ZebOS-XP VRRP SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:57 2015

Contents

Chapter 1

File Index

1.1 File List

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

Chapter 2

File Documentation

2.1 smi_vrrp.h File Reference

Provides API for managing VRRP The Virtual Router Redundancy Protocol (VRRP) allows a virtual router composed of two or more VRRP routers on the same subnet to prevent failure by providing at least one Standby virtual router if the Master virtual router fails. It is designed to eliminate the single point of failure most common in a static default routed environment. `#include "smi_client.h"`

`#include "smi_vrrp_msg.h"`

Defines

- #define **VRRP_SHOW_GLOBAL** (1 << SMI_VRRP_CTYPE_SHOW_-GLOBAL)
- #define **SMI_VRRP_ADV_INTERVAL_MIN** 5
- #define **SMI_VRRP_ADV_INTERVAL_MAX** 4095
- #define **SMI_VRRP_PRIORITY_MIN** 1
- #define **SMI_VRRP_PRIORITY_MAX** 255
- #define **SMI_VRRP_VRID_MIN** 1
- #define **SMI_VRRP_VRID_MAX** 255

Functions

- int smi_vrrp_set_session_by_ifname_ipv4_sdkapi (struct smiclient_globals ∗azg, int ipi_vrid, int vr_id, char ∗ifname)

  Updates or creates a new VRRP session on the given interface and allocates resources for the session.

- int smi_vrrp_set_session_by_ifname_ipv6_sdkapi (struct smiclient_globals ∗azg, int ipi_vrid, int vr_id, char ∗ifname)

Updates or creates a new VRRP session on the given interface and allocates resources for the session.

- int smi_vrrp_set_session_by_vlanid_ipv4_sdkapi (struct smiclient_globals ∗azg, int ipi_vrid, int vr_id, int vlan_id)

  Updates or creates a new VRRP session on the given interface and allocates resources for the session.

- int smi_vrrp_set_session_by_vlanid_ipv6_sdkapi (struct smiclient_globals ∗azg, int ipi_vrid, int vr_id, int vlan_id)

  Updates or creates a new VRRP session on the given interface and allocates resources for the session.

- int smi_vrrp_api_del_session_by_ifname (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t af_type, int vr_id, char ∗ifname)

  This function deletes the VRRP session from a specific interface associated with the provided Virtual Router ID (VR ID) Only disabled sessions can be deleted. This function deallocates the memory for the session.

- int smi_vrrp_api_virtual_ip (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t af_type, int vr_id, u_int32_t ifindex, u_int8_t ∗vip_addr, bool_t is_owner)

  This function configures the virtual IP address for a session. It receives the IP address and VR ID from the CM(Configuration Management) application, and configures the appropriate session. This function accepts the IP address from the CM application in whatever representation it uses. It assumes that the IP application also accepts addresses using this representation.

- int smi_vrrp_api_monitored_circuit (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t af_type, int vr_id, u_int32_t ifindex, char ∗if_str, int priority_delta)

  This function sets the monitored circuit for a VRRP session. It handles circuit failover for VRRP session on an interface for a monitored circuit.

- int smi_vrrp_api_priority (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t af_type, int vr_id, u_int32_t ifindex, int prio)

  This function enables the configuration of the priority of the VRRP router for a session. It receives the priority from the CM application, and configures the appropriate session. If the router is the default Master for the session (for example, it owns the virtual IP address), the priority must be configured as 255. If the router is a backup for the session, the priority must be less than 255. The default priority is 100.

- int smi_vrrp_api_unset_priority (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t af_type, int vr_id, u_int32_t ifindex)

  This function sets the priority of the VRRP router to the default value (VRRP_DEFAULT_IP_OWNER_PRIORITY or VRRP_DEFAULT_NON_IP_OWNER_PRIORITY).

- int smi_vrrp_api_switch_back_delay (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t af_type, int vr_id, u_int32_t ifindex, int switch_back_delay)

This function sets a switch-back delay timer for the master VRRP router. This feature prevents the original master VRRP router from transitioning back to the master state after coming back online until the configured delay timer has expired.

- int smi_vrrp_api_unset_switch_back_delay (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t af_type, int vr_id, u_int32_t ifindex)

  This function sets a switch-back delay to default value 0.

- int smi_vrrp_api_advt_interval (struct smiclient_globals ∗azg, int ipi_vrid, u_-int8_t af_type, int vr_id, u_int32_t ifindex, int interval)

  This function configures the advertisement interval of a virtual router. This is the length of time, in seconds, between each advertisement sent from the master to its backup(s).The master virtual router sends VRRP advertisements to other VRRP routers in the same group.

- int smi_vrrp_api_unset_advt_interval (struct smiclient_globals ∗azg, int ipi_-vrid, u_int8_t af_type, int vr_id, u_int32_t ifindex)

  This function restores the advertisement interval to its default value 1 second.

- int smi_vrrp_api_preempt_mode (struct smiclient_globals ∗azg, int ipi_vrid, u_-int8_t af_type, int vr_id, u_int32_t ifindex, int mode)

  This function enables or disables the preempt mode for a session. It receives the mode and VR ID from the CM(Configuration Management) application, and configures the appropriate session. This value must be configured the same for all VRRP routers participating in a session. The default value for this variable is TRUE.

- int smi_vrrp_api_accept_mode (struct smiclient_globals ∗azg, int ipi_vrid, u_-int8_t af_type, int vr_id, u_int32_t ifindex, int mode)

  This function sets the accept mode for a VRRP session when VRPP V3 is enabled.

- int smi_vrrp_api_enable_session (struct smiclient_globals ∗azg, int ipi_vrid, u_-int8_t af_type, int vr_id, u_int32_t ifindex)

  This function enables a VRRP session and sets default values that have not been set. It receives the VR ID from the CM application, and calls the enable function vrrp_-enable_sess defined in the VRRP module.

- int smi_vrrp_api_disable_session (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t af_type, int vr_id, u_int32_t ifindex)

  This function disables a VRRP session. It receives the VR ID from the CM(Configuration Management) application, and calls the disable function vrrp_-shutdown_sess defined in the VRRP module.

