

ZebOS-XP RIPng SMI Reference
IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:56 2015

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	smi_ripng.h File Reference	3
2.1.1	Detailed Description	13
2.1.2	Function Documentation	13
2.1.2.1	smi_ripng_aggregate_add	13
2.1.2.2	smi_ripng_aggregate_delete	14
2.1.2.3	smi_ripng_cisco_metric_behavior_set	14
2.1.2.4	smi_ripng_cisco_metric_behavior_unset	14
2.1.2.5	smi_ripng_debug	15
2.1.2.6	smi_ripng_default_metric_set	15
2.1.2.7	smi_ripng_default_metric_unset	16
2.1.2.8	smi_ripng_distance_set_default	16
2.1.2.9	smi_ripng_distance_unset_default	17
2.1.2.10	smi_ripng_distribute_list_prefix_set_sdkapi	17
2.1.2.11	smi_ripng_distribute_list_prefix_unset_sdkapi	18
2.1.2.12	smi_ripng_distribute_list_set_sdkapi	18
2.1.2.13	smi_ripng_distribute_list_unset_sdkapi	19
2.1.2.14	smi_ripng_enable_nbr_add	19
2.1.2.15	smi_ripng_enable_nbr_delete	20
2.1.2.16	smi_ripng_if_ipv6_router_set	20
2.1.2.17	smi_ripng_if_ipv6_router_unset	20
2.1.2.18	smi_ripng_if_metric_offset_set_sdkapi	21
2.1.2.19	smi_ripng_if_metric_offset_unset_sdkapi	21

2.1.2.20	smi_ripng_if_rmap_set_sdkapi	22
2.1.2.21	smi_ripng_if_rmap_unset_sdkapi	22
2.1.2.22	smi_ripng_if_split_horizon_poisoned_set	22
2.1.2.23	smi_ripng_if_split_horizon_set	23
2.1.2.24	smi_ripng_if_split_horizon_unset	23
2.1.2.25	smi_ripng_instance_set	24
2.1.2.26	smi_ripng_instance_unset	24
2.1.2.27	smi_ripng_no_debug	24
2.1.2.28	smi_ripng_offset_list_set	25
2.1.2.29	smi_ripng_offset_list_unset	26
2.1.2.30	smi_ripng_passive_if_add	26
2.1.2.31	smi_ripng_passive_if_delete	27
2.1.2.32	smi_ripng_rcvbuf_size_set	27
2.1.2.33	smi_ripng_rcvbuf_size_unset	27
2.1.2.34	smi_ripng_redistribute_metric_rmap_set	28
2.1.2.35	smi_ripng_redistribute_metric_set	28
2.1.2.36	smi_ripng_redistribute_rmap_set	29
2.1.2.37	smi_ripng_redistribute_set	30
2.1.2.38	smi_ripng_redistribute_unset	30
2.1.2.39	smi_ripng_route_default_add	31
2.1.2.40	smi_ripng_route_default_delete	31
2.1.2.41	smi_ripng_route_type_delete	32
2.1.2.42	smi_ripng_show_db	32
2.1.2.43	smi_ripng_show_db_vrf	33
2.1.2.44	smi_ripng_show_ifname	33
2.1.2.45	smi_ripng_show_ifname_vrf	34
2.1.2.46	smi_ripng_timers_set	34
2.1.2.47	smi_ripng_timers_unset	35
2.1.2.48	smi_show_debugging_ipv6_rip_api	35
2.1.2.49	smi_show_ipv6_protocols_rip_api	35

Chapter 1

File Index

1.1 File List

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

[smi_ripng.h](#) (Provides API for managing RIPng) 3

Chapter 2

File Documentation

2.1 smi_ripng.h File Reference

Provides API for managing RIPng. `#include "smi_client.h"`
`#include "smi_ripng_msg.h"`

Functions

- int [smi_ripng_if_ipv6_router_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, char *tag)
This function enables RIPng routing on the interface.
- int [smi_ripng_if_ipv6_router_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
This function disables RIPng routing on the interface.
- int [smi_ripng_if_split_horizon_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
This function enables the RIP split-horizon behavior. It helps avoid including routes in updates sent to the same gateway from which they were learned.
- int [smi_ripng_if_split_horizon_poisoned_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
This function enables RIP split-horizon poisoned reverse behavior.
- int [smi_ripng_if_split_horizon_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
This function disables the split horizon behavior. The default configuration is split-horizon poisoned.
- int [smi_ripng_if_metric_offset_set_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int metric)

This function is used to set RIP metric offset.

- int [smi_ripng_if_metric_offset_unset_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)

This function is used to unset RIP metric offset.

- int [smi_ripng_instance_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

This function establishes an instance of the IPv6 RIP router. The RIP routing process is enabled for IPv6.

- int [smi_ripng_instance_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

This function removes an instance of the IPv6 router. The routing process is disabled.

- int [smi_ripng_enable_nbr_add](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, char *ifname)

This function enables RIP routing on the specified neighbor. It is used for each connected point-to-point link.

- int [smi_ripng_enable_nbr_delete](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, char *ifname)

This function disables RIPng routing on the neighbor.

- int [smi_ripng_passive_if_add](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *ifname)

This function suppresses RIP updates. RIP broadcast on the interface is blocked.

- int [smi_ripng_passive_if_delete](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *ifname)

This function disables blocking RIP broadcast on the interface.

- int [smi_ripng_route_default_add](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

This function configures a static route for advertisement through RIP explicitly.

