

# **ZebOS-XP RIP SMI Reference**

**IP Infusion Inc.**

**Generated by Doxygen 1.6.1**

**Wed Dec 16 12:33:54 2015**



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	smi_rip2.h File Reference . . . . .	3
2.1.1	Detailed Description . . . . .	16
2.1.2	Function Documentation . . . . .	16
2.1.2.1	smi_rip2_get_global_queries . . . . .	16
2.1.2.2	smi_rip2_get_global_route_changes . . . . .	17
2.1.2.3	smi_rip2_get_if_conf_address . . . . .	17
2.1.2.4	smi_rip2_get_if_conf_auth_key . . . . .	17
2.1.2.5	smi_rip2_get_if_conf_auth_type . . . . .	18
2.1.2.6	smi_rip2_get_if_conf_default_metric . . . . .	18
2.1.2.7	smi_rip2_get_if_conf_domain . . . . .	19
2.1.2.8	smi_rip2_get_if_conf_receive . . . . .	19
2.1.2.9	smi_rip2_get_if_conf_send . . . . .	19
2.1.2.10	smi_rip2_get_if_conf_src_address . . . . .	20
2.1.2.11	smi_rip2_get_if_conf_status . . . . .	20
2.1.2.12	smi_rip2_get_if_stat_addr . . . . .	20
2.1.2.13	smi_rip2_get_if_stat_rcv_bad_packets . . . . .	21
2.1.2.14	smi_rip2_get_if_stat_rcv_bad_routes . . . . .	21
2.1.2.15	smi_rip2_get_if_stat_sent_updates . . . . .	21
2.1.2.16	smi_rip2_get_if_stat_status . . . . .	22
2.1.2.17	smi_rip2_get_peer_address . . . . .	22
2.1.2.18	smi_rip2_get_peer_domain . . . . .	23
2.1.2.19	smi_rip2_get_peer_last_update . . . . .	23

2.1.2.20	<a href="#">smi_rip2_get_peer_rcv_bad_packets</a> . . . . .	23
2.1.2.21	<a href="#">smi_rip2_get_peer_rcv_bad_routes</a> . . . . .	24
2.1.2.22	<a href="#">smi_rip2_get_peer_version</a> . . . . .	24
2.1.2.23	<a href="#">smi_rip2_set_if_conf_auth_key</a> . . . . .	25
2.1.2.24	<a href="#">smi_rip2_set_if_conf_auth_type</a> . . . . .	25
2.1.2.25	<a href="#">smi_rip2_set_if_conf_receive</a> . . . . .	25
2.1.2.26	<a href="#">smi_rip2_set_if_conf_send</a> . . . . .	26
2.1.2.27	<a href="#">smi_rip2_set_if_conf_status</a> . . . . .	26
2.1.2.28	<a href="#">smi_rip2_set_if_stat_status</a> . . . . .	27
2.1.2.29	<a href="#">smi_rip_address_family_set</a> . . . . .	27
2.1.2.30	<a href="#">smi_rip_address_family_unset</a> . . . . .	28
2.1.2.31	<a href="#">smi_rip_cisco_metric_behavior_unset</a> . . . . .	28
2.1.2.32	<a href="#">smi_rip_debug</a> . . . . .	28
2.1.2.33	<a href="#">smi_rip_default_information_delete_unset</a> . . . . .	29
2.1.2.34	<a href="#">smi_rip_default_information_originate_set</a> . . . . .	29
2.1.2.35	<a href="#">smi_rip_default_metric_set</a> . . . . .	30
2.1.2.36	<a href="#">smi_rip_default_metric_unset</a> . . . . .	30
2.1.2.37	<a href="#">smi_rip_distance_set</a> . . . . .	30
2.1.2.38	<a href="#">smi_rip_distance_set_default</a> . . . . .	31
2.1.2.39	<a href="#">smi_rip_distance_unset</a> . . . . .	31
2.1.2.40	<a href="#">smi_rip_distance_unset_default</a> . . . . .	32
2.1.2.41	<a href="#">smi_rip_distribute_list_prefix_set_sdkapi</a> . . . . .	32
2.1.2.42	<a href="#">smi_rip_distribute_list_set_sdkapi</a> . . . . .	33
2.1.2.43	<a href="#">smi_rip_distribute_list_unset_sdkapi</a> . . . . .	33
2.1.2.44	<a href="#">smi_rip_distribute_prefix_list_unset_sdkapi</a> . . . . .	34
2.1.2.45	<a href="#">smi_rip_enable_if_add</a> . . . . .	34
2.1.2.46	<a href="#">smi_rip_enable_if_delete</a> . . . . .	35
2.1.2.47	<a href="#">smi_rip_enable_nbr_add</a> . . . . .	35
2.1.2.48	<a href="#">smi_rip_enable_nbr_delete</a> . . . . .	36
2.1.2.49	<a href="#">smi_rip_enable_network_add</a> . . . . .	36
2.1.2.50	<a href="#">smi_rip_enable_network_delete</a> . . . . .	37
2.1.2.51	<a href="#">smi_rip_if_auth_key_set</a> . . . . .	37
2.1.2.52	<a href="#">smi_rip_if_auth_key_unset</a> . . . . .	38
2.1.2.53	<a href="#">smi_rip_if_auth_mode_set</a> . . . . .	38

