unset 19
2.1.2.22 smi_auth_port_reauthenticate_set 19
2.1.2.23 smi_auth_port_reauthentication_set 20
2.1.2.24 smi_auth_port_reauthentication_unset 20
2.1.2.25 smi_auth_port_server_timeout_set 21
2.1.2.26 smi_auth_port_server_timeout_unset 21
2.1.2.27 smi_auth_port_supplicant_timeout_set 21
2.1.2.28 smi_auth_port_supplicant_timeout_unset 22
2.1.2.29 smi_auth_port_tx_period_set 22
2.1.2.30 smi_auth_port_tx_period_unset 23
2.1.2.31 smi_auth_radius_client_address_set 23
2.1.2.32 smi_auth_radius_client_address_unset 24
2.1.2.33 smi_auth_radius_server_address_set 24
2.1.2.34 smi_auth_radius_server_address_unset 24
2.1.2.35 smi_auth_radius_server_deadtime_set 25
2.1.2.36 smi_auth_radius_server_deadtime_unset 25
2.1.2.37 smi_auth_radius_server_retransmit_set 26
2.1.2.38 smi_auth_radius_server_retransmit_unset 26
2.1.2.39 smi_auth_radius_server_timeout_set 26
2.1.2.40 smi_auth_radius_server_timeout_unset 27
2.1.2.41 smi_auth_radius_shared_secret_set 27
2.1.2.42 smi_auth_radius_shared_secret_unset 27
2.1.2.43 smi_auth_system_ctrl_set 28
2.1.2.44 smi_auth_system_ctrl_unset 28
2.1.2.45 smi_show_dot1x_port 28
2.1.2.46 smi_show_dot1x_port_diagnostics 29
2.1.2.47 smi_show_dot1x_port_ssn_statistics 29
2.1.2.48 smi_show_dot1x_port_statistics 30
2.1.2.49 smi_show_dot1x_system 30

Chapter 1

File Index

1.1 File List

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

smi_auth.h (Provides APIs for managing port-based network access control to LAN devices as per IEEE802.1X specification) 3

Chapter 2

File Documentation

2.1 smi_auth.h File Reference

Provides APIs for managing port-based network access control to LAN devices as per IEEE802.1X specification. `#include "smi_client.h"`

`#include "smi_auth_msg.h"`

Functions

- int smi_auth_port_initialize_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function adds a port to 802.1x management.

- int **smi_auth_port_initialize_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_reauthenticate_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  Function to do re-authentication on specific port.

- int **smi_auth_port_reauthenticate_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_ctrl_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int portCtrl)

  This function sets the port authentication mode to authorized, unauthorized or automatic.

- s_int32_t smi_auth_mac_port_ctrl_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int macCtrl, int macMode)

  Function to enable or disable MAC authentication on a Carrier Ethernet interface.

- s_int32_t smi_auth_mac_port_auth_fail_action_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int failAction, u_int16_t vlanId)

Function to specify the required action after authentication fails for any source MAC (Media Access Control). If drop-traffic is specified, data destined to that MAC is dropped. The MAC will be added to the forwarding database in Discarded mode. If restrict-vlan is specified, the unauthorized MAC is added to a restricted VLAN. The MAC will be added to the forwarding database in Forwarding mode.

- s_int32_t smi_auth_mac_port_dynamic_vlan_creation_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, bool_t action)

  Use this function to enable or disable dynamic VLAN creation after successful MAC authentication. If the user disables dynamic VLAN creation after a successful authentication, the MAC will be added to the forwarding database with the default VLAN. If the user enables dynamic VLAN creation after a successful authentication, the MAC under authentication will be added to the VLAN ID attribute in the radius server configuration-file.

- s_int32_t **smi_auth_mac_port_dynamic_vlan_creation_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, bool_t action)
- s_int32_t smi_auth_mac_port_mac_aging_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, bool_t aging_action)

  Fuction to either enable or disable MAC aging. When enabled, a MAC entry is added to the forwarding database, with aging time equal to the bridge aging time. Otherwise, the MAC entry will not be aged out. If MAC aging is disabled, the MAC entry will not be aged out.

- s_int32_t **smi_auth_mac_port_mac_aging_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, bool_t aging_action)
- s_int32_t smi_auth_mac_system_ctrl_set (struct smiclient_globals *azg, u_int32_t vrId)

  Function to enable MAC authentication globally. If MAC authentication is not enabled, other MAC authentication related APIs throw an error when called.