- int [smi_ripng_route_default_delete](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

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

- int [smi_ripng_offset_list_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *alist, char *direct_str, 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_ripng_offset_list_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *alist, char *direct_str, int metric, char *ifname)

This function removes the offset list. The offset value is removed from the routing metric.

- int [smi_ripng_default_metric_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, int metric)

This function sets the routing protocol to use the specified metric value for all redistributed routes.

- int [smi_ripng_default_metric_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

This function resets the metrics of the redistributed routes to the default value as 1.

- int [smi_ripng_redistribute_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type)

This function enables redistributing of routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIPng.

- int [smi_ripng_redistribute_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type)

This function disables the redistribution of routes into RIPng, and removes the configured route map and metric.

- int [smi_ripng_redistribute_metric_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type, int metric)

This redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIPng and sets the metric for redistribution of routes.

- int [smi_ripng_redistribute_rmap_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type, char *name)

This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIPng. It uses the route map configured for redistribution of routes.

- int [smi_ripng_redistribute_metric_rmap_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type, int metric, char *name)

This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIPng. It sets the metric for redistribution and uses the route map configured for redistribution of routes.

- int [smi_ripng_timers_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, u_int32_t update, u_int32_t timeout, u_int32_t garbage)

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

- int [smi_ripng_timers_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

This function resets three timers to default values. Update timer to 30 seconds. Timeout timer to 180 seconds. Garbage timer to 120 seconds.

- int [smi_ripng_distance_set_default](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *distance_str)

This function sets the administrative distance to the specified value. The default value for administrative distance is 120.

- int [smi_ripng_distance_unset_default](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

This function resets the administrative distance to its default value as 120.

- int [smi_ripng_recvbuf_size_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, u_int32_t bufsize)

This function sets the specified size of the RIP UDP buffer.

- int [smi_ripng_recvbuf_size_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

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

- int [smi_ripng_route_type_delete](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type)

This function clears specified data from the RIPng routing table.

- int [smi_ripng_cisco_metric_behavior_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, u_char metric_type)

This function sets the metric update as Cisco; updating the metric consistent with Cisco is enabled.

- int [smi_ripng_cisco_metric_behavior_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

This function unsets the metric update as Cisco. Updating the metric consistent with Cisco is disabled.

- int [smi_ripng_show_ifname](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int start_index, int end_index, struct list *IfNameList, int(*funpointer)(struct list *IfNameList))

This function returns the configured interface information.

- int [smi_ripng_show_ifname_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *name, int start_index, int end_index, struct list *IfNameList, int(*funpointer)(struct list *IfNameList))

This function returns the configured interface information.

- int [smi_ripng_show_db](#) (struct smiclient_globals *azg, u_int32_t vr_id, int start_index, int end_index, struct list *DbList, int(*funpointer)(struct list *DbList))

This function returns the database information.

- int [smi_ripng_show_db_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, int start_index, int end_index, struct list *DbList, int(*funpointer)(struct list *DbList))

This function returns the database information.

- int [smi_show_ipv6_protocols_rip_api](#) (struct smiclient_globals *azg, u_int32_t vr_id, struct list *ProcList, int(*funpointer)(struct list *ProcList))

This function returns the RIPNG configuration information.

- int [smi_show_debugging_ipv6_rip_api](#) (struct smiclient_globals *azg, u_int32_t vr_id, struct smi_debug_ripng *debug)

This function returns the RIPNG configuration debug information.

- int [smi_ripng_debug](#) (struct smiclient_globals *azg, u_int32_t vr_id, int debug)

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

- int [smi_ripng_no_debug](#) (struct smiclient_globals *azg, u_int32_t vr_id, int debug)

Use this function to turn off debugging options for RIPng.

- int [smi_ripng_aggregate_add](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, int plen)

This function aggregates RIP routes. A RIPng router announces a route to the aggregated prefix with a metric of 1.

- int [smi_ripng_aggregate_delete](#) (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, int plen)

This function deletes aggregates advertising routes.

- int [smi_ripng_distribute_list_set_sdkapi](#) (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)

Use this function to set distribution list for RIPng.

- int [smi_ripng_distribute_list_unset_sdkapi](#) (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)

Use this function to unset distribution list for RIPng.

- int [smi_ripng_distribute_list_prefix_set_sdkapi](#) (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)

Use this function to set distribution list for RIPng.

- int [smi_ripng_distribute_list_prefix_unset_sdkapi](#) (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)

Use this function to unset distribution list for RIPng.

- int [smi_ripng_if_rmap_set_sdkapi](#) (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)

Use this function to set route map for RIPng.

- int [smi_ripng_if_rmap_unset_sdkapi](#) (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)

Use this function to unset route map list for RIPng.