2.1.2.54	<a href="#">smi_rip_if_auth_mode_unset</a>	38
2.1.2.55	<a href="#">smi_rip_if_auth_str_set</a>	39
2.1.2.56	<a href="#">smi_rip_if_auth_str_unset</a>	39
2.1.2.57	<a href="#">smi_rip_if_receive_packet_set</a>	40
2.1.2.58	<a href="#">smi_rip_if_receive_packet_unset</a>	40
2.1.2.59	<a href="#">smi_rip_if_receive_version_type_set</a>	40
2.1.2.60	<a href="#">smi_rip_if_receive_version_unset</a>	41
2.1.2.61	<a href="#">smi_rip_if_send_packet_set</a>	41
2.1.2.62	<a href="#">smi_rip_if_send_packet_unset</a>	42
2.1.2.63	<a href="#">smi_rip_if_send_version_type_set</a>	42
2.1.2.64	<a href="#">smi_rip_if_send_version_unset</a>	42
2.1.2.65	<a href="#">smi_rip_if_split_horizon_poisoned_set</a>	43
2.1.2.66	<a href="#">smi_rip_if_split_horizon_set</a>	43
2.1.2.67	<a href="#">smi_rip_if_split_horizon_unset</a>	44
2.1.2.68	<a href="#">smi_rip_instance_set</a>	44
2.1.2.69	<a href="#">smi_rip_instance_unset</a>	44
2.1.2.70	<a href="#">smi_rip_max_route_set</a>	45
2.1.2.71	<a href="#">smi_rip_max_route_unset</a>	45
2.1.2.72	<a href="#">smi_rip_no_debug</a>	46
2.1.2.73	<a href="#">smi_rip_offset_list_set</a>	46
2.1.2.74	<a href="#">smi_rip_offset_list_unset</a>	47
2.1.2.75	<a href="#">smi_rip_passive_if_add</a>	47
2.1.2.76	<a href="#">smi_rip_passive_if_delete</a>	48
2.1.2.77	<a href="#">smi_rip_recvbuf_size_set</a>	48
2.1.2.78	<a href="#">smi_rip_recvbuf_size_unset</a>	49
2.1.2.79	<a href="#">smi_rip_redistribute_metric_rmap_set</a>	49
2.1.2.80	<a href="#">smi_rip_redistribute_metric_set</a>	50
2.1.2.81	<a href="#">smi_rip_redistribute_rmap_set</a>	50
2.1.2.82	<a href="#">smi_rip_redistribute_set</a>	51
2.1.2.83	<a href="#">smi_rip_redistribute_unset</a>	51
2.1.2.84	<a href="#">smi_rip_route_add</a>	52
2.1.2.85	<a href="#">smi_rip_route_default_add</a>	52
2.1.2.86	<a href="#">smi_rip_route_default_delete</a>	53
2.1.2.87	<a href="#">smi_rip_route_delete</a>	53