- s_int32_t **smi_auth_mac_system_ctrl_set_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- s_int32_t smi_auth_mac_system_ctrl_unset (struct smiclient_globals *azg, u_int32_t vrId)

  Function to disable MAC authentication globally.

- s_int32_t **smi_auth_mac_system_ctrl_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int smi_auth_port_ctrl_dir_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int dir)

  This function sets the packet control direction to one of the following based on the direction specified.
  - Discard receive packets from supplicant.
  - Discard receive and transmit packets from supplicant.

- int smi_auth_port_ctrl_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function deletes a port from 802.1x management.

- int **smi_auth_port_ctrl_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- s_int32_t smi_auth_port_protocol_version_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int8_t version)

  Function to set the protocol version of dot1x to 1 or 2. The protocol version must be synchronized with the Xsupplicant being used in that interface.

- s_int32_t smi_auth_port_protocol_version_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  Function to unset the protocol version of dot1x to 1 or 2. The protocol version must be synchronized with the Xsupplicant being used in that interface.

- s_int32_t **smi_auth_port_protocol_version_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_quiet_period_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int quietPeriod)

  This function sets the quiet period for an interface.

- int smi_auth_port_quiet_period_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function resets the quiet period for an interface to default (60).

- int **smi_auth_port_quiet_period_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_reauth_max_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int reauthMax)

  Sets the maximum number of reauthentication attempts after which the port will be unauthorized.

- int smi_auth_port_reauth_max_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  Function to set the reauthentication value to default value '2', after 2 times of reauthentication attempts the port will be unauthorized.

- int **smi_auth_port_reauth_max_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_tx_period_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int txPeriod)

  This function sets the seconds between successive request id attempts.

- int **smi_auth_port_tx_period_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int txPeriod)
- int smi_auth_port_tx_period_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function resets the seconds between successive request id attempts for an interface to default value (30).

- int **smi_auth_port_tx_period_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_supplicant_timeout_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int suppTimeout)

  This function sets the supplicant response timeout.

- int **smi_auth_port_supplicant_timeout_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int suppTimeout)
- int smi_auth_port_supplicant_timeout_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function resets supplicant response timeout for an interface to default value (30).

- int **smi_auth_port_supplicant_timeout_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_server_timeout_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int serverTimeout)

  This function sets authentication server response timeout.

- int **smi_auth_port_server_timeout_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int serverTimeout)
- int smi_auth_port_server_timeout_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function resets authentication server response timeout for an interface to default value (30).

- int **smi_auth_port_server_timeout_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_reauth_period_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t reauthPeriod)

  This function sets the seconds between reauthorization attempts.

- int smi_auth_port_reauth_period_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function resets the seconds between reauthorization attempts for an interface to default value (3600).

- int **smi_auth_port_reauth_period_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_reauthentication_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function enables reauthentication on a port.

- int **smi_auth_port_reauthentication_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_port_reauthentication_unset (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

  This function disables reauthentication on a port.

- int **smi_auth_port_reauthentication_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int smi_auth_key_transmit_set (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int keyTransmit)

  Function to enable or disable key transmission over an Extensible Authentication Protocol (EAP) packet between the authenticator and supplicant.

- int smi_auth_system_ctrl_set (struct smiclient_globals *azg, u_int32_t vrId)

  This function enables global port authentication.

- int **smi_auth_system_ctrl_set_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int smi_auth_system_ctrl_unset (struct smiclient_globals *azg, u_int32_t vrId)

  This function disables global port authentication.

- int **smi_auth_system_ctrl_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int smi_auth_radius_client_address_set (struct smiclient_globals *azg, u_int32_t vrId, char *hostname, int port)

  This function sets the local radius address.

- int **smi_auth_radius_client_address_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *hostname, int port)
- int smi_auth_radius_client_address_unset (struct smiclient_globals *azg, u_int32_t vrId)

  This function resets the local radius address.

- int **smi_auth_radius_client_address_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int smi_auth_radius_server_address_set (struct smiclient_globals *azg, u_int32_t vrId, char *hostname, int port, u_int16_t max_retry, u_int32_t timeout, char *key)

  Sets the IP address or hostname of the remote radius server host and assign authentication and accounting destination port numbers.

- int **smi_auth_radius_server_address_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *hostname, int port, u_int16_t max_retry, u_int32_t timeout, char *key)
- int smi_auth_radius_server_address_unset (struct smiclient_globals *azg, u_int32_t vrId, char *hostname, int port)

  This function unconfigures a specified radius server.