- int [smi_ripng_if_rmap_config_check_sdkapi](#) (struct smiclient_globals *azg, char *ifname, u_int32_t vr_id, int instance, int *status)
- int [smi_ripng_address_family_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- int [smi_ripng_address_family_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- int [smi_ripng_passive_if_add_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *ifname)
- int [smi_ripng_passive_if_delete_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *ifname)
- int [smi_ripng_route_default_add_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- int [smi_ripng_route_default_delete_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- int [smi_ripng_offset_list_set_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *alist, char *direct_str, int metric, char *ifname)
- int [smi_ripng_offset_list_unset_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *alist, char *direct_str, int metric, char *ifname)
- int [smi_ripng_default_metric_set_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, int metric)
- int [smi_ripng_default_metric_unset_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- int [smi_ripng_redistribute_set_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type)
- int [smi_ripng_redistribute_unset_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type)
- int [smi_ripng_redistribute_metric_set_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type, int metric)
- int [smi_ripng_redistribute_rmap_set_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type, char *name)
- int [smi_ripng_redistribute_metric_rmap_set_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type, int metric, char *name)
- int [smi_ripng_timers_set_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, u_int32_t update, u_int32_t timeout, u_int32_t garbage)
- int [smi_ripng_timers_unset_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- int [smi_ripng_distance_set_default_vrf](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *distance_str)

- **int smi_ripng_distance_unset_default_vrf** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_recvbuf_size_set_vrf** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, u_int32_t bufsize)
- **int smi_ripng_recvbuf_size_unset_vrf** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_recvbuf_size_get_vrf** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, u_int32_t *bufsize)
- **int smi_ripng_route_type_delete_vrf** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type)
- **int smi_ripng_cisco_metric_behavior_set_vrf** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, u_char metric_type)
- **int smi_ripng_cisco_metric_behavior_unset_vrf** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_cisco_metric_behavior_get_vrf** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, u_char *metric_type)
- **int smi_ripng_distribute_list_set_sdkapi_vrf** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_distribute_list_unset_sdkapi_vrf** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_distribute_list_prefix_set_sdkapi_vrf** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_distribute_list_prefix_unset_sdkapi_vrf** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_if_rmap_set_sdkapi_vrf** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_if_rmap_unset_sdkapi_vrf** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_if_rmap_config_check_sdkapi_vrf** (struct smiclient_globals *azg, char *ifname, u_int32_t vr_id, char *vrf_name, int *status)
- **int smi_ripng_if_ipv6_router_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, char *tag)
- **int smi_ripng_if_ipv6_router_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
- **int smi_ripng_if_split_horizon_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
- **int smi_ripng_if_split_horizon_poisoned_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
- **int smi_ripng_if_split_horizon_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
- **int smi_ripng_if_metric_offset_set_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int metric)
- **int smi_ripng_if_metric_offset_unset_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
- **int smi_ripng_instance_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

- **int smi_ripng_instance_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)
- **int smi_ripng_enable_nbr_add_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, char *ifname)
- **int smi_ripng_enable_nbr_delete_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, char *ifname)
- **int smi_ripng_passive_if_add_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *ifname)
- **int smi_ripng_passive_if_delete_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *ifname)
- **int smi_ripng_route_add_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, int plen)
- **int smi_ripng_route_delete_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, int plen)
- **int smi_ripng_route_default_add_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)
- **int smi_ripng_route_default_delete_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)
- **int smi_ripng_offset_list_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *alist, char *direct_str, int metric, char *ifname)
- **int smi_ripng_offset_list_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *alist, char *direct_str, int metric, char *ifname)
- **int smi_ripng_default_metric_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, int metric)
- **int smi_ripng_default_metric_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)
- **int smi_ripng_redistribute_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type)
- **int smi_ripng_redistribute_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type)
- **int smi_ripng_redistribute_metric_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type, int metric)
- **int smi_ripng_redistribute_rmap_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type, char *name)
- **int smi_ripng_redistribute_metric_rmap_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type, int metric, char *name)
- **int smi_ripng_timers_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, u_int32_t update, u_int32_t timeout, u_int32_t garbage)
- **int smi_ripng_timers_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)
- **int smi_ripng_distance_set_default_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *distance_str)
- **int smi_ripng_distance_unset_default_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)

- int **smi_ripng_rcvbuf_size_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, u_int32_t bufsize)
- int **smi_ripng_rcvbuf_size_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)
- int **smi_ripng_route_type_delete_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, char *route_type)
- int **smi_ripng_cisco_metric_behavior_set_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, u_char metric_type)
- int **smi_ripng_cisco_metric_behavior_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance)
- int **smi_ripng_debug_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int debug)
- int **smi_ripng_no_debug_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int debug)
- int **smi_ripng_aggregate_add_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, int plen)
- int **smi_ripng_aggregate_delete_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int instance, struct pal_in6_addr *addr, int plen)
- int **smi_ripng_distribute_list_set_sdkapi_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)
- int **smi_ripng_distribute_list_unset_sdkapi_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)
- int **smi_ripng_distribute_list_prefix_set_sdkapi_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)
- int **smi_ripng_distribute_list_prefix_unset_sdkapi_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)
- int **smi_ripng_if_rmap_set_sdkapi_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)
- int **smi_ripng_if_rmap_unset_sdkapi_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, int instance)
- int **smi_ripng_enable_nbr_add_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, struct pal_in6_addr *addr, char *ifname)
- int **smi_ripng_enable_nbr_delete_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, struct pal_in6_addr *addr, char *ifname)
- int **smi_ripng_passive_if_add_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *ifname)
- int **smi_ripng_passive_if_delete_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *ifname)
- int **smi_ripng_route_add_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, struct pal_in6_addr *addr, int plen)
- int **smi_ripng_route_delete_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, struct pal_in6_addr *addr, int plen)
- int **smi_ripng_route_default_add_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- int **smi_ripng_route_default_delete_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)

