ZebOS-XP RIP SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:54 2015

Contents

Chapter 1

File Index

1.1 File List

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

Chapter 2

File Documentation

2.1 smi_rip.h File Reference

Provides API for managing RIP. `#include "smi_client.h"`
`#include "smi_rip_msg.h"`

Defines

- #define **SMI_RIP_DEFAULT_INSTANCE** 0
- #define **SMI_RIP_DEFAULT_METRIC_MIN** 1
- #define **SMI_RIP_DEFAULT_METRIC_MAX** 16
- #define **SMI_RIP_ROUTE_PMAX_MIN** 1
- #define **SMI_RIP_ROUTE_PMAX_MAX** 65535
- #define **SMI_RIP_ROUTE_THRESHOLD_MIN** 1
- #define **SMI_RIP_ROUTE_THRESHOLD_MAX** 100
- #define **SMI_RIP_RECV_BUF_SIZE_MIN** 8192
- #define **SMI_RIP_RECV_BUF_SIZE_MAX** 2147483647
- #define **SMI_RIP_GRACE_PERIOD_MIN** 1
- #define **SMI_RIP_GRACE_PERIOD_MAX** 65535
- #define **SMI_RIP_DISTANCE_SOURCE_MIN** 1
- #define **SMI_RIP_DISTANCE_SOURCE_MAX** 255
- #define **SMI_RIP_INSTANCE_MAX** 63
- #define **SMI_RIP_RMAP_NAME_MIN** 1
- #define **SMI_RIP_RMAP_NAME_MAX** 2

Functions

- int smi_rip_if_receive_packet_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function enables the interface to receive RIP packets. This is the default setting.

- int smi_rip_if_receive_packet_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function disables receiving RIP packets on the specified interface.

- int smi_rip_if_send_packet_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function enables the interface to send RIP packets on interface ifName.

- int smi_rip_if_send_packet_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function disables the interface to send RIP packets on interface ifName.

- int smi_rip_if_receive_version_type_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, int ifRecvVersionType)

  This function enables receiving the specified version of RIP packets (version 1 or version 2) or receiving both versions of RIP packets (version 1 and version 2).

- int smi_rip_if_receive_version_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function reset the receive version to the node version.

- int smi_rip_if_send_version_type_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, int ifSendVersionType)

  This function sets sending RIP packets on an interface using version control (version type).

- int smi_rip_if_send_version_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function sets the sending version to the version of the RIP node, the default value.

- int smi_rip_if_auth_mode_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, char ∗authMode)

  This function sets the authentication mode and specifies the type of authentication mode used for RIP v2 packets.

- int smi_rip_if_auth_mode_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function resets the authentication mode.If the authentication string or authKey chain exist, the mode is set to plain text authentication.If no mode is specified, the mode is set to no authentication.

- int smi_rip_if_auth_str_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, char ∗authString)

  This function sets the authentication string or passacListName used by a authKey.

- int smi_rip_if_auth_str_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

This function disables feature to specify the authentication string or passacListName used by a authKey.

- int smi_rip_if_auth_key_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, char ∗authKey)

  This function specifies the RIP authentication authKey chain string.It enables RIPv2 authentication on an interface and specify the name of the authKey chain to be used.

- int smi_rip_if_auth_key_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function clears the authKey chain authentication and authentication is disabled.

- int smi_rip_if_split_horizon_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function enables RIP split-horizon behavior. This command helps avoid including routes in updates sent to the same gateway from which they were learned.

- int smi_rip_if_split_horizon_poisoned_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function enables RIP split-horizon poisoned reverse behavior.

- int smi_rip_if_split_horizon_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

  This function disables split horizon behavior. The default configuration is split-horizon poisoned.

- int smi_rip_instance_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This function establishes an instance of the RIP router and the RIP routing process is enabled.

- int smi_rip_instance_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This function removes an instance of the RIP router and disables the RIP routing process.

- int smi_rip_version_type_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, int version)

  Use this command to specify a RIP version used globally by the router. RIP can be run in version 1 as well as version 2 mode. Version 2 has more features than version 1 including authentication. Once the rip version is set, rip packets of that version will be received and sent on all the rip-enabled interfaces.

- int smi_rip_version_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This function resets the RIP version to default veriosn which is 2 and globally used by the router.

- int smi_rip_enable_network_add (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)

  This function enables RIP routing on the specified network. It specifies a network as one that runs Routing Information Protocol (RIP).

- int smi_rip_enable_network_delete (struct smiclient_globals ∗azg, u_int32_-t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)

  This function disables RIP routing on the specified network.

- int smi_rip_enable_if_add (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗ifName)

  This function enables RIP routing on the specified interface. If a network is not specified, the interfaces in that network will not be advertised in any RIP update.

- int smi_rip_enable_nbr_add (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr)

  This function enables RIP routing on the specified neighbor.RIP updates are sent to the unicast IP address (es) specified in the neighbor statement.

- int smi_rip_enable_nbr_delete (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr)

  This function disables RIP routing on the neighbor.

- int smi_rip_passive_if_add (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗ifName)

  This function suppresses RIP updates and blocks RIP broadcast on the interface.

- int smi_rip_passive_if_delete (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗ifName)

  This function disables blocking RIP broadcasts on the interface.

- int smi_rip_route_add (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)

  This function configures a static route for advertisement through RIP explicitly. An ideal configuration includes a static route that is redistribute via redistribute static inside a routing process.

- int smi_rip_route_delete (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)

  This function removes the specified static route.

- int smi_rip_route_default_add (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This function generates a default route into the Routing Information Protocol (RIP).

- int smi_rip_route_default_delete (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

This function disables the configuration of a default route into the Routing Information Protocol (RIP).

- int smi_rip_offset_list_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗acListName, char ∗directStr, int metric, char ∗ifName)

  This function adds an offset to in and out metrics to routes learned through RIP: specifies the offset value that is added to the routing metric.

- int smi_rip_offset_list_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗acListName, char ∗directStr, int metric, char ∗ifName)

  This function removes the offlist.

- int smi_rip_default_metric_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, int defaultMetric)

  This function sets the routing protocol to use the specified metric value for all redistributed routes. The specified default metric will be used by all routes that are redistributed.

- int smi_rip_default_metric_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This function resets the metrics assigned to redistributed routes to the default setting: 1.

- int smi_rip_redistribute_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType)

  This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIP. It also redistributes kernel, connected and static into the RIP information.

- int smi_rip_redistribute_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType)

  This function resets the learned route.

- int smi_rip_redistribute_metric_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType, int metric)

  This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIP. It also redistributes kernel, connected and static into the RIP. The configured metric is set to the redistributing routes.

- int smi_rip_redistribute_rmap_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType, char ∗name)

  This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIP. It also redistributes kernel, connected and static into the RIP. Route redistribution is set per route map.

- int smi_rip_redistribute_metric_rmap_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType, int metric, char ∗name)

  This function specifies the metric of the route map.

- int smi_rip_timers_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, u_int32_t updateTimer, u_int32_t timeoutTimer, u_int32_t garbageTimer)

  This function sets the specified time per RIP timer: update timer, timeout timer, garbage timer.

- int smi_rip_timers_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This call resets the three timers to the default values: update timer - 30 seconds, Timeout timer - 180 seconds, Garbage Timer- 120 seconds.

- int smi_rip_distance_set_default (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗distanceStr)

  This function sets the administrative distance to the specified value.