- int smi_vrrp_api_set_vmac_status_sdkapi (struct smiclient_globals ∗azg, int new_vmac_stats)

  This function enables or disables Virtual MAC (VMAC). It affects all VRRP groups in a router. On a single network segment, multiple VRRP groups can be configured, each using a different VMAC. The use of VMAC addressing allows for faster switchover when a backup router assumes the master role.

- int smi_vrrp_get_notify (struct smiclient_globals ∗azg, int ipi_vrid, int ∗notify)

  This function gets the the flag status for trap generation.

- int smi_vrrp_get_oper_state (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗oper_state)

  This function gets the current state of the virtual router.

- int smi_vrrp_get_oper_priority (struct smiclient_globals ∗azg, int ipi_vrid, u_-int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗oper_priority)

  This function gets the priority to be used for the virtual router master election process. Higher values imply higher priority.

- int smi_vrrp_get_oper_addr_count (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗oper_count)

  This function gets the number of IP addresses that are associated with this virtual router. This number is equal to the number of rows in the vrrpv3AssociatedAddrTable that correspond to a given ifIndex/VRID/IP version.

- int smi_vrrp_get_oper_master_ipaddr (struct smiclient_globals ∗azg, int ipi_-vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int8_t ∗ipaddr)

  This function gets the master IP address of the VRRP virtual router.

- int smi_vrrp_get_oper_primary_ipaddr (struct smiclient_globals ∗azg, int ipi_-vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int8_t ∗ipaddr)

  This function gets IP address that becomes the 'vr-rpv3OperationsMasterIpAddr',where there is more than one IP Address (associated IP addresses) for a given 'ifIndex'. In case where there is more than one IP address (associated IP address)for a given 'ifindex', it is used to specify the IP address that will become up.

- int smi_vrrp_get_oper_adv_interval (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗adv_int)

  This function gets the time interval, in centiseconds between sending advertisement messages. Only the master router sends VRRP advertisements.

- int smi_vrrp_get_oper_preempt_mode (struct smiclient_globals ∗azg, int ipi_-vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗preempt_mode)

  This function gets whether a higher priority virtual router will preempt a lower priority master.

- int smi_vrrp_get_oper_accept_mode (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int32_t ∗accept_mode)

  This function gets whether a virtual router in Master state will accept packets addressed to the address owner's IPv6 address as its own if it is not the IPv6 address owner.

- int smi_vrrp_get_oper_uptime (struct smiclient_globals ∗azg, int ipi_vrid, u_-int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int32_t ∗oper_uptime)

This function gets the amount of time, in TimeTicks (hundredth of a second), since this virtual router transitioned out of 'initialize'.

- int smi_vrrp_get_oper_storage_type (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int32_t ∗storage_type)

  This function gets the storage type for this conceptual row.

- int smi_vrrp_get_oper_rowstatus (struct smiclient_globals ∗azg, int ipi_vrid, u_-int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗rowstatus)

  This function gets the value of RowStatus variable should be used in accordance to installation and removal conventions for conceptual rows.

- int smi_vrrp_get_asso_storage_type (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int8_t ∗ipaddr, u_int32_t ∗type)

  This function gets the storage type for this conceptual row IP address that is associated with a virtual router.

- int smi_vrrp_get_asso_ipaddr_rowstatus (struct smiclient_globals ∗azg, int ipi_-vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int8_t ∗ipaddr, int ∗asso_rowstatus)

  This function gets the value of RowStatus variable, used according to installation and removal conventions for conceptual rows.

- int smi_vrrp_get_checksum_errors (struct smiclient_globals ∗azg, int ipi_vrid, int ∗chksum_error)

  This function gets the total number of VRRP packets received with an invalid VRRP checksum value.

- int smi_vrrp_get_version_errors (struct smiclient_globals ∗azg, int ipi_vrid, int ∗version_error)

  This function gets the total number of VRRP packets received with unknown or unsupported version number.

- int smi_vrrp_get_vrid_errors (struct smiclient_globals ∗azg, int ipi_vrid, int ∗vrid_error)

  This function gets the total number of VRRP packets received with a VRID that is not valid for any virtual router on this router.

- int smi_vrrp_get_stats_master_transitions (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_master_-transitions)

  This function gets the total number of times that this virtual router's state has transitioned to MASTER.

- int smi_vrrp_get_stats_rcvd_advertisements (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗rcvd_advert)

  This function gets the total number of VRRP advertisements received by this virtual router.

- int smi_vrrp_get_stats_adv_interval_errors (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_adv_-interval)

  This function gets the total number of VRRP advertisement packets received for which the advertisement interval is different from the vrrpv3OperationsAdvInterval configured on this virtual router.

- int smi_vrrp_get_stats_ip_ttl_errors (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_ip_ttl_errors)

  This function gets the total number of VRRP packets received by the Virtual router with IPv4 TTL (for VRRP over IPv4) or IPv6 Hop Limit (for VRRP over IPv6) not equal to 255.

- int smi_vrrp_get_stats_rcvd_pri_zero_packets (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_rcvd_-pri_zero_packets)

  This function gets the total number of VRRP packets received by the virtual router with a priority of '0'.

- int smi_vrrp_get_stats_sent_pri_zero_packets (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_sent_pri_-zero_packets)

  This function gets the total number of VRRP packets sent by the virtual router with a priority of '0'.

- int smi_vrrp_get_stats_rcvd_invalid_type_pkts (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_rcvd_-invalid_type_pkts)

  This function gets the number of VRRP packets received by the virtual router with an invalid value in the 'type' field.

- int smi_vrrp_get_stats_address_list_errors (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_address_-list_errors)

  This function gets the total number of packets received for which the address list does not match the locally configured list for the virtual router.

- int smi_vrrp_get_stats_packet_length_errors (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_packet_-length_error)

  This function gets the total number of packets received with a packet length less than the length of the VRRP header.

- int smi_vrrp_get_stats_rcvd_invalid_authentications (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_-rcvd_invalid_authentications)

  This function gets the trtal number of packets received with an unknown authentication type. char ∗ifname);.

- int smi_vrrp_get_stats_discontinuity_time (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_-discontinuity_time)

  This function gets the value of sysUpTime on the most recent occasion at which any one or more of this entry's counters suffered a discontinuity.