- **int smi_ripng_offset_list_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *alist, char *direct_str, int metric, char *ifname)
- **int smi_ripng_offset_list_unset_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *alist, char *direct_str, int metric, char *ifname)
- **int smi_ripng_default_metric_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, int metric)
- **int smi_ripng_default_metric_unset_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_redistribute_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type)
- **int smi_ripng_redistribute_unset_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type)
- **int smi_ripng_redistribute_metric_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type, int metric)
- **int smi_ripng_redistribute_rmap_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type, char *name)
- **int smi_ripng_redistribute_metric_rmap_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type, int metric, char *name)
- **int smi_ripng_timers_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, u_int32_t update, u_int32_t timeout, u_int32_t garbage)
- **int smi_ripng_timers_unset_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_distance_set_default_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *distance_str)
- **int smi_ripng_distance_unset_default_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_recvbuf_size_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, u_int32_t bufsize)
- **int smi_ripng_recvbuf_size_unset_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_route_type_delete_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, char *route_type)
- **int smi_ripng_cisco_metric_behavior_set_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, u_char metric_type)
- **int smi_ripng_cisco_metric_behavior_unset_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name)
- **int smi_ripng_aggregate_add_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, struct pal_in6_addr *addr, int plen)
- **int smi_ripng_aggregate_delete_vrf_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *vrf_name, struct pal_in6_addr *addr, int plen)
- **int smi_ripng_distribute_list_set_sdkapi_vrf_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)

- int **smi_ripng_distribute_list_unset_sdkapi_vrf_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- int **smi_ripng_distribute_list_prefix_set_sdkapi_vrf_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- int **smi_ripng_distribute_list_prefix_unset_sdkapi_vrf_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- int **smi_ripng_if_rmap_set_sdkapi_vrf_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)
- int **smi_ripng_if_rmap_unset_sdkapi_vrf_validate** (struct smiclient_globals *azg, char *aclist, int type, char *ifname, u_int32_t vr_id, char *vrf_name)

2.1.1 Detailed Description

Provides API for managing RIPng. The API provided in this file forms the basis of ZebOS RIPng 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_ripng_aggregate_add (struct smiclient_globals * azg, u_int32_t vr_id, int instance, struct pal_in6_addr * addr, int plen)

This function aggregates RIP routes. A RIPng router announces a route to the aggregated prefix with a metric of 1. smi_ripng_aggregate_add

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *addr* The aggregated route prefix
- ← *plen* The prefix length for the 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

2.1.2.2 `int smi_ripng_aggregate_delete (struct smiclient_globals * azg, u_int32_t vr_id, int instance, struct pal_in6_addr * addr, int plen)`

This function deletes aggregates advertising routes. `smi_ripng_aggregate_delete`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *addr* The aggregated route prefix
- ← *plen* The prefix length for the 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

2.1.2.3 `int smi_ripng_cisco_metric_behavior_set (struct smiclient_globals * azg, u_int32_t vr_id, int instance, u_char metric_type)`

This function sets the metric update as Cisco; updating the metric consistent with Cisco is enabled. `smi_ripng_cisco_metric_behavior_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *metric_type* The metric type (1-RIPNG_CISCO_METRIC_COMPLIANT | 0-RIPNG_RFC_METRIC_COMPLIANT)

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_TYPE_INVALID

2.1.2.4 `int smi_ripng_cisco_metric_behavior_unset (struct smiclient_globals * azg, u_int32_t vr_id, int instance)`

This function unsets the metric update as Cisco. Updating the metric consistent with Cisco is disabled. `smi_ripng_cisco_metric_behavior_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of 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_METRIC_TYPE_INVALID

2.1.2.5 int smi_ripng_debug (struct smiclient_globals * *azg*, u_int32_t *vr_id*, int *debug*)

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

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *debug* Pass debug flag as following:
 - SMI_RIPNG_DBG_ALL - Debug all RIP information
 - SMI_RIPNG_DBG_EVENTS - Debug RIP events
 - SMI_RIPNG_DBG_PACKET - Debug RIP and NSM communications
 - SMI_RIPNG_DBG_PACKET_SEND - Debug sent packets
 - SMI_RIPNG_DBG_PACKET_RECV - Debug received packets
 - SMI_RIPNG_DBG_PACKET_DETAIL - Display detailed information for the sent and received packet
 - SMI_RIPNG_DBG_PACKET_SEND_DETAIL - Display detailed information for the sent packet
 - SMI_RIPNG_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

2.1.2.6 int smi_ripng_default_metric_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, int *instance*, int *metric*)

This function sets the routing protocol to use the specified metric value for all redistributed routes. smi_ripng_default_metric_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *metric* The default metric <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_METRIC_INVALID

2.1.2.7 `int smi_ripng_default_metric_unset (struct smiclient_globals * azg, u_int32_t vr_id, int instance)`

This function resets the metrics of the redistributed routes to the default value as 1.
 smi_ripng_default_metric_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of 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

2.1.2.8 `int smi_ripng_distance_set_default (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * distance_str)`

This function sets the administrative distance to the specified value. The default value for administrative distance is 120. smi_ripng_distance_set_default

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *distance_str* The 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

**2.1.2.9 int smi_ripng_distance_unset_default (struct smiclient_globals * azg,
 u_int32_t vr_id, int instance)**

This function resets the administrative distance to its default value as 120. smi_ripng_distance_unset_default

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router ID numeric <0-255>
 ← *instance* The number of 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

**2.1.2.10 int smi_ripng_distribute_list_prefix_set_sdkapi (struct
 smiclient_globals * azg, char * aclist, int type, char * ifname,
 u_int32_t vr_id, int instance)**