- int **smi_auth_radius_server_address_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *hostname, int port)
- int smi_auth_radius_shared_secret_set (struct smiclient_globals *azg, u_int32_t vrId, char *key)

This function sets the radius server key.

- int **smi_auth_radius_shared_secret_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *key)
- int smi_auth_radius_shared_secret_unset (struct smiclient_globals *azg, u_int32_t vrId)

  This function resets the radius server key.

- int **smi_auth_radius_shared_secret_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int smi_auth_radius_server_retransmit_set (struct smiclient_globals *azg, u_int32_t vrId, u_int16_t retransmit)

  Function to specify the number of times the router transmits each radius request to the server before giving up.

- int smi_auth_radius_server_retransmit_unset (struct smiclient_globals *azg, u_int32_t vrId)

  Function to disable retransmission.

- int **smi_auth_radius_server_retransmit_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int smi_auth_radius_server_timeout_set (struct smiclient_globals *azg, u_int32_t vrId, u_int32_t timeout)

  Function to specify the number of seconds a router waits for a reply to a radius request before retransmitting the request.

- int smi_auth_radius_server_timeout_unset (struct smiclient_globals *azg, u_int32_t vrId)

  Function to specify number of seconds as 5, which is the default value, router waits for a reply to a radius request before retransmitting the request.

- int **smi_auth_radius_server_timeout_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int smi_auth_radius_server_deadtime_set (struct smiclient_globals *azg, u_int32_t vrId, u_int32_t deadtime)

  Function to specify the number of minutes a radius server, which is not responding to authentication requests, is passed over by requests for radius authentication.

- int smi_auth_radius_server_deadtime_unset (struct smiclient_globals *azg, u_int32_t vrId)

  Function to set deadtime to the default value of 0.

- int **smi_auth_radius_server_deadtime_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- s_int32_t smi_show_dot1x_system (struct smiclient_globals *azg, u_int32_t vrId, struct smi_dot1x_system *getDot1xSystem)

  Function that copies the dot1x port system for the display.

- s_int32_t smi_show_dot1x_port (struct smiclient_globals *azg, int start_index, int end_index, u_int32_t vrId, struct smi_interface *smiAllInterface, u_int32_t(*callback)(struct smi_interface *msgDetail))

  Function that copies the dot1x port info for the display.

- s_int32_t smi_show_dot1x_port_statistics (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, struct smi_auth_port_stats *smiAuthPortStats)

  Function that copies the dot1x port statistics info for the display.

- s_int32_t smi_show_dot1x_port_ssn_statistics (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, struct smi_auth_port_session_stats *smiAuthPortSsnStats)

  Function that copies the dot1x port session statistics info for the display.

- s_int32_t **smi_auth_show_port_by_ifname** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, struct smi_auth_port *portDetails)
- s_int32_t smi_show_dot1x_port_diagnostics (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, struct smi_diagnostics *smiDiag)

  Function that copies the dot1x port diagnostics information for display.

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

  Function disables the debug for auth.

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

  Function enables the debug for auth.

- int **smi_auth_port_ctrl_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int portCtrl)
- s_int32_t **smi_auth_mac_port_ctrl_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int macCtrl, int macMode)
- s_int32_t **smi_auth_mac_port_auth_fail_action_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int failAction, u_int16_t vlanId)
- int **smi_auth_port_ctrl_dir_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int dir)
- s_int32_t **smi_auth_port_protocol_version_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int8_t version)
- int **smi_auth_port_quiet_period_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int quietPeriod)
- int **smi_auth_port_reauth_max_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int reauthMax)
- int **smi_auth_port_reauth_period_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t reauthPeriod)
- int **smi_auth_key_transmit_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int keyTransmit)

- int **smi_auth_radius_server_retransmit_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, u_int16_t retransmit)
- int **smi_auth_radius_server_timeout_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, u_int32_t serverTimeout)
- int **smi_auth_radius_server_deadtime_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, u_int32_t deadtime)
- int **smi_auth_show_debugging** (struct smiclient_globals ∗azg, u_int32_t vrId, int ∗smiDebug)
- int **smi_auth_debug_validate** (struct smiclient_globals ∗azg, u_int32_t debug)

2.1.1 Detailed Description

Provides APIs for managing port-based network access control to LAN devices as per IEEE802.1X specification. 802.1X Port Authentication is a means to authenticate devices that attempts to attach to a LAN port. A point-to-point connection establishes once a device authenticates successfully. A connection fails if authentication fails. Port Authentication thus enhances security for systems as it protects against unauthorized access from either directly connected to wireless supplicants.