- int smi_vrrp_get_stats_refresh_rate (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int ∗stats_refresh_rate)

  This function gets the minimum reasonable polling interval for corresponding entry.It provides an indication of the minimum amount of time required to update the counters in corresponding entry.

- int smi_vrrp_set_notify (struct smiclient_globals ∗azg, int ipi_vrid, int notify)

  This function sets the value indicating whether this router generates SNMP notifica-tions.

- int smi_vrrp_set_oper_primary_ipaddr (struct smiclient_globals ∗azg, int ipi_-vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int8_t ∗ipaddr)

  This function sets the primary IP address of the VRRP virtual router, if multiple asso-ciated IP addresses are present.

- int smi_vrrp_set_oper_adv_interval (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int oper_adv_interval)

  This function sets the time interval between sending advertisement messages.

- int smi_vrrp_set_oper_accept_mode (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int oper_accept_mode)

  This function sets the accept mode (IPv6 only).

- int smi_vrrp_set_oper_storage_type (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int oper_storage_type)

  This function sets the value of the storage type for this VRRP virtual router.

- int smi_vrrp_set_asso_storage_type (struct smiclient_globals ∗azg, int ipi_vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int8_t ∗ipaddr, int asso_-storage_type)

  This function sets the value of the storage type for associated IP address entry.

- int smi_vrrp_set_oper_rowstatus (struct smiclient_globals ∗azg, int ipi_vrid, u_-int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, int oper_rowstatus)

  This function sets the operational row status of the VRRP virtual router.

- int smi_vrrp_set_asso_ipaddr_rowstatus (struct smiclient_globals ∗azg, int ipi_-vrid, u_int8_t aftype, u_int8_t vr_id, u_int32_t ifindex, u_int8_t ∗ipaddr, int asso_ipaddr_rowstatus)

  This function sets the row status of the associated IP address entry.

- int **_merge_vrrp_stats_list** (struct list ∗listDest, struct list ∗listSrc)
- int **_merge_vrrp_session_list** (struct list ∗listDest, struct list ∗listSrc)
- s_int32_t smi_vrrp_show_session_all (struct smiclient_globals ∗azg, struct vrrpGlobal ∗globalData, struct list ∗vrrpSessList, int(∗funPointer)(struct list ∗vrrpSessList))

  Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) characteristics of all VRRP sessions.

- s_int32_t smi_vrrp_show_session_v6 (struct smiclient_globals ∗azg, u_int8_t vr_id, char ∗ifname, struct vrrpGlobal ∗globalData, struct list ∗vrrpSessList, int(∗funPointer)(struct list ∗vrrpSessList))

  Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) characteristics of given VRRP session.

- s_int32_t smi_vrrp_show_session_v4 (struct smiclient_globals ∗azg, u_int8_t vr_id, char ∗ifname, struct vrrpGlobal ∗globalData, struct list ∗vrrpSessList, int(∗funPointer)(struct list ∗vrrpSessList))

  Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) characteristics of given VRRP session.

- s_int32_t smi_vrrp_show_statistics_all (struct smiclient_globals ∗azg, struct vrrpGlobal ∗globalData, struct list ∗vrrpStatList, int(∗funPointer)(struct list ∗vrrpStatList))

  Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) statistics of all VRRP sessions.

- s_int32_t smi_vrrp_show_statistics_v6 (struct smiclient_globals ∗azg, u_int8_t vr_id, char ∗ifname, struct vrrpGlobal ∗globalData, struct list ∗vrrpStatList, int(∗funPointer)(struct list ∗vrrpStatList))

  Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) statistics of given VRRP session.

- s_int32_t smi_vrrp_show_statistics_v4 (struct smiclient_globals ∗azg, u_int8_t vr_id, char ∗ifname, struct vrrpGlobal ∗globalData, struct list ∗vrrpStatList, int(∗funPointer)(struct list ∗vrrpStatList))

  Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) statistics of given VRRP session.

- int smi_vrrp_debug (struct smiclient_globals ∗azg, int vr_id, int debug)

  Use this function to specify debugging options for VRRP.

- int smi_vrrp_no_debug (struct smiclient_globals ∗azg, int vr_id, int debug)

  Use this function to disable debugging.

2.1.1 Detailed Description

Provides API for managing VRRP The Virtual Router Redundancy Protocol (VRRP) allows a virtual router composed of two or more VRRP routers on the same subnet to prevent failure by providing at least one Standby virtual router if the Master virtual router fails. It is designed to eliminate the single point of failure most common in a static default routed environment.

2.1.2 Function Documentation

2.1.2.1 int smi_vrrp_api_accept_mode (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*, int *mode*)

This function sets the accept mode for a VRRP session when VRPP V3 is enabled. smi_vrrp_api_accept_mode

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***mode*** Accept mode for the VRRP session (TRUE | FALSE)

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_ENABLED
VRRP_API_SET_ERR_IPV4_ENABLED

2.1.2.2 int smi_vrrp_api_advt_interval (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*, int *interval*)

This function configures the advertisement interval of a virtual router. This is the length of time, in seconds, between each advertisement sent from the master to its backup(s).The master virtual router sends VRRP advertisements to other VRRP routers in the same group. smi_vrrp_api_advt_interval

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***interval*** Advertisement interval in seconds numeric <1-10>

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_ADVT_INTVL_NOT_FACTOR_OF_FIVE
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_ENABLED

2.1.2.3 int smi_vrrp_api_del_session_by_ifname (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, char ∗ *ifname*)

This function deletes the VRRP session from a specific interface associated with the provided Virtual Router ID (VR ID) Only disabled sessions can be deleted. This function deallocates the memory for the session. smi_vrrp_api_del_session_by_ifname

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifname*** Interface name

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_INTERFACE
VRRP_API_SET_ERR_NO_EXIST
VRRP_API_SET_ERR_ENABLED

2.1.2.4 int smi_vrrp_api_disable_session (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*)

This function disables a VRRP session. It receives the VR ID from the CM(Configuration Management) application, and calls the disable function vrrp_-shutdown_sess defined in the VRRP module. smi_vrrp_api_disable_session