Use this function to set distribution list for RIPng. smi_ripng_distribute_list_prefix_set_sdkapi

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *aclist* ACCESS LIST NAME
 ← *TYPE* [in/out]
 ← *ifname*
 ← *vr_id*
 ← *instance*

Returns:

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

2.1.2.11 `int smi_ripng_distribute_list_prefix_unset_sdkapi (struct smiclient_globals * azg, char * aclist, int type, char * ifname, u_int32_t vr_id, int instance)`

Use this function to unset distribution list for RIPng. `smi_ripng_distribute_list_prefix_unset_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *aclist* ACCESS LIST NAME
- ← *TYPE* [in/out]
- ← *ifname*
- ← *vr_id*
- ← *instance*

Returns:

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

2.1.2.12 `int smi_ripng_distribute_list_set_sdkapi (struct smiclient_globals * azg, char * aclist, int type, char * ifname, u_int32_t vr_id, int instance)`

Use this function to set distribution list for RIPng. `smi_ripng_distribute_list_set_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *aclist* ACCESS LIST NAME
- ← *TYPE* [in/out]
- ← *ifname*
- ← *vr_id*
- ← *instance*

Returns:

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

2.1.2.13 `int smi_ripng_distribute_list_unset_sdkapi (struct smiclient_globals * azg, char * aclist, int type, char * ifname, u_int32_t vr_id, int instance)`

Use this function to unset distribution list for RIPng. smi_ripng_distribute_list_unset_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *aclist* ACCESS LIST NAME
- ← *TYPE* [in/out]
- ← *ifname*
- ← *vr_id*
- ← *instance*

Returns:

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

2.1.2.14 `int smi_ripng_enable_nbr_add (struct smiclient_globals * azg, u_int32_t vr_id, int instance, struct pal_in6_addr * addr, char * ifname)`

This function enables RIP routing on the specified neighbor. It is used for each connected point-to-point link. smi_ripng_enable_nbr_add

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *addr* The neighbor address
- ← *ifname* The 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_ADDRESS_INVALID
RIP_API_SET_ERR_NBR_STATIC_EXIST

2.1.2.15 `int smi_ripng_enable_nbr_delete (struct smiclient_globals * azg, u_int32_t vr_id, int instance, struct pal_in6_addr * addr, char * ifname)`

This function disables RIPng routing on the neighbor. `smi_ripng_enable_nbr_delete`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *addr* The neighbor address
- ← *ifname* The 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_ADDRESS_INVALID
 RIP_API_SET_ERR_NBR_STATIC_NOT_EXIST

2.1.2.16 `int smi_ripng_if_ipv6_router_set (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname, char * tag)`

This function enables RIPng routing on the interface. `smi_ripng_if_ipv6_router_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *ifname* Interface name string
- ← *tag* RIPng tag

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_INSTANCE_EXIST

2.1.2.17 `int smi_ripng_if_ipv6_router_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname)`

This function disables RIPng routing on the interface. `smi_ripng_if_ipv6_router_unset`

Parameters:

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

**2.1.2.18 int smi_ripng_if_metric_offset_set_sdkapi (struct smiclient_globals *
azg, u_int32_t vr_id, char * ifname, int metric)**

This function is used to set RIP metric offset. smi_ripng_if_metric_offset_set_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *ifname* Interface Name
- ← *metric* Metric offset numeric <1-16>

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_METRIC_INVALID

**2.1.2.19 int smi_ripng_if_metric_offset_unset_sdkapi (struct smiclient_globals
* azg, u_int32_t vr_id, char * ifname)**

This function is used to unset RIP metric offset. smi_ripng_if_metric_offset_unset_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *ifname* Interface name

Returns:

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

2.1.2.20 `int smi_ripng_if_rmap_set_sdkapi (struct smiclient_globals * azg, char * aclist, int type, char * ifname, u_int32_t vr_id, int instance)`

Use this function to set route map for RIPng. `smi_ripng_if_rmap_set_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *aclist* ACCESS LIST NAME
- ← *TYPE* [in/out]
- ← *ifname*
- ← *vr_id*
- ← *instance*

Returns:

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

2.1.2.21 `int smi_ripng_if_rmap_unset_sdkapi (struct smiclient_globals * azg, char * aclist, int type, char * ifname, u_int32_t vr_id, int instance)`

Use this function to unset route map list for RIPng. `smi_ripng_if_rmap_unset_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *aclist* ACCESS LIST NAME
- ← *TYPE* [in/out]
- ← *ifname*
- ← *vr_id*
- ← *instance*

Returns:

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

2.1.2.22 `int smi_ripng_if_split_horizon_poisoned_set (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname)`

This function enables RIP split-horizon poisoned reverse behavior. `smi_ripng_if_split_horizon_poisoned_set`

Parameters:

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

**2.1.2.23 int smi_ripng_if_split_horizon_set (struct smiclient_globals * azg,
u_int32_t vr_id, char * ifname)**

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

Parameters:

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

**2.1.2.24 int smi_ripng_if_split_horizon_unset (struct smiclient_globals * azg,
u_int32_t vr_id, char * ifname)**

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

Parameters:

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

2.1.2.25 `int smi_ripng_instance_set (struct smiclient_globals * azg, u_int32_t vr_id, int instance)`

This function establishes an instance of the IPv6 RIP router. The RIP routing process is enabled for IPv6. `smi_ripng_instance_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of 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