2.1.2 Function Documentation

2.1.2.1 s_int32_t smi_auth_debug_off (struct smiclient_globals ∗ *azg*, u_int32_t *debug*)

Function disables the debug for auth. smi_auth_debug_off

Parameters:

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

← ***debug*** disables the debug <0>

Returns:

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

2.1.2.2 s_int32_t smi_auth_debug_on (struct smiclient_globals ∗ *azg*, u_int32_t *debug*)

Function enables the debug for auth. smi_auth_debug_on

Parameters:

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

← ***debug*** enables the debug <1>

Returns:

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

2.1.2.3 int smi_auth_key_transmit_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *keyTransmit*)

Function to enable or disable key transmission over an Extensible Authentication Protocol (EAP) packet between the authenticator and supplicant. smi_auth_key_-transmit_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***keyTransmit*** enable (1) or disable (0)

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.4 s_int32_t smi_auth_mac_port_auth_fail_action_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *failAction*, u_int16_t *vlanId*)

Function to specify the required action after authentication fails for any source MAC (Media Access Control). If drop-traffic is specified, data destined to that MAC is dropped. The MAC will be added to the forwarding database in Discarded mode. If restrict-vlan is specified, the unauthorized MAC is added to a restricted VLAN. The MAC will be added to the forwarding database in Forwarding mode. smi_auth_mac_-port_auth_fail_action_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***failAction*** restrict-vlan (0) or drop-traffic (1)

← ***vlanId*** of the VLAN in the range of <2-4094>

Returns:

MAC_AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_MAC_NOT_ENABLED
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.5 s_int32_t smi_auth_mac_port_ctrl_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *macCtrl*, int *macMode*)

Function to enable or disable MAC authentication on a Carrier Ethernet interface. smi_auth_mac_port_ctrl_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***portCtrl*** MAC_AUTH_PORT_CTRL_ENABLE (1) or MAC_AUTH_PORT_CTRL_DISABLE (2)

← ***portMode*** MAC_AUTH_PORT_MODE_SHUTDOWN (1) or MAC_AUTH_PORT_MODE_FILTER (2)

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST
PORT_AUTH_EXIST
MAC_AUTH_NOT_EXIST

2.1.2.6 s_int32_t smi_auth_mac_port_dynamic_vlan_creation_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, bool_t *action*)

Use this function to enable or disable dynamic VLAN creation after successful MAC authentication. If the user disables dynamic VLAN creation after a successful authentication, the MAC will be added to the forwarding database with the default VLAN. If the user enables dynamic VLAN creation after a successful authentication, the MAC under authentication will be added to the VLAN ID attribute in the radius server configuration-file. smi_auth_mac_port_dynamic_vlan_creation_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***action*** enable dynamic vlan creation (1) or disable (0)

Returns:

MAC_AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_MAC_NOT_ENABLED
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.7 s_int32_t smi_auth_mac_port_mac_aging_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, bool_t *aging_action*)

Fuction to either enable or disable MAC aging. When enabled, a MAC entry is added to the forwarding database, with aging time equal to the bridge aging time. Otherwise, the MAC entry will not be aged out. If MAC aging is disabled, the MAC entry will not be aged out. smi_auth_mac_port_mac_aging_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***aging_action*** enable (1) or disable (0)

Returns:

MAC_AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_MAC_NOT_ENABLED
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.8 s_int32_t smi_auth_mac_system_ctrl_set (struct smiclient_globals * *azg*, u_int32_t *vrId*)

Function to enable MAC authentication globally. If MAC authentication is not enabled, other MAC authentication related APIs throw an error when called. smi_auth_mac_-system_ctrl_set

Parameters:

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

← ***vrId*** Virtual Router Id

Returns:

MAC_AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_MAC_NOT_ENABLED
AUTH_API_SET_ERR_VR_NOT_EXIST
PORT_AUTH_EXIST

2.1.2.9 s_int32_t smi_auth_mac_system_ctrl_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*)

Function to disable MAC authentication globally. smi_auth_mac_system_ctrl_unset

Parameters:

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

← ***vrId*** Virtual Router Id

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.10 int smi_auth_port_ctrl_dir_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *dir*)

This function sets the packet control direction to one of the following based on the direction specified.