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_DISABLED

2.1.2.5 int smi_vrrp_api_enable_session (struct smiclient_globals $*$ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*)

This function enables a VRRP session and sets default values that have not been set. It receives the VR ID from the CM application, and calls the enable function vrrp_-enable_sess defined in the VRRP module. smi_vrrp_api_enable_session

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_CONFIG_UNSET
VRRP_API_SET_ERR_PRIO_MISMATCH
VRRP_API_SET_ERR_ENABLE

2.1.2.6 int smi_vrrp_api_monitored_circuit (struct smiclient_globals $*$ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*, char $*$ *if_str*, int *priority_delta*)

This function sets the monitored circuit for a VRRP session. It handles circuit failover for VRRP session on an interface for a monitored circuit. smi_vrrp_api_monitored_-circuit

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***if_str*** Interface name

← ***priority_delta*** Priority delta (VRRP incremets/decrements) its priority value over the VRID by specified delta

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_ENABLED
VRRP_API_SET_ERR_NO_SUCH_INTERFACE
VRRP_API_ERR_CANNOT_APPLY_INT_TRACK_ON_VRRP_BINDED_-INT
VRRP_API_SET_ERR_VIP_UNSET
VRRP_API_SET_ERR_CANNOT_TRACK_OBJECT
VRRP_API_SET_ERR_PRIORDELTA_MUST_LSTHAN_CONF_PRIO

2.1.2.7 int smi_vrrp_api_preempt_mode (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*, int *mode*)

This function enables or disables the preempt mode for a session. It receives the mode and VR ID from the CM(Configuration Management) application, and configures the appropriate session. This value must be configured the same for all VRRP routers participating in a session. The default value for this variable is TRUE. smi_vrrp_api_-preempt_mode

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***mode*** The preempt mode for the VRRP session.A value of 1 results in the preempt mode being set to PAL_TRUE (enabled). Any other value of mode results in the preempt mode being disabled (PAL_FALSE)

Returns:

VRRP_OK on success, otherwise one of the following errors VRRP_API_SET_-ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_ENABLED

2.1.2.8 int smi_vrrp_api_priority (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*, int *prio*)

This function enables the configuration of the priority of the VRRP router for a session. It receives the priority from the CM application, and configures the appropriate session.

If the router is the default Master for the session (for example, it owns the virtual IP address), the priority must be configured as 255. If the router is a backup for the session, the priority must be less than 255. The default priority is 100. smi_vrrp_api_-priority

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***prio*** Priority for a VRRP session (1 -255). A value of 255 can be assigned only to the session owner

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_VIP_UNSET
VRRP_API_SET_ERR_ENABLED
VRRP_API_SET_ERR_PRIO_CANT_255
VRRP_API_SET_ERR_PRIO_MUST_255
VRRP_API_SET_ERR_PRIO_MUST_GRTR_DELTA

2.1.2.9 int smi_vrrp_api_set_vmac_status_sdkapi (struct smiclient_globals ∗ *azg*, int *new_vmac_stats*)

This function enables or disables Virtual MAC (VMAC). It affects all VRRP groups in a router. On a single network segment, multiple VRRP groups can be configured, each using a different VMAC. The use of VMAC addressing allows for faster switchover when a backup router assumes the master role. smi_vrrp_api_set_vmac_status_sdkapi

Parameters:

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

← ***new_vmac_stats*** numeric (0 | 1)

0 - Disable

1 - Enable

Returns:

VRRP_OK on success,VRRP_API_MASTER_FOUND when a search for session master instance succeeds otherwise VRRP_FAILURE

2.1.2.10 int smi_vrrp_api_switch_back_delay (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*, int *switch_back_delay*)

This function sets a switch-back delay timer for the master VRRP router. This feature prevents the original master VRRP router from transitioning back to the master state after coming back online until the configured delay timer has expired. smi_vrrp_api_-switch_back_delay

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***switch_back_delay*** Switch-back delay in milliseconds numeric <1-500000>

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERROR

2.1.2.11 int smi_vrrp_api_unset_advt_interval (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*)

This function restores the advertisement interval to its default value 1 second. smi_-vrrp_api_unset_advt_interval

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_ENABLED

2.1.2.12 int smi_vrrp_api_unset_priority (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*)

This function sets the priority of the VRRP router to the default value (VRRP_-DEFAULT_IP_OWNER_PRIORITY or VRRP_DEFAULT_NON_IP_OWNER_-PRIORITY). smi_vrrp_api_unset_priority

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_ENABLED
VRRP_API_SET_ERR_PRIO_CANT_BE_UNSET

2.1.2.13 int smi_vrrp_api_unset_switch_back_delay (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*)

This function sets a switch-back delay to default value 0. smi_vrrp_api_unset_switch_-back_delay

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_NO_SUCH_SESSION

2.1.2.14 int smi_vrrp_api_virtual_ip (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *af_type*, int *vr_id*, u_int32_t *ifindex*, u_int8_t ∗ *vip_addr*, bool_t *is_owner*)

This function configures the virtual IP address for a session. It receives the IP address and VR ID from the CM(Configuration Management) application, and configures the appropriate session. This function accepts the IP address from the CM application in whatever representation it uses. It assumes that the IP application also accepts addresses using this representation. smi_vrrp_api_virtual_ip

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***af_type*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***vip_addr*** IP address to configure as the virtual IP address

← ***is_owner*** Indicates whether the session is the owner of the IP address (TRUE) or not (FALSE). Used to determine the session priority to assign (the owner must have a priority of 255, and non-owner may not have a priority of 255).