2.1.2.88	<a href="#">smi_rip_route_type_delete</a>	53
2.1.2.89	<a href="#">smi_rip_show_db</a>	54
2.1.2.90	<a href="#">smi_rip_show_ifname</a>	54
2.1.2.91	<a href="#">smi_rip_show_ifname_vrf</a>	55
2.1.2.92	<a href="#">smi_rip_show_protocol_info_vrf</a>	55
2.1.2.93	<a href="#">smi_rip_timers_set</a>	56
2.1.2.94	<a href="#">smi_rip_timers_unset</a>	56
2.1.2.95	<a href="#">smi_rip_version_type_set</a>	57
2.1.2.96	<a href="#">smi_rip_version_unset</a>	57
2.1.2.97	<a href="#">smi_show_debug_rip</a>	57
2.1.2.98	<a href="#">smi_show_ip_rip_statistics_if</a>	58
2.2	<a href="#">smi_rip_bfd.h File Reference</a>	59
2.2.1	<a href="#">Detailed Description</a>	59
2.2.2	<a href="#">Function Documentation</a>	60
2.2.2.1	<a href="#">smi_rip_bfd_all_interfaces_set</a>	60
2.2.2.2	<a href="#">smi_rip_bfd_all_interfaces_unset</a>	60
2.2.2.3	<a href="#">smi_rip_bfd_neighbor_set</a>	60
2.2.2.4	<a href="#">smi_rip_bfd_neighbor_unset</a>	61

# Chapter 1

## File Index

### 1.1 File List

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

<a href="#">smi_rip.h</a> (Provides API for managing RIP ) . . . . .	3
<a href="#">smi_rip_bfd.h</a> (Provides API for managing RIP BFD(Bidirectional Forward- ing Detection) ) . . . . .	59





# 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\\_rcvbuf\\_size\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int instance, u\_int32\_t rcvBufsize)

*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\\_originate\\_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 vrfId, int instance, char \*ifName)
- int [smi\\_show\\_ip\\_rip\\_statistics\\_if](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, 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 vrfId, 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 vrfId, 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 vrfId, int instance, u\_int32\_t \*recvBufsize)
- int [smi\\_rip\\_clear\\_stat](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)
- int [smi\\_rip\\_if\\_receive\\_packet\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)
- int [smi\\_rip\\_if\\_receive\\_packet\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)
- int [smi\\_rip\\_if\\_send\\_packet\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)
- int [smi\\_rip\\_if\\_send\\_packet\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)
- int [smi\\_rip\\_if\\_receive\\_version\\_type\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, int ifRecvVersionType)
- int [smi\\_rip\\_if\\_receive\\_version\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)
- int [smi\\_rip\\_if\\_send\\_version\\_type\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, int ifSendVersionType)
- int [smi\\_rip\\_if\\_send\\_version\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)
- int [smi\\_rip\\_if\\_auth\\_mode\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, char \*authMode)
- int [smi\\_rip\\_if\\_auth\\_mode\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)
- int [smi\\_rip\\_if\\_auth\\_str\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, char \*authString)
- int [smi\\_rip\\_if\\_auth\\_str\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, 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 vrfId, int instance)
- int **smi\_rip\_redistribute\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, char \*routeType)
- int **smi\_rip\_redistribute\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, char \*routeType)
- int **smi\_rip\_redistribute\_metric\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, char \*routeType, int metric)
- int **smi\_rip\_redistribute\_rmap\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, char \*routeType, char \*name)
- int **smi\_rip\_redistribute\_metric\_rmap\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, char \*routeType, int metric, char \*name)
- int **smi\_rip\_timers\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, 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 vrfId, int instance)
- int **smi\_rip\_distance\_set\_default\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, char \*distanceStr)
- int **smi\_rip\_distance\_unset\_default\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance)
- int **smi\_rip\_distance\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, 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 vrfId, int instance, struct pal\_in4\_addr \*addr, int prefixLength)
- int **smi\_rip\_max\_route\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, char \*maxPrefixString, char \*thresholdStr)
- int **smi\_rip\_max\_route\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance)
- int **smi\_rip\_route\_type\_delete\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, char \*routeType)
- int **smi\_rip\_recvbuf\_size\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, u\_int32\_t recvBufSize)
- int **smi\_rip\_recvbuf\_size\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance)
- int **smi\_rip\_cisco\_metric\_behavior\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, int instance, u\_char metricType)
- int **smi\_rip\_cisco\_metric\_behavior\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrfId, 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\_originate\_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\_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\_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_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 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 vrfId, char \* vrfName)