2.1.2.26 `int smi_ripng_instance_unset (struct smiclient_globals * azg, u_int32_t vr_id, int instance)`

This function removes an instance of the IPv6 router. The routing process is disabled. `smi_ripng_instance_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of 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

2.1.2.27 `int smi_ripng_no_debug (struct smiclient_globals * azg, u_int32_t vr_id, int debug)`

Use this function to turn off debugging options for RIPng. `smi_ripng_no_debug`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>

← *debug* Pass debug flag as following:

SMI_RIPNG_DBG_ALL - Debug all RIP information
 SMI_RIPNG_DBG_EVENTS - Debug RIP events
 SMI_RIPNG_DBG_PACKET - Debug RIP and NSM communications
 SMI_RIPNG_DBG_PACKET_SEND - Debug sent packets
 SMI_RIPNG_DBG_PACKET_RECV - Debug received packets
 SMI_RIPNG_DBG_PACKET_DETAIL - Display detailed information for the sent and received packet
 SMI_RIPNG_DBG_PACKET_SEND_DETAIL - Display detailed information for the sent packet
 SMI_RIPNG_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

2.1.2.28 `int smi_ripng_offset_list_set (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * alist, char * direct_str, 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_ripng_offset_list_set`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router ID numeric <0-255>
 ← *instance* The number of instance
 ← *alist* The access list name
 ← *direct_str* The packet direction
 ← *metric* The metric or offset
 ← *ifname* The 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

2.1.2.29 `int smi_ripng_offset_list_unset (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * alist, char * direct_str, int metric, char * ifname)`

This function removes the offset list. The offset value is removed from the routing metric. `smi_ripng_offset_list_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *alist* The access list name
- ← *direct_str* The packet direction
- ← *metric* The metric or offset
- ← *ifname* The 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_DIRECTION_INVALID
 RIP_API_SET_ERR_OFFSET_LIST_NOT_EXIST

2.1.2.30 `int smi_ripng_passive_if_add (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * ifname)`

This function suppresses RIP updates. RIP broadcast on the interface is blocked. `smi_ripng_passive_if_add`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *ifname* The 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

2.1.2.31 `int smi_ripng_passive_if_delete (struct smiclient_globals * azg,
u_int32_t vr_id, int instance, char * ifname)`

This function disables blocking RIP broadcast on the interface. `smi_ripng_passive_if_delete`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *ifname* The 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

2.1.2.32 `int smi_ripng_rcvbuf_size_set (struct smiclient_globals * azg,
u_int32_t vr_id, int instance, u_int32_t bufsize)`

This function sets the specified size of the RIP UDP buffer. `smi_ripng_rcvbuf_size_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *bufsize* The size of the receiving buffer

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

2.1.2.33 `int smi_ripng_rcvbuf_size_unset (struct smiclient_globals * azg,
u_int32_t vr_id, int instance)`

This function resets the size of the RIP UDP buffer to the default value as (1024*192). `smi_ripng_rcvbuf_size_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of 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

2.1.2.34 **int smi_ripng_redistribute_metric_rmap_set (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * route_type, int metric, char * name)**

This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIPng. It sets the metric for redistribution and uses the route map configured for redistribution of routes. smi_ripng_redistribute_metric_rmap_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *route_type* The route type
 String ("kernel" | "connected" |
 "static" | "ospf" | "isis" | "bgp")
- ← *metric* The metric value numeric <0-16>
- ← *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

2.1.2.35 **int smi_ripng_redistribute_metric_set (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * route_type, int metric)**

This redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIPng and sets the metric for redistribution of routes. smi_ripng_redistribute_metric_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *route_type* 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

**2.1.2.36 int smi_ripng_redistribute_rmap_set (struct smiclient_globals * azg,
u_int32_t vr_id, int instance, char * route_type, char * name)**

This function redistributes routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIPng. It uses the route map configured for redistribution of routes. smi_ripng_redistribute_rmap_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *route_type* 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_ROUTE_PROTO_INVALID
RIP_API_SET_ERR_INVALID_VALUE

2.1.2.37 `int smi_ripng_redistribute_set (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * route_type)`

This function enables redistributing of routes learned from other routing protocols (OSPF, IS-IS, BGP) to RIPng. `smi_ripng_redistribute_set`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID numeric <0-255>
- ← **instance** The number of instance
- ← **route_type** 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

2.1.2.38 `int smi_ripng_redistribute_unset (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * route_type)`

This function disables the redistribution of routes into RIPng, and removes the configured route map and metric. `smi_ripng_redistribute_unset`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID numeric <0-255>
- ← **instance** The number of instance
- ← **route_type** 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

2.1.2.39 `int smi_ripng_route_default_add (struct smiclient_globals * azg, u_int32_t vr_id, int instance)`

This function configures a static route for advertisement through RIP explicitly. `smi_ripng_route_add`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *addr* The address of the source prefix
- ← *plen* 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

`smi_ripng_route_default_add`

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

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of 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

2.1.2.40 `int smi_ripng_route_default_delete (struct smiclient_globals * azg, u_int32_t vr_id, int instance)`

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

Parameters:

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

- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of 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

2.1.2.41 int smi_ripng_route_type_delete (struct smiclient_globals * azg, u_int32_t vr_id, int instance, char * route_type)

This function clears specified data from the RIPng routing table. smi_ripng_route_type_delete

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *route_type* 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_PROTO_INVALID