Returns:

VRRP_OK on success, otherwise one of the following errors
VRRP_API_SET_ERR_INVALID_LINKLOCAL_ADDRESS
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_API_SET_ERR_ENABLED
VRRP_API_VIP_ALREADY_CONFIGURED_ANOTHER_SESSION

2.1.2.15 int smi_vrrp_debug (struct smiclient_globals ∗ *azg*, int *vr_id*, int *debug*)

Use this function to specify debugging options for VRRP. smi_vrrp_debug

Parameters:

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

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

SMI_VRRP_DBG_ALL - Specify debugging options for all VRRP events

SMI_VRRP_DBG_EVENTS - Specify debugging options for VRRP event troubleshooting

SMI_VRRP_DBG_PACKET - Specify debugging options for VRRP packets

SMI_VRRP_DBG_PACKET_RECV - Specify the debug option set for sent packets

SMI_VRRP_DBG_PACKET_SEND - Specify the debug option set for received packets

Returns:

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

2.1.2.16 int smi_vrrp_get_asso_ipaddr_rowstatus (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int8_t ∗ *ipaddr*, int ∗ *asso_rowstatus*)

This function gets the value of RowStatus variable, used according to installation and removal conventions for conceptual rows. smi_vrrp_get_asso_ipaddr_rowstatus

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***ipaddr*** Pointer to the location of associated IP address

→ ***asso_rowstatus*** Row status value numeric (1 | 2 | 4 | 5 | 6)
1 - active
2 - notInService
4 - createAndGo
5 - createAndWait
6 - destroy

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.17 int smi_vrrp_get_asso_storage_type (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int8_t ∗ *ipaddr*, u_int32_t ∗ *type*)

This function gets the storage type for this conceptual row IP address that is associated with a virtual router. smi_vrrp_get_asso_storage_type

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***ipaddr*** Pointer to the location of associated IP address

→ ***type*** Storage type numeric <1-5>
1 - other
2 - volatile
3 - nonVolatile(default)
4 - permanent
5 - readOnly

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.18 int smi_vrrp_get_checksum_errors (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, int ∗ *chksum_error*)

This function gets the total number of VRRP packets received with an invalid VRRP checksum value. smi_vrrp_get_checksum_errors

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

→ ***chksum_error*** Number of packets with invalid VRRP checksum value

Returns:

VRRP_API_GET_SUCCESS on success otherwise none

2.1.2.19 int smi_vrrp_get_notify (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, int ∗ *notify*)

This function gets the the flag status for trap generation. smi_vrrp_get_notify

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

→ ***notify*** ZebOS virtual router ID numeric (1 | 2)
1 - Enabled
2 - Disabled

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.20 int smi_vrrp_get_oper_accept_mode (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int32_t ∗ *accept_mode*)

This function gets whether a virtual router in Master state will accept packets addressed to the address owner's IPv6 address as its own if it is not the IPv6 address owner. smi_-vrrp_get_oper_accept_mode

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***accept_mode*** Whether a virtual router in Master state will accept packets (TRUE | FALSE)

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.21 int smi_vrrp_get_oper_addr_count (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *oper_count*)

This function gets the number of IP addresses that are associated with this virtual router. This number is equal to the number of rows in the vrrpv3AssociatedAddrTable that correspond to a given ifIndex/VRID/IP version. smi_vrrp_get_oper_addr_count

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***oper_count*** Total no of associated IP addresses <0-255>

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.22 int smi_vrrp_get_oper_adv_interval (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *adv_int*)

This function gets the time interval, in centiseconds between sending advertisement messages. Only the master router sends VRRP advertisements. smi_vrrp_get_oper_-adv_interval

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***adv_int*** Advertisement time interval in centiseconds

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.23 int smi_vrrp_get_oper_master_ipaddr (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int8_t ∗ *ipaddr*)

This function gets the master IP address of the VRRP virtual router. smi_vrrp_get_-oper_master_ipaddr

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***ipaddr*** Pointer to the location with retrieved master IP address

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.24 int smi_vrrp_get_oper_preempt_mode (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *preempt_mode*)

This function gets whether a higher priority virtual router will preempt a lower priority master. smi_vrrp_get_oper_preempt_mode

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***preempt_mode*** Whether higher priority router will preempt a lower priority master (TRUE | FALSE)

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.25 int smi_vrrp_get_oper_primary_ipaddr (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int8_t ∗ *ipaddr*)

This function gets IP address that becomes the 'vrrpv3OperationsMasterIpAddr',where there is more than one IP Address (associated IP addresses) for a given 'ifIndex'. In case where there is more than one IP address (associated IP address)for a given 'ifindex', it is used to specify the IP address that will become up. smi_vrrp_get_oper_-primary_ipaddr

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***ipaddr*** Pointer to the location with retrieved Primary IP address

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.26 int smi_vrrp_get_oper_priority (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *oper_priority*)

This function gets the priority to be used for the virtual router master election process. Higher values imply higher priority. smi_vrrp_get_oper_priority

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***oper_priority*** Priority value numeric <0-255>

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.27 int smi_vrrp_get_oper_rowstatus (struct smiclient_globals $*$ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int $*$ *rowstatus*)

This function gets the value of RowStatus variable should be used in accordance to installation and removal conventions for conceptual rows. smi_vrrp_get_oper_rowstatus

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***rowstatus*** Row status value numeric (1 | 2 | 3 | 4 | 5 | 6)

1 - active
2 - notInService
3 - notReady
4 - createAndGo
5 - createAndWait
6 - destroy

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.28 int smi_vrrp_get_oper_state (struct smiclient_globals $*$ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int $*$ *oper_state*)

This function gets the current state of the virtual router. smi_vrrp_get_oper_state

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***oper_state*** VRRP state numeric <1-3>

1 - VRRP_STATE_INIT (indicates that the virtual router is waiting for a startup event)

2 - VRRP_STATE_BACKUP (indicates the virtual router is monitoring the availability of the master router)

3 - VRRP_STATE_MASTER (indicates that the virtual router is forwarding packets for IP addresses that are associated with this router)

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.29 int smi_vrrp_get_oper_storage_type (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int32_t ∗ *storage_type*)

This function gets the storage type for this conceptual row. smi_vrrp_get_oper_-storage_type

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***ipi_vrid*** ZebOS virtual router ID
← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)
← ***vr_id*** VRRP session virtual router ID (1 - 255)
← ***ifindex*** Interface index
→ ***storage_type*** Storage type numeric <1-5>
1 - other
2 - volatile
3 - nonVolatile(default)
4 - permanent
5 - readOnly

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.30 int smi_vrrp_get_oper_uptime (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int32_t ∗ *oper_uptime*)