- int smi_rip_distance_unset_default (struct smiclient_globals ∗azg, u_int32_-t vrId, int instance)

  This function resets the administrative distance configuration to the default value: 120.

- int smi_rip_distance_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗distanceStr, struct pal_in4_addr ∗addr, int prefixLength, char ∗acListName)

  This function specifies the administrative distance for the route calculation. The distance is a feature used by the routers to select the path when there are two or more different routes to the same destination from two different routing protocols.

- int smi_rip_distance_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)

  This function deletes the administrative distance that was configured for the route calculation.

- int smi_rip_max_route_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗maxPrefixString, char ∗thresholdStr)

  This function sets the maximum number of RIP routes that can be stored in the routing table. It also sets the percentage of maximum routes to generate a warning (default maximum 75%).

- int smi_rip_max_route_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This function sets the threshold value to the default threshold percentage of maximum-prefix checking.The default percentage is 75%.

- int smi_rip_route_type_delete (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType)

  This function clears specific data from the RIP routing table.

- int smi_rip_recvbuf_size_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, u_int32_t recvBufsize)

This function specifies the size of the RIP UDP buffer.

- int smi_rip_recvbuf_size_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This function resets the size of the RIP UDP buffer to the default value: (1024∗192).

- int **smi_rip_cisco_metric_behavior_set** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, u_char metricType)
- int smi_rip_cisco_metric_behavior_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)

  This function unsets updating the metric consistent with Cisco and reverts to the default metric type.

- int smi_rip2_get_global_route_changes (struct smiclient_globals ∗azg, int instance, int ∗num_of_global_routechanges)

  This function returns the number of responses sent to RIP queries from other systems.

- int smi_rip2_get_global_queries (struct smiclient_globals ∗azg, int instance, int ∗num_of_responses)

  This function returns the number of route changes made to the IP Route Database by RIP.

- int smi_rip2_get_if_stat_addr (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, struct pal_in4_addr ∗outaddr)

  This function returns the IP address of this system on the indicated subnet.

- int smi_rip2_get_if_stat_rcv_bad_packets (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗discarded_packets_count)

  This function returns the number of RIP packets received by the RIP process that were discarded .

- int smi_rip2_get_if_stat_rcv_bad_routes (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗bad_routes_count)

  This function returns the number of routes in valid RIP packets that were ignored for any reason e.g. unknown address family.

- int smi_rip2_get_if_stat_sent_updates (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗stat_sent_updates_count)

  This function returns the number of triggered RIP updates actually sent on this interface.This does not include full updates containing new information.

- int smi_rip2_get_if_stat_status (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗ifStatStatus)

  This function returns the status of the interface.

- int smi_rip2_set_if_stat_status (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifStatStatus)

  This function sets status of the specified interface.

- int smi_rip2_get_if_conf_address (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, struct pal_in4_addr ∗outaddr)

  This function returns the IP address of this system on the indicated subnet.

- int smi_rip2_get_if_conf_domain (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗if_conf_domain)

  This function returns the value inserted into the Routing Domain field of all RIP packets sent on this interface.

- int smi_rip2_get_if_conf_auth_type (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗ifConfAuthType)

  This function returns the type of authentication used on this interface.

- int smi_rip2_get_if_conf_auth_key (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, char ∗ifConfAuthKey)

  This function returns the value to be used as the authentication authKey whenever the corresponding instance of rip2_get_ifConfAuthType has a value other than noAuthentication.

- int smi_rip2_get_if_conf_send (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗ifConfSend)

  This function returns what the router sends on this interface (typically updates).

- int smi_rip2_get_if_conf_receive (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗ifConfReceive)

  This function returns the version of RIP updates that is to be accepted.

- int smi_rip2_get_if_conf_default_metric (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗ifConfDefaultMetric)

  This function returns the variable that indicates the metric that is to be used for the default route entry in RIP updates originated on this interface.

- int smi_rip2_get_if_conf_status (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗ifConfStatus)

  This function returns the status of the interface.

- int smi_rip2_get_if_conf_src_address (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, struct pal_in4_addr ∗outaddr)

  This function returns the IP address the system will use as a source address on this interface.

- int smi_rip2_set_if_conf_auth_type (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifConfAuthType)

  This function sets authentication type used on this interface.

- int smi_rip2_set_if_conf_auth_key (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, char ∗authKey)

This function sets the value to be used as the authentication authKey of the corresponding instance of RIP.

- int smi_rip2_set_if_conf_send (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifConfSend)

  This function sets the RIP version to be sent in the control packet.

- int smi_rip2_set_if_conf_receive (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifConfReceive)

  This function sets the version of RIP updates to be accepted.

- int smi_rip2_set_if_conf_status (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifConfStatus)

  This function sets status of the interface.

- int smi_rip2_get_peer_address (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, struct pal_in4_addr ∗outaddr)

  This function returns the IP address that the peer is using as its source address.

- int smi_rip2_get_peer_domain (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗peerDomain)

  This function returns the value in the Routing Domain field in RIP packets received from the peer. As domain support is deprecated, the value returned is 0.

- int smi_rip2_get_peer_last_update (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗peerLastUpdate)

  This function returns the value of sysUptime when the most recent RIP Update is received from this system.

- int smi_rip2_get_peer_version (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗peerVersion)

  This function returns the RIP version number in the header of the last RIP packet received.

- int smi_rip2_get_peer_rcv_bad_packets (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗peerRcvBadPacketsCount)

  This function returns the number of RIP response packets from this peer discarded as invalid.

- int smi_rip2_get_peer_rcv_bad_routes (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ∗peerRcvBadRoutesCount)

  This function returns the number of routes from this peer that were ignored because the entry format was invalid.

- int smi_rip_show_Ifname (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, int start_index, int end_index, struct list ∗ifNameList, int(∗funpointer)(struct list ∗ifNameList))

  This function shows the interface rip configurations.

- int smi_rip_show_db (struct smiclient_globals ∗azg, u_int32_t vrId, int start_index, int end_index, struct list ∗DbList, int(∗funpointer)(struct list ∗DbList))

  This function shows the rip database .

- int **smi_rip_show_db_vrf** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗vrfName, int start_index, int end_index, struct list ∗DbList, int(∗funpointer)(struct list ∗DbList))
- int smi_rip_debug (struct smiclient_globals ∗azg, int vrId, int debug)

  Use this function to specify the options for the displayed debugging information for RIP events, RIP packets and RIP NSM.

- int smi_rip_no_debug (struct smiclient_globals ∗azg, int vrId, int debug)

  Use this function to disable specific debugging.

- int smi_rip_default_information_originiate_set (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗origin, char ∗rmapName)

  This function is to unset address-family.

- int smi_rip_default_information_delete_unset (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗origin, char ∗rmapName)

  This function is to unset address-family.

- int smi_rip_address_family_set (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗vrfName)

  This function is to set address-family.

- int smi_rip_address_family_unset (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗vrfName)

  This function is to unset address-family.

- int smi_rip_distribute_list_set_sdkapi (struct smiclient_globals ∗azg, char ∗acListName, int type, char ∗ifName, u_int32_t vrId, int instance)

  This function sets distribution for specified interface.

- int smi_rip_distribute_list_unset_sdkapi (struct smiclient_globals ∗azg, char ∗acListName, int type, char ∗ifName, u_int32_t vrId, int instance)

  This function unsets distribution for specified interface.