- Discard receive packets from supplicant.

- Discard receive and transmit packets from supplicant. smi_auth_port_ctrl_dir_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***dir*** packet control direction, AUTH_CTRL_DIR_IN(1) or AUTH_CTRL_-DIR_BOTH (0)

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.11 int smi_auth_port_ctrl_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *portCtrl*)

This function sets the port authentication mode to authorized, unauthorized or automatic. smi_auth_port_ctrl_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***portCtrl*** Port Authentication mode, can be one of the following
AUTH_PORT_CTRL_FORCE_UNAUTHORIZED (1)
AUTH_PORT_CTRL_FORCE_AUTHORIZED (2) or
AUTH_PORT_CTRL_AUTO (3)

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.12 int smi_auth_port_ctrl_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function deletes a port from 802.1x management. smi_auth_port_ctrl_unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.13 int smi_auth_port_initialize_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function adds a port to 802.1x management. smi_auth_port_initialize_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.14 s_int32_t smi_auth_port_protocol_version_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, u_int8_t *version*)

Function to set the protocol version of dot1x to 1 or 2. The protocol version must be synchronized with the Xsupplicant being used in that interface. smi_auth_port_-protocol_version_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***version*** Indicates the EAP Over LAN (EAPOL) version <1-2>

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.15 s_int32_t smi_auth_port_protocol_version_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

Function to unset the protocol version of dot1x to 1 or 2. The protocol version must be synchronized with the Xsupplicant being used in that interface. smi_auth_port_-protocol_version_unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes

AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.16 int smi_auth_port_quiet_period_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *quietPeriod*)

This function sets the quiet period for an interface. smi_auth_port_quiet_period_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***quietPeriod*** Seconds between retrial of authentication <1-65535>

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.17 int smi_auth_port_quiet_period_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function resets the quiet period for an interface to default (60). smi_auth_port_-quiet_period_unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.18 int smi_auth_port_reauth_max_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *reauthMax*)

Sets the maximum number of reauthentication attempts after which the port will be unauthorized. smi_auth_port_reauth_max_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***max*** Indicates the maximum number of reauthentication attempts after which the port will be unauthorized <1-10>

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.19 int smi_auth_port_reauth_max_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

Function to set the reauthentication value to default value '2', after 2 times of reauthentication attempts the port will be unauthorized. smi_auth_port_reauth_max_unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.20 int smi_auth_port_reauth_period_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, u_int32_t *reauthPeriod*)

This function sets the seconds between reauthorization attempts. smi_auth_port_-reauth_period_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***reauthPeriod*** Seconds between reauthorization attempts <1-4294967295>

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
AUTH_API_SET_ERR_INVALID_VALUE
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.21 int smi_auth_port_reauth_period_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function resets the seconds between reauthorization attempts for an interface to default value (3600). smi_auth_port_reauth_period_unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
AUTH_API_SET_ERR_INVALID_VALUE
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.22 int smi_auth_port_reauthenticate_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

Function to do re-authentication on specific port. smi_auth_port_reauthenticate_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST

2.1.2.23 int smi_auth_port_reauthentication_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function enables reauthentication on a port. smi_auth_port_reauthentication_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
AUTH_API_SET_ERR_INVALID_VALUE
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.24 int smi_auth_port_reauthentication_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function disables reauthentication on a port. smi_auth_port_reauthentication_-unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
AUTH_API_SET_ERR_INVALID_VALUE
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.25 int smi_auth_port_server_timeout_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *serverTimeout*)

This function sets authentication server response timeout. smi_auth_port_server_-timeout_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***serverTimeout*** Supplicant response timeout

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
AUTH_API_SET_ERR_INVALID_VALUE
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.26 int smi_auth_port_server_timeout_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function resets authentication server response timeout for an interface to default value (30). smi_auth_port_server_timeout_unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.27 int smi_auth_port_supplicant_timeout_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *suppTimeout*)

This function sets the supplicant response timeout. smi_auth_port_supplicant_-timeout_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***suppTimeout*** Supplicant response timeout

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.28 int smi_auth_port_supplicant_timeout_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function resets supplicant response timeout for an interface to default value (30). smi_auth_port_supplicant_timeout_unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.29 int smi_auth_port_tx_period_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, int *txPeriod*)

This function sets the seconds between successive request id attempts. smi_auth_port_-tx_period_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