This function gets the amount of time, in TimeTicks (hundredth of a second), since this virtual router transitioned out of 'initialize'. smi_vrrp_get_oper_uptime

Parameters:

← ***azg*** Pointer to the SMI client global structure
← ***ipi_vrid*** ZebOS virtual router ID
← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)
← ***vr_id*** VRRP session virtual router ID (1 - 255)
← ***ifindex*** Interface index
→ ***oper_uptime*** VRRP uptime in TimeTicks (hundredth of a second)

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.31 int smi_vrrp_get_stats_address_list_errors (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_address_list_errors*)

This function gets the total number of packets received for which the address list does not match the locally configured list for the virtual router. smi_vrrp_get_stats_-address_list_errors

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_address_list_errors*** Number of VRRP received packets with address list does not match the locally configured list

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.32 int smi_vrrp_get_stats_adv_interval_errors (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_adv_interval*)

This function gets the total number of VRRP advertisement packets received for which the advertisement interval is different from the vrrpv3OperationsAdvInterval configured on this virtual router. smi_vrrp_get_stats_adv_interval_errors

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_adv_interval*** Number of VRRP received advertisements for which advertisement interval is VRRP operational advertisement configured

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.33 int smi_vrrp_get_stats_discontinuity_time (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_discontinuity_time*)

This function gets the value of sysUpTime on the most recent occasion at which any one or more of this entry's counters suffered a discontinuity. smi_vrrp_get_stats_-discontinuity_time

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_discontinuity_time*** value of sysUpTime on the most recent occasion

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.34 int smi_vrrp_get_stats_ip_ttl_errors (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_ip_ttl_errors*)

This function gets the total number of VRRP packets received by the Virtual router with IPv4 TTL (for VRRP over IPv4) or IPv6 Hop Limit (for VRRP over IPv6) not equal to 255. smi_vrrp_get_stats_ip_ttl_errors

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_ip_ttl_errors*** Number of VRRP received packets with invalid TTL value

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.35 int smi_vrrp_get_stats_master_transitions (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_master_transitions*)

This function gets the total number of times that this virtual router's state has transitioned to MASTER. smi_vrrp_get_stats_master_transitions

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_master_transitions*** Number of times router's state has transitioned to MASTER

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.36 int smi_vrrp_get_stats_packet_length_errors (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_packet_length_error*)

This function gets the total number of packets received with a packet length less than the length of the VRRP header. smi_vrrp_get_stats_packet_length_errors

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_packet_length_error*** Number of VRRP received packets with packet length less than the length of the VRRP header

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.37 int smi_vrrp_get_stats_rcvd_advertisements (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *rcvd_advert*)

This function gets the total number of VRRP advertisements received by this virtual router. smi_vrrp_get_stats_rcvd_advertisements

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***rcvd_advert*** Number of VRRP received advertisements

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.38 int smi_vrrp_get_stats_rcvd_invalid_authentications (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_rcvd_invalid_authentications*)

This function gets the trtal number of packets received with an unknown authentication type. char ∗ifname);. smi_vrrp_get_stats_rcvd_invalid_authentications

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_rcvd_invalid_authentications*** Number of VRRP received packets with unknown authentication type

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.39 int smi_vrrp_get_stats_rcvd_invalid_type_pkts (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_rcvd_invalid_type_pkts*)

This function gets the number of VRRP packets received by the virtual router with an invalid value in the 'type' field. smi_vrrp_get_stats_rcvd_invalid_type_pkts

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_rcvd_invalid_type_pkts*** Number of VRRP received packets with invalid value in the 'type' field

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.40 int smi_vrrp_get_stats_rcvd_pri_zero_packets (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_rcvd_pri_zero_packets*)

This function gets the total number of VRRP packets received by the virtual router with a priority of '0'. smi_vrrp_get_stats_rcvd_pri_zero_packets

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_rcvd_pri_zero_packets*** Number of VRRP received packets with a priority of '0'

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.41 int smi_vrrp_get_stats_refresh_rate (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_refresh_rate*)

This function gets the minimum reasonable polling interval for corresponding entry.It provides an indication of the minimum amount of time required to update the counters in corresponding entry. smi_vrrp_get_stats_refresh_rate

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_refresh_rate*** Minimum reasonable polling interval value

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.42 int smi_vrrp_get_stats_sent_pri_zero_packets (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int ∗ *stats_sent_pri_zero_packets*)

This function gets the total number of VRRP packets sent by the virtual router with a priority of '0'. smi_vrrp_get_stats_sent_pri_zero_packets

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

→ ***stats_sent_pri_zero_packets*** Number of VRRP sent packets with a priority of '0'

Returns:

VRRP_API_GET_SUCCESS on success otherwise VRRP_API_GET_ERROR

2.1.2.43 int smi_vrrp_get_version_errors (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, int ∗ *version_error*)

This function gets the total number of VRRP packets received with unknown or unsupported version number. smi_vrrp_get_version_errors

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

→ ***version_error*** Number of packets with unknown or unsupported version number

Returns:

VRRP_API_GET_SUCCESS on success otherwise none

2.1.2.44 int smi_vrrp_get_vrid_errors (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, int ∗ *vrid_error*)

This function gets the total number of VRRP packets received with a VRID that is not valid for any virtual router on this router. smi_vrrp_get_vrid_errors

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

→ ***vrid_error*** Number of packets with invalid VRID

Returns:

VRRP_API_GET_SUCCESS on success otherwise none

2.1.2.45 int smi_vrrp_no_debug (struct smiclient_globals ∗ *azg*, int *vr_id*, int *debug*)

Use this function to disable debugging. smi_vrrp_no_debug

Parameters:

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

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

SMI_VRRP_DBG_ALL - Specify debugging options for all VRRP events

SMI_VRRP_DBG_EVENTS - Specify debugging options for VRRP event troubleshooting

SMI_VRRP_DBG_PACKET - Specify debugging options for VRRP packets

SMI_VRRP_DBG_PACKET_RECV - Specify the debug option set for sent packets

SMI_VRRP_DBG_PACKET_SEND - Specify the debug option set for received packets

Returns:

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

2.1.2.46 int smi_vrrp_set_asso_ipaddr_rowstatus (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int8_t ∗ *ipaddr*, int *asso_ipaddr_rowstatus*)