This function is to set address-family. smi\_rip\_address\_family\_set

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *vrfId* 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\_originate\_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\_originate\_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 access 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 IPv2 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_rcvbuf_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\_rcvbuf\_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, Time-out 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 version 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

- smi\_rip.h, [3](#)
  - smi\_rip2\_get\_global\_queries, [16](#)
  - smi\_rip2\_get\_global\_route\_-  
changes, [17](#)
  - smi\_rip2\_get\_if\_conf\_address, [17](#)
  - smi\_rip2\_get\_if\_conf\_auth\_key, [17](#)
  - smi\_rip2\_get\_if\_conf\_auth\_type, [18](#)
  - smi\_rip2\_get\_if\_conf\_default\_-  
metric, [18](#)
  - smi\_rip2\_get\_if\_conf\_domain, [18](#)
  - smi\_rip2\_get\_if\_conf\_receive, [19](#)
  - smi\_rip2\_get\_if\_conf\_send, [19](#)
  - smi\_rip2\_get\_if\_conf\_src\_address,  
[19](#)
  - smi\_rip2\_get\_if\_conf\_status, [20](#)
  - smi\_rip2\_get\_if\_stat\_addr, [20](#)
  - smi\_rip2\_get\_if\_stat\_rcv\_bad\_-  
packets, [21](#)
  - smi\_rip2\_get\_if\_stat\_rcv\_bad\_-  
routes, [21](#)
  - smi\_rip2\_get\_if\_stat\_sent\_updates,  
[21](#)
  - smi\_rip2\_get\_if\_stat\_status, [22](#)
  - smi\_rip2\_get\_peer\_address, [22](#)
  - smi\_rip2\_get\_peer\_domain, [22](#)
  - smi\_rip2\_get\_peer\_last\_update, [23](#)
  - smi\_rip2\_get\_peer\_rcv\_bad\_-  
packets, [23](#)
  - smi\_rip2\_get\_peer\_rcv\_bad\_routes,  
[24](#)
  - smi\_rip2\_get\_peer\_version, [24](#)
  - smi\_rip2\_set\_if\_conf\_auth\_key, [24](#)
  - smi\_rip2\_set\_if\_conf\_auth\_type, [25](#)
  - smi\_rip2\_set\_if\_conf\_receive, [25](#)
  - smi\_rip2\_set\_if\_conf\_send, [26](#)
  - smi\_rip2\_set\_if\_conf\_status, [26](#)
  - smi\_rip2\_set\_if\_stat\_status, [27](#)
  - smi\_rip\_address\_family\_set, [27](#)
  - smi\_rip\_address\_family\_unset, [27](#)
  - smi\_rip\_cisco\_metric\_behavior\_-  
unset, [28](#)
  - smi\_rip\_debug, [28](#)
  - smi\_rip\_default\_information\_-  
delete\_unset, [29](#)
  - smi\_rip\_default\_information\_-  
originiate\_set, [29](#)
  - smi\_rip\_default\_metric\_set, [29](#)
  - smi\_rip\_default\_metric\_unset, [30](#)
  - smi\_rip\_distance\_set, [30](#)
  - smi\_rip\_distance\_set\_default, [31](#)
  - smi\_rip\_distance\_unset, [31](#)
  - smi\_rip\_distance\_unset\_default, [32](#)
  - smi\_rip\_distribute\_list\_prefix\_set\_-  
sdkapi, [32](#)
  - smi\_rip\_distribute\_list\_set\_sdkapi,  
[33](#)
  - smi\_rip\_distribute\_list\_unset\_-  
sdkapi, [33](#)
  - smi\_rip\_distribute\_prefix\_list\_-  
unset\_sdkapi, [34](#)
  - smi\_rip\_enable\_if\_add, [34](#)
  - smi\_rip\_enable\_if\_delete, [35](#)
  - smi\_rip\_enable\_nbr\_add, [35](#)
  - smi\_rip\_enable\_nbr\_delete, [36](#)
  - smi\_rip\_enable\_network\_add, [36](#)
  - smi\_rip\_enable\_network\_delete, [37](#)
  - smi\_rip\_if\_auth\_key\_set, [37](#)
  - smi\_rip\_if\_auth\_key\_unset, [37](#)
  - smi\_rip\_if\_auth\_mode\_set, [38](#)
  - smi\_rip\_if\_auth\_mode\_unset, [38](#)
  - smi\_rip\_if\_auth\_str\_set, [39](#)
  - smi\_rip\_if\_auth\_str\_unset, [39](#)
  - smi\_rip\_if\_receive\_packet\_set, [39](#)
  - smi\_rip\_if\_receive\_packet\_unset, [40](#)
  - smi\_rip\_if\_receive\_version\_type\_-  
set, [40](#)
  - smi\_rip\_if\_receive\_version\_unset,  
[41](#)
  - smi\_rip\_if\_send\_packet\_set, [41](#)
  - smi\_rip\_if\_send\_packet\_unset, [41](#)
  - smi\_rip\_if\_send\_version\_type\_set,  
[42](#)