- int smi_rip_distribute_list_prefix_set_sdkapi (struct smiclient_globals ∗azg, char ∗acListName, int type, char ∗ifName, u_int32_t vrId, int instance)

  This function disables RIP routing on the specified interface.

- int smi_rip_distribute_prefix_list_unset_sdkapi (struct smiclient_globals ∗azg, char ∗acListName, int type, char ∗ifName, u_int32_t vrId, int instance)

  This function unsets distribution for specified interface.

- int smi_show_debug_rip (struct smiclient_globals ∗azg, int ∗debug)

    This function shows the debugging.

- int smi_rip_enable_if_delete (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗ifName)
- int smi_show_ip_rip_statistics_if (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, int start_index, int end_index, struct list ∗ifNameList, int(∗funpointer)(struct list ∗ifNameList))

    This function shows the rip interface traffic statistics .

- int smi_rip_show_protocol_info_vrf (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗vrfName, int start_index, int end_index, struct list ∗DbList, int(∗funpointer)(struct list ∗DbList))

    This function shows the currently rip process detailed information .

- int smi_rip_show_Ifname_vrf (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, char ∗vrfName, int start_index, int end_index, struct list ∗ifNameList, int(∗funpointer)(struct list ∗ifNameList))

    This function shows the interface rip configurations.

- int **smi_rip_recvbuf_size_get** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, u_int32_t ∗recvBufsize)
- int **smi_rip_clear_stat** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_receive_packet_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_receive_packet_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_send_packet_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_send_packet_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_receive_version_type_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, int ifRecvVersionType)
- int **smi_rip_if_receive_version_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_send_version_type_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, int ifSendVersionType)
- int **smi_rip_if_send_version_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_auth_mode_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, char ∗authMode)
- int **smi_rip_if_auth_mode_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_auth_str_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, char ∗authString)
- int **smi_rip_if_auth_str_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)

- int **smi_rip_if_auth_key_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName, char ∗authKey)
- int **smi_rip_if_auth_key_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_split_horizon_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_split_horizon_poisoned_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_if_split_horizon_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_instance_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_instance_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_version_type_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, int version)
- int **smi_rip_version_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_enable_network_add_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)
- int **smi_rip_enable_network_delete_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)
- int **smi_rip_enable_if_add_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗ifName)
- int **smi_rip_enable_nbr_add_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr)
- int **smi_rip_enable_nbr_delete_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr)
- int **smi_rip_passive_if_add_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗ifName)
- int **smi_rip_passive_if_delete_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗ifName)
- int **smi_rip_route_add_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)
- int **smi_rip_route_delete_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)
- int **smi_rip_route_default_add_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_route_default_delete_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_offset_list_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗acListName, char ∗directStr, int metric, char ∗ifName)
- int **smi_rip_offset_list_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗acListName, char ∗directStr, int metric, char ∗ifName)
- int **smi_rip_default_metric_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, int metric)

- int **smi_rip_default_metric_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_redistribute_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType)
- int **smi_rip_redistribute_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType)
- int **smi_rip_redistribute_metric_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType, int metric)
- int **smi_rip_redistribute_rmap_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType, char ∗name)
- int **smi_rip_redistribute_metric_rmap_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType, int metric, char ∗name)
- int **smi_rip_timers_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, u_int32_t update, u_int32_t timeout, u_int32_t garbage)
- int **smi_rip_timers_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_distance_set_default_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗distanceStr)
- int **smi_rip_distance_unset_default_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_distance_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗distanceStr, struct pal_in4_addr ∗addr, int prefixLength, char ∗acListName)
- int **smi_rip_distance_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, struct pal_in4_addr ∗addr, int prefixLength)
- int **smi_rip_max_route_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗maxPrefixString, char ∗thresholdStr)
- int **smi_rip_max_route_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_route_type_delete_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗routeType)
- int **smi_rip_recvbuf_size_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, u_int32_t recvBufsize)
- int **smi_rip_recvbuf_size_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip_cisco_metric_behavior_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, u_char metricType)
- int **smi_rip_cisco_metric_behavior_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance)
- int **smi_rip2_set_if_stat_status_validate** (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifStatStatus)
- int **smi_rip2_set_if_conf_auth_type_validate** (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifConfAuthType)
- int **smi_rip2_set_if_conf_auth_key_validate** (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, char ∗authKey)
- int **smi_rip2_set_if_conf_send_validate** (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifConfSend)

- int **smi_rip2_set_if_conf_receive_validate** (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifConfReceive)
- int **smi_rip2_set_if_conf_status_validate** (struct smiclient_globals ∗azg, int instance, struct pal_in4_addr ∗addr, int ifConfStatus)
- int **smi_rip_debug_validate** (struct smiclient_globals ∗azg, int vrId, int debug)
- int **smi_rip_no_debug_validate** (struct smiclient_globals ∗azg, int vrId, int debug)
- int **smi_rip_distribute_list_set_sdkapi_validate** (struct smiclient_globals ∗azg, char ∗acListName, int type, char ∗ifName, u_int32_t vrId, int instance)
- int **smi_rip_distribute_list_unset_sdkapi_validate** (struct smiclient_globals ∗azg, char ∗acListName, int type, char ∗ifName, u_int32_t vrId, int instance)
- int **smi_rip_distribute_list_prefix_set_sdkapi_validate** (struct smiclient_globals ∗azg, char ∗acListName, int type, char ∗ifName, u_int32_t vrId, int instance)
- int **smi_rip_distribute_prefix_list_unset_sdkapi_validate** (struct smiclient_globals ∗azg, char ∗acListName, int type, char ∗ifName, u_int32_t vrId, int instance)
- int **smi_rip_clear_stat_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, char ∗ifName)
- int **smi_rip_default_information_originiate_set_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗origin, char ∗rmapName)
- int **smi_rip_default_information_delete_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗origin, char ∗rmapName)
- int **smi_rip_enable_if_delete_validate** (struct smiclient_globals ∗azg, u_int32_t vrId, int instance, char ∗ifName)

2.1.1 Detailed Description

Provides API for managing RIP. The API provided in this file forms the basis of ZebOS RIP management. These APIs are used by various north bound management interfaces like CLI, SNMP and SMI

2.1.2 Function Documentation

2.1.2.1 int smi_rip2_get_global_queries (struct smiclient_globals ∗ *azg*, int *instance*, int ∗ *num_of_responses*)

This function returns the number of route changes made to the IP Route Database by RIP. smi_rip2_get_global_queries

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

→ ***num_of_responses*** The number of responses

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.2 int smi_rip2_get_global_route_changes (struct smiclient_globals ∗ *azg*, int *instance*, int ∗ *num_of_global_routechanges*)

This function returns the number of responses sent to RIP queries from other systems. smi_rip2_get_global_route_changes

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

→ ***num_of_global_routechanges*** The number of route changes

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.3 int smi_rip2_get_if_conf_address (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, struct pal_in4_addr ∗ *outaddr*)

This function returns the IP address of this system on the indicated subnet. smi_rip2_get_if_conf_address

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***outaddr*** The IP address of the system on the indicated subnet

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.4 int smi_rip2_get_if_conf_auth_key (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, char ∗ *ifConfAuthKey*)

This function returns the value to be used as the authentication authKey whenever the corresponding instance of rip2_get_ifConfAuthType has a value other than noAuthentication. smi_rip2_get_if_conf_auth_key