This function sets the row status of the associated IP address entry. smi_vrrp_set_-asso_ipaddr_rowstatus

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***ipaddr*** Associated IP address

← ***asso_ipaddr_rowstatus*** Row status value numeric (1 | 2 | 4 | 5 | 6)
1 - active
2 - notInService
4 - createAndGo
5 - createAndWait
6 - destroy

Returns:

VRRP_API_SET_SUCCESS on success otherwise VRRP_API_SET_ERROR

2.1.2.47 int smi_vrrp_set_asso_storage_type (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int8_t ∗ *ipaddr*, int *asso_storage_type*)

This function sets the value of the storage type for associated IP address entry. smi_-vrrp_set_asso_storage_type

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***ipaddr*** Associated IP address

← ***asso_storage_type*** Storage type numeric <1-5>
1 - other
2 - volatile
3 - nonVolatile(default)
4 - permanent
5 - readOnly

Returns:

VRRP_API_SET_SUCCESS on success otherwise VRRP_API_SET_ERROR

2.1.2.48 int smi_vrrp_set_notify (struct smiclient_globals * *azg*, int *ipi_vrid*, int *notify*)

This function sets the value indicating whether this router generates SNMP notifications. smi_vrrp_set_notify

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***notify*** Indicates whether the notifications are enabled numeric (1 | 2)
1 - Enabled
2 - Disabled

Returns:

VRRP_API_SET_SUCCESS on success otherwise VRRP_API_SET_ERROR

2.1.2.49 int smi_vrrp_set_oper_accept_mode (struct smiclient_globals * *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int *oper_accept_mode*)

This function sets the accept mode (IPv6 only). smi_vrrp_set_oper_accept_mode

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***oper_accept_mode*** Accept mode numeric (1 | 2)
1 - True
2 - False

Returns:

VRRP_API_SET_SUCCESS on success otherwise VRRP_API_SET_ERROR

2.1.2.50 int smi_vrrp_set_oper_adv_interval (struct smiclient_globals * *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int *oper_adv_interval*)

This function sets the time interval between sending advertisement messages. smi_-vrrp_set_oper_adv_interval

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***oper_adv_interval*** Interval in centiseconds (default: 100)

Returns:

VRRP_API_SET_SUCCESS on success otherwise VRRP_API_SET_ERROR

2.1.2.51 int smi_vrrp_set_oper_primary_ipaddr (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, u_int8_t ∗ *ipaddr*)

This function sets the primary IP address of the VRRP virtual router, if multiple associated IP addresses are present. smi_vrrp_set_oper_primary_ipaddr

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***ipaddr*** Pointer to the location storing the selected primary IP address

Returns:

VRRP_API_SET_SUCCESS on success otherwise VRRP_API_SET_ERROR

2.1.2.52 int smi_vrrp_set_oper_rowstatus (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int *oper_rowstatus*)

This function sets the operational row status of the VRRP virtual router. smi_vrrp_-set_oper_rowstatus

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***ipaddr*** Associated IP address

← ***oper_rowstatus*** Row status value numeric (1 | 2 | 3 | 4 | 5 | 6)
   1 - active
   2 - notInService
   3 - notReady 4 - createAndGo
   5 - createAndWait
   6 - destroy

Returns:

VRRP_API_SET_SUCCESS on success otherwise VRRP_API_SET_ERROR

2.1.2.53 int smi_vrrp_set_oper_storage_type (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, u_int8_t *aftype*, u_int8_t *vr_id*, u_int32_t *ifindex*, int *oper_storage_type*)

This function sets the value of the storage type for this VRRP virtual router. smi_-vrrp_set_oper_storage_type

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router ID

← ***aftype*** Address family. AF_INET (IPv4) or AF_INET6 (IPv6)

← ***vr_id*** VRRP session virtual router ID (1 - 255)

← ***ifindex*** Interface index

← ***oper_storage_type*** Storage type numeric <1-5>
   1 - other
   2 - volatile
   3 - nonVolatile(default)
   4 - permanent
   5 - readOnly

Returns:

VRRP_API_SET_SUCCESS on success otherwise VRRP_API_SET_ERROR

2.1.2.54 int smi_vrrp_set_session_by_ifname_ipv4_sdkapi (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, int *vr_id*, char ∗ *ifname*)

Updates or creates a new VRRP session on the given interface and allocates resources for the session. smi_vrrp_set_session_by_ifname_ipv4_sdkapi

Parameters:

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

← ***ipi_vrid*** ZebOS Virtual Router Id

← ***vr_id*** VRRP Session virtual rounter Id

← ***ifname*** Interface name

Returns:

VRRP_OK on success, otherwise one of the following error codes
VRRP_API_SET_ERR_NO_SUCH_INTERFACE
VRRP_API_SET_ERR_L2_INTERFACE
VRRP_API_SET_ERR_SESSION_GET_OR_CRE

2.1.2.55 int smi_vrrp_set_session_by_ifname_ipv6_sdkapi (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, int *vr_id*, char ∗ *ifname*)

Updates or creates a new VRRP session on the given interface and allocates resources for the session. smi_vrrp_set_session_by_ifname_ipv6_sdkapi

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router id

← ***vr_id*** VRRP Session virtual router id

← ***ifname*** Interface name

Returns:

VRRP_OK on success, otherwise one the following errors
VRRP_API_SET_ERR_NO_SUCH_INTERFACE
VRRP_API_SET_ERR_L2_INTERFACE
VRRP_API_SET_ERR_SEESION_GET_OR_CRE

2.1.2.56 int smi_vrrp_set_session_by_vlanid_ipv4_sdkapi (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, int *vr_id*, int *vlan_id*)

Updates or creates a new VRRP session on the given interface and allocates resources for the session. smi_vrrp_set_session_by_vlanid_ipv4_sdkapi

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router id

← ***vr_id*** VRRP Session virtual router id (1 - 255)

← ***vlan_id*** VLAN Id

Returns:

VRRP_OK in success, otherwise one of the following errors VRRP_API_SET_-ERR_NO_SUCH_INTERFACE VRRP_API_SET_ERR_L2_INTERFACE VRRP_API_SET_ERR_SEESION_GET_OR_CRE