- smi\_rip\_if\_send\_version\_unset, 42
- smi\_rip\_if\_split\_horizon\_-poisoned\_set, 43
- smi\_rip\_if\_split\_horizon\_set, 43
- smi\_rip\_if\_split\_horizon\_unset, 43
- smi\_rip\_instance\_set, 44
- smi\_rip\_instance\_unset, 44
- smi\_rip\_max\_route\_set, 45
- smi\_rip\_max\_route\_unset, 45
- smi\_rip\_no\_debug, 45
- smi\_rip\_offset\_list\_set, 46
- smi\_rip\_offset\_list\_unset, 47
- smi\_rip\_passive\_if\_add, 47
- smi\_rip\_passive\_if\_delete, 48
- smi\_rip\_recvbuf\_size\_set, 48
- smi\_rip\_recvbuf\_size\_unset, 48
- smi\_rip\_redistribute\_metric\_rmap\_-set, 49
- smi\_rip\_redistribute\_metric\_set, 49
- smi\_rip\_redistribute\_rmap\_set, 50
- smi\_rip\_redistribute\_set, 51
- smi\_rip\_redistribute\_unset, 51
- smi\_rip\_route\_add, 52
- smi\_rip\_route\_default\_add, 52
- smi\_rip\_route\_default\_delete, 52
- smi\_rip\_route\_delete, 53
- smi\_rip\_route\_type\_delete, 53
- smi\_rip\_show\_db, 54
- smi\_rip\_show\_ifname, 54
- smi\_rip\_show\_ifname\_vrf, 55
- smi\_rip\_show\_protocol\_info\_vrf, 55
- smi\_rip\_timers\_set, 56
- smi\_rip\_timers\_unset, 56
- smi\_rip\_version\_type\_set, 56
- smi\_rip\_version\_unset, 57
- smi\_show\_debug\_rip, 57
- smi\_show\_ip\_rip\_statistics\_if, 58
- smi\_rip2\_get\_global\_queries  
smi\_rip.h, 16
- smi\_rip2\_get\_global\_route\_changes  
smi\_rip.h, 17
- smi\_rip2\_get\_if\_conf\_address  
smi\_rip.h, 17
- smi\_rip2\_get\_if\_conf\_auth\_key  
smi\_rip.h, 17
- smi\_rip2\_get\_if\_conf\_auth\_type  
smi\_rip.h, 18
- smi\_rip2\_get\_if\_conf\_default\_metric  
smi\_rip.h, 18
- smi\_rip2\_get\_if\_conf\_domain  
smi\_rip.h, 18
- smi\_rip2\_get\_if\_conf\_receive  
smi\_rip.h, 19
- smi\_rip2\_get\_if\_conf\_send  
smi\_rip.h, 19
- smi\_rip2\_get\_if\_conf\_src\_address  
smi\_rip.h, 19
- smi\_rip2\_get\_if\_conf\_status  
smi\_rip.h, 20
- smi\_rip2\_get\_if\_stat\_addr  
smi\_rip.h, 20
- smi\_rip2\_get\_if\_stat\_rcv\_bad\_packets  
smi\_rip.h, 21
- smi\_rip2\_get\_if\_stat\_rcv\_bad\_routes  
smi\_rip.h, 21
- smi\_rip2\_get\_if\_stat\_sent\_updates  
smi\_rip.h, 21
- smi\_rip2\_get\_if\_stat\_status  
smi\_rip.h, 22
- smi\_rip2\_get\_peer\_address  
smi\_rip.h, 22
- smi\_rip2\_get\_peer\_domain  
smi\_rip.h, 22
- smi\_rip2\_get\_peer\_last\_update  
smi\_rip.h, 23
- smi\_rip2\_get\_peer\_rcv\_bad\_packets  
smi\_rip.h, 23
- smi\_rip2\_get\_peer\_rcv\_bad\_routes  
smi\_rip.h, 24
- smi\_rip2\_get\_peer\_version  
smi\_rip.h, 24
- smi\_rip2\_set\_if\_conf\_auth\_key  
smi\_rip.h, 24
- smi\_rip2\_set\_if\_conf\_auth\_type  
smi\_rip.h, 25
- smi\_rip2\_set\_if\_conf\_receive  
smi\_rip.h, 25
- smi\_rip2\_set\_if\_conf\_send  
smi\_rip.h, 26
- smi\_rip2\_set\_if\_conf\_status  
smi\_rip.h, 26
- smi\_rip2\_set\_if\_stat\_status  
smi\_rip.h, 27
- smi\_rip\_address\_family\_set  
smi\_rip.h, 27
- smi\_rip\_address\_family\_unset  
smi\_rip.h, 27
- smi\_rip\_bfd.h, 59
- smi\_rip\_bfd\_all\_interfaces\_set, 60