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***ifConfAuthKey*** The authentication authKey value

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.5 int smi_rip2_get_if_conf_auth_type (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *ifConfAuthType*)

This function returns the type of authentication used on this interface. smi_rip2_get_-if_conf_auth_type

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***ifConfAuthType*** The authentication type

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.6 int smi_rip2_get_if_conf_default_metric (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *ifConfDefaultMetric*)

This function returns the variable that indicates the metric that is to be used for the default route entry in RIP updates originated on this interface. smi_rip2_get_if_conf_-default_metric

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***if_conf_default_metric*** The default metric

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.7 int smi_rip2_get_if_conf_domain (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *if_conf_domain*)

This function returns the value inserted into the Routing Domain field of all RIP packets sent on this interface. smi_rip2_get_if_conf_domain

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***instance*** The RIP instance ID which is set to 0
← ***addr*** The IP address of a specified interface
→ ***if_conf_domain*** The routing domain field value

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.8 int smi_rip2_get_if_conf_receive (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *ifConfReceive*)

This function returns the version of RIP updates that is to be accepted. smi_rip2_get_-if_conf_receive

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***instance*** The RIP instance ID which is set to 0
← ***addr*** The IP address of a specified interface
→ ***ifConfReceive*** The version of RIP updates

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.9 int smi_rip2_get_if_conf_send (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *ifConfSend*)

This function returns what the router sends on this interface (typically updates). smi_-rip2_get_if_confsend

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***instance*** The RIP instance ID which is set to 0
← ***addr*** The IP address of a specified interface
→ ***ifConfSend*** The updates

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.10 int smi_rip2_get_if_conf_src_address (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, struct pal_in4_addr ∗ *outaddr*)

This function returns the IP address the system will use as a source address on this interface. smi_rip2_get_if_conf_src_address

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***outaddr*** The IP address that will be used as a source address

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.11 int smi_rip2_get_if_conf_status (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *ifConfStatus*)

This function returns the status of the interface. smi_rip2_get_if_conf_status

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***ifConfStatus*** The status of the interface numeric <1-2>
1-RIP_API_STATUS_VALID, 2-RIP_API_STATUS_INVALID

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.12 int smi_rip2_get_if_stat_addr (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, struct pal_in4_addr ∗ *outaddr*)

This function returns the IP address of this system on the indicated subnet. smi_rip2_get_if_stat_addr

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***outaddr*** The IP address

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.13 int smi_rip2_get_if_stat_rcv_bad_packets (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *discarded_packets_count*)

This function returns the number of RIP packets received by the RIP process that were discarded . smi_rip2_get_if_stat_rcv_bad_packets

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***discarded_packets_count*** The number of discarded RIP packets

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.14 int smi_rip2_get_if_stat_rcv_bad_routes (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *bad_routes_count*)

This function returns the number of routes in valid RIP packets that were ignored for any reason e.g. unknown address family. smi_rip2_get_if_stat_rcv_bad_routes

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***bad_routes_count*** The number of routes ignored

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.15 int smi_rip2_get_if_stat_sent_updates (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *stat_sent_updates_count*)

This function returns the number of triggered RIP updates actually sent on this interface.This does not include full updates containing new information. smi_rip2_get_if_stat_sent_updates

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***stat_sent_updates_count*** The number of triggered RIP updates

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.16 int smi_rip2_get_if_stat_status (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *ifStatStatus*)

This function returns the status of the interface. smi_rip2_get_if_stat_status

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

→ ***ifStatStatus*** The returned status RIP_API_STATUS_VALID

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.17 int smi_rip2_get_peer_address (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, struct pal_in4_addr ∗ *outaddr*)

This function returns the IP address that the peer is using as its source address. smi_-rip2_get_peer_address

Parameters:

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

← ***instance*** RIP instance ID which is set to 0

← ***addr*** peer address

→ ***outaddr*** The address of the peer

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.18 int smi_rip2_get_peer_domain (struct smiclient_globals $*$ *azg*, int *instance*, struct pal_in4_addr $*$ *addr*, int $*$ *peerDomain*)

This function returns the value in the Routing Domain field in RIP packets received from the peer. As domain support is deprecated, the value returned is 0. smi_rip2_get_peer_domain

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The peer address

→ ***peer_domain*** The routing domain field value

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.19 int smi_rip2_get_peer_last_update (struct smiclient_globals $*$ *azg*, int *instance*, struct pal_in4_addr $*$ *addr*, int $*$ *peerLastUpdate*)

This function returns the value of sysUptime when the most recent RIP Update is received from this system. smi_rip2_get_peer_last_update

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The peer address

→ ***peer_last_update*** The sys uptime value

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.20 int smi_rip2_get_peer_rcv_bad_packets (struct smiclient_globals $*$ *azg*, int *instance*, struct pal_in4_addr $*$ *addr*, int $*$ *peerRcvBadPacketsCount*)

This function returns the number of RIP response packets from this peer discarded as invalid. smi_rip2_get_peer_rcv_bad_packets

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The peer address

→ ***peer_rcv_bad_packets_count*** The number of discarded RIP response packets

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.21 int smi_rip2_get_peer_rcv_bad_routes (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *peerRcvBadRoutesCount*)

This function returns the number of routes from this peer that were ignored because the entry format was invalid. smi_rip2_get_peer_rcv_bad_routes

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The peer address

→ ***peer_rcv_bad_routes_count*** The number of ignored routes

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.22 int smi_rip2_get_peer_version (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int ∗ *peerVersion*)

This function returns the RIP version number in the header of the last RIP packet received. smi_rip2_get_peer_version

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The peer address

→ ***peer_version*** The version number

Returns:

RIP_API_GET_SUCCESS on success, otherwise RIP_API_GET_ERROR

2.1.2.23 int smi_rip2_set_if_conf_auth_key (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, char ∗ *authKey*)

This function sets the value to be used as the authentication authKey of the corresponding instance of RIP. smi_rip2_set_if_conf_auth_key

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

← ***authKey*** The authentication authKey value

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_IF_NOT_EXIST
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_KEY_CHAIN_EXIST

2.1.2.24 int smi_rip2_set_if_conf_auth_type (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int *ifConfAuthType*)

This function sets authentication type used on this interface. smi_rip2_set_if_conf_auth_type

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

← ***ifConfAuthType*** The authentication type

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_IF_NOT_EXIST
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_AUTH_TYPE_INVALID
RIP_API_SET_ERROR

2.1.2.25 int smi_rip2_set_if_conf_receive (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int *ifConfReceive*)

This function sets the version of RIP updates to be accepted. smi_rip2_set_if_conf_recieve

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

← ***ifConfReceive*** The updates

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_IF_NOT_EXIST
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
RIP_API_SET_ERR_VERSION_INVALID
RIP_API_SET_ERR_INVALID_VALUE

2.1.2.26 int smi_rip2_set_if_conf_send (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int *ifConfSend*)

This function sets the RIP version to be sent in the control packet. smi_rip2_set_if_-conf_send

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

← ***ifConfSend*** The updates

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_IF_NOT_EXIST
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
RIP_API_SET_ERR_VERSION_INVALID
RIP_API_SET_ERR_INVALID_VALUE

2.1.2.27 int smi_rip2_set_if_conf_status (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int *ifConfStatus*)

This function sets status of the interface. smi_rip2_set_if_conf_status

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

← ***ifConfStatus*** The interface status

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_IF_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_INCONSISTENT_VALUE
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERROR

2.1.2.28 int smi_rip2_set_if_stat_status (struct smiclient_globals ∗ *azg*, int *instance*, struct pal_in4_addr ∗ *addr*, int *ifStatStatus*)

This function sets status of the specified interface. smi_rip2_set_if_stat_status

Parameters:

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

← ***instance*** The RIP instance ID which is set to 0

← ***addr*** The IP address of a specified interface

← ***ifStatStatus*** The status of the interface

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERROR
RIP_API_SET_ERR_INCONSISTENT_VALUE

2.1.2.29 int smi_rip_address_family_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *vrfName*)

This function is to set address-family. smi_rip_address_family_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***vrfName***

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST

2.1.2.30 int smi_rip_address_family_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *vrfName*)

This function is to unset address-family. smi_rip_address_family_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***vrfName***

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes RIP_API_SET_ERR_VR_NOT_EXIST

2.1.2.31 int smi_rip_cisco_metric_behavior_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function unsets updating the metric consistent with Cisco and reverts to the default metric type. smi_rip_cisco_metric_behavior_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes RIP_API_SET_ERR_VR_NOT_EXIST RIP_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.32 int smi_rip_debug (struct smiclient_globals ∗ *azg*, int *vrId*, int *debug*)

Use this function to specify the options for the displayed debugging information for RIP events, RIP packets and RIP NSM. smi_rip_debug

Parameters:

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

← ***vrId*** Virtual Router Id

← ***debug*** Pass debug flag as following:
SMI_RIP_DBG_ALL - Debug all RIP information
SMI_RIP_DBG_EVENTS - Debug RIP events
SMI_RIP_DBG_PACKET - Debug RIP and NSM communications

SMI_RIP_DBG_PACKET_SEND - Debug sent packets
SMI_RIP_DBG_PACKET_RECV - Debug received packets
SMI_RIP_DBG_PACKET_DETAIL - Display detailed information for the sent and received packet
SMI_RIP_DBG_PACKET_SEND_DETAIL - Display detailed information for the sent packet
SMI_RIP_DBG_PACKET_RECV_DETAIL - Display detailed information for the received packet

Returns:

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

2.1.2.33 int smi_rip_default_information_delete_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *origin*, char ∗ *rmapName*)

This function is to unset address-family. smi_rip_default_information_delete_unset

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***vrId*** Virtual Router ID numeric <0-255>
← ***instance*** Instance ID
← ***origin*** Default route Origination Type

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes RIP_API_SET_ERR_VR_NOT_EXIST

2.1.2.34 int smi_rip_default_information_originiate_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *origin*, char ∗ *rmapName*)

This function is to unset address-family. smi_rip_default_information_originiate_set

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***vrId*** Virtual Router ID numeric <0-255>
← ***instance*** Instance ID
← ***origin*** Default route Origination Type
← ***rmapName*** Route-Map Name

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes RIP_API_SET_ERR_VR_NOT_EXIST

2.1.2.35 int smi_rip_default_metric_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, int *instance*, int *defaultMetric*)

This function sets the routing protocol to use the specified metric value for all redistributed routes. The specified default metric will be used by all routes that are redistributed. smi_rip_default_metric_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***metric*** The metric of the offset list

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_METRIC_INVALID

2.1.2.36 int smi_rip_default_metric_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, int *instance*)

This function resets the metrics assigned to redistributed routes to the default setting: 1. smi_rip_default_metric_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.37 int smi_rip_distance_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, int *instance*, char * *distanceStr*, struct pal_in4_addr * *addr*, int *prefixLength*, char * *acListName*)

This function specifies the administrative distance for the route calculation. The distance is a feature used by the routers to select the path when there are two or more different routes to the same destination from two different routing protocols. smi_rip_-distance_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***distanceStr*** The distance value

← ***addr*** The address of source prefix

← ***prefixLength*** The prefix length for static RIP route

← ***acListName*** The acess list

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_DISTANCE_INVALID
RIP_API_SET_ERR_PREFIX_INVALID

2.1.2.38 int smi_rip_distance_set_default (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *distanceStr*)

This function sets the administrative distance to the specified value. smi_rip_distance_set_default

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***distanceStr*** The pointer to distance value String<1-255>

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_DISTANCE_INVALID

2.1.2.39 int smi_rip_distance_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, struct pal_in4_addr ∗ *addr*, int *prefixLength*)

This function deletes the administrative distance that was configured for the route calculation. smi_rip_distance_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***addr*** The address of source prefix

← ***prefixLength*** The prefix length for static RIP route

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_DISTANCE_INVALID
RIP_API_SET_ERR_PREFIX_INVALID

2.1.2.40 int smi_rip_distance_unset_default (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function resets the administrative distance configuration to the default value: 120. smi_rip_distance_unset_default

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.41 int smi_rip_distribute_list_prefix_set_sdkapi (struct smiclient_globals ∗ *azg*, char ∗ *acListName*, int *type*, char ∗ *ifName*, u_int32_t *vrId*, int *instance*)

This function disables RIP routing on the specified interface. smi_rip_distribute_list_-prefix_set_sdkapi

Parameters:

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

← ***Access*** List Name

← ***Type*** in|out

← ***ifName*** Interface name string

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_ERR_DISTRIBUTION_NOT_EXIST
RIP_API_SET_ERR_IF_NOT_EXIST

2.1.2.42 int smi_rip_distribute_list_set_sdkapi (struct smiclient_globals ∗ *azg*, char ∗ *acListName*, int *type*, char ∗ *ifName*, u_int32_t *vrId*, int *instance*)

This function sets distribution for specified interface. smi_rip_distribute_list_set_sdkapi

Parameters:

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

← ***Access*** List Name

← ***Type*** in|out

← ***ifName*** Interface name string

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_ERR_DISTRIBUTION_NOT_EXIST
RIP_API_SET_ERR_IF_NOT_EXIST

2.1.2.43 int smi_rip_distribute_list_unset_sdkapi (struct smiclient_globals ∗ *azg*, char ∗ *acListName*, int *type*, char ∗ *ifName*, u_int32_t *vrId*, int *instance*)

This function unsets distribution for specified interface. smi_rip_distribute_list_unset_sdkapi

Parameters:

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

← ***Access*** List Name

← ***Type*** in|out

← ***ifName*** Interface name string

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes RIP_API_SET_ERR_VR_NOT_EXIST RIP_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.44 int smi_rip_distribute_prefix_list_unset_sdkapi (struct smiclient_globals ∗ *azg*, char ∗ *acListName*, int *type*, char ∗ *ifName*, u_int32_t *vrId*, int *instance*)

This function unsets distribution for specified interface. smi_rip_distribute_prefix_list_unset_sdkapi

Parameters:

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

← ***Access*** List Name

← ***Type*** in|out

← ***ifName*** Interface name string

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes RIP_API_SET_ERR_VR_NOT_EXIST RIP_API_SET_ERR_PROCESS_NOT_EXIST\

2.1.2.45 int smi_rip_enable_if_add (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *ifName*)

This function enables RIP routing on the specified interface. If a network is not specified, the interfaces in that network will not be advertised in any RIP update. smi_rip_enable_if_add

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***ifName*** The interface ename for which RIP routing is enabled

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_IF_EXIST

2.1.2.46 int smi_rip_enable_if_delete (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *ifName*)

smi_rip_enable_if_delete

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_IF_NOT_EXIST

2.1.2.47 int smi_rip_enable_nbr_add (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, struct pal_in4_addr ∗ *addr*)