2.1.2.57 int smi_vrrp_set_session_by_vlanid_ipv6_sdkapi (struct smiclient_globals ∗ *azg*, int *ipi_vrid*, int *vr_id*, int *vlan_id*)

Updates or creates a new VRRP session on the given interface and allocates resources for the session. smi_vrrp_set_session_by_vlanid_ipv6_sdkapi

Parameters:

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

← ***ipi_vrid*** ZebOS virtual router id

← ***vr_id*** VRRP Session virtual rounter id (1 - 255)

← ***vlan_id*** VLAN Id

Returns:

VRRP_OK on success, otherwise one or the following errors
VRRP_API_SET_ERR_NO_SUCH_INTERFACE
VRRP_API_SET_ERR_L2_INTERFACE
VRRP_API_SET_ERR_SEESION_GET_OR_CRE

2.1.2.58 s_int32_t smi_vrrp_show_session_all (struct smiclient_globals ∗ *azg*, struct vrrpGlobal ∗ *globalData*, struct list ∗ *vrrpSessList*, int(∗)(struct list ∗vrrpSessList) *funPointer*)

Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) characteristics of all VRRP sessions. smi_vrrp_show_session_all

Parameters:

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

→ ***globalData*** Pointer to structure vrrpGlobal

→ ***vrrpSessList*** Pointer to linked list of structure vrrpSession

← ***funPointer*** Callback function pointer

Returns:

0 on success, otherwise one of the following error codes
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_ERR_GLOBAL_DATA_NULL
VRRP_ERR_MEM_ALLOC_FAIL

2.1.2.59 s_int32_t smi_vrrp_show_session_v4 (struct smiclient_globals ∗ *azg*, u_int8_t *vr_id*, char ∗ *ifname*, struct vrrpGlobal ∗ *globalData*, struct list ∗ *vrrpSessList*, int(∗)(struct list ∗vrrpSessList) *funPointer*)

Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) characteristics of given VRRP session. smi_vrrp_show_session_v4

Parameters:

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

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

← ***ifname*** Interface name

→ ***globalInfo*** Pointer to structure vrrpGlobal

→ ***vrrpSessList*** Pointer to linked list of structure vrrpSession

← ***funPointer*** Callback function pointer

Returns:

0 on success, otherwise one of the following error codes
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_API_SET_ERR_NO_SUCH_INTERFACE
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_ERR_GLOBAL_DATA_NULL
VRRP_ERR_MEM_ALLOC_FAIL

2.1.2.60 s_int32_t smi_vrrp_show_session_v6 (struct smiclient_globals ∗ *azg*, u_int8_t *vr_id*, char ∗ *ifname*, struct vrrpGlobal ∗ *globalData*, struct list ∗ *vrrpSessList*, int(∗)(struct list ∗vrrpSessList) *funPointer*)

Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) characteristics of given VRRP session. smi_vrrp_show_session_v6

Parameters:

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

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

← ***ifname*** Interface name

→ ***globalInfo*** Pointer to structure vrrpGlobal

→ ***vrrpSessList*** Pointer to linked list of structure vrrpSession

← ***funPointer*** Callback function pointer

Returns:

0 on success, otherwise one of the following error codes
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_API_SET_ERR_NO_SUCH_INTERFACE

VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_ERR_GLOBAL_DATA_NULL
VRRP_ERR_MEM_ALLOC_FAIL

2.1.2.61 s_int32_t smi_vrrp_show_statistics_all (struct smiclient_globals ∗ *azg*, struct vrrpGlobal ∗ *globalData*, struct list ∗ *vrrpStatList*, int(∗)(struct list ∗vrrpStatList) *funPointer*)

Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) statistics of all VRRP sessions. smi_vrrp_show_statistics_all

Parameters:

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

→ ***globalInfo*** Pointer to structure vrrpGlobal

→ ***vrrpStatList*** Pointer to linked list of structure vrrpStatistics

← ***funPointer*** Callback function pointer

Returns:

0 on success, otherwise one of the following error codes
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_ERR_GLOBAL_DATA_NULL
VRRP_ERR_MEM_ALLOC_FAIL

2.1.2.62 s_int32_t smi_vrrp_show_statistics_v4 (struct smiclient_globals ∗ *azg*, u_int8_t *vr_id*, char ∗ *ifname*, struct vrrpGlobal ∗ *globalData*, struct list ∗ *vrrpStatList*, int(∗)(struct list ∗vrrpStatList) *funPointer*)

Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) statistics of given VRRP session. smi_vrrp_show_statistics_v4

Parameters:

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

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

← ***ifname*** Interface name

→ ***globalInfo*** Pointer to structure vrrpGlobal

→ ***vrrpStatList*** Pointer to linked list of structure vrrpStatistics

← ***funPointer*** Callback function pointer

Returns:

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

VRRP_API_SET_ERR_NO_SUCH_INTERFACE
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_ERR_GLOBAL_DATA_NULL
VRRP_ERR_MEM_ALLOC_FAIL

2.1.2.63 s_int32_t smi_vrrp_show_statistics_v6 (struct smiclient_globals ∗ *azg*, u_int8_t *vr_id*, char ∗ *ifname*, struct vrrpGlobal ∗ *globalData*, struct list ∗ *vrrpStatList*, int(∗)(struct list ∗vrrpStatList) *funPointer*)

Shows Virtual Router Redundancy Protocol Version 3 (VRRPv3) statistics of given VRRP session. smi_vrrp_show_statistics_v6

Parameters:

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

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

← ***ifname*** Interface name

→ ***globalInfo*** Pointer to structure vrrpGlobal

→ ***vrrpStatList*** Pointer to linked list of structure vrrpStatistics

← ***funPointer*** Callback function pointer

Returns:

0 on success, otherwise one of the following error codes
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_API_SET_ERR_NO_SUCH_INTERFACE
VRRP_API_SET_ERR_NO_SUCH_SESSION
VRRP_ERR_VR_DOES_NOT_EXIST
VRRP_ERR_GLOBAL_DATA_NULL
VRRP_ERR_MEM_ALLOC_FAIL

Index