- smi\_rip\_bfd\_all\_interfaces\_unset, 60
- smi\_rip\_bfd\_neighbor\_set, 60
- smi\_rip\_bfd\_neighbor\_unset, 61
- smi\_rip\_bfd\_all\_interfaces\_set
  - smi\_rip\_bfd.h, 60
- smi\_rip\_bfd\_all\_interfaces\_unset
  - smi\_rip\_bfd.h, 60
- smi\_rip\_bfd\_neighbor\_set
  - smi\_rip\_bfd.h, 60
- smi\_rip\_bfd\_neighbor\_unset
  - smi\_rip\_bfd.h, 61
- smi\_rip\_cisco\_metric\_behavior\_unset
  - smi\_rip.h, 28
- smi\_rip\_debug
  - smi\_rip.h, 28
- smi\_rip\_default\_information\_delete\_-unset
  - smi\_rip.h, 29
- smi\_rip\_default\_information\_originiate\_-set
  - smi\_rip.h, 29
- smi\_rip\_default\_metric\_set
  - smi\_rip.h, 29
- smi\_rip\_default\_metric\_unset
  - smi\_rip.h, 30
- smi\_rip\_distance\_set
  - smi\_rip.h, 30
- smi\_rip\_distance\_set\_default
  - smi\_rip.h, 31
- smi\_rip\_distance\_unset
  - smi\_rip.h, 31
- smi\_rip\_distance\_unset\_default
  - smi\_rip.h, 32
- smi\_rip\_distribute\_list\_prefix\_set\_sdkapi
  - smi\_rip.h, 32
- smi\_rip\_distribute\_list\_set\_sdkapi
  - smi\_rip.h, 33
- smi\_rip\_distribute\_list\_unset\_sdkapi
  - smi\_rip.h, 33
- smi\_rip\_distribute\_prefix\_list\_unset\_-sdkapi
  - smi\_rip.h, 34
- smi\_rip\_enable\_if\_add
  - smi\_rip.h, 34
- smi\_rip\_enable\_if\_delete
  - smi\_rip.h, 35
- smi\_rip\_enable\_nbr\_add
  - smi\_rip.h, 35
- smi\_rip\_enable\_nbr\_delete
  - smi\_rip.h, 36
- smi\_rip\_enable\_network\_add
  - smi\_rip.h, 36
- smi\_rip\_enable\_network\_delete
  - smi\_rip.h, 37
- smi\_rip\_if\_auth\_key\_set
  - smi\_rip.h, 37
- smi\_rip\_if\_auth\_key\_unset
  - smi\_rip.h, 37
- smi\_rip\_if\_auth\_mode\_set
  - smi\_rip.h, 38
- smi\_rip\_if\_auth\_mode\_unset
  - smi\_rip.h, 38
- smi\_rip\_if\_auth\_str\_set
  - smi\_rip.h, 39
- smi\_rip\_if\_auth\_str\_unset
  - smi\_rip.h, 39
- smi\_rip\_if\_receive\_packet\_set
  - smi\_rip.h, 39
- smi\_rip\_if\_receive\_packet\_unset
  - smi\_rip.h, 40
- smi\_rip\_if\_receive\_version\_type\_set
  - smi\_rip.h, 40
- smi\_rip\_if\_receive\_version\_unset
  - smi\_rip.h, 41
- smi\_rip\_if\_send\_packet\_set
  - smi\_rip.h, 41
- smi\_rip\_if\_send\_packet\_unset
  - smi\_rip.h, 41
- smi\_rip\_if\_send\_version\_type\_set
  - smi\_rip.h, 42
- smi\_rip\_if\_send\_version\_unset
  - smi\_rip.h, 42
- smi\_rip\_if\_split\_horizon\_poisoned\_set
  - smi\_rip.h, 43
- smi\_rip\_if\_split\_horizon\_set
  - smi\_rip.h, 43
- smi\_rip\_if\_split\_horizon\_unset
  - smi\_rip.h, 43
- smi\_rip\_instance\_set
  - smi\_rip.h, 44
- smi\_rip\_instance\_unset
  - smi\_rip.h, 44
- smi\_rip\_max\_route\_set
  - smi\_rip.h, 45
- smi\_rip\_max\_route\_unset
  - smi\_rip.h, 45
- smi\_rip\_no\_debug
  - smi\_rip.h, 45
- smi\_rip\_offset\_list\_set
  - smi\_rip.h, 45