This function enables RIP routing on the specified neighbor.RIP updates are sent to the unicast IP address (es) specified in the neighbor statement. smi_rip_enable_nbr_add

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***addr*** The neighbor address in which this function enables RIP routing

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_NBR_STATIC_EXIST

2.1.2.48 int smi_rip_enable_nbr_delete (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, struct pal_in4_addr ∗ *addr*)

This function disables RIP routing on the neighbor. smi_rip_enable_nbr_delete

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***addr*** The neighbor address in which this function disables RIP routing

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_NBR_STATIC_NOT_EXIST

2.1.2.49 int smi_rip_enable_network_add (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, struct pal_in4_addr ∗ *addr*, int *prefixLength*)

This function enables RIP routing on the specified network. It specifies a network as one that runs Routing Information Protocol (RIP). smi_rip_enable_network_add

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***addr*** The network address on which this call enables RIP routing

← ***prefixLength*** The prefix length for the enabled network

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST

RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_PREFIX_INVALID

2.1.2.50 int smi_rip_enable_network_delete (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, struct pal_in4_addr ∗ *addr*, int *prefixLength*)

This function disables RIP routing on the specified network. smi_rip_enable_-network_delete

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***addr*** The network address on which this call enables RIP routing

← ***prefixLength*** The prefix length for the enabled network

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_PREFIX_INVALID
RIP_API_SET_ERR_NETWORK_NOT_EXIST

2.1.2.51 int smi_rip_if_auth_key_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, char ∗ *authKey*)

This function specifies the RIP authentication authKey chain string.It enables RIPv2 authentication on an interface and specify the name of the authKey chain to be used. smi_rip_if_auth_key_set

Parameters:

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

← ***vrId*** virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

← ***authKey*** Authentication authKey chain string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_authString_EXIST

2.1.2.52 int smi_rip_if_auth_key_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function clears the authKey chain authentication and authentication is disabled. smi_rip_if_auth_key_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.53 int smi_rip_if_auth_mode_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, char ∗ *authMode*)

This function sets the authentication mode and specifies the type of authentication mode used for RIP v2 packets. smi_rip_if_auth_mode_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

← ***authMode*** Specified authentication mode

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_AUTH_TYPE_INVALID

2.1.2.54 int smi_rip_if_auth_mode_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function resets the authentication mode.If the authentication string or authKey chain exist, the mode is set to plain text authentication.If no mode is specified, the mode is set to no authentication. smi_rip_if_auth_mode_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.55 int smi_rip_if_auth_str_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, char ∗ *authString*)

This function sets the authentication string or passacListName used by a authKey. smi_rip_if_auth_str_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

← ***authString*** Authentication string.It is either a text string or an MD5 string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_KEY_CHAIN_EXIST

2.1.2.56 int smi_rip_if_auth_str_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function disables feature to specify the authentication string or passacListName used by a authKey. smi_rip_if_auth_str_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.57 int smi_rip_if_receive_packet_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function enables the interface to receive RIP packets. This is the default setting. smi_rip_if_receive_packet_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.58 int smi_rip_if_receive_packet_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function disables receiving RIP packets on the specified interface. smi_rip_if_-receive_packet_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST

2.1.2.59 int smi_rip_if_receive_version_type_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, int *ifRecvVersionType*)

This function enables receiving the specified version of RIP packets (version 1 or version 2) or receiving both versions of RIP packets (version 1 and version 2). smi_rip_-if_receive_version_type_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

← ***ifRecvVersionType*** Version type <0-4>

0-RI_RIP_UNSPEC, 1-RI_RIP_VERSION_1,

2-RI_RIP_VERSION_2, 3-RI_RIP_VERSION_1_AND_2 -RI_RIP_VERSION_1_COMPATIBLE

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_VERSION_INVALID

2.1.2.60 int smi_rip_if_receive_version_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function reset the receive version to the node version. smi_rip_if_receive_version_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.61 int smi_rip_if_send_packet_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function enables the interface to send RIP packets on interface ifName. smi_rip_if_send_packet_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.62 int smi_rip_if_send_packet_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function disables the interface to send RIP packets on interface ifName. smi_rip_if_send_packet_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes RIP_API_SET_ERR_VR_NOT_EXIST

2.1.2.63 int smi_rip_if_send_version_type_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, int *ifSendVersionType*)

This function sets sending RIP packets on an interface using version control (version type). smi_rip_if_send_version_type_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

← ***ifSendVersionType*** Version type <0-4>

0-RI_RIP_UNSPEC, 1-RI_RIP_VERSION_1,

2-RI_RIP_VERSION_2, 3-RI_RIP_VERSION_1_AND_2 -RI_RIP_VERSION_1_COMPATIBLE

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes RIP_API_SET_ERR_VR_NOT_EXIST RIP_API_SET_ERR_VERSION_INVALID

2.1.2.64 int smi_rip_if_send_version_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function sets the sending version to the version of the RIP node, the default value. smi_rip_if_send_version_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.65 int smi_rip_if_split_horizon_poisoned_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function enables RIP split-horizon poisoned reverse behavior. smi_rip_if_split_horizon_poisoned_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.66 int smi_rip_if_split_horizon_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function enables RIP split-horizon behavior. This command helps avoid including routes in updates sent to the same gateway from which they were learned. smi_rip_if_split_horizon_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_SPLIT_HORIZON_INVALID

2.1.2.67 int smi_rip_if_split_horizon_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*)

This function disables split horizon behavior. The default configuration is split-horizon poisoned. smi_rip_if_split_horizon_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_SPLIT_HORIZON_INVALID

2.1.2.68 int smi_rip_instance_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function establishes an instance of the RIP router and the RIP routing process is enabled. 0

smi_rip_instance_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance. The value should be set to 0 (RIP_DEFAULT_INSTANCE)

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE

2.1.2.69 int smi_rip_instance_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function removes an instance of the RIP router and disables the RIP routing process. smi_rip_instance_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.70 int smi_rip_max_route_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *maxPrefixString*, char ∗ *thresholdStr*)

This function sets the maximum number of RIP routes that can be stored in the routing table. It also sets the percentage of maximum routes to generate a warning (default maximum 75%). smi_rip_max_route_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***maxPrefixString*** The maximum prefix numeric <1-65535>

← ***thresholdStr*** The threshold value numeric <1-100>

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE

2.1.2.71 int smi_rip_max_route_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function sets the threshold value to the default threshold percentage of maximum-prefix checking.The default percentage is 75%. smi_rip_max_route_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.72 int smi_rip_no_debug (struct smiclient_globals ∗ *azg*, int *vrId*, int *debug*)

Use this function to disable specific debugging. smi_rip_no_debug

Parameters:

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

← ***vrId*** Virtual Router Id

← ***debug*** Pass debug flag as following:

SMI_RIP_DBG_ALL - Debug all RIP information

SMI_RIP_DBG_EVENTS - Debug RIP events

SMI_RIP_DBG_PACKET - Debug RIP and NSM communications

SMI_RIP_DBG_PACKET_SEND - Debug sent packets

SMI_RIP_DBG_PACKET_RECV - Debug received packets

SMI_RIP_DBG_PACKET_DETAIL - Display detailed information for the sent and received packet

SMI_RIP_DBG_PACKET_SEND_DETAIL - Display detailed information for the sent packet