2.1.2.42 int smi_ripng_show_db (struct smiclient_globals * azg, u_int32_t vr_id, int start_index, int end_index, struct list * DbList, int(*) (struct list * DbList) funpointer)

This function returns the database information. smi_ripng_show_db

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *start_index*
- ← *end_index*
- *DbList* Returned list

Returns:

RESULT_OK on success, otherwise the following error code
RESULT_ERROR

2.1.2.43 `int smi_ripng_show_db_vrf (struct smiclient_globals * azg, u_int32_t vr_id, char * vrf_name, int start_index, int end_index, struct list * DbList, int(*) (struct list * DbList) funpointer)`

This function returns the database information. smi_ripng_show_db_vrf

Parameters:

← *azg* Pointer to the SMI client global structure
← *vr_id* Virtual Router ID numeric <0-255>
← *vrf_name* VRF name <all/default/VRF_NAME>
← *start_index*
← *end_index*
→ *DbList* Returned list

Returns:

RESULT_OK on success, otherwise the following error code
RESULT_ERROR

2.1.2.44 `int smi_ripng_show_ifname (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int start_index, int end_index, struct list * IfNameList, int(*) (struct list * IfNameList) funpointer)`

This function returns the configured interface information. smi_ripng_show_ifname

Parameters:

← *azg* Pointer to the SMI client global structure
← *vr_id* Virtual Router ID numeric <0-255>
← *name* Interface name
← *start_index*
← *end_index*
→ *IfNameList* Returned list

Returns:

RESULT_OK on success, otherwise the following error code
RESULT_ERROR

2.1.2.45 `int smi_ripng_show_ifname_vrf (struct smiclient_globals * azg, u_int32_t vr_id, char * vrf_name, char * name, int start_index, int end_index, struct list * IfNameList, int(*)(struct list *IfNameList) funpointer)`

This function returns the configured interface information. `smi_ripng_show_ifname_vrf`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *vrf_name* VRF name <all/default/VRF_NAME>
- ← *name* Interface name
- ← *start_index*
- ← *end_index*
- *IfNameList* Returned list

Returns:

RESULT_OK on success, otherwise the following error code
RESULT_ERROR

2.1.2.46 `int smi_ripng_timers_set (struct smiclient_globals * azg, u_int32_t vr_id, int instance, u_int32_t update, u_int32_t timeout, u_int32_t garbage)`

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

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of instance
- ← *update* The number of update timer seconds. Default is 30
- ← *timeout* The number of timeout timer seconds. Default is 180
- ← *garbage* The number of garbage timer seconds. Default is 120

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

2.1.2.47 int smi_ripng_timers_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, int *instance*)

This function resets three timers to default values. Update timer to 30 seconds. Timeout timer to 180 seconds. Garbage timer to 120 seconds. smi_ripng_timers_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *instance* The number of 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

2.1.2.48 int smi_show_debugging_ipv6_rip_api (struct smiclient_globals * *azg*, u_int32_t *vr_id*, struct smi_debug_ripng * *debug*)

This function returns the RIPNG configuration debug information. smi_show_debugging_ipv6_rip_api

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-252>
- *debug* Debug info

Returns:

RESULT_OK on success, otherwise the following error code
 RESULT_ERROR

2.1.2.49 int smi_show_ipv6_protocols_rip_api (struct smiclient_globals * *azg*, u_int32_t *vr_id*, struct list * *ProcList*, int(*) (struct list * *ProcList*) *funpointer*)

This function returns the RIPNG configuration information. smi_show_ipv6_protocols_rip_api

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-252>
- *ProcList* Returned list

Returns:

RESULT_OK on success, otherwise the following error code
RESULT_ERROR

Index

- smi_ripng.h, [3](#)
 - smi_ripng_aggregate_add, [13](#)
 - smi_ripng_aggregate_delete, [13](#)
 - smi_ripng_cisco_metric_behavior_set, [14](#)
 - smi_ripng_cisco_metric_behavior_unset, [14](#)
 - smi_ripng_debug, [15](#)
 - smi_ripng_default_metric_set, [15](#)
 - smi_ripng_default_metric_unset, [16](#)
 - smi_ripng_distance_set_default, [16](#)
 - smi_ripng_distance_unset_default, [17](#)
 - smi_ripng_distribute_list_prefix_set_sdkapi, [17](#)
 - smi_ripng_distribute_list_prefix_unset_sdkapi, [17](#)
 - smi_ripng_distribute_list_set_sdkapi, [18](#)
 - smi_ripng_distribute_list_unset_sdkapi, [18](#)
 - smi_ripng_enable_nbr_add, [19](#)
 - smi_ripng_enable_nbr_delete, [19](#)
 - smi_ripng_if_ipv6_router_set, [20](#)
 - smi_ripng_if_ipv6_router_unset, [20](#)
 - smi_ripng_if_metric_offset_set_sdkapi, [21](#)
 - smi_ripng_if_metric_offset_unset_sdkapi, [21](#)
 - smi_ripng_if_rmap_set_sdkapi, [21](#)
 - smi_ripng_if_rmap_unset_sdkapi, [22](#)
 - smi_ripng_if_split_horizon_poisoned_set, [22](#)
 - smi_ripng_if_split_horizon_set, [23](#)
 - smi_ripng_if_split_horizon_unset, [23](#)
 - smi_ripng_instance_set, [23](#)
 - smi_ripng_instance_unset, [24](#)
 - smi_ripng_no_debug, [24](#)
 - smi_ripng_offset_list_set, [25](#)
 - smi_ripng_offset_list_unset, [25](#)
 - smi_ripng_passive_if_add, [26](#)
 - smi_ripng_passive_if_delete, [26](#)
 - smi_ripng_recvbuf_size_set, [27](#)
 - smi_ripng_recvbuf_size_unset, [27](#)
 - smi_ripng_redistribute_metric_rmap_set, [28](#)
 - smi_ripng_redistribute_metric_set, [28](#)
 - smi_ripng_redistribute_rmap_set, [29](#)
 - smi_ripng_redistribute_set, [29](#)
 - smi_ripng_redistribute_unset, [30](#)
 - smi_ripng_route_default_add, [30](#)
 - smi_ripng_route_default_delete, [31](#)
 - smi_ripng_route_type_delete, [32](#)
 - smi_ripng_show_db, [32](#)
 - smi_ripng_show_db_vrf, [33](#)
 - smi_ripng_show_ifname, [33](#)
 - smi_ripng_show_ifname_vrf, [33](#)
 - smi_ripng_timers_set, [34](#)
 - smi_ripng_timers_unset, [34](#)
 - smi_show_debugging_ipv6_rip_api, [35](#)
 - smi_show_ipv6_protocols_rip_api, [35](#)
- smi_ripng_aggregate_add
 - smi_ripng.h, [13](#)
- smi_ripng_aggregate_delete
 - smi_ripng.h, [13](#)
- smi_ripng_cisco_metric_behavior_set
 - smi_ripng.h, [14](#)
- smi_ripng_cisco_metric_behavior_unset
 - smi_ripng.h, [14](#)
- smi_ripng_debug
 - smi_ripng.h, [15](#)
- smi_ripng_default_metric_set
 - smi_ripng.h, [15](#)
- smi_ripng_default_metric_unset
 - smi_ripng.h, [16](#)
- smi_ripng_distance_set_default
 - smi_ripng.h, [16](#)

- smi_ripng_distance_unset_default
smi_ripng.h, [17](#)
- smi_ripng_distribute_list_prefix_set_
- sdkapi
smi_ripng.h, [17](#)
- smi_ripng_distribute_list_prefix_unset_
- sdkapi
smi_ripng.h, [17](#)
- smi_ripng_distribute_list_set_sdkapi
smi_ripng.h, [18](#)
- smi_ripng_distribute_list_unset_sdkapi
smi_ripng.h, [18](#)
- smi_ripng_enable_nbr_add
smi_ripng.h, [19](#)
- smi_ripng_enable_nbr_delete
smi_ripng.h, [19](#)
- smi_ripng_if_ipv6_router_set
smi_ripng.h, [20](#)
- smi_ripng_if_ipv6_router_unset
smi_ripng.h, [20](#)
- smi_ripng_if_metric_offset_set_sdkapi
smi_ripng.h, [21](#)
- smi_ripng_if_metric_offset_unset_sdkapi
smi_ripng.h, [21](#)
- smi_ripng_if_rmap_set_sdkapi
smi_ripng.h, [21](#)
- smi_ripng_if_rmap_unset_sdkapi
smi_ripng.h, [22](#)
- smi_ripng_if_split_horizon_poisoned_set
smi_ripng.h, [22](#)
- smi_ripng_if_split_horizon_set
smi_ripng.h, [23](#)
- smi_ripng_if_split_horizon_unset
smi_ripng.h, [23](#)
- smi_ripng_instance_set
smi_ripng.h, [23](#)
- smi_ripng_instance_unset
smi_ripng.h, [24](#)
- smi_ripng_no_debug
smi_ripng.h, [24](#)
- smi_ripng_offset_list_set
smi_ripng.h, [25](#)
- smi_ripng_offset_list_unset
smi_ripng.h, [25](#)
- smi_ripng_passive_if_add
smi_ripng.h, [26](#)
- smi_ripng_passive_if_delete
smi_ripng.h, [26](#)
- smi_ripng_rcvbuf_size_set
smi_ripng.h, [27](#)
- smi_ripng_rcvbuf_size_unset
smi_ripng.h, [27](#)
- smi_ripng_redistribute_metric_rmap_set
smi_ripng.h, [28](#)
- smi_ripng_redistribute_metric_set
smi_ripng.h, [28](#)
- smi_ripng_redistribute_rmap_set
smi_ripng.h, [29](#)
- smi_ripng_redistribute_set
smi_ripng.h, [29](#)
- smi_ripng_redistribute_unset
smi_ripng.h, [30](#)
- smi_ripng_route_default_add
smi_ripng.h, [30](#)
- smi_ripng_route_default_delete
smi_ripng.h, [31](#)
- smi_ripng_route_type_delete
smi_ripng.h, [32](#)
- smi_ripng_show_db
smi_ripng.h, [32](#)
- smi_ripng_show_db_vrf
smi_ripng.h, [33](#)
- smi_ripng_show_ifname
smi_ripng.h, [33](#)
- smi_ripng_show_ifname_vrf
smi_ripng.h, [33](#)
- smi_ripng_timers_set
smi_ripng.h, [34](#)
- smi_ripng_timers_unset
smi_ripng.h, [34](#)
- smi_show_debugging_ipv6_rip_api
smi_ripng.h, [35](#)
- smi_show_ipv6_protocols_rip_api
smi_ripng.h, [35](#)