← ***txPeriod*** Seconds between successive request ID attempts

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.30 int smi_auth_port_tx_period_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*)

This function resets the seconds between successive request id attempts for an interface to default value (30). smi_auth_port_tx_period_unset

Parameters:

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

← ***vrId*** Virtual Router Id

← ***ifName*** Name of the port

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
MAC_AUTH_EXIST
AUTH_API_SET_ERR_PORT_NOT_EXIST

2.1.2.31 int smi_auth_radius_client_address_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *hostname*, int *port*)

This function sets the local radius address. smi_auth_radius_client_address_set

Parameters:

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

← ***vrId*** Virtual Router Id

← ***hostname*** Radius client in dotted decimal ip address string format or hostname format

← ***port*** RADIUS client port number

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.32 int smi_auth_radius_client_address_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*)

This function resets the local radius address. smi_auth_radius_client_address_unset

Parameters:

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

← ***vrId*** Virtual Router Id

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.33 int smi_auth_radius_server_address_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *hostname*, int *port*, u_int16_t *max_retry*, u_int32_t *timeout*, char * *key*)

Sets the IP address or hostname of the remote radius server host and assign authentication and accounting destination port numbers. smi_auth_radius_server_address_set

Parameters:

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

← ***vrId*** Virtual router ID

← ***hostname*** RADIUS server dotted IP address or hostname format

← ***port*** RADIUS server port number

← ***max_retry*** RADIUS server maximum retransmit attempts

← ***timeout*** RADIUS server timeout value

← ***key*** Shared secret among radius server and 802.1X client

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST
AUTH_API_SET_ERR_HOST_NOT_CONFIGURED

2.1.2.34 int smi_auth_radius_server_address_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *hostname*, int *port*)

This function unconfigures a specified radius server. smi_auth_radius_server_address_unset

Parameters:

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

← ***vrId*** Virtual router ID

← ***hostname*** RADIUS server dotted IP address or hostname format

← ***port*** RADIUS server port number

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.35 int smi_auth_radius_server_deadtime_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, u_int32_t *deadtime*)

Function to specify the number of minutes a radius server, which is not responding to authentication requests, is passed over by requests for radius authentication. smi_auth_radius_server_deadtime_set

Parameters:

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

← ***vrId*** Virtual router ID

← ***deadtime*** Length of time (in minutes) that a radius server is skipped over by transaction requests <1-1440>

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERROR

2.1.2.36 int smi_auth_radius_server_deadtime_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*)

Function to set deadtime to the default value of 0. smi_auth_radius_server_deadtime_unset

Parameters:

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

← ***vrId*** Virtual router ID

Returns:

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

2.1.2.37 int smi_auth_radius_server_retransmit_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, u_int16_t *retransmit*)

Function to specify the number of times the router transmits each radius request to the server before giving up. smi_auth_radius_server_retransmit_set

Parameters:

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

← ***vrId*** Virtual router ID

← ***retransmit*** The retransmit value <1-100>

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERROR

2.1.2.38 int smi_auth_radius_server_retransmit_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*)

Function to disable retransmission. smi_auth_radius_server_retransmit_unset

Parameters:

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

← ***vrId*** Virtual router ID

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERROR

2.1.2.39 int smi_auth_radius_server_timeout_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, u_int32_t *timeout*)

Function to specify the number of seconds a router waits for a reply to a radius request before retransmitting the request. smi_auth_radius_server_timeout_set

Parameters:

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

← ***vrId*** Virtual router ID

← ***timeout*** The number of seconds for a router to wait for a server host to reply before timing out <1-1000>

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERROR

2.1.2.40 int smi_auth_radius_server_timeout_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*)

Function to specify number of seconds as 5, which is the default value, router waits for a reply to a radius request before retransmitting the request. smi_auth_radius_server_-timeout_unset

Parameters:

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

← ***vrd*** Virtual router ID

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERROR

2.1.2.41 int smi_auth_radius_shared_secret_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *key*)

This function sets the radius server key. smi_auth_radius_shared_secret_set

Parameters:

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

← ***vrId*** Virtual router ID

← ***key*** Shared secret among radius server and 802.1X client.