SMI_RIP_DBG_PACKET_RECV_DETAIL - Display detailed information for the received packet

Returns:

0 on success, otherwise one of the following error codes RIP_API_SET_ERR_-VR_NOT_EXIST

2.1.2.73 int smi_rip_offset_list_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *acListName*, char ∗ *directStr*, int *metric*, char ∗ *ifName*)

This function adds an offset to in and out metrics to routes learned through RIP: specifies the offset value that is added to the routing metric. smi_rip_offset_list_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***acListName*** The access list name

← ***directStr*** The string of ("in" | "out")

← ***metric*** The metric of the offset numeric <0-RIP_METRIC_INFINITY>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_METRIC_INVALID
RIP_API_SET_ERR_OFFSET_LIST_NOT_EXIST

2.1.2.74 int smi_rip_offset_list_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *acListName*, char ∗ *directStr*, int *metric*, char ∗ *ifName*)

This function removes the offlist. smi_rip_offset_list_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***acListName*** The access list name

← ***directStr*** The string of ("in" | "out")

← ***metric*** The metric of the offset numeric <0-RIP_METRIC_INFINITY>

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_METRIC_INVALID
RIP_API_SET_ERR_OFFSET_LIST_NOT_EXIST

2.1.2.75 int smi_rip_passive_if_add (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *ifName*)

This function suppresses RIP updates and blocks RIP broadcast on the interface. smi_rip_passive_if_add

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_IF_EXIST

2.1.2.76 int smi_rip_passive_if_delete (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *ifName*)

This function disables blocking RIP broadcasts on the interface. smi_rip_passive_if_-delete

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***ifName*** Interface name string

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_IF_NOT_EXIST

2.1.2.77 int smi_rip_recvbuf_size_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, u_int32_t *recvBufsize*)

This function specifies the size of the RIP UDP buffer. smi_rip_recvbuf_size_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***recvBufsize*** The size of the receiving buffer numeric <8192-2147483647>

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_CANT_CHANGE_BUFFER_SIZE

2.1.2.78 int smi_rip_recvbuf_size_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function resets the size of the RIP UDP buffer to the default value: (1024∗192). smi_rip_recvbuf_size_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_CANT_CHANGE_BUFFER_SIZE

2.1.2.79 int smi_rip_redistribute_metric_rmap_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *routeType*, int *metric*, char ∗ *name*)

This function specifies the metric of the route map. smi_rip_redistribute_metric_rmap_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***routeType*** The route type
String ("kernel" | "connected" |
"static" | "ospf" | "isis" | "bgp")

← ***metric*** The metric value

← ***name*** The route map name

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_ROUTE_PROTO_INVALID
RIP_API_SET_ERR_METRIC_INVALID

2.1.2.80 int smi_rip_redistribute_metric_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *routeType*, int *metric*)

This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIP. It also redistributes kernel, connected and static into the RIP. The configured metric is set to the redistributing routes. smi_rip_redistribute_metric_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***routeType*** The route type
String ("kernel" | "connected" |
"static" | "ospf" | "isis" | "bgp")

← ***metric*** The metric value numeric <0-16>

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_ROUTE_PROTO_INVALID
RIP_API_SET_ERR_METRIC_INVALID

2.1.2.81 int smi_rip_redistribute_rmap_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *routeType*, char ∗ *name*)

This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIP. It also redistributes kernel, connected and static into the RIP. Route redistribution is set per route map. smi_rip_redistribute_rmap_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***routeType*** The route type
String ("kernel" | "connected" |
"static" | "ospf" | "isis" | "bgp")

← ***name*** The route map name

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST

RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_ROUTE_PROTO_INVALID

2.1.2.82 int smi_rip_redistribute_set (struct smiclient_globals $*$ *azg*, u_int32_t *vrId*, int *instance*, char $*$ *routeType*)

This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIP. It also redistributes kernel, connected and static into the RIP information. smi_rip_redistribute_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***routeType*** The route type
String ("kernel" | "connected" |
"static" | "ospf" | "isis" | "bgp")

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_ROUTE_PROTO_INVALID

2.1.2.83 int smi_rip_redistribute_unset (struct smiclient_globals $*$ *azg*, u_int32_t *vrId*, int *instance*, char $*$ *routeType*)

This function resets the learned route. smi_rip_redistribute_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***routeType*** The route type
String ("kernel" | "connected" |
"static" | "ospf" | "isis" | "bgp")

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST

RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_ROUTE_PROTO_INVALID

2.1.2.84 int smi_rip_route_add (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, struct pal_in4_addr ∗ *addr*, int *prefixLength*)

This function configures a static route for advertisement through RIP explicitly. An ideal configuration includes a static route that is redistribute via redistribute static inside a routing process. smi_rip_route_add

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***addr*** The static RIP route address

← ***prefixLength*** The prefix length for static RIP route

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_PREFIX_INVALID
RIP_API_SET_ERR_NETWORK_EXIST

2.1.2.85 int smi_rip_route_default_add (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function generates a default route into the Routing Information Protocol (RIP). smi_rip_route_default_add

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** The number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_NETWORK_EXIST

2.1.2.86 int smi_rip_route_default_delete (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function disables the configuration of a default route into the Routing Information Protocol (RIP). smi_rip_route_default_delete

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_NETWORK_NOT_EXIST

2.1.2.87 int smi_rip_route_delete (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, struct pal_in4_addr ∗ *addr*, int *prefixLength*)

This function removes the specified static route. smi_rip_route_delete

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***addr*** The static RIP route address

← ***prefixLength*** The prefix length for static RIP route

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_PREFIX_INVALID
RIP_API_SET_ERR_NETWORK_NOT_EXIST

2.1.2.88 int smi_rip_route_type_delete (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, char ∗ *routeType*)

This function clears specific data from the RIP routing table. smi_rip_route_type_delete

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***routeType*** The route type
String ("all", "kernel" | "connected" |
"static" | "ospf" | "isis" | "bgp")

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_INVALID_VALUE
RIP_API_SET_ERR_ROUTE_NOT_EXIST
RIP_API_SET_ERR_ROUTE_PROTO_INVALID

2.1.2.89 int smi_rip_show_db (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *start_index*, int *end_index*, struct list ∗ *DbList*, int(∗)(struct list ∗DbList) *funpointer*)

This function shows the rip database . smi_rip_show_db

Parameters:

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

← ***vrId*** The virtual router id <1-255>

in} start_index

in} end_index

→ ***DbList*** Return output list

Returns:

RESULT_OK on success, otherwise RIP_ERROR

2.1.2.90 int smi_rip_show_Ifname (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, int *start_index*, int *end_index*, struct list ∗ *ifNameList*, int(∗)(struct list ∗ifNameList) *funpointer*)

This function shows the interface rip configurations. smi_rip_show_ifname

Parameters:

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

← ***vrId*** The virtual router id <1-255>

← ***name*** The interface name

in} start_index

in} end_index

→ ***ifNameList*** Return output list

Returns:

RESULT_OK on success, otherwise RIP_ERROR

2.1.2.91 int smi_rip_show_Ifname_vrf (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, char ∗ *vrfName*, int *start_index*, int *end_index*, struct list ∗ *ifNameList*, int(∗)(struct list ∗ifNameList) *funpointer*)

This function shows the interface rip configurations. smi_rip_show_ifname_vrf

Parameters:

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

← ***vrId*** The virtual router id <1-255>

← ***name*** The interface name

← ***name*** VRF name

in} start_index

in} end_index

→ ***ifNameList*** Return output list

Returns:

RESULT_OK on success, otherwise RIP_ERROR

2.1.2.92 int smi_rip_show_protocol_info_vrf (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *vrfName*, int *start_index*, int *end_index*, struct list ∗ *DbList*, int(∗)(struct list ∗DbList) *funpointer*)

This function shows the currently rip process detailed information . smi_rip_show_protocol_info_vrf

Parameters:

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

← ***vrId*** The virtual router id <1-255>

← ***vrfName*** VRF name

in} start_index

in} end_index

→ ***DbList*** Return output list

Returns:

RESULT_OK on success, otherwise RIP_ERROR

2.1.2.93 int smi_rip_timers_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, u_int32_t *updateTimer*, u_int32_t *timeoutTimer*, u_int32_t *garbageTimer*)

This function sets the specified time per RIP timer: update timer, timeout timer, garbage timer. smi_rip_timers_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***update*** The number of update timer seconds

← ***timeout*** The number of timeout timer seconds

← ***garbage*** The number of garbage timer seconds

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.94 int smi_rip_timers_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This call resets the three timers to the default values: update timer - 30 seconds, Timeout timer - 180 seconds, Garbage Timer- 120 seconds. smi_rip_timers_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.95 int smi_rip_version_type_set (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*, int *version*)

Use this command to specify a RIP version used globally by the router. RIP can be run in version 1 as well as version 2 mode. Version 2 has more features than version 1 including authentication. Once the rip version is set, rip packets of that version will be received and sent on all the rip-enabled interfaces. smi_rip_version_type_set

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

← ***version*** Version type <1-2>
1-RIPv1, 2-RIPv2

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_VERSION_INVALID

2.1.2.96 int smi_rip_version_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, int *instance*)

This function resets the RIP version to default veriosn which is 2 and globally used by the router. smi_rip_version_unset

Parameters:

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

← ***vrId*** Virtual Router ID numeric <0-255>

← ***instance*** Number of the instance

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_VERSION_INVALID

2.1.2.97 int smi_show_debug_rip (struct smiclient_globals ∗ *azg*, int ∗ *debug*)

This function shows the debugging. smi_show_debug_rip

Parameters:

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

← ***debug*** Values to be shown during debug

Returns:

RIP_API_SHOW_SUCCESS on success, otherwise one of the following error code

2.1.2.98 int smi_show_ip_rip_statistics_if (struct smiclient_globals ∗ *azg*, u_int32_t *vrId*, char ∗ *ifName*, int *start_index*, int *end_index*, struct list ∗ *ifNameList*, int(∗)(struct list ∗ifNameList) *funpointer*)

This function shows the rip interface traffic statistics . smi_show_ip_rip_statistics_if

Parameters:

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

← ***vrId*** The virtual router id <1-255>

← ***if_name*** Interface name

in} start_index

in} end_index

→ ***DbList*** Return output list

Returns:

RESULT_OK on success, otherwise RIP_ERROR

2.2 smi_rip_bfd.h File Reference

Provides API for managing RIP BFD(Bidirectional Forwarding Detection).

```
#include "smi_client.h"
#include "smi_rip_bfd_msg.h"
```

Functions

- int smi_rip_bfd_all_interfaces_set (struct smiclient_globals ∗azg, u_int32_t vr_id, int instance)

  This function sets the BFD fall-over check for all the interfaces under a specified process.

- int smi_rip_bfd_all_interfaces_unset (struct smiclient_globals ∗azg, u_int32_t vr_id, int instance)

  This function unsets the BFD fall-over check for all the interfaces under a specified process.

- int smi_rip_bfd_neighbor_set (struct smiclient_globals ∗azg, struct pal_in4_addr ∗nbr, u_int32_t vr_id, int instance)

  This function sets the BFD fall-over check for a specific neighbor under a specified process.

- int smi_rip_bfd_neighbor_unset (struct smiclient_globals ∗azg, struct pal_in4_addr ∗nbr, u_int32_t vr_id, int instance)

  This function unsets the BFD fall-over check for a specific neighbor under a specified process.

- int **smi_rip_bfd_all_interfaces_set_validate** (struct smiclient_globals ∗azg, u_int32_t vr_id, int instance)
- int **smi_rip_bfd_all_interfaces_unset_validate** (struct smiclient_globals ∗azg, u_int32_t vr_id, int instance)
- int **smi_rip_bfd_neighbor_set_validate** (struct smiclient_globals ∗azg, struct pal_in4_addr ∗nbr, u_int32_t vr_id, int instance)
- int **smi_rip_bfd_neighbor_unset_validate** (struct smiclient_globals ∗azg, struct pal_in4_addr ∗nbr, u_int32_t vr_id, int instance)

2.2.1 Detailed Description

Provides API for managing RIP BFD(Bidirectional Forwarding Detection). The API provided in this file forms the basis of ZebOS RIP BFD management. These APIs are used by various north bound management interfaces like CLI, SNMP and SMI

2.2.2 Function Documentation

2.2.2.1 int smi_rip_bfd_all_interfaces_set (struct smiclient_globals ∗ *azg*, u_int32_t *vr_id*, int *instance*)

This function sets the BFD fall-over check for all the interfaces under a specified process. smi_rip_bfd_all_interfaces_set

Parameters:

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

← ***vr_id*** Virtual Router ID numeric <0-255>

← ***instance*** The instance identifier

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_BFD_CONF_SET

2.2.2.2 int smi_rip_bfd_all_interfaces_unset (struct smiclient_globals ∗ *azg*, u_int32_t *vr_id*, int *instance*)

This function unsets the BFD fall-over check for all the interfaces under a specified process. smi_rip_bfd_all_interfaces_unset

Parameters:

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

← ***vr_id*** Virtual Router ID numeric <0-255>

← ***instance*** The instance identifier

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_BFD_CONF_UNSET

2.2.2.3 int smi_rip_bfd_neighbor_set (struct smiclient_globals ∗ *azg*, struct pal_in4_addr ∗ *nbr*, u_int32_t *vr_id*, int *instance*)

This function sets the BFD fall-over check for a specific neighbor under a specified process. smi_rip_bfd_neighbor_set

Parameters:

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

← ***temp_nbr*** Neighbor address in IPv4 address format

← ***vr_id*** Virtual Router ID numeric <0-255>

← ***instance*** The instance identifier

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_BFD_NEIGHBOR_INVALID
RIP_API_SET_ERR_BFD_CONF_SET

2.2.2.4 int smi_rip_bfd_neighbor_unset (struct smiclient_globals ∗ *azg*, struct pal_in4_addr ∗ *nbr*, u_int32_t *vr_id*, int *instance*)

This function unsets the BFD fall-over check for a specific neighbor under a specified process. smi_rip_bfd_neighbor_unset

Parameters:

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

← ***temp_nbr*** Neighbor address in IPv4 address format

← ***vr_id*** Virtual Router ID numeric <0-255>

← ***instance*** The instance identifier

Returns:

RIP_API_SET_SUCCESS on success, otherwise one of the following error codes
RIP_API_SET_ERR_VR_NOT_EXIST
RIP_API_SET_ERR_PROCESS_NOT_EXIST
RIP_API_SET_ERR_BFD_NEIGHBOR_INVALID
RIP_API_SET_ERR_BFD_CONF_UNSET

Index