smi\_rip.h, [46](#)  
smi\_rip\_offset\_list\_unset  
    smi\_rip.h, [47](#)  
smi\_rip\_passive\_if\_add  
    smi\_rip.h, [47](#)  
smi\_rip\_passive\_if\_delete  
    smi\_rip.h, [48](#)  
smi\_rip\_recvbuf\_size\_set  
    smi\_rip.h, [48](#)  
smi\_rip\_recvbuf\_size\_unset  
    smi\_rip.h, [48](#)  
smi\_rip\_redistribute\_metric\_rmap\_set  
    smi\_rip.h, [49](#)  
smi\_rip\_redistribute\_metric\_set  
    smi\_rip.h, [49](#)  
smi\_rip\_redistribute\_rmap\_set  
    smi\_rip.h, [50](#)  
smi\_rip\_redistribute\_set  
    smi\_rip.h, [51](#)  
smi\_rip\_redistribute\_unset  
    smi\_rip.h, [51](#)  
smi\_rip\_route\_add  
    smi\_rip.h, [52](#)  
smi\_rip\_route\_default\_add  
    smi\_rip.h, [52](#)  
smi\_rip\_route\_default\_delete  
    smi\_rip.h, [52](#)  
smi\_rip\_route\_delete  
    smi\_rip.h, [53](#)  
smi\_rip\_route\_type\_delete  
    smi\_rip.h, [53](#)  
smi\_rip\_show\_db  
    smi\_rip.h, [54](#)  
smi\_rip\_show\_ifname  
    smi\_rip.h, [54](#)  
smi\_rip\_show\_ifname\_vrf  
    smi\_rip.h, [55](#)  
smi\_rip\_show\_protocol\_info\_vrf  
    smi\_rip.h, [55](#)  
smi\_rip\_timers\_set  
    smi\_rip.h, [56](#)  
smi\_rip\_timers\_unset  
    smi\_rip.h, [56](#)  
smi\_rip\_version\_type\_set  
    smi\_rip.h, [56](#)  
smi\_rip\_version\_unset  
    smi\_rip.h, [57](#)  
smi\_show\_debug\_rip  
    smi\_rip.h, [57](#)  
smi\_show\_ip\_rip\_statistics\_if  
    smi\_rip.h, [58](#)