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.42 int smi_auth_radius_shared_secret_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*)

This function resets the radius server key. smi_auth_radius_shared_secret_unset

Parameters:

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

← ***vrId*** Virtual router ID

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.43 int smi_auth_system_ctrl_set (struct smiclient_globals * *azg*, u_int32_t *vrId*)

This function enables global port authentication. smi_auth_system_ctrl_set

Parameters:

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

← ***vrId*** Virtual Router Id

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.44 int smi_auth_system_ctrl_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*)

This function disables global port authentication. smi_auth_system_ctrl_unset

Parameters:

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

← ***vrId*** Virtual Router Id

Returns:

AUTH_API_SET_SUCCESS when the function succeeds, otherwise one of the following error codes
AUTH_API_SET_ERR_VR_NOT_EXIST

2.1.2.45 s_int32_t smi_show_dot1x_port (struct smiclient_globals * *azg*, int *start_index*, int *end_index*, u_int32_t *vrId*, struct smi_interface * *smiAllInterface*, u_int32_t(*)(struct smi_interface *msgDetail) *callback*)

Function that copies the dot1x port info for the display. smi_show_dot1x_port

Parameters:

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

← ***vrId*** Virtual router ID

← ***start_index*** index from where the copy has to happen

← ***end_index*** index to where the copy should last

→ ***smiInterface*** which has the final copied components

→ ***msgDetail*** For user, to specify wat function to call upon reception of the data from the server

Returns:

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

2.1.2.46 s_int32_t smi_show_dot1x_port_diagnostics (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, struct smi_diagnostics * *smiDiag*)

Function that copies the dot1x port diagnostics information for display. smi_show_-dot1x_port_diagnostics

Parameters:

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

← ***vrId*** Virtual router ID

← ***ifName*** interface name

→ ***smiDiag*** which has the final copied components

Returns:

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

2.1.2.47 s_int32_t smi_show_dot1x_port_ssn_statistics (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, struct smi_auth_port_session_stats * *smiAuthPortSsnStats*)

Function that copies the dot1x port session statistics info for the display. smi_show_-dot1x_port_ssn_statistics

Parameters:

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

← ***vrId*** Virtual router ID

← ***ifName*** interface name

→ ***smiAuthPortSsnStats*** which has the final copied components

Returns:

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

2.1.2.48 s_int32_t smi_show_dot1x_port_statistics (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ifName*, struct smi_auth_port_stats * *smiAuthPortStats*)

Function that copies the dot1x port statistics info for the display. smi_show_dot1x_-port_statistics

Parameters:

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

← ***vrId*** Virtual router ID

← ***ifName*** interface name

→ ***smiAuthPortStats*** which has the final copied components

Returns:

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

2.1.2.49 s_int32_t smi_show_dot1x_system (struct smiclient_globals * *azg*, u_int32_t *vrId*, struct smi_dot1x_system * *getDot1xSystem*)

Function that copies the dot1x port system for the display. smi_show_dot1x_system

Parameters:

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

← ***vrId*** Virtual router ID

→ ***getDot1xSystem*** which has the final copied components

Returns:

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

Index

smi_auth.h, 3
- smi_auth_debug_off, 10
- smi_auth_debug_on, 10
- smi_auth_key_transmit_set, 11
- smi_auth_mac_port_auth_fail_-action_set, 11
- smi_auth_mac_port_ctrl_set, 12
- smi_auth_mac_port_dynamic_-vlan_creation_set, 12
- smi_auth_mac_port_mac_aging_set, 13
- smi_auth_mac_system_ctrl_set, 13
- smi_auth_mac_system_ctrl_unset, 13
- smi_auth_port_ctrl_dir_set, 14
- smi_auth_port_ctrl_set, 14
- smi_auth_port_ctrl_unset, 15
- smi_auth_port_initialize_set, 15
- smi_auth_port_protocol_version_-set, 16
- smi_auth_port_protocol_version_-unset, 16
- smi_auth_port_quiet_period_set, 17
- smi_auth_port_quiet_period_unset, 17
- smi_auth_port_reauth_max_set, 17
- smi_auth_port_reauth_max_unset, 18
- smi_auth_port_reauth_period_set, 18
- smi_auth_port_reauth_period_unset, 19
- smi_auth_port_reauthenticate_set, 19
- smi_auth_port_reauthentication_set, 20
- smi_auth_port_reauthentication_-unset, 20
- smi_auth_port_server_timeout_set, 20
- smi_auth_port_server_timeout_-unset, 21
- smi_auth_port_supplicant_-timeout_set, 21
- smi_auth_port_supplicant_-timeout_unset, 22
- smi_auth_port_tx_period_set, 22
- smi_auth_port_tx_period_unset, 23
- smi_auth_radius_client_address_set, 23
- smi_auth_radius_client_address_-unset, 23
- smi_auth_radius_server_address_-set, 24
- smi_auth_radius_server_address_-unset, 24
- smi_auth_radius_server_deadtime_-set, 25
- smi_auth_radius_server_deadtime_-unset, 25
- smi_auth_radius_server_-retransmit_set, 25
- smi_auth_radius_server_-retransmit_unset, 26
- smi_auth_radius_server_timeout_-set, 26
- smi_auth_radius_server_timeout_-unset, 27
- smi_auth_radius_shared_secret_set, 27
- smi_auth_radius_shared_secret_-unset, 27
- smi_auth_system_ctrl_set, 28
- smi_auth_system_ctrl_unset, 28
- smi_show_dot1x_port, 28
- smi_show_dot1x_port_diagnostics, 29
- smi_show_dot1x_port_ssn_-statistics, 29
- smi_show_dot1x_port_statistics, 30
- smi_show_dot1x_system, 30

smi_auth_debug_off
smi_auth.h, 10
smi_auth_debug_on
smi_auth.h, 10
smi_auth_key_transmit_set
smi_auth.h, 11
smi_auth_mac_port_auth_fail_action_set
smi_auth.h, 11
smi_auth_mac_port_ctrl_set
smi_auth.h, 12
smi_auth_mac_port_dynamic_vlan_-
creation_set
smi_auth.h, 12
smi_auth_mac_port_mac_aging_set
smi_auth.h, 13
smi_auth_mac_system_ctrl_set
smi_auth.h, 13
smi_auth_mac_system_ctrl_unset
smi_auth.h, 13
smi_auth_port_ctrl_dir_set
smi_auth.h, 14
smi_auth_port_ctrl_set
smi_auth.h, 14
smi_auth_port_ctrl_unset
smi_auth.h, 15
smi_auth_port_initialize_set
smi_auth.h, 15
smi_auth_port_protocol_version_set
smi_auth.h, 16
smi_auth_port_protocol_version_unset
smi_auth.h, 16
smi_auth_port_quiet_period_set
smi_auth.h, 17
smi_auth_port_quiet_period_unset
smi_auth.h, 17
smi_auth_port_reauth_max_set
smi_auth.h, 17
smi_auth_port_reauth_max_unset
smi_auth.h, 18
smi_auth_port_reauth_period_set
smi_auth.h, 18
smi_auth_port_reauth_period_unset
smi_auth.h, 19
smi_auth_port_reauthenticate_set
smi_auth.h, 19
smi_auth_port_reauthentication_set
smi_auth.h, 20
smi_auth_port_reauthentication_unset
smi_auth.h, 20
smi_auth_port_server_timeout_set
smi_auth.h, 20
smi_auth_port_server_timeout_unset
smi_auth.h, 21
smi_auth_port_supplicant_timeout_set
smi_auth.h, 21
smi_auth_port_supplicant_timeout_unset
smi_auth.h, 22
smi_auth_port_tx_period_set
smi_auth.h, 22
smi_auth_port_tx_period_unset
smi_auth.h, 23
smi_auth_radius_client_address_set
smi_auth.h, 23
smi_auth_radius_client_address_unset
smi_auth.h, 23
smi_auth_radius_server_address_set
smi_auth.h, 24
smi_auth_radius_server_address_unset
smi_auth.h, 24
smi_auth_radius_server_deadtime_set
smi_auth.h, 25
smi_auth_radius_server_deadtime_unset
smi_auth.h, 25
smi_auth_radius_server_retransmit_set
smi_auth.h, 25
smi_auth_radius_server_retransmit_unset
smi_auth.h, 26
smi_auth_radius_server_timeout_set
smi_auth.h, 26
smi_auth_radius_server_timeout_unset
smi_auth.h, 27
smi_auth_radius_shared_secret_set
smi_auth.h, 27
smi_auth_radius_shared_secret_unset
smi_auth.h, 27
smi_auth_system_ctrl_set
smi_auth.h, 28
smi_auth_system_ctrl_unset
smi_auth.h, 28
smi_show_dot1x_port
smi_auth.h, 28
smi_show_dot1x_port_diagnostics
smi_auth.h, 29
smi_show_dot1x_port_ssn_statistics
smi_auth.h, 29
smi_show_dot1x_port_statistics
smi_auth.h, 30
smi_show_dot1x_system
smi_auth.h, 30